

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи надання доступу до мережі
Internet сервіс провайдера”

КБГЗ - 2024

Виконав здобувач вищої освіти
IV курсу, групи КІ-20
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Серeda Є.В.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 17 » січня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Середі Єгору Віталійовичу

(прізвище, ім'я, по батькові)

- Тема роботи *Програмне забезпечення системи надання доступу до мережі Internet сервіс провайдера*
- Керівник роботи *Смірнова Тетяна Віталіївна, канд. техн. наук*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 131-02 від 01.04.2024 року
- Строк подання студентом роботи до захисту *23.05.2024 р.*
- Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи надання доступу до мережі Internet сервіс провайдера*
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.*
 - Перегляд аналогічних існуючих систем.*
 - Опис і обґрунтування проектних рішень.*
 - Етапи програмування системи.*
 - Впровадження системи в промислову експлуатацію.*
 - Висновки*
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

| | |
|--|-----------------|
| <i>Структурна схема системи</i> | <i>1 аркуш</i> |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> |

7. Дата видачі завдання « 17 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти | Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти | Примітка |
|-------|---|---|----------|
| 1. | Аналіз існуючих систем | 10.03.2024 р. | |
| 2. | Постановка задачі, оформлення ТЗ | 15.03.2024 р. | |
| 3. | Розробка моделі компонента | 20.03.2024 р. | |
| 4. | Розробка структур даних | 25.03.2024 р. | |
| 5. | Розробка алгоритмів зв'язку та відображення | 30.03.2024 р. | |
| 6. | Програмування алгоритмів | 10.04.2024 р. | |
| 7. | Оформлення ПЗ | 17.04.2024 р. | |
| 8. | Попередній захист роботи | 23.05.2024 р. | |
| | | | |
| | | | |
| | | | |
| | | | |

Дата видачі завдання
« 17 » січня 2024 р.

Підпис керівника

Смірнова Т.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2024 р.

Підпис здобувача

Середа Є.В.
(прізвище та ініціали)

АНОТАЦІЯ

Серета Є.В. Програмне забезпечення системи надання доступу до мережі Internet сервіс провайдера. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи надання доступу до мережі Internet сервіс провайдера.

Метою розробки є програмне забезпечення системи надання доступу до мережі Internet сервіс провайдера.

Результат роботи – програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі PHP, Perl.

Ключові слова: комп'ютерна інженерія, Internet сервіс провайдера

ABSTRACT

Sereda E.V. Software of the system of providing access to the Internet service provider network. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this final qualification work for the first (bachelor) level of higher education, software is developed, which is intended for the system of providing access to the Internet service provider's network.

The purpose of the development is the software of the system for providing access to the Internet service provider's network.

The result of the work is the software implementation of the system for providing access to the Internet service provider's network.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the PHP, Perl environment.

Keywords: computer engineering, Internet service provider

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

HTML (HyperText Markup Language) – мова розмітки гіпертекстових документів;

СУБД – системи управління базами даних;

TCP/IP (Transport Control Protocol/Internet Protocol) – протокол роботи мережі Інтернет;

ГМ – глобальні мережі;

ЛВМ – локальна віртуальна мережа;

ОД – остаточне обладнання даних;

ОС – операційна система;

ПК – персональний комп'ютер;

ОЗП – основний запам'ятовуючий пристрій;

URL – universal resource locator – локатор ресурсів інтернет;

PPTP – Point-to-Point-Tunneling Protocol – протокол створення захищеного каналу при доступі віддалених користувачів через публічні мережі;

ВСЗП – віртуальна система захисту інформаційного потоку;

ВСПП – віртуальні системи поділу інформаційних потоків;

ВДТ – відеодисплейний термінал;

ПЕОМ – персональна електронно обчислювальна машина;

PHP – створює HTML-сторінки з використанням спеціальних тегів, розпізнаваних аналізатором PHP;

WWW – world wide web – всесвітня мережа інтернет;

FTP – File Transport Protocol – протокол передачі файлів.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 2 |

ВСТУП

Актуальність теми. Інтернет-провайдер (ISP) надає доступ до Інтернету. Цей доступ може здійснюватися через кабельне, DSL або комутоване з'єднання. Усі пристрої, підключені до Інтернету, запускають кожен запит через Інтернет-провайдера для доступу до серверів, де вони можуть переглядати веб-сторінки та завантажувати файли. Сервери надають ці файли через свого провайдера. Прикладами провайдерів є AT&T, Comcast/Xfinity, Verizon, Cox і NetZero. Ці Інтернет-провайдери можуть бути підключені безпосередньо до дому чи підприємства або передаватися по бездротовій мережі за допомогою супутника чи іншої технології.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи надання доступу до мережі Internet сервіс провайдера.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем надання доступу до мережі Internet сервіс провайдера.
- Дослідження системи надання доступу до мережі Internet сервіс провайдера.
- Програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі надання доступу до мережі Internet сервіс провайдера.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи надання доступу до мережі Internet сервіс провайдера, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 3 |

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Більшість будинків і підприємств мають пристрій, який підключається до Інтернету. Саме через цей пристрій телефони, ноутбуки, настільні комп'ютери та інші пристрої з підключенням до Інтернету досягають решти світу – і це робиться через Інтернет-провайдера.

Ось приклад ролі, яку відіграє постачальник послуг Інтернету, коли ви завантажуєте файли та відкриваєте веб-сторінки з Інтернету.

1. Коли ви використовуєте свій ноутбук вдома для доступу до сторінки на сайті, наприклад Lifewire.com, веб-браузер використовує DNS-сервери, налаштовані на пристрої, для перекладу доменного імені Lifewire на IP-адресу, з якою воно пов'язане, що це адреса, яку Lifewire налаштовано використовувати зі своїм провайдером.

2. IP-адреса надсилається з вашого маршрутизатора до вашого провайдера, який пересилає запит провайдеру, якого використовує Lifewire.com.

3. У цей момент Інтернет-провайдер для Lifewire.com надсилає сторінку вашому Інтернет-провайдеру, який пересилає дані на ваш домашній маршрутизатор і на ваш ноутбук.

Усе це робиться швидко – зазвичай за лічені секунди. Однак, щоб це працювало, і домашня мережа, і мережа Lifewire повинні мати дійсну публічну IP-адресу, яку призначає провайдер.

Ця ж концепція застосовується під час надсилання та завантаження інших файлів, таких як відео, зображення та документи. Усе, що ви завантажуєте в Інтернеті, передається через Інтернет-провайдера.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 4 |

1.2 Область застосування

Тест швидкості Інтернету показує швидкість, яку ви отримуєте від свого провайдера. Якщо ця швидкість відрізняється від тієї, за яку ви платите, зверніться до свого провайдера та поділіться результатами.

Більшість інтернет-провайдерів надають клієнтам динамічні IP-адреси, які постійно змінюються, але компанії, які обслуговують веб-сайти, зазвичай підписуються зі статичною IP-адресою, яка не змінюється.

Інші типи провайдерів включають хостингових провайдерів, наприклад тих, які розміщують лише електронну пошту чи онлайн-сховище, а також безкоштовних або некомерційних провайдерів (іноді їх називають безкоштовними мережами), які надають безкоштовний доступ до Інтернету, як правило, супроводжується рекламою.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи надання доступу до мережі Internet сервіс провайдера, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 5 |

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Спочатку Інтернет був розроблений для використання урядами та деякими університетськими факультетами для досліджень. Згодом ці організації почали надавати доступ своїм викладачам і співробітникам, а також іншим організаціям і компаніям, які також приєдналися через непрямі та прямі зв'язки.

До 1980-х років було зроблено зусилля, щоб зробити Інтернет доступним для комерційного та загального використання. Регулятори зняли деякі обмеження на використання Інтернету до 1991 року, і широка громадськість могла отримати доступ до нього через Всесвітню павутину.

На початку було лише кілька провайдерів, таких як America Online (AOL), і вони надавали обмежений доступ через комутоване з'єднання з використанням телефонних ліній. Поступово швидкість підключення та параметри збільшувалися, щоб започаткувати Інтернет-економіку з використанням передових технологій, таких як широкосмуговий зв'язок через кабелі та DSL-модеми, доки в 1995 році обмеження на використання Інтернету нарешті не було знято з комерційного трафіку.

Типи провайдерів

Ще в 90-х провайдери були одного з трьох типів: комутовані послуги, DSL від телефонних провайдерів і широкосмугові кабельні компанії. Проте протягом наступних двох десятиліть комутовані зв'язки ставали рідшими через низьку швидкість, тоді як інші варіанти процвітали.

Давайте розглянемо різні типи інтернет-провайдерів, доступних сьогодні.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 6 |

Інтернет-провайдери комутованого доступу

Комутований доступ до Інтернету використовує поточну телефонну лінію для підключення вас до Інтернету. Щоб встановити з'єднання, потрібно набрати номер доступу (схожий на номер телефону) за допомогою модему. Коли ви в мережі, ви не можете здійснити телефонний дзвінок зі свого номера, на відміну від підключення DSL.

Комутований зв'язок легко доступний через стаціонарний зв'язок, він безпечний і економічний. Однак він дуже повільний – 56 Кбіт/с – це максимальна швидкість Інтернету, яку ви можете досягти з ним. Враховуючи передові технології, які ми використовуємо сьогодні, цієї швидкості недостатньо для більшості операцій, особливо через те, що деякі веб-сторінки чи програми не завантажуватимуться, і може виникнути час очікування перед завантаженням.

Тому люди переходять до інших варіантів. Згідно зі звітом Statista за 2021 рік, лише 1,9% домогосподарств користувалися комутованим доступом у США. Він використовується в деяких сільських районах, де немає широкосмугового зв'язку, але в інших випадках від нього відмовилися.

Постачальники DSL

Цифрова абонентська лінія (DSL) забезпечує підключення до Інтернету через телефонну лінію. Послуги широко доступні, тому що будинки вже підведені до телефонних з'єднань. Його надають традиційні телефонні компанії.

Технологія DSL використовує додаткові сигнали, які не використовують телефонні сигнали. Він використовує маршрутизатор DSL для підключення до телефонної розетки через телефонний кабель. Ці можливості дозволяють користувачам користуватися Інтернетом, навіть коли вони користуються телефоном або коли телефон дзвонить.

DSL не є спільною для користувачів і може бути найкращим варіантом, якщо у вас немає інших доступних опцій. Він пропонує середню швидкість Інтернету і не вимагає від вас інвестицій у дороге обладнання. Просто купіть модем, і ваш телефонний зв'язок дозволить вам підключитися до Інтернету. Отже,

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 7 |

коли ви купуєте з'єднання DSL, переконайтеся, що ви знаходитесь достатньо близько до постачальника послуг, оскільки швидкість впливає на відстань.

Відповідно до іншого звіту Statista, опублікованого минулого року, близько 57% споживачів у США використовують широкосмуговий зв'язок, такий як DSL і кабель, що робить його найпопулярнішим варіантом.

Провайдери широкосмугового кабельного зв'язку

Компанії кабельного телебачення зазвичай пропонують широкосмугові кабельні послуги. Широкосмугові кабелі використовують коаксіальні кабелі, які доставляють кабельне телебачення до дому. Постачальники послуг також пропонують надійні та швидкі Інтернет-послуги, які допоможуть вам виконувати різноманітні операції вдома чи в офісі. Ви можете досягти швидкості Інтернету від одного до ста Мбіт/с.

Перевага використання широкосмугових кабелів полягає в тому, що вони включають низькі затримки, що не сильно впливає на вас. Ви можете зіткнутися з меншою затримкою під час виконання певних дій, наприклад відкриття веб-сайту, перегляду відео або гри в онлайн-відеогру.

Завдяки широкій доступності широкосмугові кабелі популярні серед користувачів Інтернету. Ви також можете звернутися до наведеної вище статистики, де більше 50% споживачів використовують його для підключення до Інтернету. Однак ці послуги також залежать від вашого географічного розташування. Його спільно використовують різні користувачі у вашому районі, а виділена пропускна здатність розподіляється між усіма. Отже, чим менше користувачів або трафіку надходить від них, тим швидшим буде підключення до Інтернету.

Використання широкосмугового кабелю дає можливість заощадити гроші, дозволяючи об'єднати свої послуги, як-от Інтернет і кабельне телебачення.

Оптоволоконні інтернет-провайдери

Підключення до Інтернету, що використовує технологію волоконної оптики, є найшвидшим Інтернет-сервісом. Інтернет-провайдери, які пропонують

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 8 |

цю послугу, використовують волоконно-оптичні кабелі, виготовлені зі скляних ниток, для передачі даних зі швидкістю світла. Він не використовує мідь. Причин для використання оптоволокна багато – це середовище для пропускання світла та забезпечує такі якості, як:

- Захищеність від електромагнітних перешкод.
- Висока пропускна здатність, наприклад гігабайт на секунду.
- Вищі сигнали передачі максимум 150 миль.
- Постійність і надійність.

Оптоволокно пропонує максимальну швидкість і найменші затримки під час використання Інтернету. Ви можете очікувати незначних затримок під час онлайн-ігор, проведення відеоконференцій, потокового відео тощо. Оптоволоконне підключення створено для підтримки інтенсивного використання Інтернету з максимальною пропускну здатністю та підтримкою, що дозволяє користувачам одночасно транслювати відео, ділитися великими файлами та використовувати пристрої IoT вдома чи в офісі.

В даний час досягаються нові досягнення, такі як встановлення зв'язків між країнами через волоконно-оптичні кабелі, що проходять уздовж дна океану на неймовірних швидкостях без перешкод. Однак цей тип підключення є дорогим, тому існує менше областей і установ, де підключено волоконно-оптичне підключення.

Крім того, багато областей досі не підключені, щоб допомогти цій технології досягти споживачів. Оскільки це відносно нова концепція, докладаються зусилля, щоб підвищити її доступність за допомогою супутникових антен.

Популярним прикладом волоконно-оптичних з'єднань є Google Fiber, який пропонує повний гігабайт пропускну здатності за такою ж ціною, як і багато постачальників послуг DSL і кабельного телебачення. У звіті за 2021 рік говориться, що в багатьох країнах використання оптоволокна перевищило 30% абонентів широкопasmового зв'язку.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 9 |

Провайдери супутникового Інтернету

Провайдери послуг супутникового Інтернету використовують геостаціонарні супутники для передачі даних між Інтернетом і користувачами.

Оскільки дані передаються з космосу на землю – на відстань 22 000 миль – ці великі відстані можуть спричинити значні затримки. Насправді супутниковий Інтернет пропонує найвищу затримку відповіді серед усіх інших типів підключення. Крім того, на швидкість можуть сильно вплинути погодні умови. Швидкість також залежить від перевантаженості мережі або від того, наскільки мережа зайнята в певний час.

Іншим недоліком супутника є те, що вам може постійно заважати обмежена пропускна здатність. Це буде недостатньо надійно, якщо ви бажаєте узгодженого підключення для виконання бізнес-процесів або гри в онлайн-ігри.

Однак це єдина надія для багатьох людей у сільській місцевості, пустелях, гірських місцевостях і на фермах, які хочуть відносно високих швидкостей. Це кращий варіант, ніж мобільний Інтернет і телефонне підключення. Хоча він не такий швидкий порівняно з іншими типами з'єднання, його перевага полягає в тому, що він створює менший вплив на навколишнє середовище та пропонує більшу гнучкість визначення місця розташування. (Це також чудовий варіант для центрів відновлення після стихійного лиха.)

Провайдери Wi-Fi

Іншим типом провайдера є той, який пропонує з'єднання Wi-Fi. Кілька користувачів можуть отримати доступ до Wi-Fi з легкістю та гнучкістю, як ніде більше, оскільки модем не потрібен. Використання Wi-Fi дає вам можливість пересуватися, залишаючись на зв'язку. Сьогодні ви можете отримати з'єднання Wi-Fi за допомогою маленького кишенькового пристрою, який можна носити куди завгодно та працювати з великою кількістю незалежності. Крім того, Wi-Fi також безкоштовно пропонують уряди, готелі, ресторани та вокзали в різних частинах світу. Це надає доступ до Інтернету більшій кількості людей і допомагає людям, які не можуть дозволити собі тарифні плани Інтернету.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 10 |

Постачальники доступу

Постачальники доступу пропонують доступ до Інтернету за допомогою різних технологій для встановлення з'єднання для користувачів. Вони можуть використовувати комп'ютерні модеми, оптоволокно, телевізійні кабелі та телефонні лінії. Малі підприємства та окремі особи можуть використовувати традиційні варіанти, як-от мідні дроти для комутованого з'єднання, кабельні модеми, асиметричну цифрову абонентську лінію (ADSL) тощо.

Для великих і середніх організацій із постійними потребами у високошвидкісному Інтернеті постачальники послуг доступу пропонують Ethernet, швидші DSL, синхронну оптичну мережу (SONET) тощо. Крім того, вони також можуть використовувати супутникове підключення до Інтернету для аварійного відновлення, бездротового доступу тощо.

Хостинг провайдерів

Компанії, що надають веб-хостинги, можуть виступати в якості провайдера, щоб пропонувати такі послуги, як рішення для веб-хостингу, системи онлайн-сховища, постачальники послуг електронної пошти, хмарні рішення, серверні операції, віртуальні сервери тощо. Доступні численні хостинг-провайдери, такі як Kinsta, які пропонують виняткову продуктивність, швидкість, масштабованість і гнучкість.

Постачальники поштових скриньок

Постачальники поштових скриньок пропонують послуги з розміщення доменів електронної пошти з доступом до пам'яті для пошти. Вони пропонують сервери електронної пошти для надсилання, отримання, прийому та зберігання електронних листів для організацій і окремих осіб.

Транзитні IP-адреси

Існують різні рівні постачальників Інтернет-послуг. Вихідні провайдери мають більші мережі, ніж контрактні провайдери, які надають доступ провайдерам за контрактом, які не можуть самостійно отримати доступ до частини Інтернету. Це працює подібно до того, як клієнти платять за доступ до

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 11 |

Інтернету: провайдери, що укладають договір, платять першим провайдерам.

Різні рівні:

- Рівень 1.
- Рівень 2.
- Рівень 3.
- Мережі.
- Інтернет-провайдери.

Віртуальні провайдери

Віртуальні провайдери (VISP) купують послуги в інших провайдерів (або оптових провайдерів). Вони дозволяють клієнтам VISP користуватися Інтернетом, яким керує оптовий провайдер.

VISP може бути місцевим оператором обміну даними для голосового зв'язку або оператором віртуальної мобільної мережі.

Як працюють інтернет-провайдери?

Як обговорювалося вище, існують різні рівні провайдерів. Давайте краще розберемося, що це таке та як вони працюють, щоб надавати Інтернет-послуги кінцевим користувачам.

– Найвищі провайдери Інтернету належать до рівня 1 з доступом до всіх мереж, які Інтернет може мати через угоди про піринг. Вони з'єднують усі куточки Інтернету, і деякі з їхніх прикладів – такі популярні провайдери, як Verizon, Deutsch Telekom і Bharti.

– Інтернет-провайдери рівня 1 продають свій доступ до мережі наступному рівню, тобто рівню 2. Домівки, приватні особи, підприємства тощо можуть купувати доступ до Інтернету у рівня 2. Однак ISO рівня 1 також може вибрати продаж доступу до Інтернету кінцевим користувачам.

– Також може існувати проміжний Інтернет-провайдер рівня 3, який може купувати пропускну здатність мережі у свого Інтернет-провайдера верхнього рівня (провайдера рівня 2), а потім продавати його кінцевим користувачам.

Уявіть провайдера як шлюз або точку доступу, яка надає користувачам

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 12 |

доступ до Інтернету, як правило, за певну плату. Ось як усі провайдери співпрацюють один з одним, щоб надавати доступ до Інтернету споживачам.

Тепер давайте розберемося, що насправді готується всередині.

Коли Інтернет-трафік йде від вашої домашньої чи організаційної мережі до місця призначення для отримання інформації, він стикається з великою кількістю стрибків. Він може направляти від модему до мережі рівня 3, мережі рівня 2 і мережі рівня 1. Після отримання інформації вона знову повертається через різних провайдерів, перш ніж досягти пункту призначення.

Щоб встановити з'єднання, провайдери використовують різні технології – кабелі, телефонні лінії, DSL, супутник, волоконну оптику та Wi-Fi тощо.

Тепер кінцеві користувачі повинні мати модем і активний обліковий запис, щоб встановити з'єднання зі своїми інтернет-провайдерами. Модеми мають кабельні виходи або телефонну лінію для підключення до провайдерів.

Таким чином, після перевірки вашого облікового запису ваш провайдер призначить IP-адресу, яка буде унікальною для вашого модему. Таким чином, вам буде надано доступ до Інтернету, і ви зможете насолоджуватися вивченням Інтернету для особистого чи професійного використання.

Роль провайдера

Незалежно від того, чи маєте ви бізнес чи просто хочете досліджувати Інтернет заради розваги, стабільне та надійне підключення до Інтернету – усе, що вам потрібно. Ваш смартфон, ноутбук, бізнес-пристрої, пристрої Інтернету речей удома – усе потребує підключення до Інтернету, яке надають не хто інший, як інтернет-провайдери.

І провайдери спрощують це, виконуючи всю фонову роботу, щоб у вас було безперебійне підключення до Інтернету, звичайно, сплачуючи певні гроші, принаймні в більшості випадків. Вони допомагають вам забезпечити чудову взаємодію з клієнтами, коли вони відвідують ваш сайт, щоб переглянути ваші продукти чи послуги, купити щось або прочитати ваші статті. Усе це допоможе вам розвивати свій бізнес.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 13 |

Давайте розглянемо переваги, які ви можете очікувати від постачальника послуг Інтернету.

Сильніша безпека мережі

Питання кібербезпеки зростають швидше, ніж будь-коли. Тактика поганих акторів швидко змінюється і розвивається; отже, використання безпечного з'єднання від відомого провайдера допоможе вам захистити вашу мережу та всі ваші дані.

Найпопулярніші інтернет-провайдери використовують надійні механізми безпеки, такі як шифрування, політики конфіденційності, можливості моніторингу тощо. Функції та можливості залежать від провайдера.

Вища пропускна здатність

Вам потрібна достатня пропускна здатність для підтримки всіх ваших бізнес-операцій. Крім того, це має бути швидше. Ці два аспекти важливі навіть для особистого використання. Взявши послугу Інтернету від хорошого провайдера, ви запропонуєте достатню пропускну здатність і швидкість, щоб допомогти вам прискорити ваші бізнес-процеси, уникнувши при цьому проблем під час перегляду улюблених шоу онлайн.

Широкий вибір варіантів

Інтернет-провайдери пропонують різноманітні послуги своїм клієнтам, щоб вони могли вибрати плани відповідно до своїх потреб, наприклад для особистого чи бізнес-користування. Якщо у вас є бізнес, вам потрібен надійний план із високошвидкісним з'єднанням і пропускнуою спроможністю, щоб ви могли вибрати схожі плани залежно від розміру вашого бізнесу, великого, середнього чи малого. Однак, якщо вам це потрібно для особистого користування, ви також отримаєте різні плани на вибір.

Надійна мережа

Використання підключення до Інтернету від хорошого провайдера не підведе вас або ваших клієнтів через часті простої, затримки відповіді або затримки. Ви отримаєте більший час безперебійної роботи та безперебійне

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 14 |

з'єднання, яке є надійним для підтримки вашого бізнесу, задоволення ваших клієнтів і запропонує вам безпроблемний досвід дослідження Інтернету. Вони також мають автоматичні системи перемикання після відмови, які можуть перенаправляти трафік, якщо одне з посилань не працює.

Керує бізнесом

Пропонуючи своїм клієнтам постійний і безперебійний досвід, ви можете краще обслуговувати їх. Це сприятиме розвитку вашого бізнесу через продажі, дохід і лояльність клієнтів до вашого бренду. Вашим співробітникам також стане простіше працювати та швидко виконувати роботу; ви можете навіть спостерігати підвищення рівня продуктивності.

Глосарій ISP: деякі загальні терміни ISP

У світі інтернет-провайдерів є деякі часто використовувані терміни, про які ви повинні знати, якщо хочете краще зрозуміти цю концепцію.

– **Широкосмуговий зв'язок:** Ви можете користуватися цією послугою Інтернету в будь-який час, оскільки вона постійно працює. Його швидкість завантаження становить мінімум 25 Мбіт/с, а швидкість завантаження – мінімум 3 Мбіт/с.

– **Wi-Fi:** бездротова мережа – це підключення до Інтернету, яке дозволяє підключати до Інтернету ваші пристрої, наприклад смартфони, ноутбуки, планшети, віртуальних помічників тощо. Він створений за допомогою маршрутизатора з радіохвилями.

– **Пропускна здатність:** це кількість даних, що передаються мережею за один раз.

– **Пакет:** дані переміщуються через Інтернет у «пакетах». Якщо пакети не досягають місця призначення, це називається втратою пакетів, що призводить до затримок і зниження швидкості.

– **Обмеження обсягу даних:** Ваш тарифний план Інтернету матиме обмеження на певний обсяг даних на місяць. Деякі з планів також можуть мати необмежену кількість даних.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 15 |

– **Пакет:** якщо ви отримуєте кілька послуг від Інтернет-провайдера, наприклад підключення до Інтернету та кабельного телебачення, вони часто групуватимуть ці послуги в оплаті, утворюючи «пакет». Іншими послугами, які ви можете об'єднати, можуть бути передача голосу через Інтернет-протокол (VoIP), домашня автоматизація, протоколи підвищеної безпеки тощо.

– **ADSL:** Асиметрична цифрова абонентська лінія (ADSL) – це послуга DSL для домашнього використання. Він має високу швидкість завантаження та завантаження.

– **SDSL:** це DSL для компаній і має однакову швидкість завантаження та завантаження. Це дозволяє організаціям пришвидшити свої процеси, такі як хмарне резервне копіювання, виконання операцій тощо.

– **Узгодженість:** Узгодженість означає, наскільки надійними є інтернет-послуги провайдера з точки зору їх швидкості під час інтенсивного використання.

– **Швидкість завантаження:** коли ви шукаєте щось в Інтернеті, транслюєте відео, граєте в ігри, переглядаєте веб-сторінки, відтворюєте музику тощо, ви фактично завантажуйте дані. Швидкість, з якою ви можете завантажити цю інформацію, є вашою швидкістю завантаження.

– **Швидкість завантаження:** це швидкість, з якою інформація надсилається через Інтернет, наприклад, коли ви надсилаєте зображення друзів чи колезі.

– **Затримка:** це затримка, яку може відчувати користувач, коли запитує певну інформацію з Інтернету, наприклад, коли він завантажує веб-сторінку.

– Мбіт/с, Кбіт/с, Гбіт/с: мегабіт на секунду (Мбіт/с) – це одиниця вимірювання швидкості завантаження та завантаження в секундах, коли дані переміщуються через мережу. 1 Мбіт/с дорівнює 1000 Кбіт/с або кілобіт на секунду, а 1000 Мбіт/с дорівнює 1 гігабіту на секунду (Гбіт/с).

– **Hot Spot:** цей тип підключення до Інтернету охоплює менші території.

– **Модем:** це розшифровується як модулятор-демоулятор – електронне обладнання для взаємодії з Інтернетом, що забезпечує передачу даних за

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 16 |

допомогою комп'ютера, смартфона чи інших цифрових пристроїв. Його можна використовувати всередині того самого пристрою, який має маршрутизатор.

– **Маршрутизатор:** це з'єднує вас з модемом і використовує радіохвилі для включення бездротової мережі.

– **VoIP:** протокол голосового зв'язку (VoIP) – це технологія або програма, яка дозволяє здійснювати телефонні дзвінки або відповідати на них за допомогою широкосмугової служби, а не телефонної лінії.

– **Регулювання:** це подія, яка виникає, коли ваш інтернет-провайдер навмисно знижує швидкість вашого інтернет-з'єднання. Це може бути з багатьох причин, таких як:

- Досягнення обмеження даних.
- Велике використання даних у години пік, як-от вечір, вихідні та свята.
- Якщо інтернет-провайдер підозрює незаконну діяльність, як-от обмін файлами, торрент-завантаження тощо..
- Інтернет-провайдер вирішує зменшити швидкість сайту (наприклад, оптимізовану платформу), часто для максимізації загальної пропускної здатності.

Як вибрати хорошого інтернет-провайдера

Щоб мати відмінне підключення до Інтернету та послуги, ви повинні уважно вибирати провайдера. Ось деякі моменти, на які варто звернути увагу.

Тип підключення

Тип підключення до Інтернету – це перше, на що ви повинні звернути увагу, коли ви обираєте провайдера. Як обговорювалося вище, провайдери Інтернету бувають різних типів, які пропонують різні типи підключення до Інтернету, такі як широкосмугове, волоконно-оптичне, кабельне, комутоване, супутникове, DSL, мідне, бездротове тощо. Якщо вам потрібні неймовірно високі швидкості, ви можете вибрати оптоволокно або широкосмуговий кабель через кабель або мідь. Однак вам доведеться підтвердити, що він підтримується у вашому регіоні, про що ми й поговоримо далі.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 17 |

Доступність

Якщо ви перебуваєте в міській місцевості, ви можете вибрати будь-яке підключення залежно від використання, бюджету та вимог.

Однак, якщо ви родом із сільської місцевості, вибір наразі, на жаль, обмежений. Доступність буде важливо дослідити перед остаточним вибором провайдера.

Перевірте, який тип підключення до Інтернету доступний у вашому регіоні. Домівки та підприємства в сільській місцевості мають небагато варіантів, як-от деякі широкосмугові з'єднання від таких компаній, як Xfinity та AT&T, супутникове з'єднання, мережа 4G-LTE і навіть телефонне з'єднання.

Отже, перевірте, що доступно у вашому регіоні, а потім зверніться до відомого провайдера відповідно до ваших потреб.

Швидкість провайдера

Наступний важливий фактор – швидкість. Якщо ви керуєте бізнесом, вам потрібна виняткова швидкість, щоб усі ваші операції, системи та служби працювали безперебійно в будь-який час. Належний інтернет-провайдер забезпечить постійну роботу вашої хмарної служби резервного копіювання, щоб ви не втратили жодних даних через неочікуваний збій

Швидкість знову залежить від вашого географічного розташування. Міські споживачі можуть навіть торкнутися тисячі Мбіт/с, тоді як сільські споживачі можуть застрягти на кількох Мбіт/с.

Отже, не просто дивіться на рекламовану швидкість і пропускну здатність у вашому плані; перевірте це самі. Ви можете провести тести швидкості, щоб переконатися, на яку швидкість ви можете розраховувати у вашому регіоні, перш ніж заплатити значні суми не тому постачальнику і в кінцевому підсумку розчаруватися.

Крім того, запитайте про швидкість завантаження та завантаження, оскільки вони різні. Якщо ви ведете бізнес, вам потрібна висока швидкість як завантаження, так і завантаження. Це дозволить вам швидше переглядати веб-

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 18 |

сторінки, обмінюватися інформацією, вносити зміни на свій сайт або програму, приєднуватися до відеоконференцій тощо.

Однак, якщо ви окремий користувач, який користується Інтернетом для відпочинку, вам може знадобитися більша швидкість завантаження, ніж завантаження, оскільки вам, імовірно, знадобиться більша швидкість для завантаження інформації під час перегляду веб-сторінок, ігор, потокового відео тощо, ніж ви самі будете надсилати інформацію.

Вартість

Вартість завжди є вирішальним фактором, коли ви інвестуєте в щось. Якщо ви малий бізнес або окремий споживач, ви можете шукати доступні плани, які забезпечать достатню пропускну здатність і швидкість для виконання вашої роботи.

Однак, якщо ви є великою компанією, і вам постійно потрібне високошвидкісне з'єднання, щоб підтримувати роботу, вам найкраще підійде надійна мережа, пропускну здатність і швидкість. Якщо ви потрапляєте в цю категорію, шукайте найкращих постачальників, які можуть запропонувати подібні послуги, але майте на увазі, що це, ймовірно, матиме вищу ціну

Щоб зробити найкращий вибір, порівняйте ціни та характеристики різних провайдерів, доступних у ваших регіонах, які пропонують потрібний вам тип підключення. Виберіть той, який забезпечує найкращий баланс між ціною та характеристиками.

Чіткість обслуговування

Крім того, підключення до Інтернету від відомого провайдера матиме чіткі угоди про рівень обслуговування (SLA), які визначають відповідальність провайдера та чого очікувати, а чого ні. Це рятує вас від плутанини чи тертя. Вони окреслять такі речі, як:

- Встановлення швидкості завантаження та завантаження.
- Забезпечення відсутності постійних затримок у наданні послуг.
- Тестування швидкості передачі пакетів.

| | | | | | | | | |
|------|------|----------|--------|------|--|--|----------------------------------|------|
| | | | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | | | 19 |

- Захист мережі.
- Обмеження того, що ви не повинні робити в мережі.
- Чіткі умови оплати.
- Час, коли вони можуть відключити ваші послуги.
- Протоколи відновлення послуг.

І більше...

Угода про рівень обслуговування (SLA) устанавлює угоду між вами та вашим постачальником послуг. У ньому будуть описані всі необхідні деталі вашої послуги, такі як час безвідмовної роботи, втрата пакетів, затримка, час відповіді тощо. Це підвищує прозорість між обома сторонами та повідомляє вам, які послуги ви отримаєте без будь-яких прихованих планів від постачальника послуг.

Однак багато компаній цього не надають. І якщо ви користуєтеся послугами таких компаній, ви можете заплатити більше, ніж вимагали спочатку. Вони можуть попросити сплатити приховані платежі, зв'язати послугу або інші сюрпризи, на які ви не реєструвалися.

Тому завжди шукайте угоди про рівень обслуговування, уважно їх читайте, а потім купуйте послуги у них.

FUP

Політика добросовісного використання (FUP) – поширений термін, з яким ви, можливо, стикалися самі. Це пов'язано із справедливим спільним використанням ресурсів підключення до Інтернету.

Ви коли-небудь відчували зниження швидкості Інтернету після використання великого обсягу даних?

Якщо так, то ви стикалися з троттлінгом.

Постачальники Інтернет-послуг можуть пропонувати необмежені плани Інтернету, але знижують швидкість, якщо ви споживаєте великий обсяг даних відповідно до вашого тарифу Інтернету. Це тому, що ваш провайдер матиме обмежену пропускну здатність. Отже, якщо один користувач використовує всю швидкість, інші користувачі зіткнуться з поганим користуванням Інтернетом.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 20 |

З цієї причини провайдер знижує вашу швидкість, щоб підтримувати баланс, тож кожен матиме справедливу частку використання Інтернету. Тим не менш, це рідко впливає на звичайних користувачів Інтернету, навіть на тих, хто багато годин поспіль дивиться відео або гортає соціальні мережі.

Підтримка клієнтів

Підтримка клієнтів – це фактор, до якого не слід ставитися легковажно. Якщо ваше підключення до Інтернету часто переривається або не працює протягом кількох годин, це може мати руйнівний вплив на ваш бізнес. У цьому сучасному діловому середовищі, наповненому конкуренцією, ви не можете дозволити собі залишатися офлайн, поки ваші клієнти борються. Але все може піти не так у будь-який момент. Якщо проблема перевищує ваші можливості швидко вирішити, вам знадобиться швидка допомога, тому наявність команди клієнтів, яка чуйно реагує, окупається.

Шукайте постачальників послуг, які можуть запропонувати відмінне обслуговування клієнтів через різні канали, такі як чат, електронна пошта та телефон, з досвідченими, але доброзичливими та професійними агентами, які зможуть вирішити ваші запити за короткий час. Використовуйте такі сайти, як Reddit, Quora та Google, щоб дослідити репутацію провайдера та дізнатися, як їхні клієнти ставляться до їхніх послуг, перш ніж остаточно прийняти рішення.

Як знайти свого провайдера?

Якщо ви не знаєте, хто є вашим поточним постачальником послуг Інтернету, перевірка швидкості провайдера допоможе вам дізнатися. Тест відобразить ім'я вашого провайдера та швидкість завантаження та завантаження. Для цього ви можете спробувати інструменти перевірки швидкості ISP від Ookla.

Крім того, в Інтернеті є багато сайтів, таких як whoismyisp.org, на яких можна знайти свою IP-адресу, на більшості з яких також відображається ім'я вашого провайдера.

Поширені запитання провайдера

Ось кілька типових запитань і відповідей про Інтернет-провайдерів.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 21 |

1. Яку інформацію може бачити мій провайдер?

Ваш провайдер знатиме вашу IP-адресу. Таким чином, він може відстежувати ваші кроки в Інтернеті. Це включає вашу історію веб-перегляду, як-от контент, який ви використовуєте, сайти, які ви відвідуєте, тощо.

Закони про збереження даних дозволяють провайдерам відстежувати свою клієнтську базу. Фактично, деякі країни навіть дозволяють продавати дані користувачів третім особам.

Використання проксі-сервера або VPN може допомогти уникнути цієї проблеми, маскуючи вашу справжню IP-адресу іншою IP-адресою. Ви також можете розглянути можливість переходу на безпечний, орієнтований на конфіденційність веб-переглядач і пошукову систему, щоб замести сліди.

2. Яка різниця між національним і місцевим Інтернет-провайдером?

Національний провайдер Інтернету надає послуги в Інтернеті споживачам, які знаходяться в різних частинах країни. Вони мають потужну інфраструктуру та можливості для задоволення потреб клієнтів за конкурентними цінами. Вони надійні, але споживачам із деяких сільських районів може бути важко отримати до них доступ.

З іншого боку, місцеві інтернет-провайдери пропонують підключення до Інтернету та послуги для невеликої кількості місцевих територій, включаючи сільські регіони. Вони також мають надійні можливості для задоволення потреб клієнтів, але їхні зони обслуговування відносно менші, ніж у національних провайдерів.

3. Скільки коштують послуги Інтернету?

Вартість інтернет-послуг відрізняється від провайдера до провайдера. Вартість може відрізнятись залежно від постачальника послуг, швидкості тарифного плану, місцезнаходження, типу підключення, функцій тощо. Плани з розширеним підключенням, як-от оптоволокно та широкопasmовий доступ, можуть коштувати чималі копійки. Подібним чином підключення до Інтернету з наймовірно високою швидкістю буде коштувати дорожче, ніж підключення до

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 22 |

Інтернету з середньою або низькою швидкістю.

У середньому послуги оптоволоконного, супутникового або кабельного Інтернету можуть коштувати від 20 до сотень доларів на місяць, тоді як DSL коштує менше – у середньому близько 50 доларів на місяць.

4. Яке обладнання мені потрібно для доступу до Інтернету?

Основним необхідним обладнанням є модем для зв'язку з Інтернетом. Окрім цього, вам знадобиться цифровий пристрій, як-от ноутбук, смартфон, ігрова консоль тощо, із можливістю підключення до Інтернету.

Для бездротових пристроїв, таких як планшети чи смартфони, вам знадобиться маршрутизатор Wi-Fi. Зараз використовується шлюз, у якому розміщуються як маршрутизатор, так і модем.

Нарешті, вам знадобиться коаксіальний кабель або кабель Ethernet для підключення пристроїв до маршрутизатора.

В інших налаштуваннях може знадобитися більше обладнання, наприклад:

- Тарілка для підключення до супутника.
- Термінал оптичної мережі (ONT) у випадку волокна.
- Лінійні фільтри у випадку DSL.

Ви повинні подякувати постачальникам послуг Інтернету за те, що вони відкрили для вас величезний світ Інтернету, наповнений можливостями та розвагами. Вони допомагають вам підтримувати ваш бізнес, пропонуючи підключення до Інтернету та послуги, щоб ви могли безперервно обслуговувати своїх клієнтів. Будь то покупки в Інтернеті, спілкування з друзями з усього світу в соціальних мережах або гра в онлайн-ігри, інтернет-провайдери надають вам можливості, які допоможуть вам безпечно, швидко та надійно переглядати Інтернет. Отже, коли ви обираєте провайдера, зверніть увагу на тип підключення, доступний у вашій місцевості, швидкість і пропускну здатність, підтримку клієнтів, SLA та інші пункти, виділені вище, щоб переконатися, що ви вибрали правильний виходячи з ваших потреб і бюджету.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 23 |

скрипт виконується на сервері, а клієнтові передається результат його роботи, тоді як в JavaScript, код повністю передається на клієнтську машину і лише там виконується браузером.

При написанні на інших мовах програмування, наприклад, на Perl або C – замість того, щоб створювати програму, яка займається формуванням HTML-коду і містить незліченну безліч призначених для цього команд, PHP-програміст має можливість створювати HTML-код з декількома упровадженими командами PHP. Код PHP відділяється спеціальними початковим і кінцевим тегами, які дозволяють процесору PHP визначати початок і кінець ділянки HTML-коду, що містить PHP-скрипт.

Значною відмінністю PHP від якого-небудь коду, що виконується на стороні клієнта, наприклад, Javascript, є те, що PHP-скрипти виконуються на сервері. PHP-скрипт розміщений на сервері, клієнт отримує тільки результат виконання скрипта, причому клієнт не має можливості з'ясувати, який саме код виконується. Існує також можливість конфігурувати власний сервер так, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не в змозі дізнатися, чи отримують вони звичайний HTML-файл, чи це результат виконання скрипта.

Мова програмування PHP досить проста для освоєння, але разом з тим здатна задовольнити запити професійних програмістів. Хоча PHP, головним чином, призначений для роботи в середовищі web-серверів, область його застосування не обмежується тільки цим.

Можливості мови PHP дуже великі. Головним чином, область застосування PHP сфокусована на написання скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти дані форм, генерувати динамічні сторінки або посилати і приймати cookies. Окрім вищенаведеного PHP здатний виконувати і безліч інших завдань.

Існують три основні області, де використовується PHP.

1. Створення скриптів для виконання на стороні сервера. Саме таким чином мова програмування PHP найширше використовується. Все, що може знадобитися програмісту – це парсер PHP (у вигляді програми CGI або серверного модуля), вебсервер і браузер. Для того, щоб існувала можливість проглядати результати виконання PHP-скриптів у браузері, потрібен працюючий веб-сервер і встановлений PHP.

2. Створення скриптів для виконання в командному рядку. Існує можливість створити PHP-скрипт, здатний запускатися незалежно від веб-серверу та браузера. Все, що потрібно – це парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою cron (на платформах Unix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в завданнях простої обробки даних.

Існує величезна кількість документації і списків розсилки, до яких можна звернутися у разі виникнення яких-небудь питань.

Perl – високорівнева, інтерпретована, динамічна мова програмування загального призначення. Perl запозичує можливості з багатьох інших мов програмування, як то C, shell scripting, AWK та sed. Мова надає потужні можливості для обробки тексту без довільних обмежень на довжину даних багатьох сучасних інструментів Unix, полегшуючи процес маніпуляції текстових файлів. Використовується для програмування графіки, системного адміністрування, у мережному програмуванні, у написанні програмного забезпечення, яке взаємодіє з базами даних, у програмуванні CGI для веб. Perl за свою гнучкість і потужність отримав прізвисько «швейцарського армійського ножа мов програмування».

Perl – мова програмування загального призначення, котра на початку розроблялась, як інструмент для обробки тексту, і тепер використовується для вирішення дуже широкого кола завдань, включно із системним

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 26 |

адмініструванням, веб-розробкою, розробкою мережного програмного забезпечення, та програмного забезпечення з графічним інтерфейсом користувача.

Мова віддає перевагу практичному (простота у використанні, ефективність, повнота) над гарним (крихітність, елегантність, мінімалістичність). Вона має багато можливостей, включно із підтримкою кількох парадигм програмування(процедурне програмування, об'єктно-орієнтоване програмування, а також функціональне програмування), управлінням пам'яттю, вбудованою підтримкою системи обробки текстів, і величезною колекцією сторонніх модулів.

Загальна структура програм на мові Perl була запозичена у C. Perl є типовою процедурною мовою, зі змінними, виразами, присвоюваннями, дужками-розділювачами, блоками, циклами та підпрограмами.

Perl також запозичив деякі особливості з shell-програмування. Назви усіх змінних розпочинаються зі спеціальних знаків, кожен з яких характеризує певний тип даних (наприклад, скаляри, масиви та хеш-таблиці). Важливо зазначити, що це дозволяє правильно інтерпретувати назви змінних безпосередньо у текстових рядках. Perl має багато вбудованих функцій, які забезпечують програміста інструментарієм, котрий часто використовується у shell-програмуванні (хоча, багато з цих інструментів реалізовується за допомогою зовнішніх інструментів), як то сортування або виклик зовнішніх програм.

Perl запозичив списки з мови Lisp, асоціативні масиви (хеші) з AWK, і регулярні вирази з sed. Це спростило процес парсингу і обробки великих обсягів текстової інформації.

У Perl 5, була додана підтримка комплексних структур даних, функції першого класу, і об'єктно-орієнтованої моделі програмування. Що включає у себе посилання, пакунки, та інше. Серед найбільших нововведень у Perl 5 слід відмітити наявність модулів, які дозволяють повторно використовувати один і той же код.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 27 |

Усі версії Perl здійснюють автоматичне типізування змінних і управління пам'яттю. Інтерпретатор знає про тип та способи зберігання усіх об'єктів даних у програмі.

Дизайн Perl можна описати, як відповідь на три основні тенденції в комп'ютерній індустрії: зниження вартості апаратного забезпечення, зростання вартості робочої сили, а також вдосконалення технології компіляторів. Багато ранніх мов програмування, як то Fortran та C, були розроблені таким чином, щоб якомога ефективніше використовувати на той час дороге апаратне забезпечення. У протилежність, Perl було розроблено для підвищення ефективності роботи дорогих в наш час програмістів.

Perl має багато можливостей, які збільшують ефективність програміста за рахунок інтенсивного використання мікропроцесора та великих обсягів оперативної пам'яті. Серед них: автоматичне управління пам'яттю; динамічна типізація; стрічки, списки, та хеші; регулярні вирази; самоаналіз; та функція eval().

Perl підтримує мовні конструкції, які є короткими і природними для людей при їхньому читанні і написанні, навіть якщо це ускладнює реалізацію інтерпретатора Perl.

Синтаксис Perl має багато спільного з синтаксисом мов Cі, Awk, Sed і Shell. Перший рядок початкового коду може починатися з «#!/Шлях/до/Perl [-ключі]» – що указує системі шлях до інтерпретатора Perl для виконання програми в *NIX системах і виконання їх на Веб-сервері.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи надання доступу до мережі Internet сервіс провайдера.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 28 |

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 29 |

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

У наступному розділі докладніше буде розглянуто технології доступу.

Кабель

Послуги кабельного Інтернету надаються через ті самі коаксіальні кабелі, які спочатку були встановлені для передачі аналогового відео на телевізори. Більшість провайдерів кабельного Інтернету використовують стандарт під назвою DOCSIS (специфікація інтерфейсу служби передачі даних через кабель), який є міжнародним телекомунікаційним стандартом, який дозволяє додавати передачу даних із високою пропускнуою здатністю до існуючої системи коаксіального кабельного телебачення.

Переваги

Широко розгортається в міських і приміських зонах, може досягати швидкості до 10 Гб вниз і 2 Гб вгору за допомогою наявних кабелів

Недоліки

Через обмежену загальну ємність провайдери кабельного телебачення зазвичай пропонують набагато нижчу швидкість завантаження та завантаження. Небагато провайдерів використовують коаксіальний кабель для нових розгортань, оскільки оптоволокно має значно більшу пропускну здатність

Стільниковий (включаючи 5G)

Послуга стільникового зв'язку – це технологія зв'язку, у якій остання лінія зв'язку забезпечується бездротовим зв'язком, а територія землі, яка має забезпечуватися радіозв'язком, поділяється на стільники відповідно до рельєфу та характеристик прийому. Є чотири основні стільникові технології, які передають дані; широкосмуговий доступ до 2G, 3G, 4G і 5G.

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 30 |

Переваги

Вежі охоплюють широкі території; мобільність означає, що ви можете взяти свій пристрій куди завгодно, 5G запропонує вищі швидкості в майбутньому

Недоліки

Послуги, як правило, мають обмеження на передачу даних (навіть «необмежені» плани), покриття в сільській місцевості та швидкість можуть бути нерівними, 5G навряд чи вирішить розрив широкопasmового зв'язку в сільській місцевості

Супутник

Супутниковий доступ до Інтернету забезпечують супутники зв'язку на орбіті Землі. Традиційні супутники зв'язку запускаються на геосинхронну орбіту на висоту 22 236 миль, що передбачає затримку в обігу приблизно півсекунди для даних, що проходять через супутник.

Переваги

Покриття майже скрізь; такі компанії, як Space X, Amazon і OneWeb, працюють над запуском супутників на низьку навколоземну орбіту для боротьби з проблемами затримки

Недоліки

Традиційні супутники мають високу затримку та низькі обмеження даних; Супутники на низькій навколоземній орбіті – амбітні проекти з невизначеним майбутнім

Стаціонарний бездротовий зв'язок

Стаціонарний бездротовий зв'язок – це робота пристроїв або систем бездротового зв'язку, які використовуються для з'єднання двох фіксованих місць, наприклад між вежею та будівлею. Хоча зараз компанії стільникового зв'язку пропонують послуги, які відповідають цьому визначенню, технології стільникового зв'язку 2G/3G/4G/5G зазвичай не можна класифікувати як фіксований бездротовий зв'язок, оскільки стільникова технологія була розроблена для мобільних пристроїв і має значні обмеження при використанні

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 31 |

для фіксованих місць. Нестільниковий фіксований бездротовий зв'язок може використовувати різні технології та частоти.

Переваги

Найменш дорогий варіант розгортання; більшість частот фіксованого бездротового зв'язку, не пов'язаних зі стільниковим зв'язком, забезпечують прийнятне обслуговування на відкритих площах

Недоліки

Швидкість набагато нижча, ніж у послуг дротового зв'язку; частоти, що використовуються, можуть забезпечити низьку швидкість і покриття в районах Мічигану з значним рельєфом або лісовим покривом

DSL

Послуга цифрової абонентської лінії (DSL) – це сімейство технологій, які використовуються для підключення до Інтернету через телефонні лінії, спочатку призначені для аналогового голосу.

Переваги

Послуга надається за допомогою стандартних телефонних ліній, що робить її широкодоступною

Недоліки

Доступні швидкості значною мірою залежать від відстані від точки розподілу та якості телефонної лінії, тому DSL часто дуже повільний у сільській місцевості

Волоконно-оптичні

Оптоволокно до дому – це найшвидший спосіб надати послуги Інтернету для проживання та бізнесу. Магістралі Інтернету використовують оптоволоконний кабель як з'єднання «останньої милі», оскільки він усуває будь-які вузькі місця між кінцевим користувачем та магістралями Інтернету. Усі інші технології доступу до Інтернету, включаючи кабельний, DSL, стільниковий зв'язок, фіксований бездротовий зв'язок і навіть супутник, використовують волоконно-оптичні кабелі для передачі даних, доки це волокно не досягне їх

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 32 |

інфраструктури.

Переваги

Оптоволокно – це найшвидша технологія передачі даних, яка підтримує кілька терабіт пропускної здатності в одному кабелі та пропонує надійність у майбутньому на десятиліття вперед

Недоліки

Розгортання оптоволокна вимагає прокладання нових кабелів на стовпах або під землею, що є дорогим завданням

Технології стільникових даних

І 3G, і 4G використовувалися для надання домашнього широкосмугового зв'язку. Однак бізнес-моделі чотирьох основних операторів стільникового зв'язку в США – Verizon, AT&T, Sprint і T-Mobile – розподіляють дані таким чином, що робить використання цих послуг для основного домашнього підключення до Інтернету дорогим або неможливим. Хоча всі оператори пропонують певну форму послуги, яка називається необмеженою, усі ці плани мають обмеження щодо використання даних на пристрої з точкою доступу або на пристроях, «підключених» до смартфона. Після певного обсягу даних з клієнта або стягується плата за одиницю використаних додаткових даних, або втрачається пріоритет (уповільнюється, коли стільникова мережа зайнята) або регулюється (обмежується меншою швидкістю, наприклад 0,6 Мб).

Багато операторів стільникового зв'язку розгортають послуги 5G. Незважаючи на те, що 5G покращить послуги стільникового зв'язку в сільській місцевості, це навряд чи стане срібною кулею для сільського широкосмугового зв'язку. Важливо розуміти, що 5G фактично охоплює дві основні технології:

- 1) Удосконалення традиційної стільникової служби.
- 2) Високошвидкісна послуга «міліметрової хвилі».

Спектри міліметрових хвиль обіцяють забезпечити набагато вищі швидкості – до 1 Гбіт/с – але мають набагато «коротший» діапазон, ніж типові стільникові сигнали. Це означає, що пристрої мають бути набагато ближче до веж

стілньикового зв'язку, щоб отримувати сигнали, і ці вежі мають бути під'єднані до оптоволоконної мережі.

Хоча ця технологія буде корисною в міських умовах, у сільській місцевості міліметрові хвилі 5G не є значно дешевшими, ніж оптоволокло до бордюру, і не змінюють фінансову модель для розгортання в сільській місцевості.

Двома головними удосконаленнями традиційних послуг стільникового зв'язку, які принесе 5G, є можливість для одного стільникового вузла мати набагато більше індивідуальних з'єднань і можливість збільшити швидкість на одному пристрої, підключивши його до кількох стільникових сайтів. Оскільки сільські сайти стільникового зв'язку знаходяться в районах з низькою щільністю, підключення до кількох окремих пристроїв не є великою потребою, і ймовірність того, що один пристрій буде в зоні дії кількох сайтів стільникового зв'язку, низька. Таким чином, хоча 5G, ймовірно, значно покращить послуги стільникового зв'язку в міських районах, навряд чи вирішить проблему широкосмугового доступу в сільській місцевості.

1. Супутник

Супутниковий доступ до Інтернету забезпечують супутники зв'язку на орбіті Землі. Традиційні супутники зв'язку запускаються на геосинхронну орбіту на висоту 22 236 миль, що передбачає затримку в обігу приблизно півсекунди для даних, що проходять через супутник. Це робить супутникове обслуговування проблематичним для додатків у реальному часі, таких як голосовий або відеочат та ігри.

В останні роки такі компанії, як SpaceX, Amazon і OneWeb, запропонували запустити велику кількість супутників на низькій навколоземній орбіті, щоб вирішити ці проблеми. Ці супутники знаходяться на висоті лише кілька сотень миль і тому можуть передавати дані з низькою затримкою. Ці амбітні проекти потребують запуску у багато разів більшої кількості супутників, які зараз знаходяться на орбіті. Вони можуть бути успішними, але до такого успіху можуть залишитися роки, а прогнозовані швидкості залишатимуть сільських жителів на

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 34 |

порядок чи більше позаду їхніх сусідів із дротовим доступом. SpaceX почала запускати пілотні супутники в 2019 році.

2. Стационарний бездротовий зв'язок

Стационарний бездротовий зв'язок – це робота пристроїв або систем бездротового зв'язку, які використовуються для з'єднання двох фіксованих місць, наприклад між вежею та будівлею. Хоча зараз компанії стільникового зв'язку пропонують послуги, які відповідають цьому визначенню, технології стільникового зв'язку 2G/3G/4G/5G зазвичай не можна класифікувати як фіксований бездротовий зв'язок, оскільки стільникова технологія була розроблена для мобільних пристроїв і має значні обмеження при використанні для фіксованих місць. Нестільниковий фіксований бездротовий зв'язок може використовувати різні технології та частоти.

Однією спільною є те, що частоти, які використовуються для більшості нестільникових фіксованих бездротових мереж, мають проблеми з пересуванням по місцевості та листям, тобто вони можуть забезпечити чудове обслуговування на рівнинних відкритих територіях, але можуть бути викликані в районах Мічигану з пересіченою місцевістю або значним деревним покривом.

3. DSL

Послуга цифрової абонентської лінії – це сімейство технологій, які використовуються для підключення до Інтернету через телефонні лінії, спочатку призначені для аналогового голосу. Послуги DSL можуть надаватися одночасно з послугами дротового телефонного зв'язку на одній телефонній лінії, оскільки DSL використовує для передачі даних більш високі діапазони частот. На території клієнта DSL-фільтр на кожній розетці, що не підтримує DSL, блокує будь-які високочастотні перешкоди, щоб забезпечити можливість одночасного використання голосових і DSL-служб. Більшість DSL, розгорнутих у Сполучених Штатах, є формою асиметричного DSL (ADSL), де швидкість завантаження нижча за швидкість завантаження.

Основна проблема з DSL полягає в тому, що через те, що сигнал

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 35 |

передається через неекрановані дроти, він піддається погіршенню сигналу, а доступні швидкості значною мірою залежать від відстані від точки розповсюдження – як правило, від мультиплексора доступу DSL або DSLAM.

4. Волоконно-оптичні

Оптоволокно до дому – це найшвидший спосіб надати послуги Інтернету для проживання та бізнесу. Магістралі Інтернету, які використовують волоконно-оптичний кабель як з'єднання «останньої милі», усувають будь-які вузькі місця між кінцевим користувачем та магістралями Інтернету. Усі інші технології доступу до Інтернету, включно з кабельним, DSL, стільниковим зв'язком, фіксованим бездротовим зв'язком і навіть супутниковим зв'язком, використовують волоконно-оптичні кабелі для передачі даних, поки це волокно не досягне їх інфраструктури. Волоконно-оптичні кабелі використовують світло для передачі даних замість електромагнітних сигналів. Результатом цього є набагато більша швидкість – 1 Гбіт, 10 Гбіт і 100 Гбіт – звичайні для волоконно-оптичних мереж – на відстанях у багато миль.

Поточний рекорд швидкості за стандартним волоконно-оптичним кабелем належить компанії NEC, яка надіслала понад 100 Тб (100 000 Гб) на 165 км.²⁶

Компанії, які продають оптоволоконний широкопугувий зв'язок, часто називають себе «100% оптоволоконними мережами». Цей термін вводить в оману. FCC визнає кілька рівнів послуг оптоволоконного широкопугувого зв'язку, і більшість переходить на іншу технологію, як-от коаксіальний кабель або навіть мідні телефонні лінії, у певний момент між офісом постачальника послуг Інтернету та роз'ємами домашнього модема. Тільки оптоволокно до дому (FTTH) є справді 100% оптоволоконом.

Мережі "волокно до дому" сьогодні зазвичай розгортаються з використанням технології гігабітної пасивної оптичної мережі (GPON) або технології активного Ethernet. Обидві технології мають різні переваги. GPON не потребує спеціального волокна для кожного клієнта, і його економічніше розгортати на великій території. Активна мережа Ethernet використовує окреме

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 36 |

волокно для кожного клієнта, тому швидкість для кожного клієнта обмежена лише електронікою на кожному кінці.

На практиці багато спільнот розгортають GPON з кількома додатковими волокнами, щоб забезпечити можливість мати виділені волокна для кількох великих клієнтів, якщо вони з'являться. Для GPON пропускна здатність одного з'єднання становить 2,4 Гб униз, 1,2 Гб угору, спільно використовується між 128 клієнтами. NGPON2 є відносно новим стандартом, який дозволяє до 10 Гб симетричних з'єднань для кожного клієнта, підключеного до PON.

3.2 Розробка структурної схеми

Чим швидше ваше підключення до Інтернету, тим краще, і це особливо вірно, якщо ви є одним із дедалі більшої кількості віддалених працівників – службовців, фрілансерів і власників бізнесу – які працюють поза домом.

Коли ви працюєте вдома, ви повинні працювати так само добре, або краще, ніж ваші офісні та віддалені колеги. Швидкий Інтернет-сервіс допоможе швидше виконувати роботу. Повільне підключення до Інтернету призводить до зниження продуктивності через час, витрачений на очікування завантаження та завантаження інформації.

Порівняйте швидкість завантаження та завантаження

Технологія пройшла довгий шлях від часів комутованих послуг від таких компаній, як AOL, Prodigy та CompuServe. Кабельні, волоконно-оптичні та супутникові послуги тепер конкурують із (і загалом надають набагато швидші послуги, ніж) телефонні та DSL-послуги. Ці компанії пропонують аналогічні швидкості передачі даних і послуги за конкурентними цінами.

Вибираючи інтернет-провайдера, порівняйте ціни конкурентів на тарифні плани з аналогічною швидкістю. Наприклад, якщо телефонна компанія пропонує тарифний план із швидкістю завантаження 15 Мбіт/с і швидкістю передавання 5 Мбіт/с, порівняйте його з найближчим доступним тарифним планом із такою ж

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 37 |

швидкістю від кабельної компанії.

Ознайомтеся з умовами контракту, ціною пакета та комерційним використанням

Також враховуйте фактори, окрім швидкості.

– **Умови контракту:** багато компаній вимагають щонайменше однорічного контракту, але інші – щомісяця. Деякі стягують комісію за скасування послуги, коли ви змінюєте постачальника.

– **Ціни на пакетні послуги:** деякі компанії пропонують хорошу, але обмежену в часі угоду, якщо ви користуєтеся послугами телефону, Інтернету та телебачення від одного постачальника. Наприклад, ваш постачальник послуг кабельного телебачення може запропонувати спеціальну ціну, якщо ви додасте послугу Інтернет до свого облікового запису.

– **Бізнес-плани та зручність використання:** деякі провайдери пропонують спеціальні швидкості передачі даних, функції електронної пошти, безпеку та ціни для малого бізнесу. Повідомте свого постачальника послуг про свої плани на випадок, якщо послуга має будь-які обмеження щодо використання в бізнесі чи роботі з дому, наприклад, наскрізні VPN під час використання віртуальної приватної мережі для віддаленої роботи.

Порівняйте спеціальні додатки та інші функції

Деякі служби пропонують безкоштовні бездротові маршрутизатори; інші вимагають від вас придбання обладнання. Деякі постачальники послуг також пропонують безкоштовні послуги онлайн-резервного копіювання, розміщення персональних веб-сайтів, підписки на антивірусні програми та інші додаткові компоненти зі своїми інтернет-планами. У порівняння включіть вартість додаткового апаратного забезпечення, програмного забезпечення та послуг.

Корисним і безкоштовним доповненням для мобільних працівників є безкоштовний доступ до точки доступу Wi-Fi, доступний від основних провайдерів у певних зонах обслуговування. Це забезпечує доступ до Інтернету, коли ви перебуваєте вдома та в інших місцях.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 38 |

Дослідіть обслуговування клієнтів і надійність

Важливим показником є надійність. Той самий інтернет-провайдер в одній частині країни може мати кращі чи гірші показники надійності послуг і задоволеності клієнтів в іншій місцевості.

На рисунку 3.1 зображена структурна схема системи. На який детально розглянута розроблена система. Розглянемо роботу розробленої системи на прикладі запиту Інтернету користувача №N.

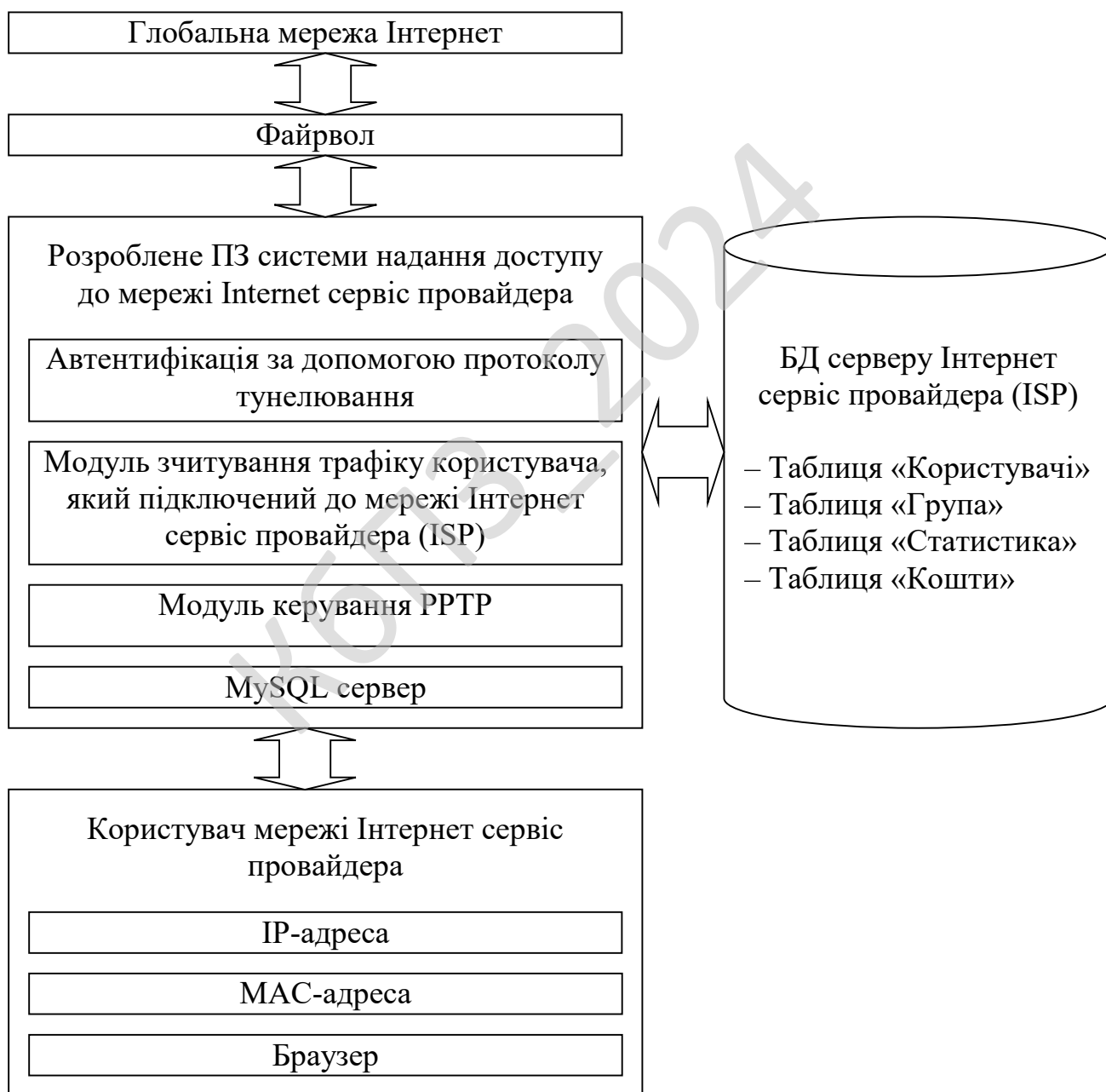


Рисунок 3.1 – Структурна схема системи

Користувач N через мережу через WEB браузер робочої станції посилає запит до мережі Інтернет сервіс провайдера (ISP) з даними IP адреси. На сервері розроблене ПЗ проводить автентифікацію робочої станції за допомогою протоколу тунелювання та далі проводить запит сторінки через брандмауер сервера. В цей час розроблене ПЗ проводить зчитування трафіку мережі, через PPTP, керування відбувається через відповідний модуль. Всі дані роботи користувача у Інтернеті заносяться до бази даних сервера ISP. Основні таблиці це «Таблиця Кошти» де зберігаються дані витрачених коштів, «Таблиця Користувачі» де знаходиться інформація користувача, «Таблиця Статистика» статистичні дані та «Таблиця Група» де зберігаються встановлені привілеї користувача.

Адміністратор ISP за допомогою розробленого бакалаврського ПЗ проводить моніторинг мережі. При необхідності переглядає базу даних. Основне призначення системи це моніторинг трафіку користувачів, які підключені до ISP з можливістю детального перегляду статистики роботи користувачів, розрахунок коштів, інформацію Інтернет серфінгу користувача та ін.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Існують наступні функціональні можливості:

- Облік НТТР– трафіку (керована маршрутизація до проксі-сервера).
- Тарифний облік можливий у двох варіантах: обмеження використання трафіку в годину при оплаті за часом і оплата за фактом використання трафіку й витраченого часу.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|-----------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 40 |



Рисунок 3.2 – Функціональна схема системи

– Підтримка гнучкої цінової політики. Можна встановити незалежну ціну на кожний комп'ютер, у т.ч. залежно від часу доби. Підтримка погодинних і пакетних тарифів (у т.ч. "нічний режим"). Підтримка автоматичних знижок по днях тижня й часу доби. Можливість дати клієнтові безкоштовні додаткові 5 хвилин у випадку апаратних проблем (зберігається в журналі, налаштовується адміністратором).

– Підтримка передоплати й оплати по факту. Можливість створити предоплачений сеанс як на певний час, так на задану суму.

– Можливість "розбивки" залишку засобів по різних комп'ютерах.

– Ведення каси з повним журналом подій. Збереження історії й статистики по кожному менеджері (адміністраторові). Підбиття статистики по кожному окремо взятому комп'ютеру.

– Підтримка дисконтів (у т.ч. обмежених за часом) для постійних клієнтів.

– Можливість задати індивідуальну цінову політику для різних комп'ютерів (у т.ч. виділити VIP і low-end комп'ютери), а також залежно від часу доби.

– Висока стійкість до злому.

– Можливість очистити всі завдання, запущені клієнтом (у т.ч. "завислі" ігри).

– Вбудовані модулі складу й продажів (облік супутніх товарів і послуг Cd-диски, напої і т.д., актуальних для Інтернет-кафе).

– Журналювання всіх входів у систему адміністраторів і менеджерів ISP.

– Можливість "заморожування" сеансу зі збереженням даних користувача й повного "заморожування" його програм.

– "Очищення" сеансу від завислих процесів з комп'ютера менеджера.

– Авто очищення системи ISP від несистемних процесів між сеансами.

– Можливість відправлення повідомлень користувачеві (відображаються в TRAY), робити тихіше звук на його комп'ютері, штрафувати за непристойну поведінку.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 42 |

- Автоматичне завершення небажаних і небезпечних процесів ISP (список налаштовується адміністратором Інтернет-кафе / комп'ютерного клубу).
- Докладне журналювання всіх дій адміністраторів (менеджерів) для хазяїна Інтернет-кафе / комп'ютерного клубу.
- Розблокування комп'ютерів ISP з використанням подвійного коду безпеки (пароля) у випадку порушення зв'язку із сервером.
- Звукові оповіщення про порушення працездатності локальної мережі або зв'язку із сервером.
- Сеанси відновлюються навіть при збої/перезавантаженню сервера або клієнта. Сеанс зникне тільки в тому випадку, якщо одночасно виключити й сервер, і клієнт, але навіть у цьому випадку дані можна буде відновити вручну по журналу подій.
- Комп'ютери можуть працювати в off-line режимі (з виключеним сервером). При цьому фінансові транзакції обробляються коректно.
- За замовчуванням використовує обмежений робочий стіл Windows у якості клієнтської оболонки, що створює додаткові зручності для користувача в силу звичності.
- При бажанні можна замість робочого стола встановити будь-яку іншу оболонку сторонньої розробки.
- Висока чутливість на дії користувача, тому що зв'язок здійснюється по протоколу TCP.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку 3.3. Після початку роботи проходить автентифікація з використанням протоколу тунелювання та після

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 43 |

завантаження демону РРТР з можливістю моніторингу демону РРТР ми потрапляємо до інтерфейсу ПЗ.

Звідки ми можемо потрапити до налаштування ПЗ, статистика роботи та моніторинг локальної мережі з доступом до БД роботи локальної мережі через mySQL сервер.

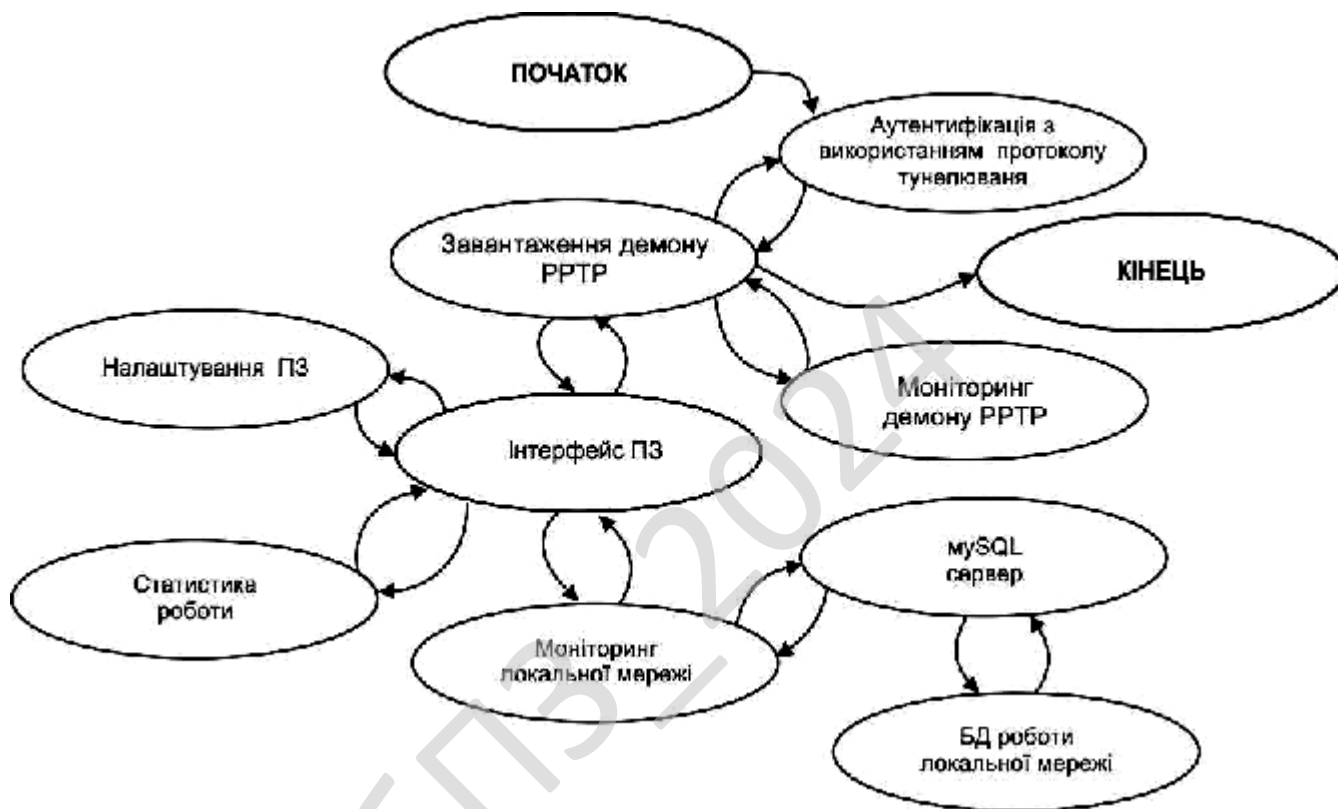


Рисунок 3.3 – Діаграма взаємодії процесів

РРТР це тунельний протокол типу точка-точка, що дозволяє комп'ютеру встановлювати захищене з'єднання з сервером за рахунок створення спеціального тунелю в стандартній, незахищеній мережі. РРТР поміщає (інкапсулює) кадри PPP в IP-пакети для передачі по глобальній IP-мережі, наприклад Інтернет.

РРТР може також використовуватися для організації тунелю між двома локальними мережами. РРТР використовує додаткове TCP-з'єднання для обслуговування тунелю.

Протоколи тунелювання потрібні тому, що деякі протоколи не можуть маршрутизуватися через певні мережі. Кадри PPP, які не маршрутизуються через мережу Інтернет, самостійно ці кадри не можуть знайти правильний маршрут через Інтернет до одержувача, для цього їм потрібна підтримка. Саме для цього і потрібні протоколи тунелювання. Вони дозволяють проводити інкапсуляцію пакетів PPP, тим самим кадри правильно доходять до одержувача.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ_2024

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 45 |

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему програми. Її робота складається з виконання наступних кроків:

- Ініціалізація серверу.
- Підключення та запуск на виконання системних скриптів.
- Створення скрипкового каналу на стороні серверу.

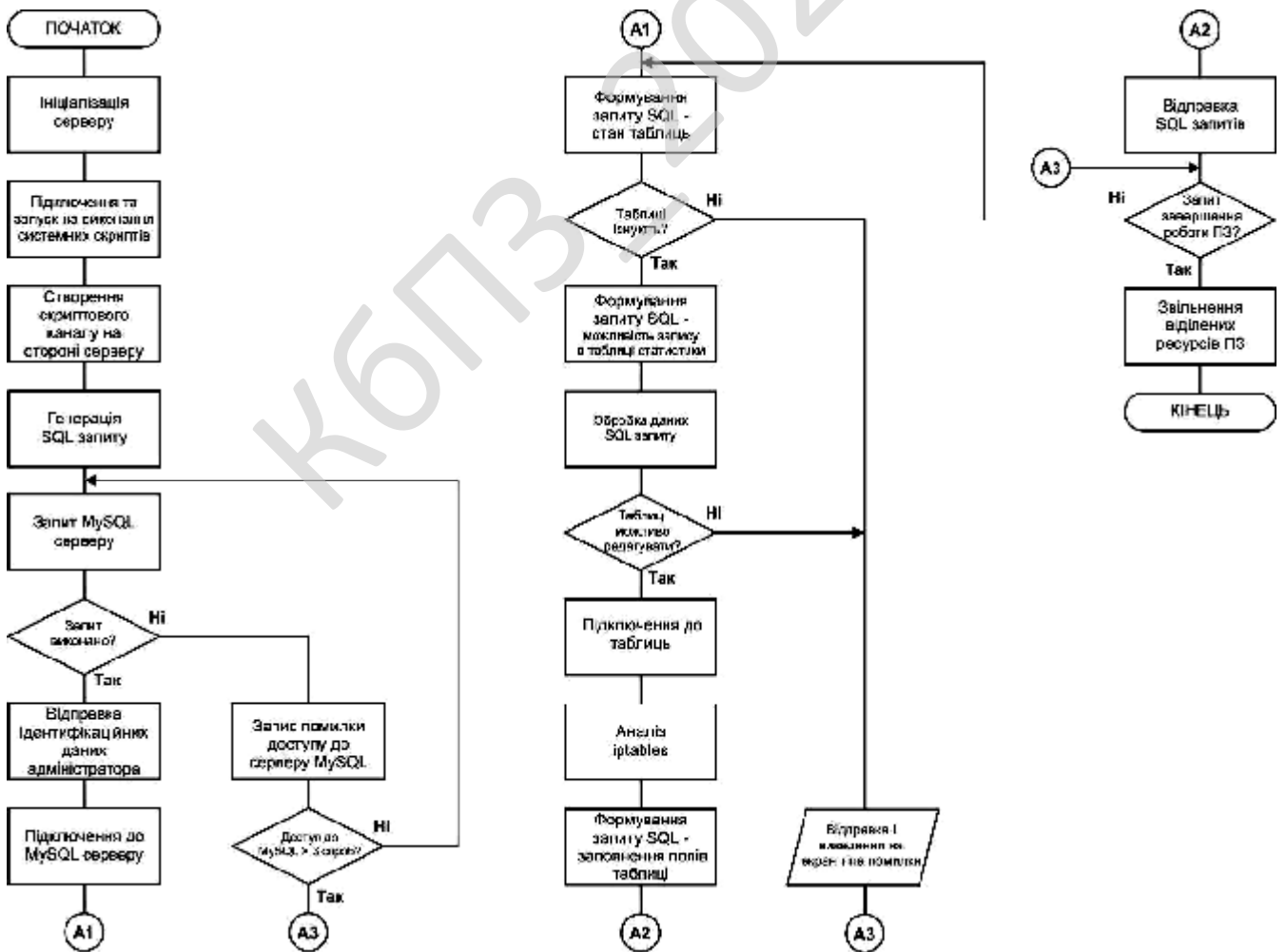


Рисунок 4.1 – Блок-схема програми

- Генерація SQL запиту.
- Запит MySQL серверу (запит).
- Відправка ідентифікаційних даних адміністратора.
- Підключення до MySQL серверу.
- Формування запиту SQL – стан таблиць (запит).
- Формування запиту SQL – можливість запису в таблиці статистики.
- Обробка даних SQL запиту.
- Запит таблиці можливо редагувати?.
- Підключення до таблиць.
- Аналіз iptables.
- Формування запиту SQL – заповнення полів таблиці.
- Відправка SQL запитів.
- Запит завершення роботи ПЗ – звільнення виділених ресурсів ПЗ.

На рисунку 4.2 наведено блок-схему інтерфейсу клієнтського ПЗ. Її робота складається з виконання наступних кроків:

- Ініціалізація клієнтського ПЗ.
- Підключення та запуск на виконання системних скриптів.
- Створення скрипкового каналу на стороні клієнта.
- Генерація SQL запиту.
- Запит MySQL серверу.
- Відправка ідентифікаційних даних користувача ПК.
- Підключення до MySQL серверу.
- Формування запиту SQL – доступ до читання таблиці статистики (запит).
- Підключення до таблиць.
- Формування запиту SQL – статистика по користувачу.
- Формування коду нижньої панелі.
- Формування коду лівої панелі.
- Формування коду верхньої панелі.
- Створення HTML коду сторінки.

- Відправка даних користувачу.
- Запит завершення роботи ПЗ – звільнення виділених ресурсів ПЗ.

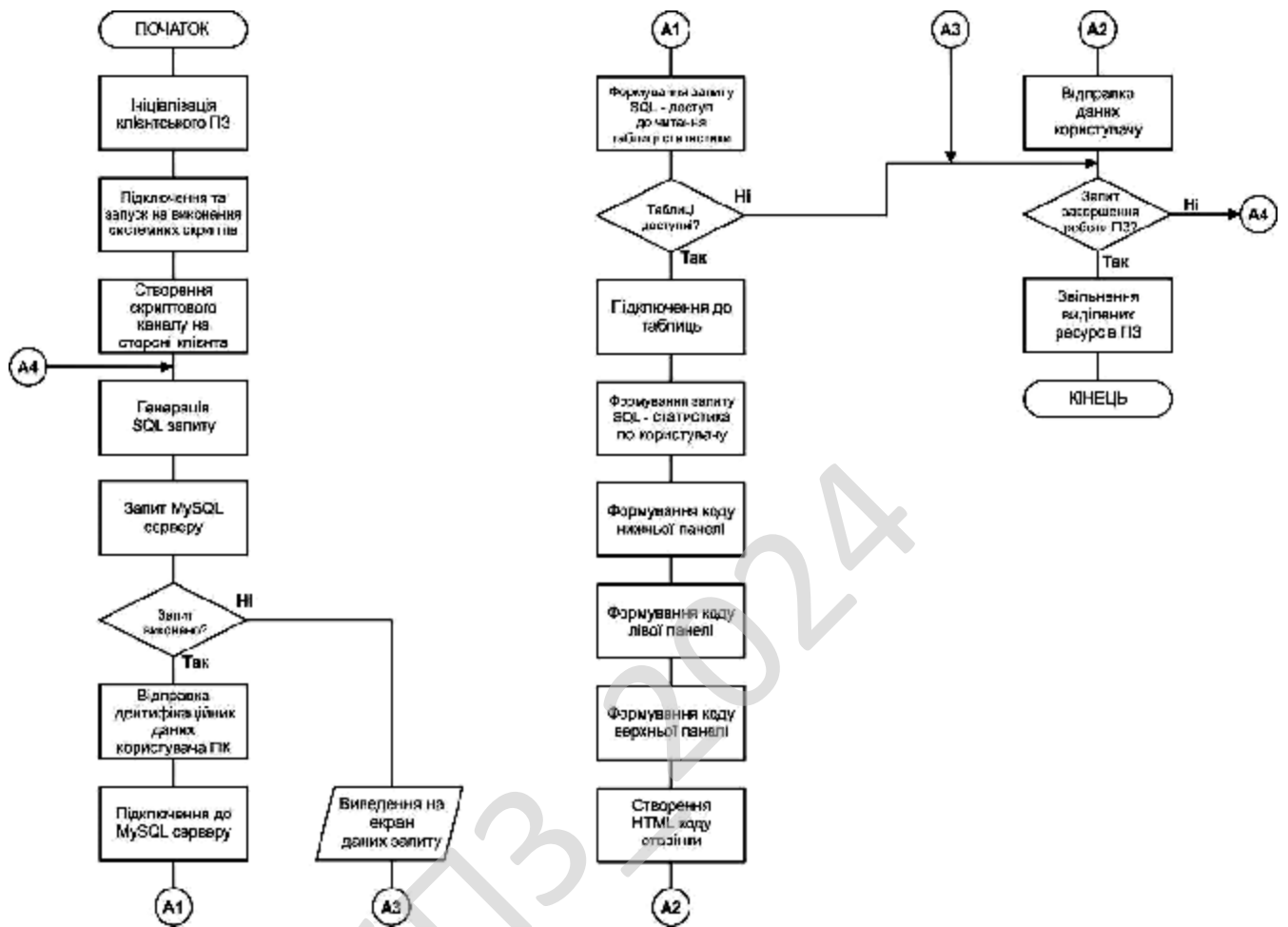


Рисунок 4.2 – Блок-схема інтерфейсу клієнтського ПЗ

Розглянемо розроблені алгоритми та методи.

При роботі із системою моніторингу трафіку локальної мережі. Виникла проблема роботи з XML. Інформація на сервері зберігається саме в цьому форматі. Тому був розроблений перетворювач XML в HTML на мові програмування PHP.

Як неважко припустити по його назві, мова HTML (Hypertext Markup Language) ставиться до числа так званих мов розмітки тексту (markup languages).

Під терміном "розмітка" розуміється загальна службова інформація, яка не виводиться разом з документом, але визначає, як повинні виглядати ті або інші фрагменти документа.

Наприклад можна зажадати, щоб яке-небудь слово виводилося жирним або курсивним шрифтом, вивести окремий абзац особливим шрифтом або оформляти заголовки збільшеним шрифтом. Текстовий редактор теж використовує особливу форму розмітки для вистави тих атрибутів форматування, які я вибираю. Таким чином, у ньому теж використовується особливий різновид мови розмітки.

У наші дні існує безліч різних мов розмітки. Наприклад, у комунікаційних програмах особлива форма розмітки визначає зміст кожного пакета з нулів і одиниць, що пересилається в Інтернеті. Коли ми підкреслюємо слова в книзі, це теж можна вважати свого роду розміткою. Втім, будь-яка мова розмітки повинен вирішувати два важливі завдання:

1. мова визначає синтаксис розмітки. Наприклад, у відповідності зі специфікацією HTML конструкція `text` визначає синтаксично правильну розмітку тексту, а конструкція `<xr5t>text</x4rt>` вважається неправильної через розбіжність відкриваючого й закриваючого тегів.

2. мова визначає зміст розмітки. Звичайно, ви знаєте, що команда `text` виводить слово `text` жирним шрифтом. У цьому випадку визначається зміст, пов'язаний з оголошенням деякого компонента документа.

Стрімкий розвиток Web за останні кілька років наочно показує, що самою популярною мовою розмітки тексту є HTML.

Перетворення XML в HTML.

```
<?
Class XMLHTML {
VAR $xmlparser;
VAR $tagcolor ="#800000";
VAR $datacolor ="#0000ff";
function XMLHTML() {
$this->xmlparser = xml_parser_create();
xml_set_object($this->xmlparser, $this);
xml_set_element_handler($this->xmlparser, "starttag", "endtag");
```

| | | | | | | |
|------|------|-----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № док.ум. | Підпис | Дата | | 49 |


```

    if (/login=(.*)/) { $login=&urldecode ($1); }
    if (/pass=(.*)/) { $password=&urldecode ($1); }
}

# Відкриваємо базу даних і перевіряємо логін і пароль.
open(INFO,$file) ||die;
@data=;
close(INFO);
foreach $string(@data) {
    @item=split(/&/,$string);
    foreach (@item) {
        if (($item[0] eq $login) && ($item[1] eq $password)) {
# Якщо все нормально, переходимо в користувацький каталог.
        print "Content-type:text/html\n\n";
        print <<HTML;
        <html><body bgcolor="e6e8fa">
# Вітаємо користувача.
        <p><h2 align=center><font color="ff0000">Hello,$login!</font></h2>
        <p><center>Welcome to your home directory!
        <p>Your URL is <a href="$basedir$login">$basedir$login.</a></center>
        HTML
# Список директорії
        $userdir=$dir.$login;
        chdir ("$userdir");
# Відкриваємо каталог і читаємо список файлів у масив.
        opendir(DIR,"$userdir") || die "Cannot open $userdir!";
        while (@files=readdir(DIR)) {
# Якщо каталог містить підкаталоги, виводимо їх
# окремо, а також не показуємо
# каталоги "." і ".." Друкуємо шапку таблиці.
        print <<HTML;
        <p><center>
        <table bgcolor="\bfbfbf\" width=600 border cellspacing=0
        cellpadding=0 nowrap>
        <tr><td colspan=5 align=center nowrap>
        <b><font color="ff0000">Directories</font></b></td></tr>
        <tr><td>.</td><td align=center><b>List</b>
        </td><td><b>Size</b>
        <td><td><b>Last accessed</b></td><td><b>Last modified</b></td>
#HTML
        foreach $file(@files) {
# Статистика файлів - розмір, час останнього звернення й модифікації.
        $size=(stat("$userdir/$file")) [7];

```

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|-----------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 53 |

```

    $atime=localtime((stat("$userdir/$file"))[8]);
    $mtime=localtime((stat("$userdir/$file"))[9]);
# друкуємо список підкаталогів.
    if ( -d "$userdir/$file" && "$file" ne "." && "$file" ne "..") {
    print "<tr><td width=30><img src=\"$url/image/folder.gif\">
    </td><td width=100 align=left>$file</td>\n";
    print "<td width=50>",$size,"</td><td width=200>",$atime,"</td>
    <td width=200>",$mtime,"</td></tr>\n";
    }
    }
    print "</table>\n";
# Ту ж операцію проводимо для файлів. Друкуємо шапку таблиці.
    print <<HTML;
    <p><table bgcolor=\"$bfbfbf\" width=600 border
    cellspacing=0 cellpadding=0>
    <tr><td colspan=5 align=center><b><font color="ff0000">Files</font>
    </b></td></tr>
    <tr><td>.</td><td><b>List</b><td><b>Size</b>
    <td><b>Last accessed</b><td><b>Last modified</b></tr>
#HTML
    foreach $file(@files) {
    $size=(stat("$userdir/$file"))[7];
    $atime=localtime((stat("$userdir/$file"))[8]);
    $mtime=localtime((stat("$userdir/$file"))[9]);
    if (!-d "$userdir/$file" && "$file" ne "." && "$file" ne "..") {
# поміщаємо знайдені файли в масив
        push (@dir,"$userdir/$file");
# підраховуємо їх кількість.
        $number=@dir;
# Видаємо інформацію.
        print "<tr><td width=30><img src=\"$url/image/page.gif\"></td>
        <td width=100><a href=\"$basedir$login/$file\">",$file,"</a></td>\n";
        print "<td width=50>",$size,"</td>\n";
        print "<td width=200>",$atime,"</td>
        <td width=200>",$mtime,"</td></tr>\n";
        }
        }
    print "</table>\n";
    print "<p><center><b><font color=\"$0000ff\">There are",
    $number," files in this directory.</b></font></center>\n";
    }

```

Алгоритми були створені для системи моніторингу трафіку локальної мережі. Читання й запис файлів і каталогів є чи не головною перевагою Perl. Практично будь-який Perl – скрипт використовує або запис у файли, або пошук певних рядків у файлі, або запис або завантаження файлів у каталоги.

Файл можна відкривати для читання, запису, додавання або пошуку певних рядків:

1. Відкрити файл для читання – `open (FILE,"$file");`
2. Відкрити файл для запису – `open (FILE,">$file");`
3. Відкрити файл для додавання – `open (FILE,">>$file");`
4. Відкрити файл для читання й додавання – `open (FILE,"+<$file").`

Де:

– FILE це дескриптор або описатель файлу, тобто ім'я, під яким він фігурує в програмі. Може бути будь-яким рядком символів, рекомендується набирати прописними буквами щоб уникнути неоднозначностей;

– \$file змінна для файлу, містить звичайне ім'я файлу й шлях до нього. Переважно оголосити цю змінну на початку скрипта. Це зручно, якщо файл перебуває десь глибоко в підкаталогах, приміром, незручно весь час набирати `/usr/local/htdocs/main/pages/file.html`. Зміст файлу можна читати порядково або в масив:

```
#!/usr/local/bin/perl
# прочитати один рядок з файлу.
open (STAT,"$file");
$count=;
close (STAT);

-----

#!/usr/local/bin/perl
# прочитати файл у масив.
open (STAT,"$file");
@count=;
close (STAT);
```

Розглянемо сценарій реєстрації користувача на веб-сервері. Ім'я користувача і його пароль записуються в текстовий файл і використовуються для його наступної автентифікації.


```

<TD><INPUT TYPE="password" NAME="pass" SIZE="20">
<TR><td><font color="ff0000">*</font><TD><b>E-mail:</b>
<TD><INPUT TYPE="text" NAME="email" SIZE="20">
<TR><TD colspan=3><p><center>
<INPUT TYPE="submit" VALUE="Submit"></center>
</TABLE></center>
</FORM>
# HTML
# Декодуємо поля форми
    else {
        foreach (@fields=split(/&/,$query)) {
            if (/login=(.*)/) { $login=&urldecode ($1); }
            if (/pass=(.*)/) { $password=&urldecode ($1); }
            if (/email=(.*)/) { $email=&urldecode ($1); }
        }
# Перевіряємо, не чи існує дане ім'я в системі.
        open(INFO,"login.txt") ||die;
# читаємо рядок в масив.
        @data=<INFO>;
        close(INFO);
        foreach $string(@data) {
            @item=split(/&/,$string);
# Розбиваємо рядок на частини
            foreach (@item) {
                if ($item[0] eq $login) {
# Порівнюємо отримане ім'я з першим полем файлу
# для кожного рядка і якщо таке знайдене видаємо
# помилку.
                print <<HTML;
                <h2 align=center><font color="ff0000">Error!</font></h2>
                <p><center><b>The name <font color="ff0000">$login</font> already
                    exists in the system. <p>Please,go back and choose another name.</b>
                <p><form><input type="button" value="Back" onclick=
                    "history.back()"></form>
                </center>
                HTML
                exit;
                }
                }
                }
# Якщо ім'я не знайдене,відкриваємо базу даних і додаємо інформацію.
                if ($item[0] ne $login) {

```

| | | | | | | | |
|------|------|----------|--------|------|--|----------------------------------|------|
| | | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | | 57 |

```

open(DATA,">>login.txt");
$string=join('&', $login, $password, $email, scalar localtime, $ENV
           {'REMOTE_ADDR'});
print DATA "$string\n";
close(DATA);
# Створюємо домашній каталог користувача й переходимо в нього.
mkdir("$userdir/$login",0700);
chdir("$userdir/login");
opendir(USER,"$userdir/$login");
# Поміщаємо файл index.html у каталог користувача.
open(IN,">$userdir/$login/index.html");
print IN "This is the test!\n";
close(IN);
closedir(USER);
# Зміст файлу може бути будь-яким,це тільки для прикладу.
# Генеруємо відповідь користувачеві.
print <<HTML;
<p><h1 align=center><font color="ff0000">Congratulations!</font></h1>
<p><b>Your registration was successful and your data were added to
our database.Thank you for your time.</b>
<p><center><b><font color="ff0000">
You entered:</font>(print this page and keep it in safe place)</b>
<p><table>
<tr><td><b>Your login name:</b><td><font color="0000ff">$login</font>
<tr><td><b>Your password:</b><td><font color="0000ff">$password</font>
<tr><td><b>Your e-mail address:</b>
<td><font color="0000ff">$email</font>
</table></center>
HTML
}
}

```

Скрипт видає відповідь у вигляді html сторінки, що містить всю інформацію, введена користувачем.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою Threefish – в криптографії симетричний блоковий криптоалгоритм, розроблений автором Blowfish та Twofish, американським криптографом Брюсом

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 58 |

Шнайером 2008 року для використання в геш-функції Skein і як універсальну заміну наявним блоковим шифрам. Основними принципами розробки шифру були: мінімальне використання пам'яті, необхідна для використання в геш-функції стійкість до атак, простота реалізації та оптимізація під 64-розрядні процесори.

Структура алгоритму

Threefish має дуже просту структуру і може бути використаний для заміни алгоритмів блочного шифрування, будучи швидким і гнучким шифром, що працюють в довільному режимі шифрування. Threefish S-блоки не використовуює, заснований на комбінації інструкцій виключаючого або, складання і циклічного зсуву.

Як і AES, шифр реалізований у вигляді підстановочно-перестановочної мережі на оборотних операціях, не будучи шифром мережі Фейстел.

Алгоритм передбачає використання tweak-значення, свого роду вектора ініціалізації, дозволяючи змінювати таким чином значення виходу, без зміни ключа, що має позитивний ефект як для реалізації нових режимів шифрування, так і на криптостійкості алгоритму.

Як результат думки авторів, що кілька складних раундів часто гірше ніж застосування великого числа простих раундів, алгоритм має нетрадиційно велику кількість раундів – 72 або 80 при ключі 1024 біт, проте, за заявою творців, його швидкісні характеристики випереджають AES приблизно вдвічі. Варто зауважити, що через 64-бітної структури шифру, дана заява має місцево лише на 64-розрядної архітектури. Тому, Threefish, як і Skein [1], заснований на ньому, на 32-розрядних процесорах показує значно гірші результати ніж на «рідному» обладнанні.

Ядром шифру є проста функція «MIX», перетворювальна два 64-бітових беззнакових числа, в процесі якої відбувається складання, циклічний зсув (ROL / ROR), і додавання по модулю 2 (XOR).

Нижче представлений код MIX-функції для Threefish-1024 [2]:

```
<syntaxhighlight lang="C">
```

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 59 |

```

// Константи для циклічного зсуву
int R16 [8] [8] = {
    {55, 43, 37, 40, 16, 22, 38, 12},
    {25, 25, 46, 13, 14, 13, 52, 57},
    {33, 8, 18, 57, 21, 12, 32, 54},
    {34, 43, 25, 60, 44, 9, 59, 34},
    {28, 7, 47, 48, 51, 9, 35, 41},
    {17, 6, 18, 25, 43, 42, 40, 15},
    {58, 7, 32, 45, 19, 18, 2, 56},
    {47, 49, 27, 58, 37, 48, 53, 56},
};
// D - раунд, j - індекс в таблиці циклічного зсуву
void mix (int j, int d) {
    unsigned long long rotl;
    y [0] = x [0] + x [1];
    rotl = R16 [d% 8] [j];
    y [1] = (x [1] << rotl) | (x [1] >> (64 - rotl));
    y [1] ^ = y [0];
}
</Source>

```

Процедура розшифрування обернена процедурі зашифрування і містить зворотну функцію DEMIX. Кожен з 72 раундів Threefish-256 і Threefish-512 має чотири MIX перетворення, Threefish-1024 – вісім звернень до MIX функції.

Безпека

За заявою авторів, алгоритм має більш високий рівень безпеки, ніж AES. Існує атака на 25 з 72 раундів Threefish, в той час як для AES – на 6 з 10. Threefish має показник фактора безпеки 2.9, в свою чергу, AES всього 1.7 [3] Для досягнення повної дифузії, шифру Threefish-256 досить 9 раундів, Threefish-512 – 10 раундів і Threefish-1024 – 11 раундів. Виходячи з цього, 72 і 80 раундів відповідно в середньому, забезпечать кращі результати, ніж існуючі шифри. [4] У той же час, алгоритм має набагато простішу структуру і функцію перетворення, проте виконання 72-80 раундів, на думку дослідників, забезпечує необхідну стійкість. Вживаний розмір ключа від 256 до 1024 біт зводить нанівець можливість повного перебору паролів при так званій атаці грубою силою (brute force attack) на сучасному обладнанні.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 60 |

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс користувача програми складається з таких логічних блоків:

- Меню програми яке складається з розділів: Файл; ЛМ (локальна мережа); Аналіз; Довідка.
- Функціональних клавiш, а саме: Підключитись; Відключитись; Статистика доступу; Налаштування; Журнал роботи програми.
- Обрання основної мережної карти.
- Поточної інформації.
- Кнопки сканування – отримання поточних даних.

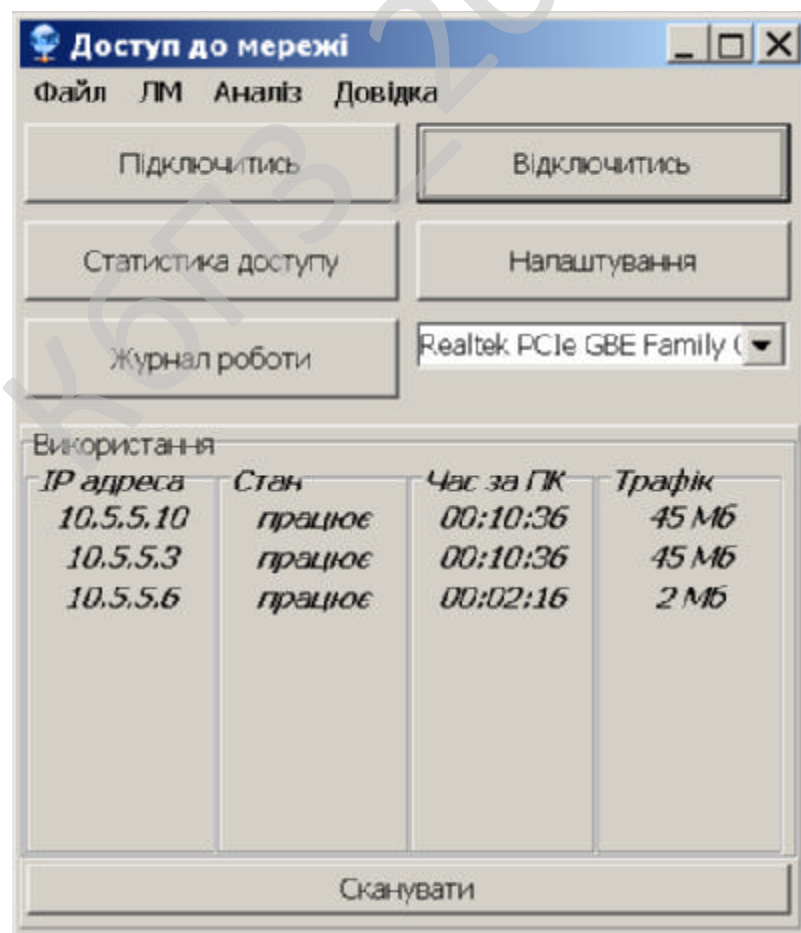


Рисунок 5.1 – Головне вікно програми

Перевагою розробленого ПЗ є те, що воно використовує безпечний протокол РРТР. Протокол РРТР дозволяє створювати захищені канали для обміну даними по різних мережевих протоколах – IP, IPX або NetBEUI.

Дані цих протоколів інкапсулюються за допомогою протоколу РРТР в пакети протоколу IP, за допомогою якого переносяться в зашифрованому вигляді через мережу TCP / IP. Інкапсулюється вихідний кадр PPP, тому протокол РРТР можна віднести до класу протоколів інкапсуляції каналного рівня в мережевий.

На рисунку 5.2 зображено форму авторського права.

Обрано тип розповсюдження – безплатне програмне забезпечення (Freeware). Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без вихідних кодів. Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

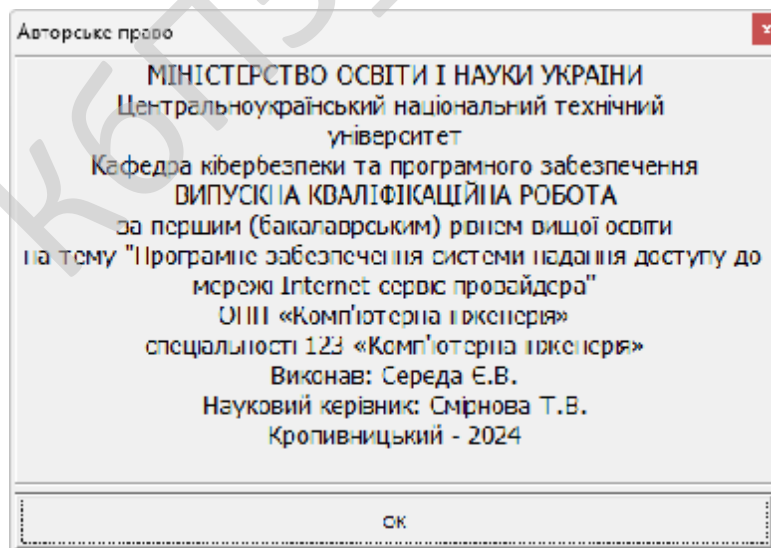


Рисунок 5.2 – Довідка програми

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи надання доступу до мережі Internet сервіс провайдера.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем надання доступу до мережі Internet сервіс провайдера.

– Досліджена система надання доступу до мережі Internet сервіс провайдера.

– На основі отриманих результатів досліджень створена програмна реалізація системи надання доступу до мережі Internet сервіс провайдера.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання надання доступу до мережі Internet сервіс провайдера.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня PHP, Perl. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи надання доступу до мережі Internet сервіс провайдера. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|-----------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 63 |

загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Threefish.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

КБПЗ_2024

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 64 |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
2. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
3. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
4. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
5. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
6. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
7. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
8. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings* Volume 3156, 2022, Pages 390-399.
9. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 65 |

10. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

11. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

12. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

13. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

14. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

15. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

16. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

17. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties».

International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

18. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

19. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.517-522.

20. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

21. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

22. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv*, Ukraine, 2-6 July, 2019, P. 395-399.

23. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

24. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising

Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

25. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.*

26. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering.* – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

27. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем ІР-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

28. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

29. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

30. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

31. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 68 |

захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.*

32. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.*

33. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95*

34. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.*

35. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.*

36. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

37. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія. – Х.: ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.*

38. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.*

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 69 |

39. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

40. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

41. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х.: Вид. Рожко С.Г. 2019. С. 123-139

42. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

43. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

44. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

45. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 70 |

46. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

47. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

48. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

49. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

50. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

51. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

52. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

53. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

| | | | | | | |
|------|------|----------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата | | 71 |

Додаток А
(обов'язковий)

Технічне завдання

Зміст

| | |
|---|---|
| 1 Найменування та область застосування..... | 2 |
| 2 Підстава для розробки..... | 2 |
| 3 Мета та призначення розробки..... | 2 |
| 4 Джерела розробки..... | 2 |
| 5 Технічні вимоги..... | 2 |
| 5.1 Вміст проекту..... | 2 |
| 5.2 Показники призначення..... | 3 |
| 5.3 Вимоги до функціональних характеристик..... | 3 |
| 5.4 Вимоги до архітектури..... | 3 |
| 5.5 Вимоги до надійності..... | 3 |
| 5.6 Умови експлуатації..... | 4 |
| 5.7 Вимоги до складу та параметрів технічних засобів..... | 4 |
| 5.8 Вимоги до інформаційної і програмної сумісності..... | 4 |
| 5.8.1 Обладнання..... | 4 |
| 5.8.2 Мова програмування..... | 4 |
| 5.8.3 Вхідні дані..... | 5 |
| 5.8.4 Вихідні дані..... | 5 |
| 6 Вимоги до програмної документації..... | 5 |
| 7 Перелік документів, що розробляються..... | 5 |
| 8 Етапи розробки..... | 6 |
| 9 Порядок контролю та приймання..... | 6 |

| | | | | | | | | |
|-----------|---------------|-------------|--------|------|---|------|-------|---------|
| | | | | | ВКРБ-123.24.0014.00.00.ТЗ | | | |
| Вим. | Арк. | № документа | Підпис | Дата | | | | |
| Розробив | Серета Є.В. | | | | Програмне забезпечення системи надання доступу до мережі Internet сервіс провайдера | Літ. | Аркуш | Аркушів |
| Перевірів | Смірнова Т.В. | | | | | Б | 1 | 6 |
| Н. Контр. | Коваленко А.С | | | | ЦНТУ КІ-20 | | | |
| Затв. | Смірнов О.А. | | | | | | | |

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи надання доступу до мережі Internet сервіс провайдера.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 131-02 від 01.04.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи надання доступу до мережі Internet сервіс провайдера.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

| | | | | | | |
|------|------|-------------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 2 |

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи надання доступу до мережі Internet сервіс провайдера;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

| | | | | | | |
|------|------|-------------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 3 |

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище PHP, Perl.

| | | | | | | |
|------|------|-------------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 2 |

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 71 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

| | | | | | | |
|------|------|-------------|--------|------|----------------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 5 |

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 14.06.2024 р.

| | | | | | | |
|------|------|-------------|--------|------|---------------------------|------|
| | | | | | ВКРБ-123.24.0014.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 6 |

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
першим (бакалаврським) рівнем вищої освіти

_____ Смірнова Т.В.

*Програмне забезпечення системи надання доступу до мережі Internet сервіс
провайдера*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 33

Літера: РП

Кропивницький – 2024 року

ВИХІДНИЙ КОД МОДУЛЯ АДМІНІСТРУВАННЯ

```
<?php
//Середа Єгор Віталійович
// ЦНТУ, кафедра КБПЗ, КІ-20, 2024

    $path=getenv("DOCUMENT_ROOT");
    $path.="MONITORING /"; // шлях до папки зі скриптами
    include( "$path"."common.php" ); // підключення модулю common.php
    include( "$path"."global.php" ); // підключення модулю global.php
    Top("./", "сторінка адміністратора");
    ConnectToBase($conf{'host'}, $conf{'username'}, $conf{'password'},
    $conf{'database'});
    $CurrentDateDB = GetCurrentDateDB();
    if ($action=="search") // виклик пошуку?
    {
    search();
    }
    elseif ($action=="show") // виклик відображення інтерфейсу
    {
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="incbalance") // виклик відображення балансу
    {
    IncBalance ();
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="update") // виклик оновлення
    {
    UpdateUser ();
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="delete") // виклик видалення
    {
    if ($is_js_confirmed)
    {
    DeleteUser ();
    }
    ShowSearchForm(GetMonths());
    }
    elseif ($action=="adduserform")
    {
    AddUserForm ( );
    }
    elseif ($action=="adduser") // виклик додавання користувача
    {
    AddUser( );
    ShowSearchForm(GetMonths());
    }
    elseif ($action=="addgroup")
    {
    AddGroup ();
    EditGroup ();
    }elseif ($action=="editgroup")
    {
    EditGroup ();
    }
    elseif ($action=="delgroup")
    {
    DelGroup ();
    EditGroup ();
    }
    elseif ($action=="updategroup")
    {
    UpdateGroup ();
    EditGroup ();
    }
    elseif ($action=="stat")
    {
```

```
    ShowStat ();  
}  
elseif ($action=="deleteold")  
{  
    DeleteOld ();  
} else {  
    ShowSearchForm(GetMonths ());  
}  
Footer();  
?>
```

К6П3_2024

ВИХІДНИЙ КОД WWW-ІНТЕРФЕЙСУ

```
<?php
//Середа Єгор Віталійович
// ЦНТУ, кафедра КБПЗ, КІ-20, 2024
include( "./common.php" ); // підключення модулю common.php
include( "./global.php" ); // підключення модулю global.php
// початок формування HTML коду
Тор("./", "сторінка користувача.");
// підключення та робота з БД, формування HTML коду
ConnectToBase($conf{'host'}, $conf{'username'}, $conf{'password'},
              $conf{'database'});
// Дані користувачів, формування HTML коду
ShowUser(ResolveUser());
// кінець формування HTML коду
Footer();
?>
```

КБПЗ_2024

ФАЙЛ MONITORING.PL - ВИХІДНИЙ КОД МОДУЛЯ ЗЧИТУВАННЯ ТРАФІКА

```
#!/usr/bin/perl
# Diplom
# Серета Єгор Віталійович
# ЦНТУ, кафедра КБПЗ, КІ-20, 2024
use DBI;
use Common;
LoadConfig();
GetDate();
ConnectToDB();
TestNextMonth();
PutTrafToDBOutput();
PutTrafToDBForward();
SubtractAbon();
TurnOffAccess();
TurnOnAccess();
KillUserOutput();
KillUserForward();
Disconnect();
```

КБПЗ_2024

ФАЙЛ COMMON.PM - ПАКЕТНИЙ ФАЙЛ ПРОЕКТУ

```

package Common;
BEGIN {
//Середа Єгор Віталійович
// ЦНТУ, кафедра КБПЗ, КІ-20, 2024
    use Exporter ();
    @ISA = 'Exporter';
    @EXPORT = qw(&SubtractAbon &ConvertMoneyToDB &TurnOnAccess &TurnOffAccess
&SubtractMoney &GetDate &PutTrafToDBOutput &PutTrafToDBForward &LoadConfig
&ConnectToDB &TestNextMonth &KillUserOutput &KillUserForward &Disconnect);
}
// завантаження конфігурації
sub LoadConfig
{
for $file("./Monitoring.conf")
    {
        // повернення коду читання файлу
        unless ($return = do $file) {
            warn "couldn't parse $file: $@" if $@;
            warn "couldn't do $file: $!" unless defined $return;
            warn "couldn't run $file" unless $return;
        }
    }
}

// читання таблиць БД
sub TurnOffAccess
{
    my $query_counter = ("SELECT `ip` FROM `counter` WHERE
        `moneymonth`='n' AND `yearmonth`='$date'");
    my $sth_counter = $dbh -> prepare($query_counter);
    $sth_counter -> execute;
    $sth_counter -> bind_columns(undef, \($ip));
    while ($sth_counter -> fetch)
    {
        my $query = ("UPDATE `user` SET access = 'n' WHERE `ip` = '$ip' AND
            `access`='y'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        $sth -> finish;
    }
    $sth_counter -> finish;
my $query = ("UPDATE `user` SET access = 'n' WHERE `balance`<=0 AND
    `access`='y'");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    $sth -> finish;
}
sub TurnOnAccess
{
    my $query_counter = ("SELECT `ip` FROM counter WHERE
        `moneymonth`='y' AND `yearmonth`='$date'");
    my $sth_counter = $dbh -> prepare($query_counter);
    $sth_counter -> execute;
    my ($ip);
    $sth_counter -> bind_columns(undef, \($ip));
    while ($sth_counter -> fetch)
    {
        my $query = ("UPDATE `user` SET access = 'y' WHERE `balance` > 0 AND
`ip` = '$ip' AND `access`='n'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        $sth -> finish;
    }
    $sth_counter -> finish;
}

// Запис до таблиць БД

```

```

sub ConvertMoneyToDB
{
    $money = $_[0];
    $newmoney = $money*1000000000;
return $newmoney; // повернення коду затрачених коштів
}
sub SubtractAbon
{
    $ip = $_[0];
    # $balancelimit = ConvertMoneyToDB(100);
    my $query_user = ("SELECT `ip`,`group`,`balance` FROM user");
    #print "$query_user\n";
    my $sth_user = $dbh -> prepare($query_user);
    $sth_user -> execute;
    $sth_user -> bind_columns(undef,\($ip,$group,$balance));
    while( $sth_user -> fetch )
    {
        my $query_counter = ("SELECT `moneymonth`,`counter` FROM `counter`
                               WHERE `yearmonth`='$date' AND `ip`='$ip'");
        my $sth_counter = $dbh -> prepare($query_counter);
        $sth_counter -> execute;
        my ($moneymonth,$counter);
        $sth_counter -> bind_columns(undef,\($moneymonth,$counter));
        $sth_counter -> fetch;
        $sth_counter -> finish;
        if ($moneymonth eq 'n')
        {
            my $query_group = ("SELECT `trafcost`,`abon`,`abontraf` FROM
                                `group` WHERE `group`='$group'");
            my $sth_group = $dbh -> prepare($query_group);
            $sth_group -> execute;
            my ($trafcost,$abon,$abontraf);
            $sth_group -> bind_columns(undef,\($trafcost,$abon,$abontraf));
            $sth_group -> fetch;
            $sth_group -> finish;
            if ($balance >= ConvertMoneyToDB($abon))
            {
                my $MoneyForAbon = ConvertMoneyToDB($abon)-1;
                my $query_abon = ("UPDATE `user` SET balance = balance -
                                   $MoneyForAbon WHERE `ip`='$ip'");
                my $sth_abon = $dbh -> prepare($query_abon);
                $sth_abon -> execute;
                $sth_abon -> finish;
            }
        }
    }
    $sth_user -> finish;
}
sub SubtractMoney
{
    $ip = $_[0];
    $traffic = $_[1];
    my $query_user = ("SELECT `group` FROM user WHERE `ip`='$ip'");
    #print "$query_user\n";
    my $sth_user = $dbh -> prepare($query_user);
    $sth_user -> execute;
    my ($group);
    $sth_user -> bind_columns(undef,\($group));
    $sth_user -> fetch;
    $sth_user -> finish;
    my $query_group = ("SELECT `trafcost`,`abontraf` FROM `group`
                        WHERE `group`='$group'");
    my $sth_group = $dbh -> prepare($query_group);
    $sth_group -> execute;
}

```

```

my ($trafcost, $abontraf);
$sth_group -> bind_columns(undef, \($trafcost, $abontraf));
$sth_group -> fetch;
$sth_group -> finish;
    my $query_counter = ("SELECT `moneymonth`,`counter` FROM `counter`
                          WHERE `yearmonth`='$date' AND `ip`='$ip'");
    my $sth_counter = $dbh -> prepare($query_counter);
$sth_counter -> execute;
my ($moneymonth, $counter);
$sth_counter -> bind_columns(undef, \($moneymonth, $counter));
$sth_counter -> fetch;
$sth_counter -> finish;
if ($counter > $abontraf*1048576)
{
    my $ByteCost = $trafcost/(100*1024*1024);
    my $MoneyToSubtract = ConvertMoneyToDB($ByteCost*$traffic);
my $query_balance = ("UPDATE `user` SET balance = balance - $MoneyToSubtract
                     WHERE `ip`='$ip'");
    my $sth_balance = $dbh -> prepare($query_balance);
    $sth_balance -> execute;
    $sth_balance -> finish;
}
}
sub KillUserForward
{
    my $query = ("SELECT ip,access FROM user");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    my ($ip, $access);
    $sth -> bind_columns(undef, \($ip, $access));
    while($sth -> fetch)
    {
        #print "$ip $access\n";
        if ($access eq "n")
        {
            my $AllreadyOff='n';
            open (KILLHFORWARD, "iptables -L FORWARD -n |");
            while (my $s=<KILLHFORWARD>)
            {
                if (($.>2))
                {
                    @ar = split /\s+/, $s;
                    #print "$ar[0] $ar[3]\n";
                    if (($ar[0] eq "REJECT") && ($ar[3] eq "$conf{'network'}"."$ip"))
                    {
                        $AllreadyOff = 'y';
                    }
                }
            }
            close(KILLHFORWARD);
            if ($AllreadyOff eq 'n')
            {
                $string = "iptables -I FORWARD -s ".$conf{'network'}.$ip." -j REJECT";
                system($string);
            }
            else {
                my $AllreadyOff='n';
                open (KILLHFORWARD, "iptables -L FORWARD -n |");
                while (my $s=<KILLHFORWARD>)
                {
                    if (($.>2))
                    {
                        @ar = split /\s+/, $s;
                        #print "$ar[0] $ar[3]\n";
                        if (($ar[0] eq "REJECT") && ($ar[3] eq
"$conf{'network'}"."$ip"))
                        {
                            $AllreadyOff = 'y';
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    }
    close(KILLHFORWARD);
    if ($AllreadyOff eq 'y')
    {
$string = "iptables -D FORWARD -s ".$conf{'network'}. $ip." -j REJECT";
        system($string);
    }
}
}
$sth -> finish;
}
sub KillUserOutput
{
    my $query = ("SELECT ip,access FROM user");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    my ($ip,$access);
    $sth -> bind_columns(undef,\($ip,$access));
    while($sth -> fetch)
    {
        print "$ip $access\n";
        if ($access eq "n")
        {
            my $AllreadyOff='n';
            open (KILLHOUTPUT, "iptables -L OUTPUT -n |");
            while (my $s=<KILLHOUTPUT>)
            {
                if (($.>2))
                {
                    {
                        @ar = split /\s+/, $s;
                        #print "$ar[0] $ar[3]\n";
                    if (($ar[0] eq "REJECT") && ($ar[4] eq "$conf{'network'}. "$ip"))
                    {
                        $AllreadyOff = 'y';
                    }
                }
            }
            close(KILLHOUTPUT);
            if ($AllreadyOff eq 'n')
            {
my $string = "iptables -I OUTPUT -d ".$conf{'network'}. $ip." -p tcp --sport 3128
                -j REJECT";
                system($string);
            }
            } else {
                my $AllreadyOff='n';
                open (KILLHOUTPUT, "iptables -L OUTPUT -n |");
                while (my $s=<KILLHOUTPUT>)
                {
                    if (($.>2))
                    {
                        {
                            @ar = split /\s+/, $s;
                            #print "$ar[0] $ar[3]\n";
                        if (($ar[0] eq "REJECT") && ($ar[4] eq "$conf{'network'}. "$ip"))
                        {
                            $AllreadyOff = 'y';
                        }
                    }
                }
            }
            close(KILLHOUTPUT);
            if ($AllreadyOff eq 'y')
            {
                my $string = "iptables -D OUTPUT -d ".$conf{'network'}. $ip." -
                p tcp --sport 3128 -j REJECT";
                system($string);
            }
        }
    }
}
}

```

```

        $sth -> finish;
    }

    sub ConnectToDB
    {
        $datasource = "DBI:mysql:". $conf{ 'DBName' }. ":". $conf{ 'host' };
        $dbh = DBI -> connect($datasource, $conf{ 'username' }, $conf{ 'password' });
    }

    sub GetDate
    {
        open (DATE_H, "date +%y%m |");
        $date = <DATE_H>;
    }

    sub TestNextMonth
    {
        my $query = ("SELECT ip FROM counter WHERE yearmonth='$date'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        my $numrows = $sth -> rows;
        $sth -> finish;
        if ($numrows == 0)
        {
            for (my =1;$i<=254;$i++)
            {
                my $query = ("INSERT INTO counter (ip, yearmonth) VALUES
($i, '$date')");
                my $sth = $dbh -> prepare($query);
                $sth -> execute;
                $sth -> finish;
            }
        }
    }

    sub PutTrafToDBOutput
    {
        $count=0;
        open (FH, "iptables -L -Z OUTPUT -nvx |");
        print "$date\n";
        while(my $s=<FH>)
        {
            if (($.>2))
            {
                # $s =~ s/\s+/_/g;
                @ar = split /\s+/, $s;
                if (($ar[9] =~ /\$conf{'network'}\d+/)&&($ar[3] ne "REJECT"))
                {
                    @ip = split /\./, $ar[9];
                    if ($ar[2] ne "0")
                    {
                        if (($ip[3] >= 2) && ($ip[3] <= 254))
                        {
                            my $query = ("UPDATE counter SET counter = counter + $ar[2] WHERE
ip='$ip[3]' AND yearmonth='$date'");
                            my $sth = $dbh -> prepare($query);
                            $sth -> execute;
                            $sth -> finish;
                            SubtractMoney("$ip[3]", "$ar[2]");
                        }
                        $count+=$ar[2];
                    }
                }
            }
        }
        print "\nOverall OUTPUT: $count\n";
        close(FH);
    }

    sub PutTrafToDBForward
    {
        $count=0;
        open (FH, "iptables -L -Z FORWARD -nvx |");
    }

```

```

print "$date\n";
while(my $s=<FH>)
{
    if (($.>2))
    {
        # $s =~ s/\s+/_/g;
        @ar = split /\s+/, $s;
        if (($ar[9] =~ /\$conf{'network'}\d+/)&&($ar[3] ne "REJECT"))
        {
            @ip = split /\./, $ar[9];
            if ($ar[2] ne "0")
            {
                if (($ip[3] >= 2) && ($ip[3] <= 254))
                {
                    my $query = ("UPDATE counter SET counter = counter + $ar[2] WHERE
ip='$ip[3]' AND yearmonth='$date'");
                    my $sth = $dbh -> prepare($query);
                    $sth -> execute;
                    $sth -> finish;
                    SubtractMoney("$ip[3]","$ar[2]");
                }
                $count+=$ar[2];
            }
        }
    }
}
print "\nOverall FORWARD: $count\n";
close(FH);
}
sub Disconnect
{
    $dbh -> disconnect;
}
return 1; // повернення коду
END { }

```

ВИХІДНИЙ КОД ФАЙЛА COMMON.PHP

```

<?php
//Середа Єгор Віталійович
// ЦНТУ, кафедра КБПЗ, КІ-20, 2024
// функція підключення та роботи з БД
function ConnectToBase($host,$username,$password,$database)
{
    global $link;
    if (!($link = mysql_pconnect( $host, $username, $password)))
    {
        echo ("<br>Соединение с базой не установлено<br>");
        exit();
    }
// вибор БД
    if (!mysql_select_db($database, $link))
    {
        echo ( "Помилка при виклику БД. ".mysql_errno($link)."
            ".mysql_error($link) );
        exit();
    }
}

// Функція відображення форми пошуку користувачів
function ShowSearchForm($month)
{
    ?>
<table cellpadding="0" cellspacing="0" border="0">
<tr><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <form method="GET" action=?echo ("$_PHP_SELF");?>
    <tr class="even"><td height=30>
        Login:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="login">
    </td>
    </tr>
    <tr class="even"><td height=30>
        IP:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="ip">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Ім'я:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="name">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Адреса:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="address">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Доступ:
    </td>
    <td bgcolor="#ffffff">
        <input type="radio" name="access" value="y">Єсть&nbsp;
        <input type="radio" name="access" value="n">Нет&nbsp;
        <input type="radio" name="access" value="-1" checked>-
    </td>

```

```

</tr>
<tr class="even"><td height=30>
    Місяць:
</td>
<td bgcolor="#ffffff">
    <select name="month">
<?
        $i=1;
        while ($month[$i])
        {
            if (GetCurrentDateDB()==$month[$i])
            {
                echo "<OPTION VALUE=\"\$month[$i]\"
SELECTED>".ConvertDateDB($month[$i])."</option>";
            } else {
                echo "<OPTION
VALUE=\"\$month[$i]\">".ConvertDateDB($month[$i])."</option>";
            }
            $i++;
        }
?>
    </select>
</td>
</tr>
<tr class="odd"><td></td><td height=30 align="left">
    <input type="submit" value="Искать">
</td></tr>
    <input type="hidden" name="action" value="search" >
</form>
</table>
</td></tr>
</table>
</td>
<td valign="top">
<table cellpadding="0" cellspacing="0" border="0" width="300">
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=adduserform");?>">Додати нового
користувача</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=editgroup");?>">Редагування груп</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=stat");?>">Статистика</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=deleteold");?>">Видалення інформації</a>
</td></tr>
</table>
</td></tr>
</table>
<?
}

// Функція видалення
function DeleteOld()
{
    global $delete, $link, $month;
    $month_ar = GetMonths();
?>
<table cellpadding="0" cellspacing="2" border="0">
<tr class="odd"><td>
<a href="<? echo ("\$PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd"><td>
Видалити інформацію о трафіке та платежах за:
</td></tr>

```

```

<tr class="odd"><td>
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
    <select name="month">
<?
        $i=1;
        while ($month_ar[$i])
        {
            if (GetCurrentDateDB()!=$month_ar[$i])
            {
                echo "<OPTION
VALUE=\"$_month_ar[$i]\">".ConvertDateDB($month_ar[$i])."</option>";
            }
            $i++;
        }
?>
    </select>
</td></tr>
<tr><td>
<br>
        <input type="hidden" name="action" value="deleteold">
        <input type="hidden" name="delete" value="1">
        <input type="submit" value="Удалить">
</form>
</td></tr>
</table>
<?
    if (($delete)&&($month))
    {
        $query1="DELETE FROM `counter` WHERE `yearmonth`='$_month.'";
        $query2="DELETE FROM `money` WHERE `date` LIKE '$_month.%'";
        echo $query1."<br>".$query2;
        if (!( $result1 = mysql_query($query1, $link ) ))
        {
            echo ( "Помилка при виконанні $query1
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        if (!( $result2 = mysql_query($query2, $link ) ))
        {
            echo ( "Помилка при виконанні $query2
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
    }
}

// Функція пошуку
function Search()
{
    global $name, $login, $ip, $address, $access, $month, $link, $sortBy;
    if ($ip != "")
    {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE `ip`='$ip'";
    }
    elseif ($login != "")
    {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE `login` like '%$login%'";
    } else {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE 1=1";
        if ($name != "") {$query_user.=" AND `name` like '%$name%'"; }
        if ($address != "") {$query_user.=" AND `address` like
'%$address%'";}
        if ($access !="-1") {$query_user.=" AND `access` like
'$access'";}
    }
    if ($sortBy=="ip_asc")

```

```

{
$query_user.=" ORDER BY ip ASC";
} elseif ($sortby=="ip_desc")
{
$query_user.=" ORDER BY ip DESC";
} elseif ($sortby=="login_desc")
{
$query_user.=" ORDER BY login DESC";
} elseif ($sortby=="login_asc")
{
$query_user.=" ORDER BY login ASC";
} elseif ($sortby=="group_desc")
{
$query_user.=" ORDER BY `group` DESC";
} elseif ($sortby=="group_asc")
{
$query_user.=" ORDER BY `group` ASC";
} elseif ($sortby=="access_desc")
{
$query_user.=" ORDER BY `access` DESC";
} elseif ($sortby=="access_asc")
{
$query_user.=" ORDER BY `access` ASC";
} elseif ($sortby=="balance_desc")
{
$query_user.=" ORDER BY balance DESC";
} elseif ($sortby=="balance_asc")
{
$query_user.=" ORDER BY balance ASC";
} elseif ($sortby=="address_desc")
{
$query_user.=" ORDER BY address DESC";
} elseif ($sortby=="address_asc")
{
$query_user.=" ORDER BY address ASC";
} elseif ($sortby=="name_desc")
{
$query_user.=" ORDER BY name DESC";
} elseif ($sortby=="name_asc")
{
$query_user.=" ORDER BY name ASC";
} else {
$query_user.=" ORDER BY ip ASC";
}
if ($sortby == "traffic_asc")
{
$query_counter = "SELECT `ip`, `counter`, `moneymonth` FROM `counter`
WHERE `yearmonth`='".$$.month.'" AND `counter` > 0 ORDER BY 'counter' ASC";
if (!( $result_counter = mysql_query( $query_counter, $link ) ))
{
echo ( "Помилка при виконанні $query_counter ".mysql_errno($link)."
".mysql_error($link) );
exit();
}
} elseif ($sortby == "traffic_desc")
{
$query_counter = "SELECT `ip`, `counter`, `moneymonth` FROM `counter`
WHERE `yearmonth`='".$$.month.'" AND `counter` > 0 ORDER BY 'counter' DESC";
if (!( $result_counter = mysql_query( $query_counter, $link ) ))
{
echo ( "Помилка при виконанні $query_counter
".mysql_errno($link)." ".mysql_error($link) );
exit();
}
}
if (!( $result_user = mysql_query( $query_user, $link ) ))
{
echo ( "Помилка при виконанні $query_user
".mysql_errno($link)." ".mysql_error($link) );
}
}

```

```

        exit ();
    }
    $backst = getenv("QUERY_STRING");
    $row_user = mysql_fetch_object( $result_user );
    if ( mysql_num_rows($result_user)==1 )
    {
        ShowUserAdmin($row_user -> ip,$month);
    } elseif (mysql_num_rows($result_user) > 1)
    {
        ?>
        <table cellpadding="0" cellspacing="0" border="0">
        <tr class="odd"><td>
        <a href="<? echo ("$_PHP_SELF"."?action=");?>">Вернуться</a>
        </td></tr>
        <tr class="edge"><td>
        <table cellpadding="2" cellspacing="1" border="0">
        <tr class="top" align="center">
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="ip_asc") { echo "sortby=ip_desc"; }
        else { echo "sortby=ip_asc"; }?>>IP</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="login_asc") { echo
        "sortby=login_desc"; } else { echo "sortby=login_asc"; }?>>Login</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="name_asc") { echo "sortby=name_desc";
        } else { echo "sortby=name_asc"; }?>>Имя</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="group_asc") { echo
        "sortby=group_desc"; } else { echo "sortby=group_asc"; }?>>Группа</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="address_asc") { echo
        "sortby=address_desc"; } else { echo "sortby=address_asc"; }?>>Адрес</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="access_desc") { echo
        "sortby=access_asc"; } else { echo "sortby=access_desc"; }?>>Доступ</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="traffic_desc") { echo
        "sortby=traffic_asc"; } else { echo "sortby=traffic_desc"; }?>>Трафик</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="balance_asc") { echo
        "sortby=balance_desc"; } else { echo "sortby=balance_asc";
        }?>>Баланс (грн.)</a></td>
        <td>user.balance</td>
        <td>Абонентка</td>
        <td></td></tr>
        <?
        if (($sortby=="traffic_asc") || ($sortby=="traffic_desc"))
        {
            $i=1;
            mysql_data_seek( $result_user, 0);
            while (( $row_counter = mysql_fetch_object( $result_counter ))
            {
                $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
                `balance`, `group` FROM `user` WHERE `ip`='".$row_counter -> ip.'"";
            if (!( $result_user = mysql_query($query_user, $link ) ))
            {
                echo ( "Помилка при виконанні $query_user ".mysql_errno($link)."
                ".mysql_error($link) );
                exit ();
            }
            $row_user = mysql_fetch_object( $result_user );

```

```

        if ($i%2) {$class="odd";} else {$class="even";}
        $i++;
        printf("<tr class=\"\$class\"><td>%s</td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td
align=\"center\">%s</td><td>%s</td><td align=\"center\">%s</td><td
align=\"right\">%s</td><td align=\"right\">%s</td><td align=\"right\">%s</td><td
align=\"center\">%s</td><td><a href=\"\$PHP_SELF?action=delete&ip=%s\"
onclick=\"return confirmLink(this, 'видалити користувача
`s`s`)\">>Видалити</a></td></tr>\n",

$row_user -> ip, $row_user -> ip, $month, $row_user -> login, $row_user -> ip,
$month, $row_user -> name, $row_user -> group, $row_user -> address, $row_user -
> access, number_format($row_counter -> counter,0, chr(44), "."),
ConvertMoneyFromDB($row_user -> balance), $row_user -> balance, $row_counter ->
moneymonth, $row_user -> ip, $row_user -> login);
    }
} else {
    $i=1;
    mysql_data_seek( $result_user, 0);
    while (( $row_user = mysql_fetch_object( $result_user ))
        {
            $query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`=\".$row_user -> ip.\" AND `yearmonth`=\".$month;
            if (!( $result_counter =
mysql_query($query_counter, $link ) )
                {
                    echo ( "Помилка при виконанні
$query_counter ".mysql_errno($link)." ".mysql_error($link) );
                    exit();
                }
            $row_counter = mysql_fetch_object(
$result_counter );
            if ($i%2) {$class="odd";} else {$class="even";}
            $i++;
            printf("<tr class=\"\$class\"><td>%s</td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td
align=\"center\">%s</td><td>%s</td><td align=\"center\">%s</td><td
align=\"right\">%s</td><td align=\"right\">%s</td><td align=\"right\">%s</td><td
align=\"center\">%s</td><td><a href=\"\$PHP_SELF?action=delete&ip=%s\"
onclick=\"return confirmLink(this, 'видалити користувача
`s`s`)\">>Видалити</a></td></tr>\n",
                $row_user -> ip, $row_user -> ip, $month, $row_user ->
login, $row_user -> ip, $month, $row_user -> name, $row_user -> group, $row_user
-> address, $row_user -> access, number_format($row_counter -> counter,0,
chr(44), "."), ConvertMoneyFromDB($row_user -> balance), $row_user -> balance,
$row_counter -> moneymonth, $row_user -> ip, $row_user -> login);
        }
}
?>
</table>
</td></tr>
</table>
<?
} else
{
?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("\$PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd"><td>
Нема користувачів, відповідних пошукові
</td></tr>
</table>
<?>
mysql_free_result( $result_user );
    if ($result_counter)

```

```
        {
            mysql_free_result( $result_counter );
        }
echo $query_user;
echo "<br>".$query_counter;
}

// Функція конвертування коштів
function ConvertMoneyFromDB($money)
{
    $NewMoney = number_format($money/1000000000, 2, chr(44), ".");
    return $NewMoney; // повернення коду коштів
}

// Функція конвертування коштів
function ConvertMoneyToDB($money)
{
    $NewMoney = $money*1000000000;
    return $NewMoney; // повернення коду коштів
}

// Функція конвертування дати у БД
function ConvertDateDBMoney($dateDB)
{
    $date_ar = explode(" ", $dateDB);
    $time = $date_ar[1];
    $year = substr($date_ar[0], 0, 2);
    $month = substr($date_ar[0], 2, 2);
    $date = substr($date_ar[0], -2);
    switch ( $month ) {
        case "1":
            $monthstring = "січня";
            break;
        case "2":
            $monthstring = "лютого";
            break;
        case "3":
            $monthstring = "березня";
            break;
        case "4":
            $monthstring = "квітня";
            break;
        case "5":
            $monthstring = "травня";
            break;
        case "6":
            $monthstring = "червня";
            break;
        case "7":
            $monthstring = "липня";
            break;
        case "8":
            $monthstring = "серпня";
            break;
        case "9":
            $monthstring = "вересня";
            break;
        case "10":
            $monthstring = "жовтня";
            break;
        case "11":
            $monthstring = "листопаду";
            break;
        case "12":
            $monthstring = "грудня";
            break;
        default:
            $monthstring = "-1";
    }
}
```

```

$newdate = $date." ".$monthstring." 20".$year." ".$time;
return $newdate; // повернення коду дати
}

```

```

// Функція конвертування дати
function ConvertDateDB($DateDB)
{

```

```

$year = substr($DateDB,0,2);
$month = substr($DateDB,-2);
    switch ( $month ) {
        case "1":
            $monthstring = "січень";
            break;
        case "2":
            $monthstring = "лютий";
            break;
        case "3":
            $monthstring = "березень";
            break;
        case "4":
            $monthstring = "квітень";
            break;
        case "5":
            $monthstring = "травень";
            break;
        case "6":
            $monthstring = "червень";
            break;
        case "7":
            $monthstring = "липень";
            break;
        case "8":
            $monthstring = "серпень";
            break;
        case "9":
            $monthstring = "вересень";
            break;
        case "10":
            $monthstring = "жовтень";
            break;
        case "11":
            $monthstring = "листопад";
            break;
        case "12":
            $monthstring = "грудень";
            break;
        default:
            $monthstring = "-1";
    }

```

```

$newdate = $monthstring." 20".$year;
return $newdate; // повернення коду дати
}

```

```

// Функція отримання поточного місяця
function GetMonths()
{

```

```

    global $link;
    $query = "SELECT yearmonth FROM counter WHERE `ip`=1";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
//друкування результату
    $i = 1;

```

```

while (( $row = mysql_fetch_object( $result ))
{
    $month[$i] = $row -> yearmonth;
    $i++;
}
mysql_free_result( $result );
return $month; // повернення коду місяця
}

// Функція лічильника
function ShowCounter($ip, $dateDB)
{
    global $link;
    $query = "SELECT ip,counter, yearmonth FROM counter WHERE `ip`=$ip AND
`yearmonth`=$dateDB";
    if (!( $result = mysql_query($query, $link ) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    //друкування результату
    echo("<table cellpadding=0 cellspacing=5><tr><td colspan=2
class=left>");
    echo("</td></tr>");
    while (( $row = mysql_fetch_object( $result ))
    {
        printf("<tr><td>%s</td><td>%s</td><td>%s</td></tr>\n",
$row -> yearmonth, $row -> ip, $row -> counter);
    }
    echo("</table>");
    mysql_free_result( $result );
}

// Функція роботи з трафіком
function TrafRemainder($balance, $group, $counter, $moneymonth)
{
    global $link;
    $query_group = "SELECT `trafcost`,`abontraf` FROM `group` WHERE
`group`='$.$group.'";
    if (!( $result_group = mysql_query($query_group, $link ) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $row_group = mysql_fetch_object( $result_group );
    if ($row_group -> trafcost > 0)
    {
        if ($moneymonth=="y")
        {
            if ($counter < ($row_group -> abontraf*1048576))
            {
                $TrafRemainder = number_format(($row_group -> abontraf*1048576-
$counter)/1048576+ConvertMoneyFromDB(($balance)/(($row_group -> trafcost)/100),
2, chr(44), ".");
            } else {
                $TrafRemainder = ConvertMoneyFromDB(($balance)/(($row_group ->
trafcost)/100));
            }
        } else { $TrafRemainder = 0; }
    } else {
        $TrafRemainder = "-";
    }
}
return $TrafRemainder; // повернення коду трафіку
}

// Функція відображення сумарної статистики
function ShowStat()

```

```

{
    global $link, $monthtoshow;
$query1 = "SELECT `ip` FROM `user` WHERE `access`='y'";
$query2 = "SELECT `ip` FROM `user`";
if ($monthtoshow)
{
$query3 = "SELECT `money` FROM `money` WHERE `date` LIKE '". $monthtoshow. "%'";
$query4 = "SELECT `counter` FROM `counter` WHERE `yearmonth` LIKE
'". $monthtoshow. "'";
} else {
$query3 = "SELECT `money` FROM `money` WHERE `date` LIKE
'".GetCurrentDateDB()."%'";
$query4 = "SELECT `counter` FROM `counter` WHERE `yearmonth` LIKE
'".GetCurrentDateDB()."%'";
}

    if (!( $result1 = mysql_query($query1, $link ) ))
    {
        echo ( "Помилка при виконанні $query1
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result2 = mysql_query($query2, $link ) ))
    {
        echo ( "Помилка при виконанні $query2
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result3 = mysql_query($query3, $link ) ))
    {
        echo ( "Помилка при виконанні $query3
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result4 = mysql_query($query4, $link ) ))
    {
        echo ( "Помилка при виконанні $query4
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $activeusers=mysql_num_rows($result1);
    $allusers=mysql_num_rows($result2);
    while ( $row_money = mysql_fetch_object( $result3 ) )
    {
        $allmoney+=$row_money -> money;
    }
    while ( $row_counter = mysql_fetch_object( $result4 ) )
    {
        $alltraffic+=$row_counter -> counter;
    }
    while ( $row_user = mysql_fetch_object( $result2 ) )
    {
        if ($monthtoshow)
        {
            $query5 = "SELECT `counter` FROM `counter` WHERE `ip`='". $row_user -> ip.'" AND
`yearmonth` LIKE '". $monthtoshow. "'";
        } else {
            $query5 = "SELECT `counter` FROM `counter` WHERE `ip`='". $row_user -> ip.'"
AND `yearmonth` LIKE '".GetCurrentDateDB()."%'";
        }
        if (!( $result5 = mysql_query($query5, $link ) ))
        {
            echo ( "Помилка при виконанні $query5
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        $row_counterusers = mysql_fetch_object( $result5 );
        $alltrafficusers+=$row_counterusers -> counter;
    }
?>

```

```

<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF"."?action=");?>">Вернуться</a>
</td></tr>
<tr class="odd"><td valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <tr class="odd">
    <td height=30>Кількість активних користувачів</td><td width="30"
align="center"><? echo $activeusers;?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальна кількість користувачів</td><td width="30"
align="center"><? echo $allusers;?></td>
    </tr>
    <tr class="odd">
    <td height=30>Кількість зібраних грошей (грн.)</td><td width="30"
align="center"><? echo number_format($allmoney,0, chr(44), ".");?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальний трафік:</td><td width="30" align="center"><? echo
number_format($alltraffic,0, chr(44), ".");?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальний трафік (використовуючи таблицю user):</td><td
width="30" align="center"><? echo number_format($alltrafficusers,0, chr(44),
".");?></td>
    </tr>
</table>
</td>
</tr>
<tr class="odd"><td>
    <form method="GET" action="<? echo ("$_PHP_SELF");?>">
    <select name="monthtoshow">
<?
    $month = GetMonths();
    $i=1;
    while ($month[$i])
    {
    if (GetCurrentDateDB()==$month[$i])
    {
    echo "<OPTION VALUE=\"\$month[$i]\"
SELECTED>".ConvertDateDB($month[$i])."</option>";
    } else {
    echo "<OPTION
VALUE=\"\$month[$i]\">".ConvertDateDB($month[$i])."</option>";
    }
    $i++;
    }
?>
    </select>
    <input type="hidden" name="action" value="stat">
    <br>
    <input type="submit" value="Показать">
    </form>
</td></tr>
</table>
<?
}

// Функція оновлення групи
function UpdateGroup()
{
    global $link, $group, $abon, $abontraf, $trafcost, $message;
    $query = "UPDATE `group` SET `abon`='".$abon."',
`abontraf`='".$abontraf."', `trafcost`='".$trafcost.'" WHERE
`group`='".$group.'"';
    if (!( $result = mysql_query($query, $link ) ))
    {

```

```

        echo ( "Помилка при виконанні $query ".mysql_errno($link)." ".
            mysql_error($link) );
            exit();
        } else {
            $message="Updated";
        }
    }

// Функція видалення групи
function DelGroup()
{
    global $link, $group, $message;
    $query = "DELETE FROM `group` WHERE `group`='".$group."'";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    } else {
        $message="Група ".$group." удалена";
    }
}

// Функція додавання групи
function AddGroup()
{
    global $link, $groupname;
    if ($groupname)
    {
        $query_add = "INSERT INTO `group` ( `group` ) VALUES ( '$groupname' )";
        if (!( $result = mysql_query($query_add, $link) ))
        {
            echo ( "Помилка при виконанні $query_add
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
    }
}

// Функція редагування групи
function EditGroup()
{
    global $link, $group, $message;
    ?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF.">action=");?>">Вернуться</a>
</td></tr>
<tr class="odd"><td valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="3" cellspacing="1" border="0">
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
<tr class="even"><td height=30>
    Група:
</td>
<td bgcolor="#ffffff" width="110">
<select name="group">
<?
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    while ( $row_group = mysql_fetch_object( $result_group ) )
    {
        if ($row_group -> group == $group)

```

```

        {
        ?>
        <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
        <?
        } else {
        ?>
        <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
        <?
        }
        }
?>

</select>
</td></tr>
<tr class="odd"><td></td><td height=30 align="left">
<input type="submit" value="Показати">
<input type="hidden" name="action" value="editgroup" >
</form>
</td></tr>
<tr class="even"><td colspan="2">
Додати групу:
</td></tr>
<tr class="odd"><td>
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
<input type="text" size=8 name="groupname">
</td><td>
<input type="hidden" name="action" value="addgroup" >
<input type="submit" value="Додати">
</form>
</td></tr>
</table>
</td></tr>
<? if ($group)
{?>
<tr class="odd"><td>
<a href="<?echo ("$_PHP_SELF"."?action=delgroup&group=$group")?>"
onclick="return confirmLink(this, 'видалити групу `<? echo
$group;?>`')">Видалити групу <?echo $group;?></a>
</td></tr>
<?}?>
</table>
</td>

<td width="500" align="right" valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr><td>
<?
    if ($group != "")
    {
        $query_group = "SELECT `trafcost`,`abon`,`abontraf` FROM `group` WHERE
`group`='$group'";
        if (!( $result_group = mysql_query($query_group, $link ) ) )
        {
            echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        $row_group = mysql_fetch_object( $result_group );
    ?>

    <table cellpadding="0" cellspacing="0" border="0">
    <tr class="edge"><td>
    <form name="updategroup" method="POST" action="<?echo
("$_PHP_SELF");?>>
    <table cellpadding="2" cellspacing="1" border="0">
        <tr class="odd"><td>Вартість трафіка (грн. за 100
Мбайт):</td><td align="right"><input type="text" name="trafcost" value="<? echo
$row_group -> trafcost; ?>" size="2"></td></tr>

```

```

                <tr class="odd"><td>Абонентська плата (грн. за
місяць):</td><td align="right"><input type="text" name="abon" value="<? echo
$row_group -> abon; ?>" size="2"></td></tr>
                <tr class="odd"><td>Кількість трафіка, вхідного в
абонентську плату (Мбайт):</td><td align="right"><input type="text"
name="abontraf" value="<? echo $row_group -> abontraf; ?>" size="2"></td></tr>
                <tr class="odd"><td colspan="2" align="right"><input
type="submit" value="Відіслати"></td></tr>
            </table>
        </td></tr>
        <input type="hidden" name="group" value="<? echo $group; ?>">
        <input type="hidden" name="action" value="updategroup">
    </form>
</table>
<?
}
?>
</td></tr>
</table>
</td>
<td valign="top" align="right" width="70"><font color="red"><? echo
$message;?></td>
</tr>
</table>
<?
}

// Функція додавання користувача
function AddUser()
{
    global $link, $login, $ip, $name, $address, $moneyinet, $moneynetwork,
$balance, $group, $trafficebegin, $SubtractAb;
    $balanceDB = ConvertMoneyToDB($balance);
    $query_user = "INSERT INTO `user` ( `login`, `ip`, `name`, `address`,
`moneyinet`, `moneynetwork`, `balance`, `group`) VALUES ( '$login', '$ip',
'$name', '$address', '$moneyinet', '$moneynetwork', '$balanceDB', '$group' )";
    if (!( $result = mysql_query($query_user, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_user
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if ($trafficebegin)
    {
        $query_counter = "UPDATE `counter` SET `counter`='$trafficebegin',
`moneymonth`='$SubtractAb' WHERE `ip`='$ip' AND
`yearmonth`='".GetCurrentDateDB()."''";
    } else {
        $query_counter = "UPDATE `counter` SET `counter`='0',
`moneymonth`='$SubtractAb' WHERE `ip`='$ip' AND
`yearmonth`='".GetCurrentDateDB()."''";
    }
    if (!( $result_counter = mysql_query($query_counter, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_counter
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $date = GetCurrentDateDBMoney();
    $query_money = "INSERT INTO `money` ( `ip`, `money`, `date`) VALUES (
'$ip', '$balance', '$date' )";
    if (!( $result = mysql_query($query_money, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_money
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
}
}

```

```

// Функція форми користувача
function AddUserForm()
{
    global $link;
?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <form method="GET" action=<?echo ("$_PHP_SELF");?>>
    <tr class="even"><td height=30>
        Login:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="login" size="12">
    </td>
    <td bgcolor="#ffffff">
        маленькими літерами
    </td>
    </tr>
<tr class="even"><td height=30>
        IP:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="ip" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться тільки остання частина ip-адреса
    </td>
    </tr>
<tr class="even"><td height=30>
        Имя:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="name" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться у вигляді: Фамилія Ім'я
    </td>
    </tr>
<tr class="even"><td height=30>
        Адреса:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="address" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться в вигляді: 15/1 7-2-3
    </td>
    </tr>
<tr class="even"><td height=30>
        Деньги за сеть:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="moneynetwork" size="12">
    </td>
    <td bgcolor="#ffffff"></td>
    </tr>
<tr class="even"><td height=30>
        Деньги за инет:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="moneyinet" size="12">
    </td>
    <td bgcolor="#ffffff"></td>
    </tr>

```

```

</tr>
<tr class="even"><td height=30>
    Баланс:
</td>
<td bgcolor="#ffffff">
    <input type="text" name="balance" size="12">
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Вилучити абонентську плату?
</td>
<td bgcolor="#ffffff">
    <input type="radio" name="SubtractAb" value="n" checked>Да&nbsp;
    <input type="radio" name="SubtractAb" value="y">Нет&nbsp;
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Початковий трафік (байт):
</td>
<td bgcolor="#ffffff">
    <input type="text" name="trafficbegin" value="0" size="12">
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Група:
</td>
<td bgcolor="#ffffff">
    <select name="group">
<?
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    while ( $row_group = mysql_fetch_object( $result_group ) )
    {
        if ($row_group -> group=="comp")
        {
            ?>
            <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
            <?
            } else {
            ?>
            <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
            <?
            }
        }
    }
?>
    </select>
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="odd"><td></td><td height=30 align="left">
    <input type="submit" value="Додати">
</td>
<td bgcolor="#ffffff"></td>
</tr>
    <input type="hidden" name="action" value="adduser" >
</form>
</table>
</td></tr>
</table>

```

```

</td>
</tr>
</table>
<?
}

// Функція видалення користувача
function DeleteUser ()
{
    global $link, $ip;
    $query = "DELETE FROM `user` WHERE `ip`='". $ip. "'";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query ".mysql_errno($link)."
            ".mysql_error($link) );
        exit();
    }
    $query_counter = "UPDATE `counter` SET `counter`='0', `moneymonth`='n'
WHERE `ip`='$ip' AND `yearmonth`='".GetCurrentDateDB()." ";
    if (!( $result_counter = mysql_query($query_counter, $link) ))
    {
        echo ( "Помилка при виконанні $query_counter
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
}

// Функція оновлення даних користувача
function UpdateUser ()
{
    global $link, $message, $ip, $login, $group, $address, $moneynetwork,
$moneynet;
    $query = "UPDATE `user` SET `login`='". $login. "', `group`='". $group. "',
`address`='". $address. "', `moneynetwork`='". $moneynetwork. "',
`moneynet`='". $moneynet. "' WHERE `ip`='". $ip;
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    } else {
        $message = "Updated";
    }
}

// Функція додавання до балансу
function IncBalance ()
{
    global $link, $message, $ip, $incbalance ;
    if ($incbalance)
    {
        $query = "UPDATE `user` SET balance = balance +
".ConvertMoneyToDB($incbalance)." WHERE `ip`='". $ip;
        if (!( $result = mysql_query($query, $link) ))
        {
            echo ( "Помилка при виконанні $query ".mysql_errno($link)."
            ".mysql_error($link) );
            exit();
        }
        $date = GetCurrentDateDBMoney();
        $query_money = "INSERT INTO `money` ( `ip`, `money`, `date`) VALUES (
            '$ip', '$incbalance', '$date' )";
        if (!( $result = mysql_query($query_money, $link) ))
        {
            echo ( "Помилка при виконанні $query_money
            ".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        } else {
            $message = "Updated";
        }
    }
}

```

```

    }
}

// Функція відображення даних адміністратора
function ShowUserAdmin($ip, $month)
{
    global $link, $message, $showpayment;
    $query_user = "SELECT `name`, `login`, `access`, `balance`, `group`, `address`,
        `moneynetwork`, `moneyinet` FROM user WHERE `ip`=$ip";
    if (!( $result_user = mysql_query($query_user, $link) ))
    {
        echo ( "Помилка при виконанні $query_user
            ".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $row_user = mysql_fetch_object( $result_user );
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group ".mysql_errno($link)."
            ".mysql_error($link) );
        exit();
    }
}

?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo (" $PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd" valign="top"><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<form name="update" method="POST" action=<?echo (" $PHP_SELF");?>>
<table cellpadding="2" cellspacing="1" border="0">
<?
                $query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`=". $ip ." AND `yearmonth`=". $month;
                if (!( $result_counter = mysql_query($query_counter, $link) ))
                {
                    echo ( "Помилка при виконанні $query_counter
                        ".mysql_errno($link)." ".mysql_error($link) );
                    exit();
                }
                $row_counter = mysql_fetch_object( $result_counter );
?>
<tr class="odd"><td>Дані за:</td><td align="right"><? echo
ConvertDateDB($month);?></td></tr>
<tr class="even"><td>IP:</td><td align="right"><? echo $ip;?></td></tr>
<tr class="odd"><td>Login:</td><td align="right"><input type="text" name="login"
value="<? echo $row_user -> login; ?>" size="8"></td></tr>
<tr class="even"><td>Група:</td><td align="right">

                <select name="group">
?>
                while ( $row_group = mysql_fetch_object( $result_group ) )
                {
                    if ($row_group -> group==$row_user -> group)
                    {
                        ?>
                        <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
                        <?
                    } else {
                        ?>
                        <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
                        <?
                    }
                }
?>

```

```

    }
  }
?>
    </select>
        </td></tr>
        <tr class="odd"><td>Имя:</td><td align="right"><? echo
$row_user -> name; ?></td></tr>
        <tr class="even"><td>Адрес:</td><td align="right"><input
type="text" name="address" value="<? echo $row_user -> address; ?>"
size="8"></td></tr>
        <tr class="odd"><td>Гроші за мережу:</td><td
align="right"><input type="text" name="moneynetwork" value="<? echo $row_user ->
moneynetwork; ?>" size="2"></td></tr>
        <tr class="even"><td>Гроші за инет:</td><td
align="right"><input type="text" name="moneynet" value="<? echo $row_user ->
moneynet; ?>" size="2"></td></tr>
        <tr class="odd"><td>Трафік (байт):</td><td
align="right"><? echo number_format($row_counter -> counter,0, chr(44), ".");
?></td></tr>
        <tr class="even"><td>Баланс (грн.):</td><td
align="right"><font color="red"><? echo ConvertMoneyFromDB($row_user ->
balance); ?></font></td></tr>
        <tr class="odd"><td>balance:</td><td align="right"><?
echo $row_user -> balance; ?></td></tr>
        <tr class="even"><td>Залишилося (МБ):</td><td
align="right"><? echo TrafRemainder($row_user -> balance, $row_user -> group,
$row_counter -> counter, $row_counter -> moneymonth); ?></td></tr>
        <tr class="odd"><td>Доступ:</td><td align="right"><?
echo $row_user -> access; ?></td></tr>
        <tr class="even"><td>Абон. плата:</td><td
align="right"><? echo $row_counter -> moneymonth; ?></td></tr>
        <tr class="odd"><td colspan=2 align="center"><input
type="submit" value="Відправити"></td></tr>
        <input type=hidden name="ip" value="<? echo $ip;?>" >
        <input type=hidden name="month" value="<? echo $month;?>" >
        <input type=hidden name="action" value="update" >
        </form>
    </table>
</td></tr>
</table>
</td>
    <td></td>
    <td valign="top" align="right" width="300">
    <table cellpadding="0" cellspacing="0" border="0">
    <tr class="edge"><td>
        <table cellpadding="2" cellspacing="1" border="0">
        <form name="incbalance" method="POST" action=<?echo ("$_PHP_SELF");?>>
        <tr class="odd">
            <td>Баланс (грн.):</td>
            <td><font color="red"><? echo ConvertMoneyFromDB($row_user -> balance);
?></font></td>
        </tr>
        <tr class="odd">
            <td>Додати (грн.):</td>
            <td><input type="text" name="incbalance" size="8"></td>
        </tr>
        <tr><td bgcolor="#ffffff" height=40 colspan=2 align="center">
            <input type=hidden name="ip" value="<? echo $ip;?>" >
            <input type=hidden name="month" value="<? echo $month;?>" >
            <input type=hidden name="action" value="incbalance" >
            <input type="submit" value="Відправити">
        </td></tr>
        </form>
    </table>
</td></tr>
</table>

<table>
    <tr class="odd" colspan="2">

```

```

<td colspan="2" align="right">
<?
if ($showpayment)
{?>
<a href="<? echo ("$_PHP_SELF"."?action=show&ip=$ip&month=$month");?>">Вилучити
платежі</a>
</td></tr>
</table>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="2" cellspacing="1" border="0">
<tr class="odd"><td>Дата</td><td>руб.</td></tr>
<?
$query_money = "SELECT money, date FROM money WHERE `ip`=".$ip." AND `date` like
'".$month."%";
if (!( $result_money = mysql_query($query_money, $link ) ))
{
echo ("Помилка при виконанні $query_money ".mysql_errno($link)."
.mysql_error($link) );
exit();
}
while ($row_money = mysql_fetch_object( $result_money ))
{
echo "<tr class=\"odd\"><td>".ConvertDateDBMoney($row_money ->
date)."</td><td>".$row_money -> money."</td></tr>";
$allmoney+=$row_money -> money;
}
echo "<tr class=\"odd\"><td>Итого:</td><td>".$allmoney."</td></tr>";
echo "</table>";
} else {?>
<a href="<? echo
("$_PHP_SELF"."?action=show&ip=$ip&month=$month&showpayment=1");?>">
Дивитися платежі</a>
<?>?>
</td></tr>
</table>
</td>
<td valign="top"><font color="red"><? echo $message;?></td>
</tr>
</table>
<?
}

// Функція відображення користувача
function ShowUser($ip)
{
global $link;
$query_user = "SELECT `name`, `login`, `access`, `balance`, `group` FROM
user WHERE `ip`=$ip";
if (!( $result_user = mysql_query($query_user, $link ) ))
{
echo ( "Помилка при виконанні $query ".mysql_errno($link)."
.mysql_error($link) );
exit();
}
$row_user = mysql_fetch_object( $result_user );
//друкування результату
?>

<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="2" cellspacing="1" border="0">
<?
$query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`=".$ip." AND `yearmonth`=".GetCurrentDateDB();
if (!( $result_counter =
mysql_query($query_counter, $link ) ))
{

```

```

echo ( "Помилка при виконанні
$query_counter ".mysql_errno($link)." ".mysql_error($link) );
exit();
}
$row_counter = mysql_fetch_object(
$result_counter );
// виведення даних на екран
printf("
<tr class=\"odd\"><td>Дані за:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Login:</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Ім'я:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Трафік (байт):</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Баланс (грн.):</td><td align=\"right\"><font
color=\"red\">%s</font></td></tr>
<tr class=\"even\"><td>Залишилося (Мб):</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Доступ:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Абон. плата:</td><td align=\"right\">%s</td></tr>
\n",
ConvertDateDB(GetCurrentDateDB()), $row_user -> login, $row_user -> name,
number_format($row_counter -> counter,0, chr(44), "."),
ConvertMoneyFromDB($row_user -> balance), TrafRemainder($row_user -> balance,
$row_user -> group, $row_counter -> counter, $row_counter -> moneymonth),
$row_user -> access, $row_counter -> moneymonth);
?>
</table>
</td></tr>
</table>
<?
}

// Функція попередження
function ResolveUser()
{
global $network, $conf;
$host = getenv('REMOTE_ADDR');
$ip=explode(".", $host);
$network = $ip[0].".$ip[1].".$ip[2].".";
if ($network != $conf{'network'})
{
echo ("Зареєструйтеся в VPN.");
exit();
}
return $ip[3]; // повернення коду IP
}

// Функція роботи з датою
function GetCurrentDateDB()
{
$current_date=time();
$dateDB=date("ym", $current_date);
return $dateDB; // повернення коду дати у БД
}

function GetCurrentDateDBMoney()
{
$current_date=time();
$date=date("ymd H:i:s", $current_date);
return $date; // повернення коду дати внесення коштів
}

function Stuff()
{
$current_date=time();
echo $current_date."<br>";
$date=date("ymd H:i:s", $current_date);
$dateDB=date("ym", $current_date);
echo $date."<br>".$dateDB."<br>";
echo ConvertDateDBMoney($date);
}

```

```
// функція формування початку HTML сторінки
function Top($path,$pagename)
{
?>
<html>
<head>
<title>Моніторинг трафіку -> <? echo $pagename ?></title>
<META HTTP-EQUIV="Expires" CONTENT="Fri, Jan 01 1900 00:00:00 GMT">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1251">
<link rel="stylesheet" type="text/css" href="<? echo $path; ?>Monitoring.css">

<script type="text/javascript" language="javascript">
<!--
// коди та текст помилок
var errorMsg0 = 'Потрібно значення для форми!';
var errorMsg1 = 'Це не число!';
var errorMsg2 = 'Неприпустима кількість рядків!';
var noDropDbMsg = 'Команда "Видалити БД" відключена.';
var confirmMsg = 'Ви дійсно бажаєте ';
//-->
</script>
<script src="<? echo $path;?>functions.js" type="text/javascript"
language="javascript"></script>
</head>
<body>
<?
}

// формування HTML коду кінця сторінки
function Footer()
{
?>
</body>
</html>
<?
}
?>
```