

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  
“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“ Дослідження та програмна реалізація системи веб-чату для**  
**обміну даними в мережі Інтернет”**

Виконав здобувач вищої освіти  
II курсу, групи KI22M-1  
ОПП «Комп'ютерна інженерія»  
спеціальності 123 «Комп'ютерна інженерія»  
\_\_\_\_\_ Амбросьєв Д.А  
« \_\_\_\_ » \_\_\_\_\_ 2023р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Пархоменко Ю.М.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Освітній ступінь магістр  
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  
“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Амбросьєву Дмитру Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи веб-чату для обміну даними в мережі Інтернет

2. Керівник роботи Пархоменко Юрій Михайлович, канд. техн. наук, доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 34-13 від 04.08.23

3. Строк подання роботи до захисту 20.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є програмне дослідження та програмна реалізація системи веб-чату

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання. 7. Економічна ефективність

2. Перегляд аналогічних існуючих систем. розробленої програми.

3. Опис і обґрунтування проектних рішень. 8. Заходи з охорони праці та техніки

4. Етапи програмування системи. безпеки.

5. Впровадження системи в промислову експлуатацію. 9. Висновки.

експлуатацію.

6. Наукова новизна

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна 1 аркуш

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

Діаграма процесів 1 аркуш

Показники економічної ефективності 1 аркуш

## 6. Консультанти по роботі, із зазначенням розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В., к.т.н., доцент	09.11.2023 р.	17.11.2023 р.
Охорона праці	Оришака О.В., к.т.н., доцент	03.11.2023 р.	21.11.2023 р.

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	12.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	18.10.2023 р.	
3.	Розробка моделі компонента	23.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	32.10.2023 р.	
6.	Програмування алгоритмів	11.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	16.11.2023 р.	
9.	Оформлення ПЗ	18.11.2023 р.	
10.	Попередній захист роботи	04.12.2023 р.	

Дата видачі завдання  
«\_\_»\_\_\_\_\_20 р.

Підпис керівника

\_\_\_\_\_ (прізвище та ініціали)

Завдання прийнято до виконання  
«\_\_»\_\_\_\_\_20 р.

Підпис здобувача

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

**Амбросьєв Д.А. Дослідження та програмна реалізація системи веб-чату для обміну даними в мережі Інтернет. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.**

В даній магістерській роботі розроблено програмне забезпечення, яке призначено для впровадження системи веб-чату для обміну даними в мережі Інтернет.

Метою розробки є дослідження та програмна реалізація системи веб-чату для обміну даними в корпоративних мережах.

Об'єктом дослідження є процес розробки веб-чатів.

Предметом дослідження є методи систем веб-чату для спілкування в корпоративних мережах.

Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація веб-застосунку засобами технології семантичного вебу та мов програмування.

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 у браузері Chrome, Firefox, Safari.

Програму розроблено в середовищі PHP 7.3 та JS.

**Ключові слова:** комп'ютерна інженерія, веб-сайт, семантичний веб.

## ABSTRACT

**Ambrosiev D.A. Research and software implementation of a web chat system for data exchange on the Internet. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.**

In this master's thesis, software was developed, which is intended for the implementation of a web chat system for data exchange on the Internet.

The goal of development is research and software implementation of a web chat system for data exchange in corporate networks.

The object of research is the process of developing web chats.

The subject of the research is methods of web chat systems for communication in corporate networks.

Research methods are based on coding theory methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of a web application using semantic web technology and programming languages.

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 in Chrome, Firefox, and Safari browsers.

The program was developed in PHP 7.3 and JS environment.

**Keywords:** computer engineering, website, semantic web.



7.3	Визначення чисельності виконавців і планового фонду зарплати.....	65
7.4	Розрахунок капітальних вкладень та амортизаційних відрахувань.....	69
7.5	Визначення собівартості розробки та ціни програмної продукції.....	74
7.6	Визначення об'єму капітальних вкладень у споживача .....	77
7.7	Визначення експлуатаційних витрат .....	78
7.8	Визначення економічної ефективності програмної продукції .....	78
7.9	Висновки .....	81
8	ЗАХОДИ ЩОДО ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ.....	82
8.1	Вступ.....	82
8.2	Шкідливі і небезпечні фактори при роботі з комп'ютером.....	84
8.3	Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	81
8.4	Розробка заходів з умов поліпшення охорони праці.....	87
8.5	Розрахункова частина .....	88
9	ОСНОВНІ ВИСНОВКИ.....	92
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	92

КБПЗ-2023

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>2</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

- ПЗ – Програмне забезпечення.
- AJAX – Asynchronous Javascript and XML – підхід до побудови інтерактивних користувацьких інтерфейсів веб-додатків.
- CSS3 – Cascading Style Sheets 3 – каскадні таблиці стилів третього покоління.
- HTML – HyperText Markup Language – мова розмітки гіпертексту.
- PHP – Hypertext Preprocessor – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків.
- БД – База даних.
- PDO – PHP Data Objects – розширення для PHP, надає розробнику простий і універсальний інтерфейс для доступу до різних баз даних.
- JSON – JavaScript Object Notation – текстовий формат обміну даними.
- MD5 – Message Digest 5 – 128-бітний алгоритм хешування.
- SQL – Structured query language – мова структурованих запитів.
- ПК – Персональний комп'ютер.
- ПЕОМ – Загальна назва для обчислювальних машин, що є електронними.
- СУБД – системи управління базами даних.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>3</b>

## ВСТУП

**Актуальність теми.** Інтернет зайняв важливе місце в житті сучасної людини. Сьогодні вже неможливо уявити життя без комп'ютера і, звичайно, доступу до всесвітньої павутини. Мільярди людей постійно «блукають» по Інтернету в пошуках необхідної інформації.

На сьогоднішній день сучасне уявлення про Інтернет істотно змінилося з тих пір, як сфера інформаційних технологій почала широко розвиватися. Інтернет зазнав великих змін, які в свою чергу дозволяють побудувати принципово нові уявлення про Інтернет-технології та їх застосування. В даний час кожен веб-ресурс в Інтернеті побудований на інноваційних принципах. Існує багато веб-технологій, які використовуються для побудови та створення веб-ресурсів (сайтів).

Створення веб-сайту – це створення маркетингового інструменту, призначеного для стимулювання попиту на певні послуги чи товари, або інформаційного ресурсу, який спрямований на донесення необхідної інформації до цільової аудиторії, або створення сервісного ресурсу, який би вирішував завдання надання певні послуги зацікавленим відвідувачам. При цьому варто пам'ятати, що розробка сайту, здатного якісно виконувати всі ці функції, є досить складним процесом і вимагає високого професійного рівня виконання.

Якщо на сайті є онлайн-чат, то ресурс отримує від цього багато переваг.

Розвиток всесвітньої павутини диктує певні закони: щоб сторінка стала корисною та відвідуваною, вона має бути інтерактивною. Для цього призначений скрипт «онлайн-консультант». З його допомогою відвідувач отримує підтримку в пошуку необхідної інформації та технічної підтримки.

Придатність і зручність. Зараз, коли Інтернет став невід'ємною частиною життєдіяльності практично кожного громадянина, жителі замість того, щоб телефонувати чи писати повідомлення, набагато більше готові спілкуватися за

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

допомогою інтерактивного спілкування.

Скрипт «онлайн-консультант» дає можливість одним натисканням миші зв'язатися зі співробітником компанії і відразу почати пряме спілкування.

Ваша служба технічної підтримки буде весь час справна і працюватиме набагато ефективніше, адже вам доведеться постійно та швидко реагувати на запити відвідувачів. Використовуючи онлайн-чат на сайті, можна досягти високого рівня конверсії трафіку. За допомогою онлайн-чату консультант встановлює живий контакт в режимі реального часу і всі питання вирішуються швидше. Слід також додати, що в Інтернеті присутність на сторінці системи онлайн-чату давно стала ознакою доброго тону.

Отже, виходячи з вищесказаного, розробка програмного забезпечення систем веб-чату для обміну даними в мережі Інтернет є актуальним завданням, яке потребує вирішення в даній магістерській роботі.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи веб-чату для обміну інформацією в мережі Інтернет.

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

- огляд існуючих систем веб-чату;
- дослідження аналогічних системи веб-чату;
- програмна реалізація системи веб-чату з сучасними технологіями.

*Об'єктом дослідження* є процес передачі даних в системі веб-чат.

*Предметом дослідження* є методи реалізації систем передачі даних в мережі Інтернет.

*Методи дослідження* базуються на методах теорії кодування, методах математичної статистики, методах розробки веб-застосунків.

**Наукова новизна отриманих результатів.** У процесі написання завдань, визначених цілями дослідження, отримано наступні результати:

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

- вдосконалена система передачі даних онлайн-чату з розширенням підсистеми безпеки передачі даних;
- переглянуто технологічну передачу миттєвих повідомлень у режимі реального часу;
- розроблено вітчизняний продукт для передачі повідомлень із захищеною підсистемою передачі даних, який має більші можливості, на відміну від існуючих аналогів, і дозволить досягти економічного ефекту від впровадження.

**Практична цінність отриманих результатів** полягає в тому, що запропоновані алгоритми дозволяють успішно складати завдання для потреб передачі даних у режимі реального часу.

**Достовірність наукових результатів** підтверджується теоретичними положеннями, даними комп'ютерного моделювання, комплексними дослідженнями параметрів на функціональній розрахунковій межі, а також відповідністю отриманих результатів конкретним результатам, наведеним у науковій літературі.

Отже, виходячи з вищевикладеного, дослідження та програмна реалізація системи веб-чату є актуальним завданням, яке потребує вирішення в даній магістерській роботі.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

У даній магістерській роботі детально розглянуто та представлено технологію побудови інформаційної системи обміну даними. Завданням було створити веб-ресурс, який забезпечує можливість обміну повідомленнями між користувачами, які зареєстровані або авторизовані в системі, завдяки технології асинхронного обміну даними, що дозволить відправляти, редагувати та видаляти повідомлення в асинхронній формі без перевантаження сторінки. Це дозволить мені використовувати в своїй роботі інноваційний спосіб створення веб-додатків та створити сучасний програмний продукт, систему обміну повідомленнями між користувачами та службу підтримки на сайті.

## 1.1 Призначення системи

Веб-чати для обміну даними використовуються в різних галузях і для різних цілей. Нижче наведено приклади застосувань веб-чатів для обміну даними:

– корпоративний комунікаційний інструмент: Великі компанії та організації можуть використовувати веб-чати для обміну даними між співробітниками. Це може включати в себе обмін текстовими повідомленнями, відеоконференції, аудіо-дзвінки і спільну роботу над документами;

– клієнтська підтримка: Багато компаній надають можливість клієнтам звертатися до них через веб-чат для отримання підтримки та вирішення питань. Це може включати в себе чат-ботів, які надають відповіді на поширені запитання;

– онлайн-торгівля: Веб-чати також використовуються в онлайн-торгівлі для обміну інформацією між покупцями і продавцями. Вони можуть використовуватися для запитів про товари, надання консультацій та обміну інформацією про замовлення;

– медицина: Веб-чати в медицині можуть використовуватися для обміну

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

медичною інформацією між лікарями і пацієнтами. Це може включати в себе консультації в режимі реального часу, нагадування про лікування і обмін даними про результати аналізів;

– освіта: У сфері освіти веб-чати можуть бути використані для спілкування вчителів і студентів, обміну навчальними матеріалами та завданнями, а також для вирішення питань і проблем;

– фінанси: У фінансовому секторі веб-чати можуть використовуватися для обміну інформацією про банківські операції, запитання щодо фінансових продуктів та послуг, а також для підтримки клієнтів;

– технічна підтримка: Компанії в галузі технічної підтримки можуть використовувати веб-чати для вирішення проблем і запитань клієнтів щодо продуктів і послуг;

– соціальні мережі: Соціальні мережі також надають можливість спілкування через веб-чати для обміну повідомленнями, фотографіями і відео зі своїми друзями та підписниками.

Ще одним каналом спілкування є класичні консультації в онлайн-чаті на сайті. Якщо клієнт знаходиться на сайті компанії, то зручно запитувати прямо там. І ця система створена спеціально для технічної підтримки користувачів. Ця система призначена для приватного спілкування в чаті зі службою підтримки.

Оскільки офіційна інформація, така як (логін, пароль, номери облікових записів), може передаватися через нього, існуючі системи, розроблені для цього, слід встановлювати та використовувати дуже обережно.

## 1.2 Область застосування

Основною метою розробки цього веб-додатку є розробка приватного (закритого) чату для технічної підтримки користувачів.

Приватний онлайн чат - це форма комунікації через Інтернет, яка передбачає обмін текстовими повідомленнями, аудіо- або відеовикликами між

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

двома або більше особами в режимі реального часу, при цьому доступ до чату обмежений тільки для певних учасників і не доступний для публічного перегляду або участі. Основні характеристики приватного онлайн чату включають:

- приватність: Приватний онлайн чат забезпечує конфіденційність та обмежений доступ до розмови лише для учасників, які брали участь у чаті або мають відповідний доступ:

- реальний час: Чати в реальному часі дозволяють учасникам спілкуватися один з одним миттєво, без затримок;

- різноманітність медіа: В приватних чатах можна обмінюватися не лише текстовими повідомленнями, але і медіафайлами, такими як фотографії, відео, аудіозаписи тощо;

- онлайн-комунікація: Цей вид чатів використовується для спілкування між людьми, які перебувають в різних місцях і можуть бути підключені до Інтернету;

- особистість учасників: У приватних чатах зазвичай відома особистість усіх учасників, що відрізняє їх від анонімних чатів;

- захист від спаму та несанкціонованого доступу: Для забезпечення безпеки і приватності чату можуть використовуватися різні методи аутентифікації і контролю доступу.

Окрім збільшення продажів, онлайн-чат також можна використовувати для технічної (і не тільки) підтримки клієнтів. Чим у цьому відношенні онлайн-чат кращий за традиційні методи спілкування? По-перше, це допомагає користувачеві вирішити проблему швидше, ніж електронна пошта. По-друге, на відміну від телефонного дзвінка, оператор технічної підтримки не обмежується однією розмовою за раз: маючи відповідний досвід, оператор технічної підтримки може вести відразу кілька активних діалогів.

Отже, виходячи з вище сказаного, розробка програмного забезпечення веб-чату для обміну даними в мережі Інтернет є актуальним завданням, яке потребує вирішення в даній магістерській роботі.

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

### 2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми магістерської роботи

Основна функціональність системи обміну даними подібна до комунікаційних систем типу чат, які зараз використовуються в усіх соціальних мережах, веб-чатах, поштових службах та системах онлайн-зв'язку. Всі ці системи працюють за одним схожим принципом: автоматичне оновлення контенту без перевантаження сайту, це основа моєї магістерської роботи. Крім цього ще однією задаєю є проведення аналізу аналогічних систем.

#### phpBB Simple Chat v2.1

Аналогом розробки по темі моєї роботи є публікація по темі Ajax Simple Chat v2.1. Розробка російських розробників, що дозволяє обмінюватися повідомленнями в Інтернеті і використовувати інтеграцію в свій веб-ресурс. Простий чат працює як для phpBB, так і для phpBBек. Ця розмова поширюється з відкритим кодом, тому є можливість її редагувати або доповнювати. Широко використовується на форумах і сайтах, де є велика кількість клієнтів і необхідно організувати принцип обміну повідомленнями між зареєстрованими користувачами. Чат повністю розроблений на основі технології Ajax.

Порівнюючи його структуру роботи та функціональність, я можу зробити висновок, що алгоритм роботи типовий для веб-проектів такого типу. У розробці використовується схожий алгоритм роботи системи, дурний чат, який так само був розроблений у темі мого дипломного дизайну.

Використання phpBB Simple Chat v2.1 є безліцензійним, а використання з відкритим кодом дозволено як для власних потреб, так і для модифікації продукту.

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Алгоритм роботи phpBB Simple Chat v2.1:

- аутентифікація в системі користувача;
- увійти в профіль користувача;
- перевірка наявності спаму повідомлень;
- система сповіщення про повідомлення;
- відправка повідомлень;
- видалення повідомлень;
- редагування повідомлень.

Алгоритм роботи системи phpBB Simple Chat дуже простий і схожий на алгоритм роботи програми, розробленої в цьому дипломному проекті. Є можливість редагувати та видаляти повідомлення, відправка здійснюється повністю в автономному режимі, не перевантажуючи сайт.

Важливою відмінністю буде те, що надходження нових повідомлень сповіщається звуковим сповіщенням, яке не підтримується в додатку для створення дипломного проекту. Інтегрувати phpBB Simple Chat у ваш власний веб-сайт дуже просто, а також використовувати його як окремий модуль форуму або для власного веб-сайту. Прогресивною частиною буде той факт, що існує можливість адаптації майже до різних веб-систем, створених або розроблених за допомогою підходу Ajax. Основною особливістю phpBB Simple Chat є використання доступу Ajax і надсилання даних на сервер у фоновому режимі.

### **AjaxChat – чат для CMS Joomla**

AjaxChat — це компонент програми Joomla, який використовується як система обміну повідомленнями між користувачами сайту. Є можливість відправляти повідомлення, редагувати, видаляти і створювати окремі кімнати для користувачів з різними правами доступу. Система повністю налаштовується під сайт, тобто легко підключається до веб-ресурсу без зміни частин коду. У інтерфейсі можна змінити мову інтерфейсу та встановити колір повідомлень.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Повністю підтримується технологією Ajax, повідомлення оновлюються асинхронно без перевантаження веб-сайту, що доводить схожість цієї розробки з магістерською роботою.

Алгоритм компонентів:

- реєстрація, авторизація в системі;
- профіль користувача;
- відправка, видалення, редагування повідомлень;
- укорінення системи;
- видалити історію.

Порівнюючи роботу цієї системи з тією, яку я розробив у дипломному проєкті, я дійшов висновку, що загальний функціонал і алгоритм роботи дуже схожі, а використання підходу Ajax нічим не відрізняється. Для порівняння функціональності роботи була використана тестова розсилка повідомлень між двома користувачами в мережі, кожен з яких був авторизований в системі. Результат роботи програми, яку я розробив у випускному проєкті, і AJAX CHAT був подібним, повідомлення надсилалися без затримки в системі обміну повідомленнями та ідентифікувалися в базі даних відразу після надсилання, сповіщення організовувалися за допомогою ідентифікаторів статусу повідомлень, що схоже на мою програму для дизайн. Повністю настроювана, некомерційна версія з відкритим кодом.

### **Chatgoo**

Подібний веб-чат був створений командою розробників проєкту Joomla на основі php miskl і підтримується технологією Ajax. Використовується як модуль для підключення до веб-ресурсу, не має ліцензії та поширюється у відкритому коді. Основною функцією є обмін повідомленнями між користувачами сайту та створення чатів.

Алгоритм роботи Chatgoo (AJAX):

- реєстрація та авторизація користувачів;
- профільна частина користувача;

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

- кімната для спілкування;
- відправка повідомлень;
- видалення повідомлень;
- редагування повідомлень;

Алгоритм системи дуже простий і розділений на модулі. Частина модулів відповідає за інтерфейс користувача, а інша частина — за профіль користувача. Зручний принцип управління профільною частиною користувача. Є можливість вибрати колір для повідомлень, все це створює зручну і просту комунікаційну модель сайту.

### **Universal Ajax chat**

Універсальний чат для системи управління контентом Joomla. Використовується для організації системи спілкування користувача на сайті, підключається як окремий модуль. Класичний приклад веб-додатку, розробленого на php 5 з підтримкою бази даних MySQL і повною підтримкою інтерфейсу користувача ajax. Алгоритм роботи нагадує звичайну систему спілкування типу чату з діалогом користувача, який ведеться окремо в кімнатах спілкування.

Універсальний алгоритм роботи Ajax chat:

- реєстрація та авторизація користувачів;
- профільна частина користувача;
- кімната для спілкування;
- відправка повідомлень;
- видалення повідомлень;
- редагування повідомлень.

Як бачимо, існує багато готових рішень для вирішення задачі дослідження та проектування. Враховуючи всі конструктивні та компонентні рішення, можна зробити висновок, що системи обміну інформацією, тобто системи зв'язку для веб-сайту, використовують загальний інтерфейс користувача та принцип обміну даними.

Використовуючи алгоритм чату Universal Ajax, ми бачимо, що основним

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

підключенням є авторизація користувача, а потім обмін повідомленнями.

Алгоритм роботи звичайної системи обміну повідомленнями для сайту не відрізняється від наведених вище прикладів додатків, що дозволяє інтегрувати їх під абсолютно будь-який сайт або додаток. Наприклад, чат Universal Ajax — це звичайна система обміну повідомленнями між користувачами, однак реалізація програмної частини суттєво відрізняється від AJAX через модульний принцип побудови програмного коду. Додаток, розроблений у цьому дипломному проекті, матиме модульну структуру програми. Програмна реалізація всіх зазначених додатків полягає у використанні новітніх технологій php, JavaScript, та mysql, що дозволяє адаптувати додаток абсолютно до будь-якого веб-ресурсу, який проектується, де використовується система Ajax для обміну даними.

Отже, для проектування нашої системи обміну інформацією як приклад буде використано програмне забезпечення чату phpBB Simple.

Структура робочого алгоритму phpBB Simple chat нагадує алгоритм роботи програми, розробленої в цій магістерській роботі, а функціональна частина, що відповідає за обмін даними між користувачами, ілюструє кінцевий результат розробки програми.

Серед інших особливостей можна виділити простоту інтерфейсу користувача та систему авторизації, яка, наприклад, у Smchat та чаті Universal Ajax розроблена досить складно на рівні програмного забезпечення та алгоритму роботи.

Основна інформація, пов'язана з темою магістерської роботи, зосереджена в пошукових системах, таких як Google і Wikipedia. За допомогою пошукового інтерфейсу було знайдено багато публікацій і матеріалів на тему магістерського дослідження. Теоретичні основи та аспекти теми проектування оприлюднені сервісом Wikipedia, що дало змогу побачити сучасний стан публікацій на тему дослідження. Обговорюються основні визначення та джерела інформації, які останнім часом постійно оновлюються.

Існує багато готових додатків і рішень, проектів впровадження, які

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

використовують технологію Ajax як основу систем обміну інформацією.

Використання готових рішень для вирішення робочого завдання, як приклад було взято алгоритм чату phpBB Simple. Цей варіант є чудовим прикладом, який демонструє роботу системи обміну інформацією про користувачів сайту. Основна частина полягає в тому, що необхідно організувати систему обміну інформацією, тобто створити веб-додаток типу «чат», як приклад у соціальних мережах. Зосередження на таких готових рішеннях дозволило оцінити чат phpBB Simple як відмінний аналог для розробки симпатичних веб-додатків.

Готові рішення для простих чатів phpBB можна побачити в Інтернеті, де після першої реєстрації можна приступити до тестування чату.

## **2.2 Обґрунтування вибору засобів для побудови системи та мови програмування**

### **Аналіз технологій для створення веб-додатків**

Створення веб-сайту та веб-додатку – це створення маркетингового інструменту, призначеного для стимулювання попиту на певні послуги чи товари, або інформаційного ресурсу, який має на меті донести необхідну інформацію до цільової аудиторії, або створення сервісного ресурсу, який би вирішував завдання надання певних послуг зацікавленим відвідувачам.

При цьому варто пам'ятати, що розробка сайту, здатного якісно виконувати всі ці функції, є досить складним процесом і вимагає високого професійного рівня виконання.

Динаміка розвитку в області веб-розробки створює потребу в наукових дослідженнях якості веб-продуктів, тобто веб-сайтів і веб-порталів, з метою формування об'єктивних рекомендацій.

Одним із ключових моментів у розвитку Всесвітньої павутини є веб-розробка – процес створення веб-сайту чи веб-додатку [9]. Термін включає

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

розробку додатків для електронної комерції, веб-дизайн, веб-програмування на стороні клієнта та на сервері та конфігурацію веб-сервера.

Основними етапами веб-розробки є:

- дизайн сайту (збір та аналіз вимог, створення технічного завдання, дизайн інтерфейсу);
- розробка креативної та дизайнерської концепції сайту;
- створення макетів сторінок, мультимедійних та FLASH елементів;
- верстка та дизайн сторінки;
- програмування (розробка функціональних засобів) або інтеграція в систему управління контентом (CMS);
- оптимізація та розміщення матеріалів сайту;
- випробування та налаштування;
- відкриття хостингового проекту;
- підтримка функціонального веб-сайту або його програмної бази;
- подальше просування сайту в мережі та підвищення його рейтингу.

Розглянемо та порівняємо існуючі технології створення сайтів.

### **Ручний метод**

HTML (HyperText Markup Language) — стандартна мова для розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створено за допомогою мови HTML (або XHTML). Документ HTML обробляється в браузері і відображається на екрані у звичному для людини вигляді.

У більшості випадків автор документа суворо визначає зовнішній вигляд документа. У випадку HTML читач, спираючись на можливості веб-браузера, може певною мірою контролювати зовнішній вигляд документа (але не його зміст). HTML дозволяє вказати, де має бути заголовок або абзац у документі за допомогою тегів HTML, а потім дозволяє веб-браузеру інтерпретувати ці теги. Наприклад, один веб-браузер може розпізнати початок знака абзацу та представити документ у потрібному форматі, а інший – ні та відображає документ в одному рядку. Користувачі деяких веб-браузерів також мають можливість

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

регулювати розмір і тип шрифту, колір та інші параметри, які впливають на відображення документа [17].

Теги HTML можна умовно розділити на дві категорії:

- теги, які визначають, як веб-браузер відобразатиме текст документа в цілому;
- теги, які описують загальні властивості документа, наприклад назву або автора документа.

Документи HTML можна створювати за допомогою будь-якого текстового редактора або спеціалізованих редакторів і конвертерів HTML. Вибір редактора, який буде використовуватися для створення документів HTML, залежить виключно від концепції зручності та особистих уподобань кожного автора.

Наприклад, такі візуальні редактори HTML, як «Atom», «TiniMCE» або «NicEdit» дозволяють створювати документи в графічному вигляді за допомогою технології VISIVIG (What You See Is What You Get) [3].

З іншого боку, більшість традиційних інструментів створення документів мають конвертери, які дозволяють конвертувати документи у формат HTML.

Основна перевага HTML полягає в тому, що ваш документ можна переглядати в різних типах веб-браузерів і на різних платформах.

Cascading Style Sheets [17] (скорочено CSS) — це спеціальна мова, яка використовується для опису сторінок, написаних на мовах розмітки даних [9].

CSS найчастіше використовується для візуального представлення сторінок, написаних у HTML і XML, але формат CSS можна застосовувати до інших типів XML-документів.

Таблицю стилів CSS можна вставити безпосередньо в сторінку HTML - це внутрішня таблиця стилів. Або ви можете створити його в окремому файлі, а вже потім прикріпити посилання на потрібну HTML-сторінку - це зовнішня таблиця стилів. Зовнішня таблиця повинна бути пов'язана з основним HTML-документом за допомогою спеціальних тегів:

`<link rel="stylesheet" type="text/css" href="/style.css">`,

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

де style.css – це ім'я файлу, що містить таблицю CSS. Завдяки цьому, стиль, описаний у зовнішній таблиці CSS, можна використовувати повторно скільки завгодно разів [17].

**Конкретні переваги CSS:**

- керування відображенням кількох документів за допомогою однієї таблиці стилів;
- більш точний контроль зовнішнього вигляду сторінки;
- різноманітні заставки для різних носіїв (полотно, друк тощо);
- складна та складна техніка проектування [17].

Роблячи висновки з попередньої інформації, можна зробити висновок, що використовувати HTML і CSS зручно і не складно у використанні.

**Переваги посилань HTML + CSS наступні:**

- не вимагає додаткового програмного забезпечення;
- надійність сайту, малоімовірна можливість злому, оскільки немає прямого доступу до бази даних чи конфігураційного файлу;
- швидке завантаження та економія трафіку на хостингу порівняно з SMS;
- кращий спосіб опису вмісту та підвищення функціональності документів HTML.

**Недоліки:**

- обмежені можливості при створенні сайту;
- відсутність динамічних елементів;
- велика кількість часу, витраченого на написання необробленого коду;
- переважно для простих сайтів.

**Використання CMS**

Останнім часом великою популярністю користуються CMS системи - системи управління контентом або системи управління контентом (англ. Content management system, CMS) - інформаційна система або комп'ютерна програма для забезпечення та організації спільного процесу створення, редагування та управління контентом (від англ. content) .

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

### Основні функції CMS:

- створення контенту. Це набір завдань, які виконують копірайтери, фотографи, художники-оформлювачі, відеопродюсери та звукорежисери, маркетологи, юристи та інші люди, які надають оригінальний контент користувачам сайту;

- збір і адаптація вмісту з існуючих джерел;

- класифікація та індексація контенту. Вміст має бути описаний формальними характеристиками (наприклад, дата створення, автор) і класифікаційними даними (наприклад, тематична категорія або ключові слова). Ця діяльність описується як пов'язування вмісту та метаданих;

- огляд змісту. Необхідно для всіх типів опублікованого вмісту;

- схвалення. Офіційне схвалення опублікованого контенту є важливою складовою юридичної відповідальності за нього;

- перетворення вмісту. Тексти, графіку, звуки та інші форми вмісту слід конвертувати у формат, який найбільше підходить або використовується в цій системі CMS. Зберігання вмісту. Зазвичай вміст зберігається у файлах або базі даних. Для більш складних програм вміст підлягає контролю версій програмного забезпечення (SMC).

- тестування та перевірка контенту;

- тестування готовності контенту – вид тестування, який передбачає перевірку (перевірку) повноти та цілісності великої кількості контенту (наприклад, інформації про різні аспекти нової послуги);

- публікація. Він враховує всі фізичні аспекти публікації контенту, включаючи дублювання контенту на різних серверах;

- підтримка, оновлення та контроль змін. Це включає моніторинг опублікованого вмісту та реагування на сигнали та потребу в змінах;

- відкликання та архівування. Відкликання може статися з кількох причин - наприклад, втрата актуальності вмісту, втрата законних прав на вміст, низька

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

частота відвідувань, поява новішого вмісту тощо. Будь-який відкликаний вміст архівується.

**Звіти та аналізи.** Він включає різні форми звітності та аналізу, щоб краще обслуговувати користувачів, покращувати зовнішній вигляд порталу.

Система керування вмістом може містити широкий спектр даних: документи, фільми, фотографії, номери телефонів, наукові дані тощо. Така система часто використовується для зберігання, керування, перегляду та публікації документації. Контроль версій є однією з його головних переваг, коли вміст змінюється групою людей.

Система управління - це програма, яка надає інструменти для додавання, редагування та видалення інформації на сайті.

Більшість сучасних CMS мають модульну архітектуру, що дозволяє адміністратору вибирати та налаштовувати потрібні йому компоненти.

Типові модулі:

- динамічне меню;
- блог;
- новини;
- опитування;
- пошук по сайту;
- статистика відвідувань;
- гостьова книга тощо.

Сайти, організовані за допомогою системи управління контентом, засновані на наступних технологіях: веб-сервер, сховище даних (найчастіше СУБД, наприклад MySQL або PostgreSQL, але є і не-SQL CMS), веб-додаток для забезпечення роботи самої системи, візуальний (VISIVIG) редактор сторінок, файловий менеджер з веб-інтерфейсом для керування файлами сайту, система керування правами користувачів і редакторів сайту.

Існують різні системи управління сайтом, в тому числі платні і безкоштовні, побудовані на різних технологіях. Кожен сайт має панель

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

управління, яка є лише частиною всієї програми, достатньою для управління сайтом.

Найпоширенішими є такі технологічні платформи, які використовуються як основа веб-додатку, що реалізує роботу CMS: PHP, Perl, .NET.

Існує термін контент-менеджер, який вказує на вид професійної діяльності - редактор сайту або співробітник, який працює з CMS.

Більшість сучасних систем керування контентом реалізовано у вигляді візуального (VISIVIG) редактора – програми, яка створює HTML-код зі спеціальної спрощеної розмітки, що дозволяє користувачеві легше форматувати текст.

Нижче наведено приклади CMS в алфавітному порядку. Наявність у цьому списку не свідчить про популярність системи:

- Drupal;
- Етоміт CMS;
- eZ public;
- Бітрикс;
- Joomla;
- Magento;
- WordPress.

CMS рішень багато, але при виборі варто спиратися на наступні параметри:

- послуга, призначена для користувача - наявність окремих функцій і модулів, зрозумілість і доступність для користувача;
- продуктивність – використання технологій, що дозволяють збільшити пропускну здатність і надійність системи;
- сумісність (апаратна та програмна) – можливість функціонування системи на різних платформах, сумісність із СУБД, можливість підключення додаткових модулів;
- масштабованість – можливість розвитку та розширення системи.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

## Аналіз засобів реалізації баз даних

Робота з базами даних є однією з основних складових процесу динамічного програмування сайту. Бази даних для веб-сайтів використовуються для зберігання різноманітної інформації. Спрощені бази даних являють собою певний набір взаємозалежних таблиць. Розміри таблиць у базах даних різні, а їх кількість довільна. Бази даних на сервері накопичують статистичну інформацію, необхідну для роботи сайту [3].

До найпопулярніших мережевих баз даних належать SQL, MySQL, Oracle Database та інші. Вибір правильної системи управління базами даних (СУБД) визначається вимогами до інформаційних характеристик і функціональних можливостей системи.

Однією з найпоширеніших систем керування базами даних сьогодні є MySQL [3], яка є альтернативою комерційним системам. Він в основному використовується для створення динамічних веб-сторінок, оскільки має відмінну підтримку різних мов програмування. MySQL — це компактний багатопоточний сервер баз даних. Серед переваг сервера MySQL виділимо простоту встановлення та використання, підтримку паралельної роботи значної кількості користувачів. Кількість рядків у таблицях даних може досягати 50 млн. Висока швидкість виконання команд, наявність простої та ефективної системи безпеки зумовлюють зростання популярності MySQL. Однак головною перевагою СУБД MySQL є можливість її безкоштовного використання.

Система управління об'єктно-реляційними базами даних Oracle (Oracle Database) розроблена для операційних систем Windows, Unix, Linux і MacOS. Oracle Database, на відміну від MySQL, має більш широку сферу застосування [3]. СУБД Oracle широко відома як у нас, так і в світі. На його основі побудовано багато сучасних інформаційних систем. Однак в процесі програмування сайтів для малого та середнього бізнесу, через високу вартість, використовується рідко. Крім того, знайти хостинг з підтримкою такої СУБД досить складно.

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

SQL (Structured Query Language) — декларативна мова програмування для взаємодії інтерфейсу користувача з базами даних, яка реалізує процеси створення запитів, оновлення та управління реляційними базами даних, створення схеми бази даних та її модифікації, а також система контролю доступу до інформаційних ресурсів. SQL може формувати інтерактивні запити або, будучи вбудованим у прикладні програми, діяти як інструкції з керування даними. Крім того, стандарт SQL підтримує функції визначення процесу зміни, перевірки та захисту даних.

У таблиці 2.1 зібрані результати критеріального аналізу сучасних СУБД.

Таблиця 2.1 - Порівняльна характеристика СКБД

		SQL	Oracle	MySQL
1	Надійність	+	+	+
2	Швидкодія	-	+	+
3	Простота	-	-	+
4	Зручність використання	+/-	+	+

Аналізуючи основні можливості предметної СУБД, можна зробити висновок про зручність використання інструменту MySQL для створення динамічних веб-сторінок, оскільки сервер MySQL підтримується різними мовами програмування, в тому числі PHP. MySQL відповідає основним вимогам до СУБД у розробці веб-сайтів, оскільки вона надійна, швидка, проста в установці та проста у використанні. Завдяки своїй доступності, швидкості та безпеки MySQL забезпечує ефективний доступ до баз даних з Інтернету.

### Вибір бази даних

Для обробки всіх даних користувача буде використовуватися реляційна база даних MySQL. Дані зберігатимуться в таблицях, тобто матрицях із рядками та стовпцями. Дані будуть зберігатися у двох таблицях. Таблиця user буде

відповідати за зберігання даних про всіх користувачів в системі, тобто про їх статуси в системі, аутентифікатори.

Таблиця повідомлень відповідатиме за всі повідомлення в системі обміну повідомленнями. Між цими двома таблицями існуватиме реляційний зв'язок, який означає, що дані з однієї таблиці посилаються на дані з іншої таблиці. Зв'язок використовуватиметься за ідентифікаторами користувачів. Кожен користувач у таблиці матиме унікальний ідентифікатор, який використовується для ідентифікації користувача під час процесу входу. Після реєстрації користувача в системі всі його дані ідентифікуються спеціальним ID користувача. При пересиланні повідомлення між користувачами створюється система обміну інформацією, тобто всі повідомлення зберігаються під спеціальними ідентифікаторами в таблиці повідомлень.

Структура таблиці користувача — це окремі поля, які мають різні типи даних для кожного окремого запису інформації. Таблиця повідомлень стосується безпосередньо зберігання повідомлень користувача, тобто зберігання тексту повідомлення, усіх ідентифікаторів користувачів і шляхів повідомлень між користувачами.

Працювати з базою даних MySQL дуже зручно і просто, використання реляційної таблиці дозволяє легко шукати дані та ідентифікувати їх у дуже зручний спосіб.

### **Використання мови розмітки html**

HTML (HyperText Markup Language) — стандартна мова для розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створено за допомогою мови HTML (або XHTML). Документ HTML обробляється в браузері і відображається на екрані у звичному для людини вигляді.

Загальний вигляд веб-додатку розроблено з використанням мови гіпертекстової розмітки HTML. У нашому випадку створення «фреймворку» веб-додатку полягає у створенні так званого макета сайту. Верстка сайту передбачає створення покрокової структури веб-сторінки, яка буде відображатися в браузері

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

користувача. Існує так звана «блочна верстка» сайту, тобто створюються блоки, які наповнюються контентом, як правило, графічною або текстовою інформацією. HTML існує як основний інструмент для створення моделей веб-сторінок, а також як універсальний метод проектування веб-додатків. Створено блочну модель, яка дозволяє розвивати динаміку сайту, тобто інтерактивність та мультимедійність. Крім того, на цьому етапі створюється конструкція всіх гіперпосилань для веб-сторінки. Загальна структура сайту. Перегляд програми складається з документів html, які користувач завантажує та переглядає у вікні веб-браузера. Структура цих сторінок повністю залежить від дизайну та макета. Нижче наведено структуру веб-сторінки нашого веб-додатку.

Конструкція веб-сайту складатиметься з наступних частин:

- оголошення типу документа (англ. Document type declaration, Doctype), на початку документа, в якому визначається тип документа (DTD).
- обкладинка документа (розташована всередині елемента заголовка), яка містить загальну технічну інформацію або додаткову інформацію про документ, яка не відображається безпосередньо в браузері;
- тіло документа (може бути в елементах основної частини або фрейму), що містить основну інформацію про документ.

Таким чином, загальна структура веб-сторінки складатиметься з оголошення, заголовка документа та тіла документа.

### **Використання мови програмування PHP**

PHP скрипт. Це сценарна мова програмування, призначена для створення HTML-сторінок на веб-сервері та роботи з базами даних[15]. На даний момент підтримується практично всіма представниками хостингу, входить в «стандартний» набір для створення сайтів (LAMP - Linux, Apache, MySQL, PHP)

Є можливість розвитку та підключення додаткового розширення. Можливості PHP дуже великі. PHP в основному використовується для написання сценаріїв, які виконуються на стороні сервера; таким чином, PHP може робити все, що може робити будь-яка інша програма CGI (наприклад, обробка цих форм,

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

створення динамічних сторінок, надсилання та отримання файлів cookie). Але PHP також дозволяє виконувати багато інших завдань.

Є три основні сфери, де використовується PHP [15]: створення сценаріїв для виконання на стороні сервера; створення сценаріїв для виконання в командному рядку; створення клієнтських додатків з графічним інтерфейсом користувача.

#### Переваги PHP:

- доступний для більшості операційних систем, включаючи Linux, багатьох модифікацій Unix (таких як HP - UKS, Solaris і OpenBSD), Microsoft Windows, Mac OS X, RISC OS і багатьох інших;
- включає підтримку більшості веб-серверів (для більшості серверів PHP постачається як модуль, для інших, які підтримують стандарт CGI, PHP може функціонувати як процесор CGI);
- підтримує великий обсяг баз даних;
- підтримує роботу DBKS на абстрактному рівні (тому ви можете працювати з будь-якою базою даних за допомогою DBKS); ODBC (тобто ви можете працювати з будь-якою базою даних, яка підтримує цей стандарт);
- підтримує «спілкування» з іншими службами за різними протоколами: LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows), а також багато інших;
- підтримує стандарт VDDKS для обміну складними структурами даних;
- підтримує об'єкти Java, дозволяє використовувати їх як об'єкти PHP;
- дозволяє створювати зображення, PDF-файли, Flash-відео, створені «на льоту»; може виводити будь-які текстові дані (XHTML, інші файли XML); автоматично генерується та зберігається у файловій системі сервера;
- включає засоби обробки текстової інформації, починаючи з розширених регулярних виразів Perl або POSIX і закінчуючи документами XML [15].

Основні недоліки PHP: низька оптимізація рекурсії; більшість модулів PHP не підтримують безпеку потоків.

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Програмування веб-додатків буде безпосередньо реалізовано мовою php. Як було сказано раніше, мова програмування php є основним інструментом для створення динамічних веб-сторінок.

Основні функціональні частини веб-додатку:

- реєстрація користувача;
- авторизація користувача;
- підтримка сесії користувача;
- скасування підписки користувача;
- відправка повідомлень;
- видалення повідомлень;
- редагування повідомлень;
- індикатор повідомлення;
- індикатор стану користувача.

Усі зазначені функціональні частини веб-додатку будуть запрограмовані як модульна структура, тобто як окремі функціональні частини, які будуть пов'язані зі структурою веб-додатку та функціонуватимуть як єдине ціле. Кожна функціональна частина є окремою частиною програмного коду, ця частина коду виражається як окрема функція на мові програмування php. Структура коду відповідатиме наведеній вище функціональній структурі. Як відомо, файли, які використовують код php, мають розширення php. Усі основні частини коду, які забезпечують роботу програми, будуть раціонально згруповані у файли та згодом створять до них загальний функціональний доступ. У цьому випадку всі функції, що відповідають за внутрішню роботу програми, включаючи виведення контексту на веб-сторінку, будуть об'єднані в одному файлі.

### **Використання JavaScript**

JavaScript[13] — це сценарна мова програмування, орієнтована на прототип. Є діалектом мови ECMAScript. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів програми.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27



або офлайн. Тобто користувач, який зареєстрований або авторизований в системі, має статус онлайн, а користувач, який не зареєстрований або не авторизований в системі, має статус офлайн.

Індикатор повідомлення - це числове значення кількості нових вхідних повідомлень для користувача, який знаходиться в системі. Видалення, редагування та надсилання повідомлень відбувається в синхронному режимі роботи користувача та веб-додатку, тобто користувач може видалити повідомлення, а нове повідомлення прийде без перевантаження веб-сторінки.

Усі ці запити безпосередньо адресуватимуться базі даних відповідно до даних, що зберігаються в таблицях, а пізніше вони оновлюватимуться з бази даних уже на вмісті веб-сторінки.

### **2.3 Розгорнута постановка завдання**

Згідно з технічним завданням на магістерську роботу, реалізації підлягає програмне забезпечення веб-чату для обміну даними в мережі Інтернет. В процесі розробки магістерської роботи необхідно виконати наступний обсяг проекту:

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

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) сформувані висновки про виконаний обсяг робіт та одержані результати.

КБПЗ\_2023

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Система обміну інформацією в роботі представлена як веб-додаток для обміну інформацією між користувачами сайту. Система обміну інформацією повинна бути побудована на основі технології Ајах, що створює принципово нову проблему проектування цього розділу. Між користувачами веб-сайту має відбуватися обмін інформацією, тобто мають бути певні діалоги, за допомогою яких відбуватиметься взаємодія між користувачами веб-ресурсів. В результаті в системі реалізовано інтерфейс для відправки повідомлень, тобто всі повідомлення відправляються в діалозі користувача, тому що це дуже зручно, особливо коли повідомлень багато. На цьому етапі проектування система обміну інформацією буде представлена як діалогова система для обміну повідомленнями між користувачами, саме діалоговою, тому що реалізація обміну повідомленнями має відбуватися у формі діалогу. Транзакція повідомлень в системі повинна відбуватися негайно без затримки, тобто відразу після того, як користувач відправить запит на відправку. Усі повідомлення, якими керує користувач, повинні зберігатися в системі обміну повідомленнями. Система обміну повідомленнями — це веб-база даних. Отже, це буде основний вибір для збереження всіх повідомлень у системі для кожного користувача окремо. Повідомлення не слід видаляти відразу після того, як користувач їх прочитає. Повідомлення застаріють і зберігаються в системі обміну повідомленнями. Постановка задачі на цьому етапі проектування досить розпливчата, тому кінцевий результат задачі може істотно відрізнятись від початкового. Реляційна база даних найкраще підходить для зберігання інформації про всі повідомлення в системі обміну. Основною проблемою цієї частини дизайну є використання технології Ајах. Використання технології Ајах дає можливість відправляти запити

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

користувачів у так званому «тихому режимі». Під час надсилання повідомлення з діалогового вікна має бути можливість надсилати дані на сервер без перевантаження сторінки, що дозволяє Ажах. Зрештою, без використання Ажах підключення до сервера відбувається лише тоді, коли сайт оновлюється, це дуже часто буває при класичній моделі створення веб-додатків без використання підходу Ажах. У цьому випадку втрачаються всі заповнені поля і форми, і це дуже незручний метод оформлення для нашого прикладу. Ця проблема є ключовим принципом аналізу на цьому етапі.

Організація системи керування користувачем є ключовим аспектом дизайну. Робота програми неможлива без авторизації або реєстрації користувача. Створення частини профілю для користувача є обов'язковою складовою функціонування системи.

Профільна частина користувача дозволяє авторизований доступ до системи, тобто лише після входу користувача в систему йому надається можливість працювати зі своїми даними. Тобто необхідно створити можливість входу в систему та реєстрації для досягнення авторизованого доступу. Інформація про користувачів у системі буде зберігатися в реляційній базі даних. Загальний аналіз цієї проблеми показує, що необхідно розробити алгоритм авторизації та реєстрації в системі.

Отже, як ми бачимо, серед головних проблем дизайну – використання технології Ажах і частина профілю користувача. Ці аспекти будуть розроблені в майбутньому в рамках проекту. Ми можемо створити чітку модель дерева проблем, яка дозволить побачити частини системи проектування в цілому окремо.

Основною метою розвитку нашого проекту буде створення системи обміну інформацією на основі технології Ажах. Основні функції роботи нашої системи будуть:

- вибір середовища програмування;
- авторизація та реєстрація користувачів у системі;
- профільна частина користувача;

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

- обмін повідомленнями;
- кросбраузерність;
- побудова сесії підтримки користувача;
- збереження сеансу користувача;
- статуси користувачів;
- статуси та прихід повідомлень;
- розлогінення користувачів;
- організація діалогів на основі Ajax;
- захист профільних даних користувача;
- автоматичне оновлення індикаторів повідомлень. (впровадження Ajax);
- автоматичне оновлення індикаторів Користувачів. (впровадження Ajax);
- створення кімнати спілкування для користувача;
- аикористання фреймворка jquery.(Кросплатформений);
- використання js.

Робота додатку з використанням Ajax представлено на рисунку 3.1

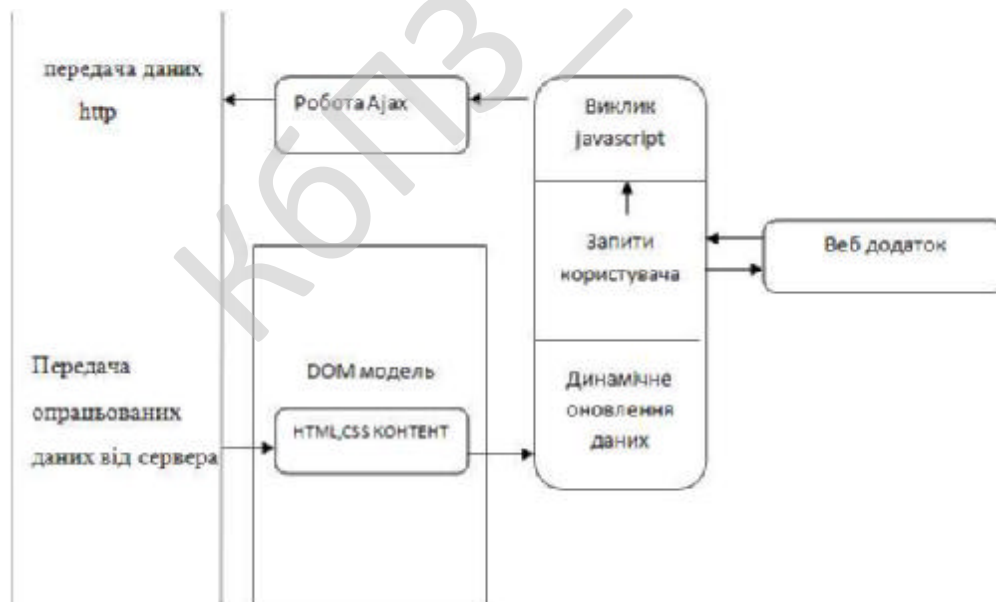


Рисунок 3.1 - Робота додатку з використанням технології Ajax

На рисунку 3.2 показаний концептуальний алгоритм веб-додатку. Алгоритм роботи просто демонструє роботу веб-додатку, розробленого в цій

дипломній роботі. Як бачимо зі схеми, основним прецедентом є користувач, а в алгоритмі працює кілька гілок, які в залежності від вибору користувача виконують певні дії. Користувач за допомогою браузера свого операційного блоку завантажує веб-ресурс за вказаною адресою та виконує потрібні йому дії. Користувач має можливість зареєструватися або авторизуватися в системі веб-додатків, про що свідчить схема алгоритму двох гілок, які слідують за прецедентом. Під час авторизації користувач входить у свою профільну частину веб-додатку, а в другому випадку, коли користувач вирішує зареєструватися в системі, йому надається можливість створити власний профіль у системі. Для цього він виконує всі вказівки системи, а потім для нього вже створюється спеціальний профіль в системі. Після виконання всіх вищезазначених кроків алгоритму користувач потрапляє на сторінку власного профілю користувача, тобто на персональну веб-сторінку у веб-додатку. Потім користувачеві надається можливість вибору наступних дій, тобто надсилання повідомлень, видалення повідомлень, редагування і, нарешті, в кінці роботи, є можливість вийти з власного профілю користувача, тобто вийти з системи.

Такий алгоритм роботи є класичним при створенні веб-додатків такого типу.

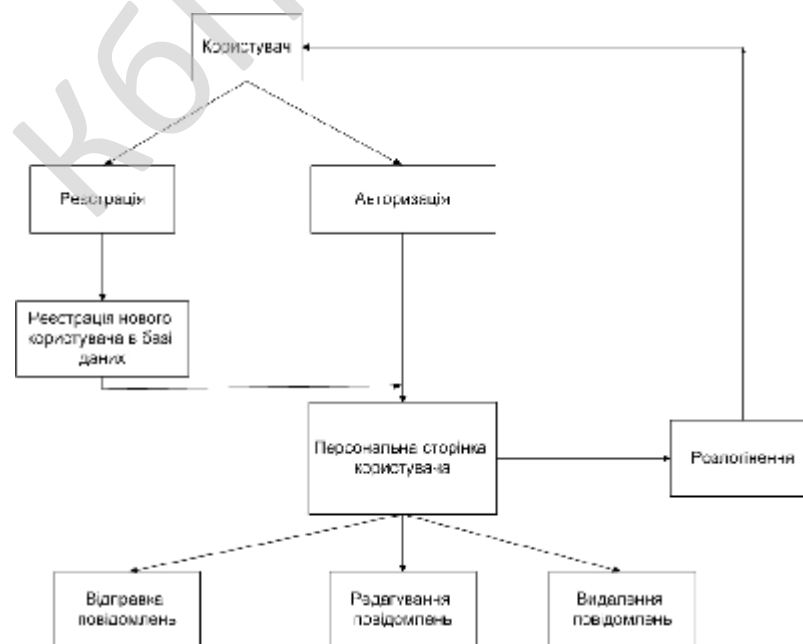


Рисунок 3.2 – Концептуальний алгоритм роботи веб-додатку

### 3.2 Розробка структурної схеми

Структурна схема — це сукупність об'єктів, частин і зв'язків між ними. Він призначений для відображення загальної структури, тобто її основних блоків, вузлів, частин і основних зв'язків між ними.

У нашому випадку потрібно проаналізувати систему обміну інформацією для онлайн-чату. Необхідно чітко побудувати алгоритм системи, проаналізувати структуру роботи та виділити основні аспекти вирішення поставленого завдання.

Розглянемо взаємне розміщення і певний зв'язок складових частин цілого проекту. Як згадувалося раніше, структура веб-додатку складається з наступних частин:

- реєстрація користувача;
- авторизація користувача;
- підтримка сесій користувача;
- вихід користувача;
- відправка повідомлень;
- видалення повідомлень;
- редагування повідомлень;
- індикатор повідомлення;
- індикатор статусу користувача.

Щоб вирішити задачу програмування всіх структурних частин веб-додатку, необхідно встановити зв'язок між ними. Перш за все, взаємозв'язок відображається між такими компонентами:

#### **Реєстрація та авторизація користувачів**

З точки зору програмування, було б доцільно створити реєстрацію та авторизацію користувачів як пріоритет. Тобто ми створюємо систему автентифікованого доступу користувачів і на її основі створюємо повнофункціональний веб-сайт. Реєстрація та авторизація користувачів завжди розглядаються як єдиний функціональний модуль веб-сайту.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

## **Підтримка сеансів користувачів і виходу з системи**

Після реєстрації користувача в системі або авторизації йому надається певне підключення до сайту. Це з'єднання будується за допомогою сеансу користувача. Сеанс користувача та вихід із системи існують як два компоненти, оскільки їхні функції працюють разом. Користувач може будь-коли припинити з'єднання з веб-сайтом, тобто припинити виконання сеансу користувача[22].

## **Надсилання, видалення та редагування повідомлень**

Ці три частини структури веб-додатку об'єднані в один загальний модуль, який використовується в частині профілю користувача. Програмування цих трьох функцій буде зручним, якщо ви об'єднаєте їх на цій стадії проектування як один модуль обробки повідомлень користувача.

## **Індикатор повідомлень та індикатор стану користувача**

Ці дві функції системи існують як метод сповіщення інформації про повідомлення та користувачів на льоту. Програмування цих двох частин тепер доречно розглядати як окрему функціональну залежність веб-сайту.

Отже, після розгляду структури веб-додатку на функціональному рівні спрощується взаємозв'язок усіх частин системи.

Структурну схему розроблюваної системи чату зображено на рисунку 3.3.

На схемі можна побачити, що сайт має головну сторінку, з якої користувач може перейти на сторінку реєстрації, відновлення пароля, сторінку про нас, сторінку контактів, профіль користувача, а також сторінку панелі адміністратора та меню онлайн-підтримки. Крім того, на головній сторінці користувач може увійти або вийти зі свого облікового запису.

Робоча схема просто демонструє роботу веб-додатку, розробленого в цій дипломній роботі. Як бачимо зі схеми, основним прецедентом є користувач, а в алгоритмі працює кілька гілок, які в залежності від вибору користувача виконують певні дії.

Користувач за допомогою браузера свого операційного блоку завантажує веб-ресурс за вказаною адресою та виконує потрібні йому дії. Користувач має

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

можливість зареєструватися або авторизуватися в системі веб-додатків, про що свідчить схема алгоритму двох гілок, які слідуєть за прецедентом.

Під час авторизації користувач входить у свою профільну частину веб-додатку, а в другому випадку, коли користувач вирішує зареєструватися в системі, йому надається можливість створити власний профіль у системі. Для цього він виконує всі вказівки системи, а потім для нього вже створюється спеціальний профіль в системі.

Після виконання всіх вищезазначених кроків алгоритму користувач потрапляє на сторінку власного профілю користувача, тобто на персональну веб-сторінку у веб-додатку. Потім користувачеві надається можливість вибору наступних дій, тобто надсилання повідомлень, видалення повідомлень, редагування і, нарешті, в кінці роботи, є можливість вийти з власного профілю користувача, тобто вийти з системи.

КБПЗ - 2023

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

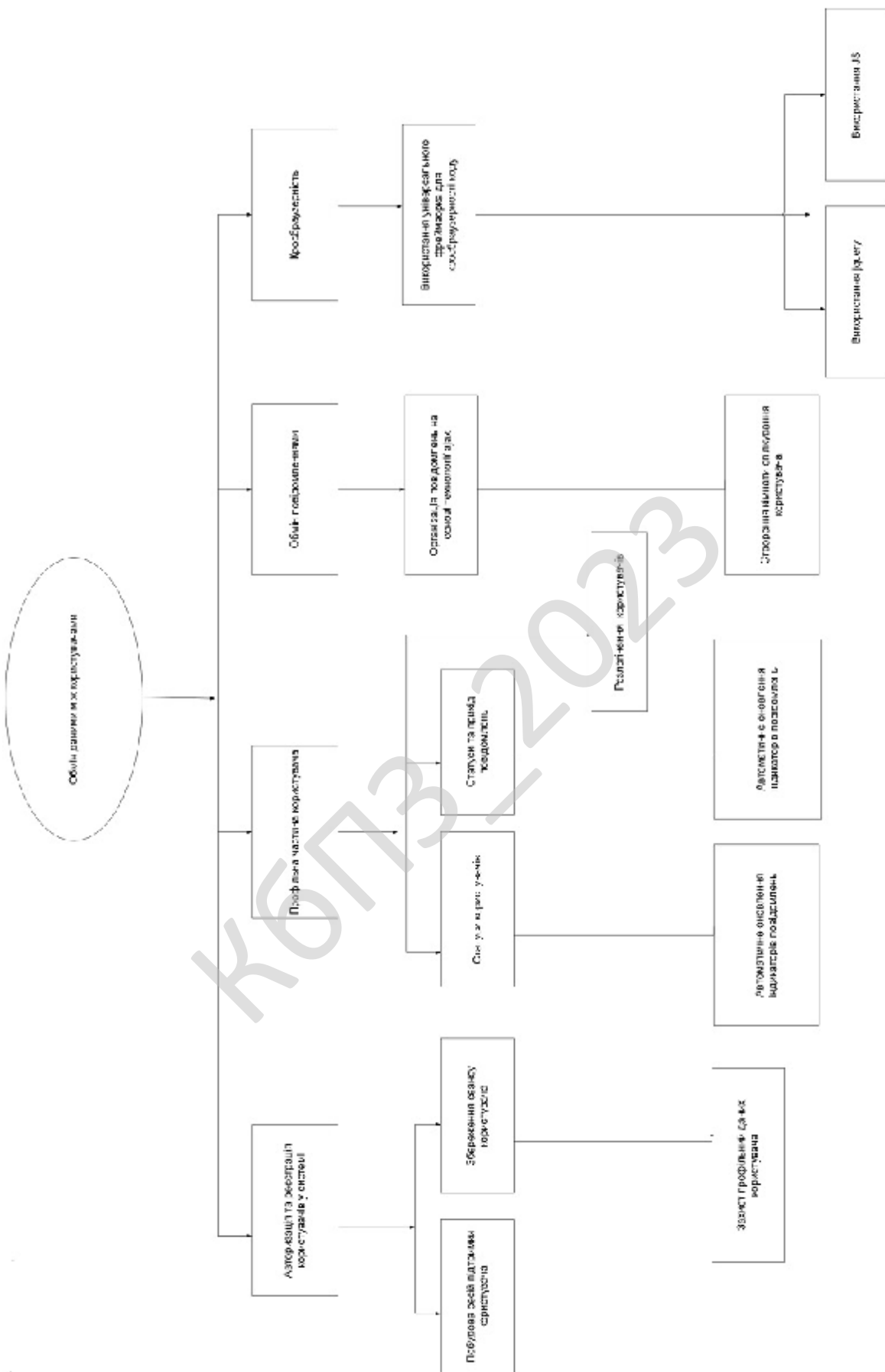


Рисунок 3.3 - Структурна схема системи онлайн-чату

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-123.23.0001.00.00.ПЗ

Арк.

38

### 3.3 Розробка функціональної схеми

Функціональну схему сайту зображено на рисунку 3.4.

Функціональна схема складається з таких блоків:

- блок головної сторінки;
- блок реєстрації користувачів;
- блок авторизації користувачів;
- блок бази даних сайту;
- блок сторінки профілю користувача;
- блок адміністративної панелі;
- блок сторінки контактів та інформації про розробника сайту.

Функціональна складова об'єкта проектування визначає його основні можливості та функції використання, які він матиме. Інформаційна система обміну даними реалізує функцію авторизації та реєстрації користувачів у системі, надсилання, редагування та видалення повідомлень у системі в синхронному режимі за технологією AJAX.

#### Реєстрація користувача

Багато веб-сайтів змушують користувача надати певну інформацію про себе. Після того, як зроблено запит і отримано інформацію про відвідувача, її потрібно певним чином зберегти та зв'язати з відвідувачем, коли він наступного разу спробує скористатися веб-сайтом. Оскільки способів реєстрації на сайті багато, я реалізував класичний спосіб, коли користувач реєструється під своїм іменем та персональним паролем. Пароль позначається латинськими літерами. Логін, тобто ім'я користувача, яке буде використовуватися в системі, має містити латинські літери відповідно до стандартів веб-дизайну для відносин. Реєстрація є основним кроком, який повинні виконати всі користувачі, яким надано доступ до певних даних у системі веб-додатків. Потужні та сучасні веб-проекти не можуть існувати без системи керування користувачами.

Щоб зареєструвати користувача в системі, необхідно заповнити поля

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

введення, тобто поле, де потрібно вказати ім'я користувача та поле, де потрібно вказати пароль.

Реєстрація необхідна для запам'ятовування користувача, який пізніше буде входити в систему за допомогою реєстраційних даних, що зберігаються в базі даних. Реєстрація в системі проводиться кожного разу для нових користувачів. Після заповнення користувачем всіх необхідних даних проводиться його реєстрація, у зв'язку з чим в базі даних створюються нові поля з інформацією про нового користувача.

Після реєстрації нового користувача в системі йому можна присвоїти відповідний статус та інформацію у власному профілі.

### **Авторизація користувача**

Авторизація користувача в системі здійснюється на підставі автентичності користувача, включаючи його логін і пароль. Перш ніж користувач зможе використовувати інформацію на сайті, йому необхідно пройти авторизацію. На будь-якому сучасному веб-сайті авторизація є основним компонентом дизайну ресурсу. Потужні веб-ресурси просто не можуть існувати без авторизації їх користувачів, оскільки це призведе до несанкціонованого доступу до інформації. Авторизація за структурою дуже схожа на реєстрацію користувача в системі. Авторизація передбачає перевірку прав і наявність певного правила доступу користувача до того чи іншого інформаційного ресурсу, за рахунок перевірки введених користувачем даних на сайті. Тобто перевіряється відповідність логіна та пароля, введених користувачем у базі зареєстрованих користувачів.

### **Надсилання повідомлень**

Для обміну повідомленнями між користувачами користувач повинен бути авторизований або зареєстрований в системі. Після цього користувач може надсилати повідомлення іншим користувачам, які також зареєстровані в системі. Відправка повідомлень реалізована асинхронним способом, без перевантаження сторінки, на якій знаходиться користувач, на відміну від класичного способу. Повідомлення можна надсилати багатьом користувачам, але послідовно. Кожен

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

користувач може надсилати повідомлення багатьом користувачам і отримувати повідомлення від кількох користувачів. Користувач буде проінформований про новий статус повідомлення асинхронно, тобто немає необхідності перевантажувати сайт, на якому він перебуває. У процесі роботи, щоб користувач міг бачити надходження нових повідомлень до нього, буде використовуватися асинхронний запит до бази даних, що містить нові повідомлення для користувача. Це пов'язано з принципом «AJAX», який в таких випадках забезпечує безперервний процес роботи користувача і одночасно обробку запиту користувача.

### **Редагування повідомлень**

Редагування повідомлення — це функція зміни вмісту повідомлення з відповідним збереженням у базі даних, тобто заміна старого повідомлення на нове та надсилання користувачеві нового модифікованого повідомлення замість попереднього повідомлення, яке ми відповідно змінили.

Редагувати можна лише надіслані повідомлення. Звичайно, щоб мати можливість редагувати повідомлення, вам необхідно пройти етап авторизації або реєстрації, тобто ввести свій профіль в системі. Після цього ми зможемо редагувати та надсилати всі надіслані повідомлення користувачеві.

Нове повідомлення автоматично замінюється старим, яке ми редагували, і надсилається асинхронно, без перевантаження сторінки.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>41</b>

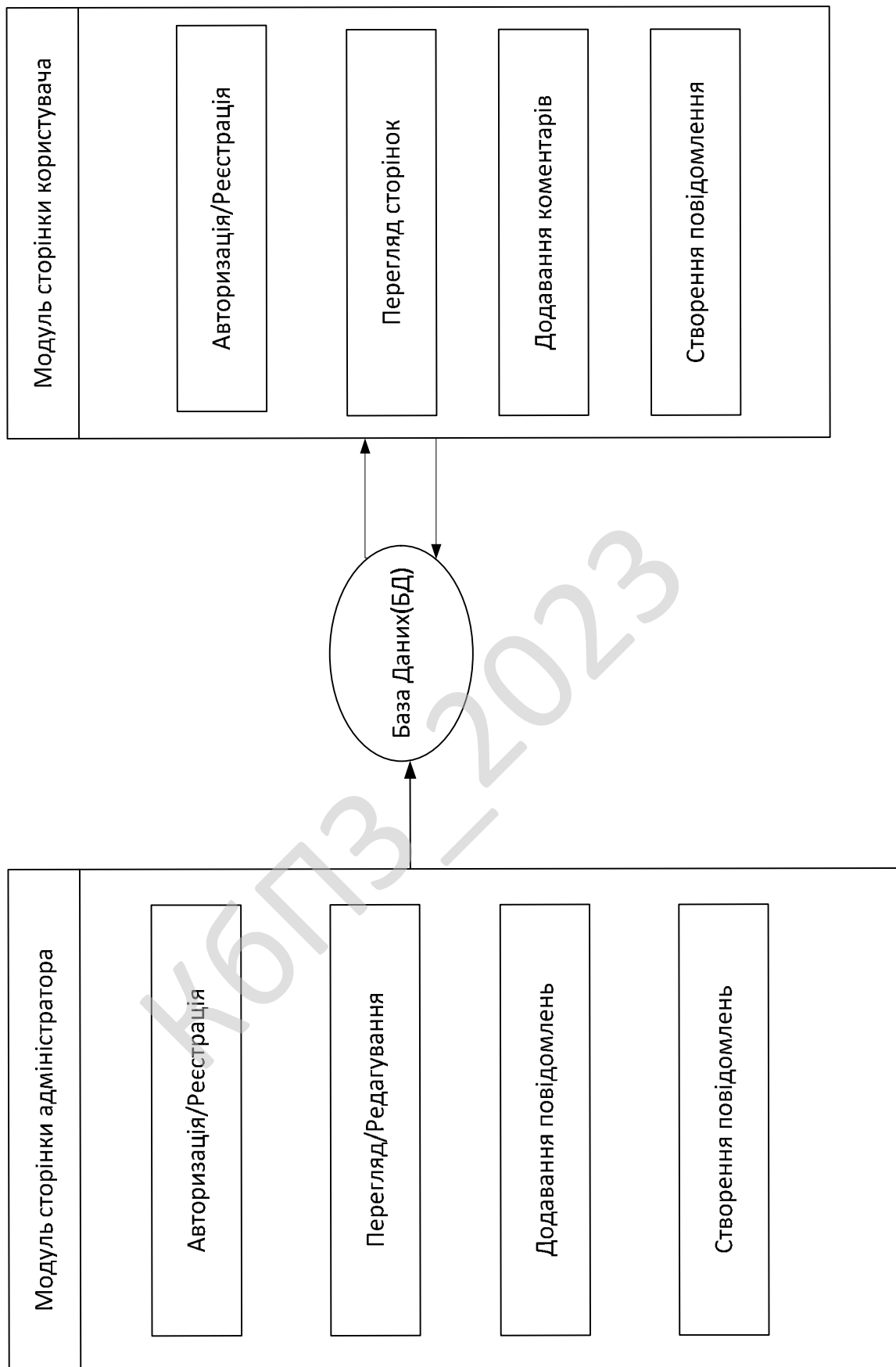


Рисунок 3.4 – Функціональна схема системи

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 магістерської роботи, наведена на рисунку 3.5.

КБПЗ\_2023

					VKPM-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43



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

### 4.1 Розробка блок-схем та опис алгоритмів функціонування системи

На рисунку 4.1 зображено блок-схему основної програми. Її робота полягає у виконанні наступних кроків.

Після завантаження сайту відкривається відповідна сторінка сайту.

Потім для роботи з сайтом необхідно зареєструватися в системі. Перевіряється авторизація користувача, якщо користувач не авторизований, на екрані відображається помилка щодо запитуваної авторизації. Якщо користувач авторизований, вміст відображається.

Потім система очікує від користувача певної дії. Щоб зв'язатися з користувачем з онлайн-підтримкою, необхідно зайти у відповідне меню сайту.

Коли з'явиться форма авторизації, користувач може або перейти на сторінку реєстрації нового облікового запису, якщо такої немає, або ввести свої дані, тобто логін і пароль для авторизації. Після натискання кнопки авторизації система перевіряє правильність заповнення полів. Якщо поля заповнені неправильно, система виведе на екран повідомлення про помилку. Якщо поля заповнені правильно, система перевірить наявність введених користувачем даних у таблиці користувачів бази даних. Якщо такий користувач є, система автоматично авторизує користувача на сайті, інакше видасть повідомлення про помилку та чекає подальших дій користувача.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

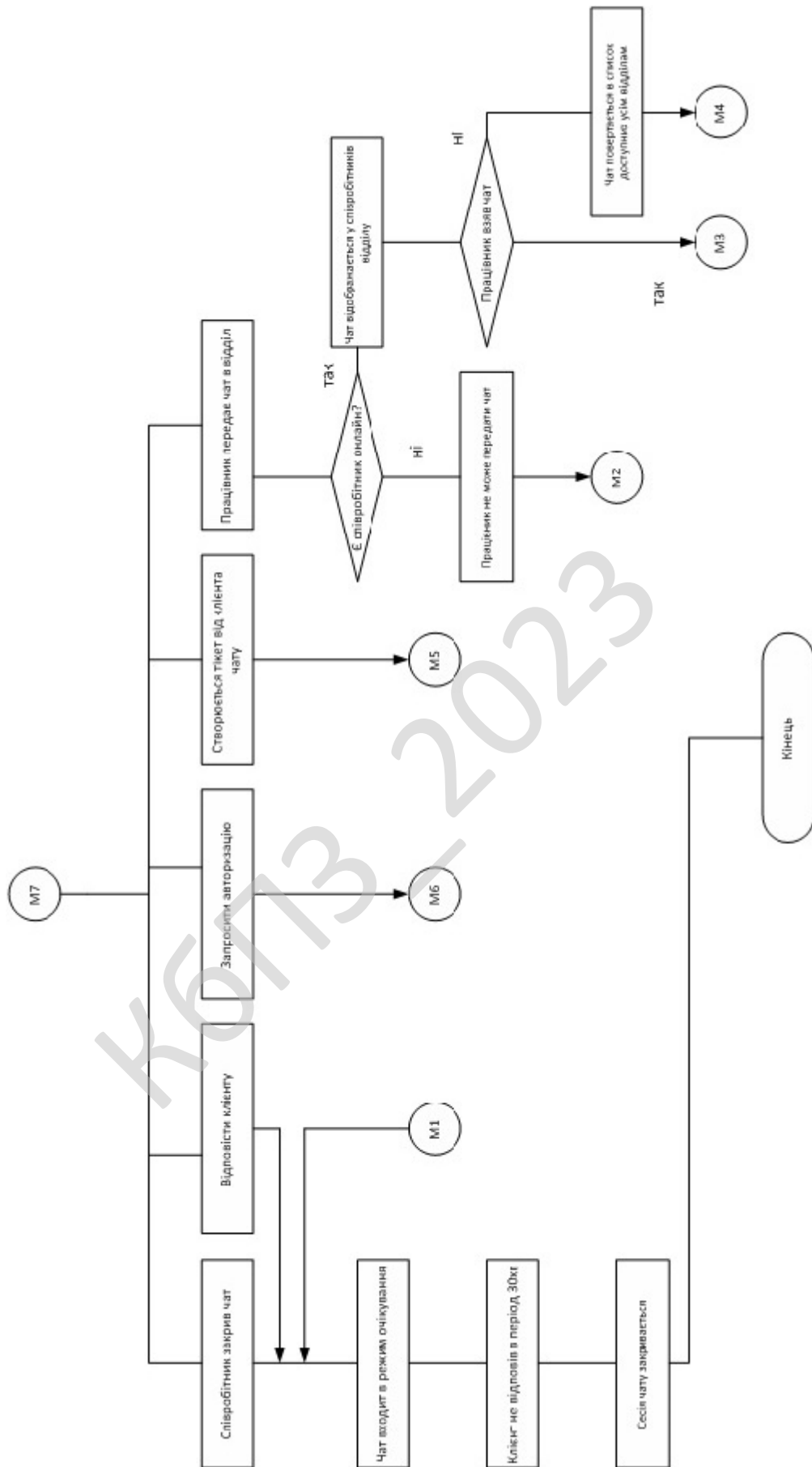


Рисунок 4.1 – Блок схема роботи головної програми

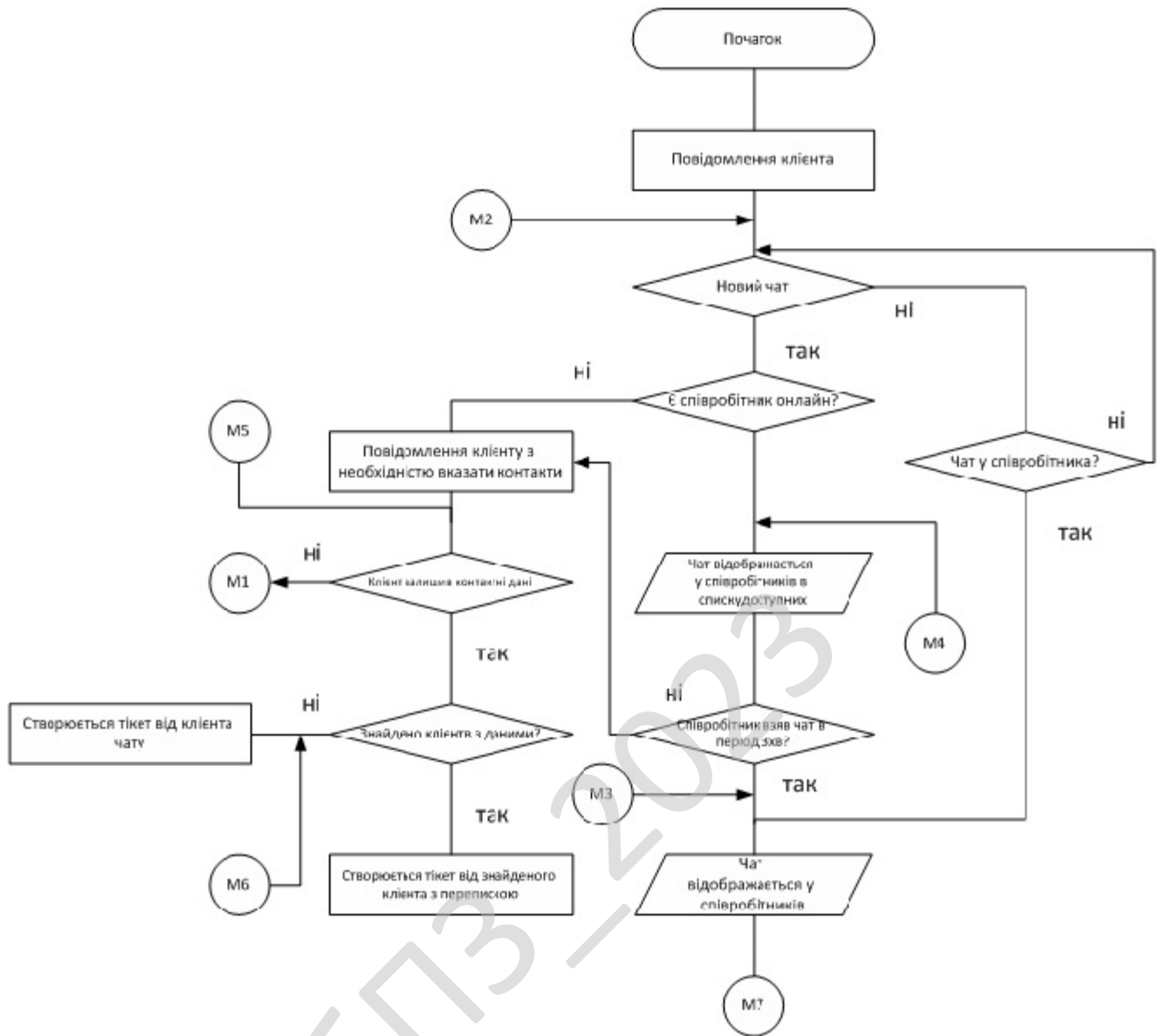


Рисунок 4.1, аркуш 2

### Програмування авторизації користувача

Авторизація буде здійснюватися на основі введення авторизаційних даних, які використовуються персональним логіном і паролем користувача.

Алгоритм роботи авторизації користувача.

1. Введення персональних даних користувача, тобто логіна та пароля.
2. Перевірка правильності введених даних користувача, тобто чи всі поля введення заповнені.

3. Перевірка наявності користувача в базі та відповідності введених даних авторизації (розгалужена гілка алгоритму).

4. Увійти в особистий профіль користувача.

Розгалуження алгоритму означає наявність варіантів подій залежно від результату.

На рисунку 4.2 зображено блок-схему системи авторизації користувачів.

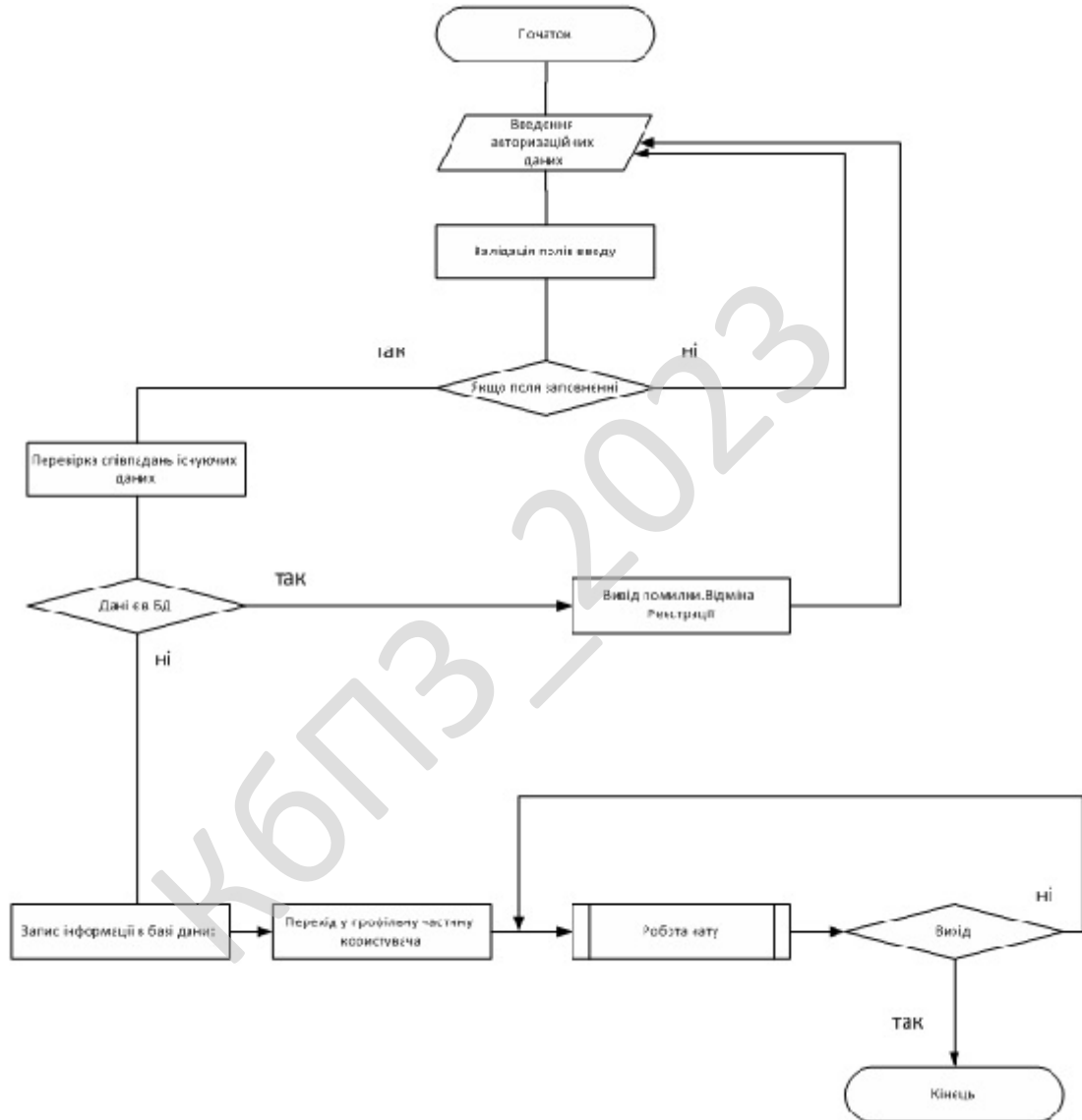


Рисунок 4.2 – Блок-схема алгоритму роботи авторизації користувача



```

$zaput="insert      into      register      (username,password,status)
values('".$$name."', '".$$password."', '0')";
$result=mysql_query($zaput);
if(!$result){echo"Помилка запису даних";}
}

```

### **Програмування профільної частини користувача**

Програмування профілю користувача буде базуватися на технології Ajax. фреймворк jquery буде використано для реалізації технології Ajax. Це дуже зручний фреймворк, який дозволяє використовувати всі можливості javascript для роботи з технологією Ajax.

Розділ профілю користувача міститиме набір програмованих функцій. У цій частині програмування веб-додатків буде реалізована технологія Ajax.

Програмування профілю умовно ділиться на наступні розділи:

- відправка повідомлень;
- редагування повідомлень;
- видалення повідомлень;
- оновлення статусу користувача та індикаторів повідомлень;
- розгортання.

### **Функція програмування обміну повідомленнями**

Передача повідомлень між користувачами буде здійснюватися в асинхронному режимі. Кожне повідомлення зберігається в базі даних відповідно до індивідуального ідентифікатора користувача.

Нижче наведено блок-схему функції обміну повідомленнями.

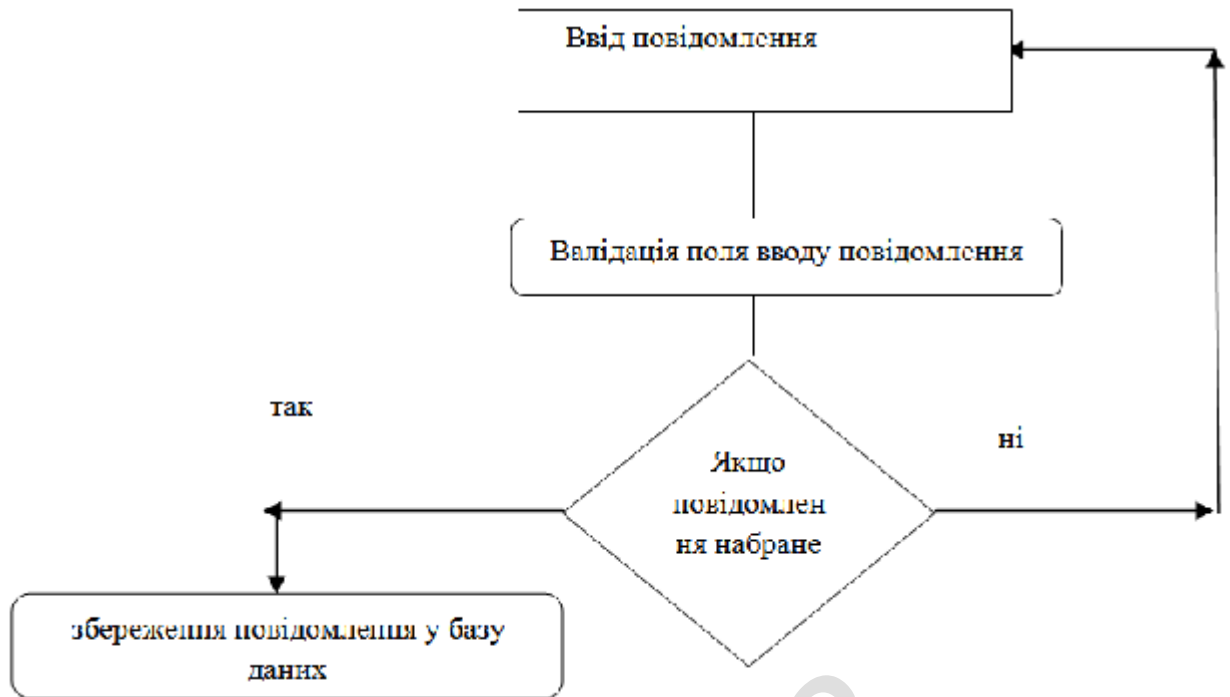


Рисунок 4.3 – Блок схема роботи функціоналу відправки повідомлень

Повідомлення надсилаються безпосередньо після реєстрації або авторизації користувача в системі. За обмін повідомленнями між користувачами відповідає файл `send.php`. Під час надсилання повідомлення виконується код javascript у файлі `output.js`, який знаходиться в папці `javascript` у файловій структурі веб-проекту. Цей код наведено в списку програм.

Відправка повідомлення починається з натискання користувачем кнопки «Відправити», але перед відправкою перевіряється, чи не порожнє поле введення повідомлення. Якщо повідомлення не введено, але користувач намагається його надіслати, виникне помилка. Потім запускається подія, запрограмована фреймворком JavaScript `jQuery`. Ця подія записується так: `$("#send_to").click(function(){}).` Тобто подія запускається, коли користувач натискає певний елемент веб-сторінки. У нашому випадку це Елемент "send\_to", який визначає кнопку для надсилання повідомлення. Подія викликає функцію `sendMess`.

Ця функція використовує асинхронний запит і надсилає дані на сервер у фоновому режимі, тобто немає перевантаження веб-сторінки, як у класична модель. Функція sendMess використовує функцію Ajax '\$.post()'. Функція '\$.post()' надсилає вхідне повідомлення на сервер для обробки у файл send\_mes.php Цей файл відповідає за збереження повідомлення користувача в базі даних Вміст файлу send\_mess.php показано в лістингу.

Отже, файл send\_mess.php використовує функції php для збереження нового повідомлення користувача в базі даних. Збереження повідомлення в базі даних здійснюється за допомогою функції send\_mess, запрограмованої у файлі send\_mess.php. У цю функцію як параметри передається адресна книга та текст відправника повідомлення. Дані Уся ця інформація передається в базу даних, де зберігається безпосередньо як ряд даних користувача.

### **Програмування функції видалення повідомлення**

Повідомлення будуть видалятися синхронно з роботою користувача. За видалення повідомлення відповідає файл user\_message.js, який зберігається в папці javascript файлової структури програми. Повідомлення видаляє лише той користувач, який його видалив. Якщо обидва користувачі видалили одне й те саме повідомлення, воно видаляється з бази даних, інакше воно зберігається. Нижче наведено блок-схему функції видалення повідомлень користувача.

Тому перед видаленням повідомлення перевіряють, чи воно стосується іншого користувача. Якщо обидва користувачі видаляють повідомлення, воно видаляється з бази даних.

Програмна реалізація функціоналу видалення повідомлень користувача. У базі даних, щоб визначити видалення повідомлень, є два поля, а саме delete\_to і delete\_from. У цих полях зберігається ідентифікатор користувача, який видалив повідомлення, тому код визначає, як саме слід видаляти повідомлення.

```
$(document).ready(function() {  
    $(".delete").click(function() {  
        var id=$(this).parent().find(".id_message").text();  
        var delete_from=$("#delete_from").text();
```

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52



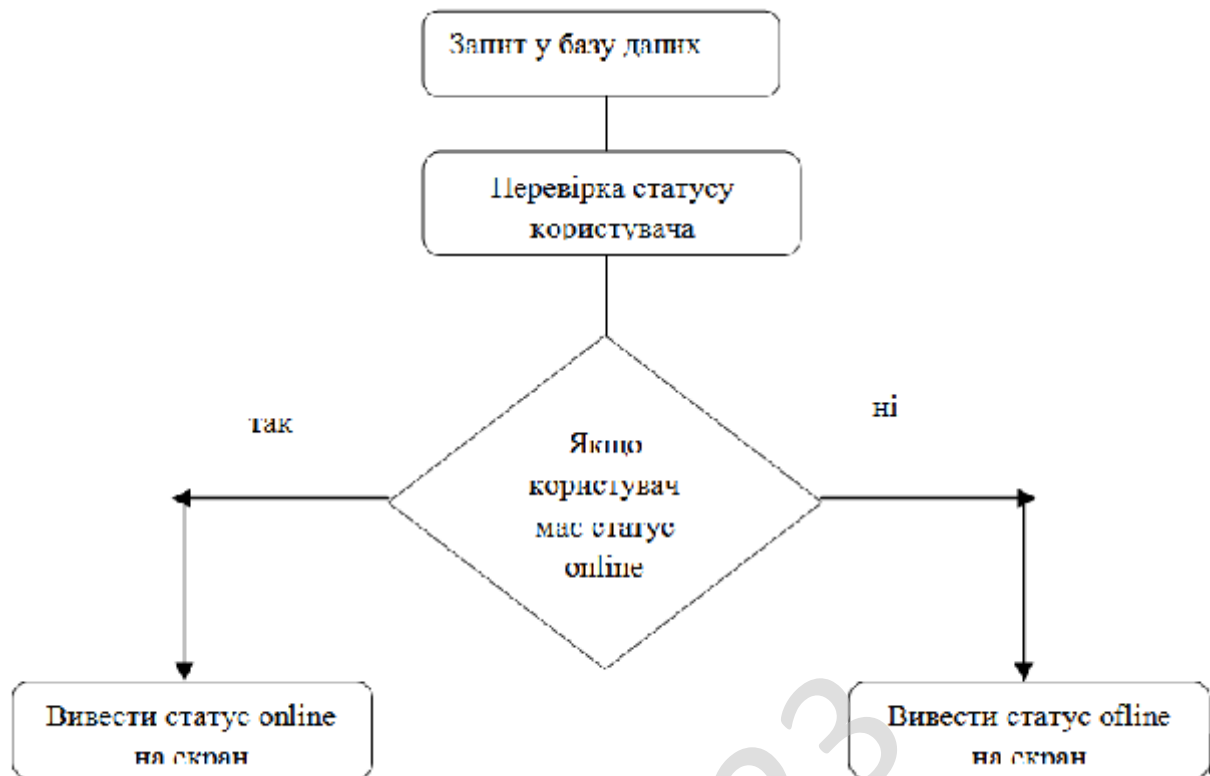


Рисунок 4.5 – Блок схема функціоналу оновлення статусу користувачів

Так, якщо користувач авторизований або зареєстрований в системі, його статус в базі змінюється на 1, в іншому випадку за замовчуванням залишається 0. За відображення статусу користувача відповідає файл output.js, який знаходиться в папці javascript. У цьому файлі є функція setInterval, яка дозволяє робити проміжні запити до веб-сервера. У нашому випадку функція оновлює статус користувача кожні 3 секунди, тобто робить синхронний запит і повертає результат. Нижче наведено код для функції оновлення статусу користувача.

```

setInterval('$("#vsi_korustuvachi").load("update.php",{user1:$("#user1").text()});',3000);

```

Тому ця функція посилається на файл update.php, який, у свою чергу, робить запити до бази даних. Програмний код файлу update.php наочно демонструє його функціональність.







## 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розробленим програмним забезпеченням даної магістерської роботи є створення веб-сайту з необхідним функціоналом, а саме:

- проектувати структуру веб-сайту та структуру бази даних;
- реалізувати клієнтську частину сайту;
- розробити модуль для перегляду сторінок сайту;
- провести пошук по сайту;
- забезпечити можливість реєстрації користувачів на сайті;
- писати скрипти для авторизації користувачів;
- реалізувати онлайн-чат;
- написати програмні модулі для забезпечення обмеженого доступу до адміністративної системи;
- наповнити сайт вихідною інформацією.

Код розробленої системи наведено у додатку В.

Щоб ним скористатися, необхідно спочатку перейти за посиланням:  
<http://fortuna.com>.

Перейшовши за цим посиланням, ви побачите головну сторінку сайту, яка зображена на рисунку 5.1.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>58</b>

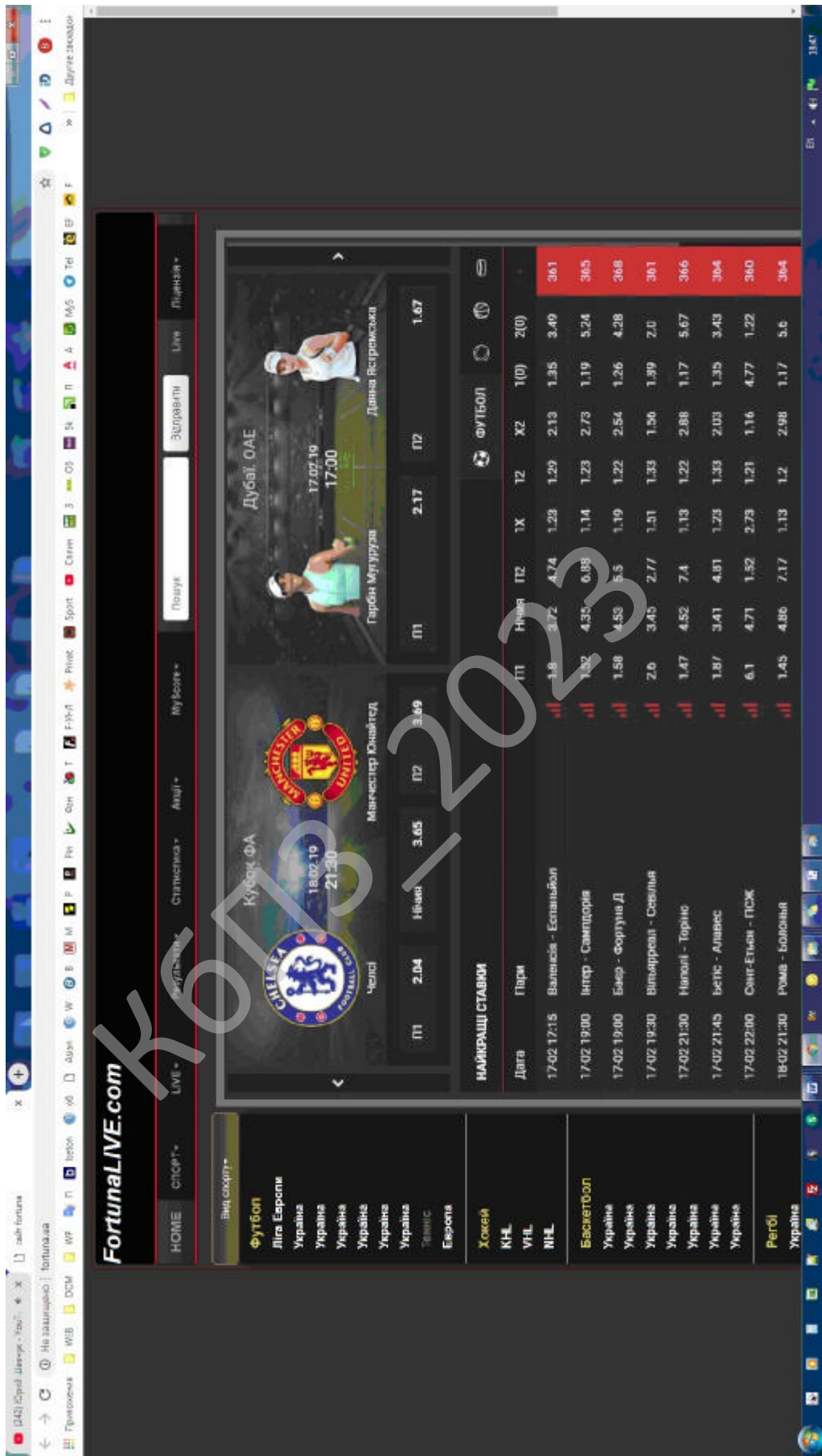


Рисунок 5.1 – Головна сторінка сайту

Для роботи з сайтом потрібно зайти в меню Login та зареєструватися або авторизуватися.

Форма данного меню представлено на рисунку 5.2.

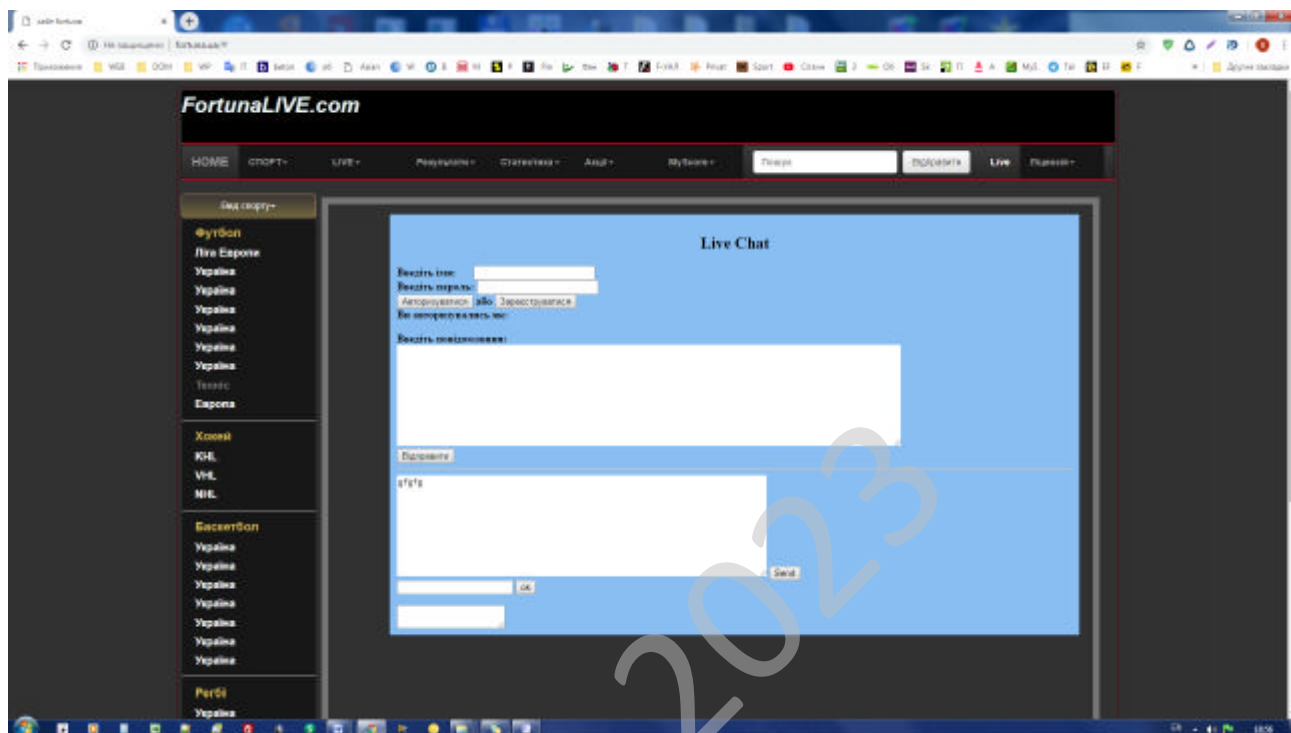


Рисунок 5.2 – Форма авторизації на сайті та онлайн чат

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

## 6 НАУКОВА НОВИЗНА

У магістерській роботі розроблено програмне забезпечення, яке призначено для миттєвого обміну даними в корпоративній мережі.

*Метою розробки є дослідження та програмна реалізація системи передачі повідомлень в режимі реального часу.*

*Об'єктом дослідження є процес передачі даних в мережі Інтернет в режимі реального часу.*

*Предметом дослідження є методи реалізації систем миттєвої передачі повідомлень в мережі Інтернет.*

*Методи дослідження базуються на методах математичної статистики, методах захисту інформації, методах розробки програмного забезпечення.*

**Наукова новизна отриманих результатів.** У процесі написання завдань, визначених цілями дослідження, отримано наступні результати:

- розроблено систему передачі миттєвих повідомлень в Інтернеті;
- проведено огляд комунікаційних технологій передачі даних в мережі Інтернет;
- розроблено вітчизняний продукт системи онлайн-чату з підсистемою безпечної передачі даних, який має більш потужні можливості, на відміну від існуючих аналогів.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

## 7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

### 7.1 Техніко-економічне обґрунтування теми магістерської роботи

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи веб-чату для обміну даними в мережі Інтернет.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
Кількість розроблених програм період, шт.	N	1
Кількість екземплярів програм, шт.	Ne	80 (ост. цифра № зал *10 <sup>1</sup> )
Запланований термін розробки, днів	Frq	60 (3 місяці)
Група задачі підсистеми управління (1-6)	–	1
Ступінь новизни задачі (А, Б, В, Г)	–	Б
Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
Кількість макетів вхідної інформації	–	3
Кількість форм вихідної інформації.	–	4
Мова програмування (1-6)	–	3
Попередній досвід (1-6)	–	3
Гнучкість проекту ПП (1-6)	–	3
Детальність проекту ПП (1-6)	–	2
Рівень спрацьованості колективу (1-6)	–	2
Ступінь вимірності процесів (1-6)	–	3
Необхідна надійність програмного забезпечення (1-6)	–	2
Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
Складність кінцевого програмного продукту (1-6)	–	2
Необхідний рівень забезпечення повторного використання (1-6)	–	2
Документованість відповідно до планованого життєвого циклу (1-6)	–	2
Вимоги до швидкодії ПП (1-6)	–	2
Обмеження на розміри основного сховища даних (1-6)	–	2
Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
Професійний рівень аналітиків (1-6)	–	2
Професійний рівень програмістів (1-6)	–	2
Постійність складу команди розробників (1-6)	–	2
Досвід розробки додатків (1-6)	–	2
Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
Досвід роботи з програмними інструментами розробки (1-6)	–	3
Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
Вартість ПЗ у розробника (НМА), грн.	–	80000 (2 ост. цифри № зал*10 <sup>4</sup> )
Норматив додаткової зарплати, % :	Нд	10
Норматив відрахувань у соціальні фонди, %	Нс	22
Норматив загальногосподарських витрат, %	Нг	15
Норматив витрат на освоєння нових мов програмування, %	Нп	15
Рівень рентабельності програмної продукції, %	Ре	50
Ставка податку на додану вартість, %	Ндв	20

## 7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

де:  $A$  – коефіцієнт Боема,  $A = 2,45$ ;

$Size$  – загальний об'єм відлагодженого програмного коду, тис. рядків;

$B$  – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де:  $W_i$  – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де:  $\prod V_j$  – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{рп} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де:  $C$  – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

$S$  – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{рп} = 0,3 \cdot 2,85 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 87 = 168 \text{ люд/день.}$$

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

### 7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{нз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де:  $F_{pq}$  – плановий фонд робочого часу одного спеціаліста, днів;

$T_{нз}$  – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор – маршрутизатор	155	2	310	5
Кабельні господарства ЛВС на 1 м. п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 <sub>ч</sub>	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

$$\Phi_{др}^c = \frac{3_ч \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{др}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2012 R2, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	8000	24000
Продакт-менеджер	0,25	7000	5250
Інженер-програміст	3,8	7000	79800
Інженер-електронщик	1,2	7000	25200
Інженер-системотехнік	0,25	7000	5250
Адміністратор мережі	0,5	7000	10500
Системний програміст	0,25	7000	5250
Дизайнер WEB	0,25	7000	5250
Інженер-верстальник	0,25	7000	5250
Бухгалтер-економіст	0,5	7000	10500
Всього за період розробки	$R_{cn} = 8,25$	-	$\Phi_{роб} = 176250$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{сд} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де:  $\Phi_{роб}$  – загальна сума зарплати за плановий період, грн.

$$z_{сд} = \frac{176250}{8,25 \cdot 60} = 356 \text{ грн.}$$

#### 7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$B_{yд} = R_{cn}^1 S_y \Pi_{пл}, \quad (7.9)$$

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

де:  $R_{cn}^1$  – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

$S_y$  – питома площа на одне робоче місце,  $m^2$ ;

$C_{пл}$  – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ $m^2$ . Враховуючи, що курс складає 1 у.о. = 25 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $m^2$ . На кожне робоче місце у середньому потрібно  $8m^2$ . З урахуванням цього:

$$B_{yd} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{не} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де:  $C_m$  – ціна меблів для одного робочого місця, грн.

$$I_{не} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом Інтернет магазину Компбест за 17.11.20 – джерело <https://compbest.com.ua>.

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		Acer X2632G DT
Процесор	Intel Core i3-4130 (2(4) ядра по 3.40 GHz)	-
Системна плата	X4660G VX4660G-I5840H1 4x USB 2.0, 2x USB 3.0, 5x Audio Ports, Com Port, 2x PS/2, LAN (RJ-45); DVI, HDMI	-
Відеокарта	AMD Radeon RX 550 4GB GDDR5 Red Dragon PowerColor (AXRX 550 4GBD5-DH)	-
Жорсткий диск	HDD Seagate Barracuda 750 Gb 7200 32Mb SATAII ST3750528AS (ST3750528AS) + 120 GB SSD	-
Оперативна пам'ять	DIMM 4096Mb DDR3 PC3-10600 CL9 Transcend JetRam, non-Reg., no-ECC , CL 9 (2 модулі)	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	GRESSO GE-7525, 500W (120mm big fan), 2xIDE, full-ATX,БЖ 2xSATA, 1xFDD, Air Duct, 2xUSB 2.0, Mic+Audio, silver/black	-

					<b>БКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D ( 5ms, 300/3000:1, 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	968	96,8	1064,8
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	200241,8

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
Будівлі	1280000	-	-
Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
Обчислювальна техніка	200242	-	-
Всього по групі	200242	50	100121
Нематеріальні активи			
Нематеріальні активи	80000	10	8000
Група 5, 6			
Вимірювальні пристрої	9031	25	2257,75
Транспортні засоби	122500	20	24500
Господарський інвентар	28000	25	7000
Всього по групі	159531	-	20000
Разом	$K_p = 1847773$		$A_p = 198521$

Примітка: вартість автомобіля (Suzuki GSX-R 600, 2010г) взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot69649.html>, складає 2900 USD, що враховуючи курс 25 складає 122500 грн.

## 7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де:  $N_e$  – кількість екземплярів програм, шт.

$$Z_o = 356 \cdot 209 / 80 = 930 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де:  $H_q$  – норматив додаткової зарплати, %.

$$Z_d = 930 \cdot 10 \cdot 0,01 = 93 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом  $H_c = 22\%$  від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де:  $H_c$  – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(930+93) = 225 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом  $H_z = 15\%$  від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де:  $H_z$  – загальногосподарські витрати, %.

$$G_{ocn} = 930 \cdot 15 \cdot 0,01 = 140 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де:  $Z_{M1}$  – вартість паперу, грн.;

$Z_{M2}$  – вартість запам'ятовуючих пристроїв, грн.;

$Z_{M3}$  – вартість фарби, картриджей, тонеру, грн.;

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

$N_e$  – кількість екземплярів програм, шт.

Згідно виданих викладачем норм приймаємо одну пачку паперу на квартал. Тоді, враховуючи, що вартість пачки паперу складає  $Ц_n = 105$  грн., визначаємо вартість паперу за період розробки  $N_m = 3$  міс:

$$З_{M1} = Ц_n \cdot N_m. \quad (7.16)$$

$$З_{M1} = 105 \cdot 1 = 105 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$З_{M2} = \sum Ц_{\delta}, \quad (7.17)$$

де:  $Ц_{\delta}$  – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 2 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 2 грн./шт.

$$З_{M2} = 80 \cdot 12 = 960 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$З_{M3} = \sum Ц_{з.}, \quad (7.18)$$

де:  $Ц_{з.}$  – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$З_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$З_M = (105 + 960 + 1702) / 80 = 34 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ( $H_n = 15\%$ ) від основної зарплати виконавців:

$$O_n = З_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де:  $H_n$  – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 930 \cdot 15 \cdot 0,01 = 140 \text{ грн.}$$

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ( $N_e = 80$  прим.):

$$A_m = \frac{A_p \cdot N_{\text{міс}}}{N_e \cdot 12}, \quad (7.20)$$

де:  $A_p$  – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 198521 \cdot 3 / (80 \cdot 12) = 620 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1. Основна зарплата виконавців	$Z_o$	930
2. Додаткова зарплата виконавців	$Z_d$	93
3. Відрахування на соціальні потреби	$C_{oc}$	225
4. Загальногосподарські витрати	$G_{ocn}$	140
5. Витрати на матеріали	$Z_M$	34
6. Освоєння нових операційних систем, мов програмування	$O_n$	140
7. Амортизація основних фондів	$A_m$	620
8. Повна собівартість програмного забезпечення	$C_n$	2182
9. Плановий прибуток	$P_p$	1091
10. Ціна підприємства $C_n = C_n + P_p$	$C_n$	3273
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{\text{дв}} \cdot C_n$	$ПДВ$	654,6
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	$C$	3927,6

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 930 + 93 + 225 + 140 + 34 + 140 + 630 = 2182 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності ( $P_n$ ) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де:  $P_n$  – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 2182 = 1091 \text{ грн.}$$

## 7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.9.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	3928
Всього капітальних витрат	–	3928

## 7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на відновлення працездатності системи після збоїв та профілактичні роботи	$Z_p$	21901	4380
Витрати на амортизацію	$Z_{ам}$	0	982
Всього витрат за рік	$I$	21901	5362

Витрати на відновлення працездатності системи після збоїв та профілактичні роботи:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де:  $T_p$  – кількість годин обслуговування сегменту мережі за рік, год.;

$Z_z$  – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 48 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 68 \cdot 1,1 \cdot 1,22 = 21901 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 48 \cdot 68 \cdot 1,1 \cdot 1,22 = 4380 \text{ грн}.$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів.

Для розрахунку складаємо таблицю 7.12.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн., за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	3928	–	982
Всього відрахувань	-	–	3928	–	982

### 7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.24)$$

де:  $K_p$  – балансова вартість основних фондів розробника, грн.;  $E_p$  – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (3273 - 2182) \cdot 80 - (0,05 \cdot 1408000 + 0,3 \cdot 200242 + 0,25 \cdot 37031 + 0,2 \cdot 122500 + 0,1 \cdot 80000) \cdot 3/12 = 44222 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.25)$$

де:  $K_p^*$  – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_e = \frac{439773}{(3273 - 2182) \cdot 80 \cdot 12 / 3} = 1,26 \text{ років.}$$

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\delta} - I_n) - E_n(K_n - K_{\delta}), \quad (7.26)$$

де:  $I_{\delta}$ ,  $I_n$  – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\delta}$ ,  $K_n$  – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (21901 - 5362) - 0,25 \cdot 3928 = 15557 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\delta}}{I_{\delta} - I_n}, \quad (7.27)$$

$$T_{cn} = \frac{3928}{21901 - 5362} = 0,24 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1	2	3
1. Кількість екземплярів програми	Прим.	80
2. Повна собівартість розробленої програми	Грн.	2182
3. Ціна розробленої програми	Грн.	3273
4. Плановий прибуток від реалізації розробленої програми	Грн.	1091
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1847773
7. Загальний прибуток від реалізації програмної продукції	Грн.	87280

Продовження таблиці 7.13

1	2	3
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	44222
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,26
10.Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3928
11.Величина економічного ефекту у користувача програмної продукції	Грн.	15557
12.Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,24

### 7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Протягом усієї історії людство приділяє прискіпливу увагу безпеці життя. Охорона праці є складовою частиною безпеки життя.

Законом України “Про охорону праці” регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98, та

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

«Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення впливу комп'ютера на організм програміста визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

## 8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- монотонність праці;

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>84</b>

- електромагнітні електромагнітні (у т.ч. високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шуми;
- статичні навантаження на кістково-м'язовий апарат;

### 8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 - Розміри приміщення

Найменування	Значення, м
Ширина	2,4
Довжина	3
Висота	2,8

Таблиця 8.2 - Площа та обсяг приміщення, на одного працюючого

Геометрич на характеристика	Одиниц я виміру	Норматив не значення*	Фактичне значення
Площа, S	м <sup>2</sup>	не менше 6.0	7,2
Обсяг, V	м <sup>3</sup>	не менше 20.0	20,1

\* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працює 1 людина. За даними, які наведено у табл. 8.1 та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста відповідають нормативним вимогам (Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86



приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно - ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

#### **8.4 Розробка заходів з умов поліпшення охорони праці**

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору розтіканню електричного струму на землю).

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

## 8.5 Розрахункова частина

Початкові дані для розрахунку штучного захисного заземлення:

Тип заземлення: робоче заземлення нульової точки трансформатора. Напруга — 220/380 В. Розташування заземлюючих електродів — по контуру.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача методом коефіцієнта використання заземлювачів.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта — чорнозем, нижнього шару ґрунта — глина. Умовна товщина верхнього шару ґрунта:  $H=0,8$  м. Для захисного заземлення: застосовуються вертикальні електроди — прутки довжиною  $L=2$  м. Відстань між вертикальними заземлювачами (електродами)  $A=2$  м. Діаметр вертикального електрода (прутка)  $D=60$  мм, Тип горизонтального заземлювача: металева полоса. Розміри перетину з'єднуючої полоси:  $60 \times 6$  мм. ( $b=60$  мм.). Опір заземлювача, який нормується:  $R_{3H} = 4$  Ом. Глибина закладення горизонтального контура заземлення  $t=0,6$  м.

Розрахунок захисного заземлення можна автоматизувати за допомогою програми, сирцевий код якої опублікован на стр.13-16 Метододичних вказівок до виконання розрахунків з викор. персон. ЕОМ IBM сумісного типу / Охорона праці. Ч. 1. Захисне заземлення / Кіровоград. ін-т с.-г. Машинобуд.; URL : <http://dSPACE.kntu.kr.ua/jspui/handle/123456789/4358>, або будь якої відповідної іншої.

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89





## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті магістерської роботи, призначене для реалізації системи онлайн-чату.

Вітчизняні розробки в цій сфері представлені в Україні недостатньо.

Це програмне забезпечення має простий, зрозумілий і зручний інтерфейс користувача, який забезпечує легкість освоєння роботи програмного продукту, зручність у використанні і не вимагає спеціальних спеціальних знань.

Під час підготовки завдання, яке було поставлене в магістерській роботі, я набула навичок та принципів розв'язування задач даного типу. Під час створення завдання засвоїв основні принципи та підходи до розробки засобів автоматизації розв'язання задач даного завдання. Основною метою мого дизайну було використання системного аналізу, декомпозиції проблем, пошуку оптимальних шляхів вирішення проблем, а також використання Ажах при створенні веб-додатків такого типу. Судячи з кінцевого результату проектування, рішення задачі по темі дипломного проектування реалізовано повністю. Я дізнався принцип інтеграції технології AJAX у веб-ресурс та основні програмні реалізації AJAX, які працюють на стороні клієнтської машини. Вивчав методи декомпозиції задач і складних комплексних задач. Вивчив основні підходи до побудови програмних структур на мові php, а також javascript. Ми розглянули утиліти веб-браузера в різних операційних системах для створення веб-ресурсів для кількох браузерів. Для того, щоб користуватися базою даних, я опанував синтаксис mysql і основні функції реляційних таблиць. Для роботи з технологією Ажах я опанував використання javascript фреймворку jquery. З метою перевірки роботи системи я вивчив підхід до створення тестового алгоритму веб-додатку.

В якості остаточного результату дизайну я розробив веб-додаток за технологією Ажах, який є повноцінною системою діалогового спілкування для сайту. Коли я використовував Ажах, я навчився працювати з javascript і

					ВКРМ-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

програмувати динамічні функції на мові php, щоб реалізувати їх у веб-ресурсі.

Існують також шляхи вдосконалення цього програмного рішення, які пропонуються в цій роботі. Код програми можна змінити для режиму сумісності з різними браузерами портативних пристроїв. Це дозволить веб-додатку працювати на будь-якому пристрої, який має вбудований веб-браузер і доступ до мережі.

Магістерська робота дала мені можливість вивчити новітні технології програмування інтернет-сайтів та опанувати сучасні методи та підходи до проектування веб-додатків високого ступеня складності.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень і повністю відповідає вимогам технічного завдання.

Запропонована програма має загальний економічний ефект з моменту впровадження у виробництво в сумі 12878 грн. З урахуванням витрат на тестування програми та обладнання коефіцієнт укорочення становить 0,24 роки.

КБПЗ-2023

					VKPM-123.23.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мова програмування JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.javascript.ru/>
2. Introduction to Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.dev/>
3. Серверне програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Learn/Server-side>
4. Node.js v14.0.0 Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/api/>
5. М. Кантелон , М. Хартер, Т. Головайчук, Н. Райлих // “Node.js в действии”.
6. Янг А., Мек Б., Кантелон М. // “Node.js в действии. 2-е издание”.
7. John Resig, Bear Bibeault, Josip Maras // “Secrets of theJavaScript Ninja”.
8. Express [Електронний ресурс] – Режим доступу до ресурсу: <http://expressjs.com/>
9. Фреймворк AngularJS [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/AngularJS>
10. AngularJS [Електронний ресурс] – Режим доступу до ресурсу: <https://angularjs.org/>
11. Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>
12. React.js [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/>
13. AngularJS MVC [Електронний ресурс] – Режим доступу до ресурсу: [https://www.tutorialspoint.com/angularjs/angularjs\\_mvc\\_architecture.htm](https://www.tutorialspoint.com/angularjs/angularjs_mvc_architecture.htm)
14. Vue.js [Електронний ресурс] – Режим доступу до ресурсу: <https://vuejs.org/>

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

15. Angular 2 [Електронний ресурс] – Режим доступу до ресурсу:  
<https://angular.io/>

16. Односторінковий застосунок [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Односторінковий\\_застосунок](https://uk.wikipedia.org/wiki/Односторінковий_застосунок)

17. Что такое MVC [Електронний ресурс] – Режим доступу до ресурсу:  
<https://ru.hexlet.io/blog/posts/что-такое-mvc-rasskazyvaem-prostymi-slovami>

18. Build Node.js Apps [Електронний ресурс] – Режим доступу до ресурсу:  
<https://code.visualstudio.com/docs/nodejs/nodejs-tutorial>

19. REST [Електронний ресурс] – Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/REST>

20. What is REST [Електронний ресурс] – Режим доступу до ресурсу:  
<https://restfulapi.net/>

21. Тренды веб-разработки [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/451572/>

22. Веб приложение [Електронний ресурс] – Режим доступу до ресурсу:  
<https://webcase.com.ua/blog/cho-takoe-web-prilozhenie-vse-vidy/>

23. Мова розмітки гіпертексту [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML>

24. HTML [Електронний ресурс] – Режим доступу до ресурсу:  
<https://developer.mozilla.org/uk/docs/Web/HTML>

25. Каскадні таблиці стилів [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/CSS>

26. Стек MEAN [Електронний ресурс] – Режим доступу до ресурсу:  
[https://uk.wikipedia.org/wiki/MEAN\\_\(веброзробка\)](https://uk.wikipedia.org/wiki/MEAN_(веброзробка))

27. MongoDB [Електронний ресурс] – Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/MongoDB>

28. Веб-технології для розробників [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/uk/docs/Web>

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95



43. Postman [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.postman.com/>

44. SQL [Електронний ресурс] – Режим доступу до ресурсу:  
<https://uk.wikipedia.org/wiki/SQL>

45. SPA (Single-page application) [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

46. Охорона праці [Електронний ресурс]: реферат – Режим доступу до ресурсу: [https://revolution.allbest.ru/life/00468031\\_0.html](https://revolution.allbest.ru/life/00468031_0.html)

47. Природне і штучне освітлення ДБН В.2.5-28:2018: державні будівельні норми України [Електронний ресурс] / Ю. Громадський, С. Облакевич, М.Громадський, Г. Фаренюк, Є. Фаренюк, О. Підгорний, О. Сергейчук, Є.Рейцен, В. Єгорченков, Л. Коваль, Д. Радомцев, В. Злоба, Н. Кучеренко, Г.Кожушко, О. Гончар, О. Козенко, Б. Шабашкевіч, Ю. Добровольський, В.Акіменко, С. Гозак, А. Яригін, В. Назаренко, В. Мартиросова, В. Сорокін, Є.Пугачов - Київ 2018 - Режим доступу до ресурсу: [https://ledeffect.com.ua/images/\\_branding/dbn2018.pdf](https://ledeffect.com.ua/images/_branding/dbn2018.pdf)

48. Розрахунок світлодіодного освітлення кімнати [Електронний ресурс] – Режим доступу до ресурсу: <https://luxled.biz.ua/rozrahynok-svitlodoidnogo-osvitlennja-kimnatu-v-kvarturi-abo-bydunky>

49. Охорона праці, охорона праці та безпека в надзвичайних ситуаціях : метод. вказ. до викон. розділів у дипломних роботах / [укл. В.М. Челябієва, О.Л. Гуменюк] - Чернігів ЧДТУ 2013 - [Електронний ресурс] – Режим доступу до ресурсу: [http://ir.stu.cn.ua/bitstream/handle/123456789/12461/Охорона праці та безпека. в надзв. ситуац;метод.вказ..pdf?sequence=1&isAllowed=y](http://ir.stu.cn.ua/bitstream/handle/123456789/12461/Охорона_праці_та_безпека._в_надзв._ситуац;метод.вказ..pdf?sequence=1&isAllowed=y)

50. Освітленість робочих місць [Електронний ресурс] – Режим доступу до ресурсу:[https://ua-referat.com/Освітленість\\_робочих\\_місць\\_сучасні\\_підходи\\_до\\_вимірів\\_і\\_оцінки](https://ua-referat.com/Освітленість_робочих_місць_сучасні_підходи_до_вимірів_і_оцінки)

					<b>ВКРМ-123.23.0001.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					<b>ВКРМ-123.23.0001.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Амбросьев Д.А.				Дослідження та програмна реалізація системи веб-чату для обміну даними в мережі Інтернет	Літ.	Аркуш	Аркушів
Перевірів	Пархоменко Ю.					Б	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22М-1			
Затв.	Смірнов О.А.							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку програмного забезпечення системи веб-чату обміну даними в мережі Інтернет.

## 2 Підстава для розробки

Підставою для розробки служить завдання на магістерську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 34-13 від 04.08.2023 року).

## 3 Мета та призначення розробки

Метою магістерської роботи є дослідження та програмна реалізація системи веб-чату для обміну даними в мережі Інтернет.

## 4 Джерела розробки

Джерелом цієї магістерської роботи є відносна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.23.0001.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- надійну передачу даних;
- 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.

					ВКРМ-123.23.0001.00.00.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 архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10/11 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 XP/Vista/7/8/10/11.

### 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 та JS

					ВКРМ-123.23.0001.00.00.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 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці та техніки безпеки в магістерській роботі повинен бути розглянутий аналіз умов праці програміста та розрахунок штучного захисного заземлення.

					<b>ВКРМ-123.23.0001.00.00.ТЗ</b>	Арк.
						5
Вим.	Арк.	№ документа	Підпис	Дата		

## 9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 97 аркушів.

## 10 Етапи розробки

10.1 Збір і обробка інформації по темі магістерської роботи. Постановка задачі на виконання магістерської роботи (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень магістерської роботи.

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 9 Порядок контролю та приймання

11.1 Подання магістерської роботи на попередній захист 10.12.2023 р.

1.2 Подання магістерської роботи на захист 15.12.2023 р.

					<b>ВКРМ-123.23.0001.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

ЗАТВЕРДЖУЮ

Керівник магістерської роботи

\_\_\_\_\_ Пархоменко Ю.М

*Дослідження та програмна реалізація системи веб-чату для обміну даними  
в мережі Інтернет*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 24

Літера: РП

Кропивницький – 2023 року

## ДОДАТОК

### Лістинг 1. Програмний код створення таблиць бази даних.

```

Create database exchange,
Use exchange,
CREATE TABLE `message` (
  `id` INT(5) NOT NULL AUTO_INCREMENT,
  `to_mes` INT(5) NOT NULL,
  `from_mes` INT(5) NOT NULL,
  `text` VARCHAR(50) NOT NULL,
  `status` INT(11) NOT NULL,
  `data` VARCHAR(50) NULL DEFAULT NULL,
  `delete_to` INT(10) NOT NULL,
  `delete_from` INT(10) NOT NULL,
  PRIMARY KEY (`id`)
)ENGINE=InnoDB;
CREATE TABLE `register` (
  `id` INT(5) NOT NULL AUTO_INCREMENT,
  `username` VARCHAR(40) NOT NULL,
  `password` VARCHAR(40) NOT NULL,
  `status` INT(5) NOT NULL,
  PRIMARY KEY (`id`)
)
ENGINE=InnoDB;

```

### Лістинг 2. Програмний код файлу add\_user.php

```

<?require_once('function.php');
$name=$_POST['name'];
$password=$_POST['password'];
session_start();
if(!$name || !$password){echo "Не введені обов'язкові поля.";exit;}
if(strlen($name)>16){echo "Імя містить забагато символів. Мак 16.";exit;}
if(strlen($password)>16){echo "Пароль містить забагато символів. Мак 16.";exit;}
if(strlen($password)<6){echo "Пароль містить недостатньо символів. Мін 6.";exit;}register($name,$password);
$_SESSION['valid_user']=$name;do_html_header();
echo "Ви успішно зареєстровані у ситемі.Перейдіть на сторінку для зареєстрованих користувачів.";
do_html_url('member.php','Перейти.');
```

### Лістинг 3. Функція javascript для функції відправки повідомлення.

```

$("#send_to").click(function(){
var id_rewrite_message=$("#id_rewrite_message").text();
$("#id_rewrite_message").remove();
$("#message_send").show();
if($("#text").val()==""){$("#message_send").text("Введіть текст повідомлення!");}
else{
sendMess(to,youText.val(),id_rewrite_message);
$("#message_send").text("Повідомлення відправлено!");
$("#message_send").hide(700);
$("#text").each(function () {
$(this).val($(this).data('defvalue'));
});});});
function sendMess(to,text,id_rewrite_message){
$.post("send_mess.php",{text:text,to:to,id_rewrite_message:id_rewrite_message});
}

```

## Лістинг 4. Файл `send_mess.php` відповідає за збереження повідомлення у базі даних.

```
<?require_once('function.php');
session_start();
db_connect();
$message=$_POST["text"];
$to=$_POST["to"];
$id_rewrite_message=$_POST['id_rewrite_message'];
if(isset($id_rewrite_message))
{$from=user_id($_SESSION["valid_user"]);
send_mess($from,$to,$message,$id_rewrite_message);
}else{
$from=user_id($_SESSION["valid_user"]);
$id_rewrite_message=0;
send_mess($from,$to,$message,$id_rewrite_message);}?>
```

## Лістинг 5. Функція оновлення статусу користувачів `output_user_to_messages`.

```
function output_users_to_messages($log_user)
{db_connect();
mysql_select_db("user");
$zaput="select username,id,status from register where username
!='$log_user' ";
$result=mysql_query($zaput);
while ($row = mysql_fetch_assoc($result)) {
$user=$row['id'];
$user2=user_id($log_user);
$zaput2="select id from message where status=0 and to_mes='$user2' and
from_mes='$user' ";
$result2=mysql_query($zaput2);
$i=0;
while ($row2 = mysql_fetch_assoc($result2))
{$i=$i+1;}
if($row2['status']==1){ ?>
<font color=blue>
<?echo "Online ";
?>
</font>
<?}else{echo "Offline ";}
echo"<a href='send.php?to=".$row2["id"]."'>".$row2['username']. "</a>";
if($i==0){echo " 0";echo "<br>";}
else{
echo "+".$i;echo "<br>";
}}}
```

### Головні частини програмного коду веб додатку.

#### Файл `output.js` який відповідає за динаміку інтерфесу користувача.

```
$(document).ready(function() {
var youText=$("#text");
var to=$("#to").text();
$(function () {
$("#text").each(function () {
$(this).data('defvalue', this.value);});
});
```

```

});$("#send_to").click(function(){
var id_rewrite_message=$("#id_rewrite_message").text();
$("#id_rewrite_message").remove();
$("#message_send").show();
if($("#text").val()==""){$("#message_send").text("Введіть текст
повідомлення!");}
else{sendMess(to,youText.val(),id_rewrite_message);
$("#message_send").text("Повідомлення відправлено!");
$("#message_send").hide(700);
$("#text").each(function () {$ (this).val ($ (this).data ('defvalue'));});});
function sendMess(to,text,id_rewrite_message){
$.post("send_mess.php",{text: text,to:
to,id_rewrite_message:id_rewrite_message});
setInterval ('$ ("#message_my").load ("mess.php",{user:$ ("#my_user").text ()})
;',1000);
setInterval ('$ ("#vsi_korustuvachi").load ("update.php",{user1:$ ("#user1").t
ext ()});',3000);
setInterval ('$ ("#get_text").load ("message_user.php",{to:$ ("#to").text (),my
_user:$ ("#my_user").text ()});',1000); });

```

**Файл user\_message.js, який відповідає за видалення та редагування повідомлень користувачів.**

```

$(document).ready(function(){
$(".delete").click(function(){
var id=$(this).parent().find(".id_message").text();
var delete_from=$("#delete_from").text();
$.post("delete.php",{id:id,delete_from:delete_from});
});
$(".rewrite").live('click',function(){
var text=$(this).parent().find(".rewrite_2").text();
var id=$(this).parent().find(".id_message").text();
$("#text").text(text);
$("#id_rewrite_message").text(id);
});
});

```

**Файл member.php що відповідає за роботу профільної частини користувача.**

```

<?
require_once('function.php');
session_start();
@$username=$_POST['username'];
@$password=$_POST['password'];
if($username && $password)
{
try {login($username,$password);$ _SESSION['valid_user']=$username;
} catch (Exception $e)
{do_html_header();
echo "Вхід неможливий невірний логін та пароль.";
do_html_url('login.php','Вхід');
do_html_footer();
exit;}}//далі має бути система спілкування.
@$user=user_id($_SESSION["valid_user"]);
if(reg_user()){
do_html_header();
?><div id="content">
<div id="header"></div>

```

```

<div id="vnesh">
<div id="vnutr">
<div id="left">
<div id=user></div>
</div>
<div id="right">
<div id=menu>
<p style="display:none;" id=my_user><?echo $user;?></p>
</div>
</div>
<div id="center">
<div id=alluser>
<span>Написати повідомлення:</span><br>
<?
output_users_to_messages($_SESSION["valid_user"]);
?>
</div>
</div>
<div id=podporka></div>
</div>
</div>
<div id="footer"></div>
</div>
<?
do_html_footer();
}
?>

```

**Файл data\_valid.php який містить в собі головні функції роботи веб ужитку.**

```

<?
function db_connect()
{
@ $result=mysql_pconnect('localhost','user','password');
//mysql_query("set character_set_client='cp1251'");
//mysql_query("set character_set_results='cp1251'");
//mysql_query("set collation_connection='cp1251_general_ci'");
if(!$result){echo "Не вдалось зєднатись ыз базуо даних";exit;}
}
function register($name,$password)
{db_connect();
mysql_select_db("user");
$zaput="select * from register where username='$name' " ;
$result=mysql_query($zaput);
if(mysql_num_rows($result)>0){echo "Немає даних";exit;}
$zaput="insert into register (username,password,status)
values('".$name."','".$password."','".$0')";
$result=mysql_query($zaput);
if(!$result){echo"Невдалий результат.";}
}
function login($username, $password)
{
db_connect();mysql_select_db("user");
$zaput="select * from register where username='$username' and
password='$password'";
$result=mysql_query($zaput);
if(!$result)throw new Exception('Помилка');
if(mysql_num_rows($result)>0) return ;
else throw new Exception(' Помилка');
};function reg_user()
{global $valid_user;
if(isset($_SESSION['valid_user']))
{echo '-'.($_SESSION['valid_user']).'.';

```

```

$status=1;
$username=$_SESSION['valid_user'];
status_user($status,$username);
?>
<a href="logout.php">Вийти<a/><br><hr><?return true;
}else
{do_html_heading();echo "Поверніться та введіть обов'язкові
поля.";do_html_url('login.php','Вхід');do_html_footer();return
false;}}function status_user($status,$username){
db_connect();mysql_select_db("user");$zaput="update register set
status='$status' where username='$username'";mysql_query($zaput);}
function user_id($username){db_connect();
mysql_select_db("user");
$zaput="select id from register where username='$username'";
$result=mysql_query($zaput);
$row=mysql_fetch_assoc($result);
return $row['id'];
}function user_log($id)
{db_connect();
mysql_select_db("user");
$zaput="select username from register where id='$id'";
$result=mysql_query($zaput);$row=mysql_fetch_assoc($result);return
$row['username'];}
function output_users_to_messages($log_user)
{db_connect();
mysql_select_db("user");
$zaput="select username,id,status from register where username
!='$log_user' ";
$result=mysql_query($zaput);
while ($row = mysql_fetch_assoc($result)) {
$user=$row['id'];
$user2=user_id($log_user);
$zaput2="select id from message where status=0 and to_mes='$user2' and
from_mes='$user' ";
$result2=mysql_query($zaput2);
$i=0;while ($row2 = mysql_fetch_assoc($result2))
{
$i=$i+1;
}
if($row['status']==1){
?><font color=blue>
<?
echo "Online ";
?>
</font>
<?
}else{echo "Offline ";}
echo"<a href='send.php?to=".$row["id"]."'>".$row['username']. "</a>";
if($i==0){echo " 0";echo "<br>";}else{echo"+".$i;echo "<br>";}}}
function send_mess($from,$to,$message,$id_rewrite_message)
{db_connect();
mysql_select_db("user");
$data=date('m.d.y');
if($id_rewrite_message==false){
$zaput="insert into message (to_mes,from_mes,text,status,data,
delete_to,delete_from) values
('".$to."','".$from."','".$message."','".$0','".$ data."','".$0','0')";
$result=mysql_query($zaput);
$zaput1="update message set status=1 where to_mes='$from' and
from_mes='$to'";
$result=mysql_query($zaput1);
}else {$zaput="update message set text='$message', status=0 where
id='$id_rewrite_message'";
$result=mysql_query($zaput);}}?>

```

**Файл add\_user.php** слугує додаванням нового користувача у базу даних.

```
<?
require_once('function.php');
$name=$_POST['name'];
$password=$_POST['password'];
session_start();
if(!$name || !$password){echo "Не введені обов'язкові поля.";exit;}
if(strlen($name)>16){echo "Імя містить забагато символів. Мак 16.";exit;}
if(strlen($password)>16){echo "Пароль містить забагато символів. Мак 16.";exit;}
if(strlen($password)<6){echo "Пароль містить недостатньо символів. Мін 6.";exit;}
register($name,$password);
$_SESSION['valid_user']=$name;
do_html_header();
echo "Ви успішно зареєстровані у системі.Перейдіть на сторінку для зареєстрованих користувачів.";
do_html_url('member.php','Перейти. ');
do_html_footer();
?>
```

**Файл delete.php** видаляє коористувацькі повідомлення.

```
<?
require_once('function.php');
$id=$_POST['id'];//повідомлення
$delete_from=$_POST['delete_from'];//користувач який видаляє
db_connect();
mysql_select_db("user");
$zaput="select delete_from from mesage where id = '$id'";
$result1=mysql_query($zaput);
while ($row = mysql_fetch_array($result1))
{if($row['delete_from']>0)
{$zaput="update mesage set delete_to ='$delete_from' where id = '$id'";
mysql_query($zaput);
$zaput1="update mesage set status=1 where id='$id'";
mysql_query($zaput1);
$zaput2="delete from mesage where id = '$id'";
mysql_query($zaput2);
}else
{$zaput="update mesage set delete_from='$delete_from' where id = '$id'";
mysql_query($zaput);
}}?>
```

**Файл function.php** підключає головні функціональні файли роботи веб-додатку.

```
<?
require_once('output.php');
require_once('data_valid.php');
?>
```

## Файл login.php Головний стартовий файл роботи веб додатку.

```
<?
require_once('function.php');
do_html_header();
login_form();
do_html_footer();
?>
```

## Файл test.php існує для тестової перевірки роботи системи видалення

```
<script src=jquery-1.6.4.min.js type=text/javascript></script>
<script type=text/javascript>
$(document).ready(function() {
$("#del").click(function() {
alert($("#this").attr('value'))});});}</script>
<?
require_once('function.php');
$to=1;
$my=2;
db_connect();
mysql_select_db("message");
$zaput="select text,to_mes,status,id from message where to_mes='$to' and
from_mes='$my' || to_mes='$my' and from_mes='$to' " ;
$result=mysql_query($zaput);
while($row = mysql_fetch_assoc($result))
{echo user_log($to).":".$row['text']." <br>";
?>
<input type=submit id="delete" class="del" value=Видалити /><br>
<?>>
```

## Файл test.html

```
{{ header }}
<div id="checkout-checkout" class="container">
  <ul class="breadcrumb">
    {% for breadcrumb in breadcrumbs %}
    <li><a href="{{ breadcrumb.href }}">{{ breadcrumb.text }}</a></li>
    {% endfor %}
  </ul>
  {% if error_warning %}
  <div class="alert alert-danger alert-dismissible"><i class="fa fa-
exclamation-circle"></i> {{ error_warning }}
  <button type="button" class="close" data-
dismiss="alert">&times;</button>
  </div>
  {% endif %}
  <div class="row">{{ column_left }}
  {% if column_left and column_right %}
  {% set class = 'col-sm-6' %}
  {% elseif column_left or column_right %}
  {% set class = 'col-sm-9' %}
  {% else %}
  {% set class = 'col-sm-12' %}
  {% endif %}
  <div id="content" class="{{ class }}">{{ content_top }}
  <h1>{{ heading_title }}</h1>
  <div class="panel-group" id="accordion">
    <div class="panel panel-default">
      <div class="panel-heading">
        <h4 class="panel-title">{{ text_checkout_option }}</h4>
      </div>
```

```

    <div class="panel-collapse collapse" id="collapse-checkout-
option">
    <div class="panel-body"></div>
    </div>
    </div>
    {% if not logged and account != 'guest' %}
    <div class="panel panel-default">
    <div class="panel-heading">
    <h4 class="panel-title">{{ text_checkout_account }}</h4>
    </div>
    <div class="panel-collapse collapse" id="collapse-payment-
address">
    <div class="panel-body"></div>
    </div>
    </div>
    {% else %}
    <div class="panel panel-default">
    <div class="panel-heading">
    <h4 class="panel-title">{{ text_checkout_payment_address
}}</h4>
    </div>
    <div class="panel-collapse collapse" id="collapse-payment-
address">
    <div class="panel-body"></div>
    </div>
    </div>
    {% endif %}
    {% if shipping_required %}
    <div class="panel panel-default">
    <div class="panel-heading">
    <h4 class="panel-title">{{ text_checkout_shipping_address
}}</h4>
    </div>
    <div class="panel-collapse collapse" id="collapse-shipping-
address">
    <div class="panel-body"></div>
    </div>
    </div>
    <div class="panel panel-default">
    <div class="panel-heading">
    <h4 class="panel-title">{{ text_checkout_shipping_method
}}</h4>
    </div>
    <div class="panel-collapse collapse" id="collapse-shipping-
method">
    <div class="panel-body"></div>
    </div>
    </div>
    {% endif %}
    <div class="panel panel-default">
    <div class="panel-heading">
    <h4 class="panel-title">{{ text_checkout_payment_method
}}</h4>
    </div>
    <div class="panel-collapse collapse" id="collapse-payment-
method">
    <div class="panel-body"></div>
    </div>
    </div>
    <div class="panel panel-default">
    <div class="panel-heading">
    <h4 class="panel-title">{{ text_checkout_confirm }}</h4>
    </div>
    <div class="panel-collapse collapse" id="collapse-checkout-
confirm">
    <div class="panel-body"></div>
    </div>
    </div>
    </div>

```

```

        {{ content_bottom }}</div>
    {{ column_right }}</div>
</div>
<script type="text/javascript"><!--
$(document).on('change', 'input[name=\'account\']', function() {
    if ($('#collapse-payment-address').parent().find('.panel-heading
.panel-title > *').is('a')) {
        if (this.value == 'register') {
            $('#collapse-payment-address').parent().find('.panel-
heading .panel-title').html('<a href="#collapse-payment-address" data-
toggle="collapse" data-parent="#accordion" class="accordion-toggle">{{
text_checkout_account }} <i class="fa fa-caret-down"></i></a>');
        } else {
            $('#collapse-payment-address').parent().find('.panel-
heading .panel-title').html('<a href="#collapse-payment-address" data-
toggle="collapse" data-parent="#accordion" class="accordion-toggle">{{
text_checkout_payment_address }} <i class="fa fa-caret-down"></i></a>');
        }
    } else {
        if (this.value == 'register') {
            $('#collapse-payment-address').parent().find('.panel-
heading .panel-title').html('{{ text_checkout_account }}');
        } else {
            $('#collapse-payment-address').parent().find('.panel-
heading .panel-title').html('{{ text_checkout_payment_address }}');
        }
    }
});

{% if not logged %}
$(document).ready(function() {
    $.ajax({
        url: 'index.php?route=checkout/login',
        dataType: 'html',
        success: function(html) {
            $('#collapse-checkout-option .panel-body').html(html);

            $('#collapse-checkout-option').parent().find('.panel-
heading .panel-title').html('<a href="#collapse-checkout-option" data-
toggle="collapse" data-parent="#accordion" class="accordion-toggle">{{
text_checkout_option }} <i class="fa fa-caret-down"></i></a>');

            $('a[href=\'#collapse-checkout-
option\']').trigger('click');
        },
        error: function(xhr, ajaxOptions, thrownError) {
            alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
        }
    });
});
{% else %}
$(document).ready(function() {
    $.ajax({
        url: 'index.php?route=checkout/payment_address',
        dataType: 'html',
        success: function(html) {
            $('#collapse-payment-address .panel-body').html(html);

            $('#collapse-payment-address').parent().find('.panel-
heading .panel-title').html('<a href="#collapse-payment-address" data-
toggle="collapse" data-parent="#accordion" class="accordion-toggle">{{
text_checkout_payment_address }} <i class="fa fa-caret-down"></i></a>');

            $('a[href=\'#collapse-payment-
address\']').trigger('click');
        },
        error: function(xhr, ajaxOptions, thrownError) {

```

```

                alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
            }
        });
    });
{% endif %}

// Checkout
$(document).delegate('#button-account', 'click', function() {
    $.ajax({
        url: 'index.php?route=checkout/' +
$('input[name=\'account\']:checked').val(),
        dataType: 'html',
        beforeSend: function() {
            $('#button-account').button('loading');
        },
        complete: function() {
            $('#button-account').button('reset');
        },
        success: function(html) {
            $('.alert-dismissible, .text-danger').remove();
            $('.form-group').removeClass('has-error');

            $('#collapse-payment-address .panel-body').html(html);

            if ($('#input[name=\'account\']:checked').val() ==
'register') {
                $('#collapse-payment-
address').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-payment-address" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{ text_checkout_account }}
<i class="fa fa-caret-down"></i></a>');
            } else {
                $('#collapse-payment-
address').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-payment-address" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_payment_address }} <i class="fa fa-caret-down"></i></a>');
            }

            $('a[href=\'#collapse-payment-
address\"]').trigger('click');
        },
        error: function(xhr, ajaxOptions, thrownError) {
            alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
        }
    });
});

// Login
$(document).delegate('#button-login', 'click', function() {
    $.ajax({
        url: 'index.php?route=checkout/login/save',
        type: 'post',
        data: $('#collapse-checkout-option :input'),
        dataType: 'json',
        beforeSend: function() {
            $('#button-login').button('loading');
        },
        complete: function() {
            $('#button-login').button('reset');
        },
        success: function(json) {
            $('.alert-dismissible, .text-danger').remove();
            $('.form-group').removeClass('has-error');

            if (json['redirect']) {
                location = json['redirect'];
            }
        }
    });
});

```

```

    } else if (json['error']) {
        $('#collapse-checkout-option .panel-body').prepend('<div
class="alert alert-danger alert-dismissible"><i class="fa fa-exclamation-
circle"></i> ' + json['error']['warning'] + '<button type="button"
class="close" data-dismiss="alert">&times;</button></div>');

        // Highlight any found errors
        $('input[name=\'email\']').parent().addClass('has-
error');

        $('input[name=\'password\']').parent().addClass('has-error');
    }
},
error: function(xhr, ajaxOptions, thrownError) {
    alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
}
});
});

// Register
$(document).delegate('#button-register', 'click', function() {
    $.ajax({
        url: 'index.php?route=checkout/register/save',
        type: 'post',
        data: $('#collapse-payment-address input[type=\'text\'],
#collapse-payment-address input[type=\'date\'], #collapse-payment-address
input[type=\'datetime-local\'], #collapse-payment-address
input[type=\'time\'], #collapse-payment-address input[type=\'password\'],
#collapse-payment-address input[type=\'hidden\'], #collapse-payment-
address input[type=\'checkbox\']:checked, #collapse-payment-address
input[type=\'radio\']:checked, #collapse-payment-address textarea,
#collapse-payment-address select'),
        dataType: 'json',
        beforeSend: function() {
            $('#button-register').button('loading');
        },
        success: function(json) {
            $('.alert-dismissible, .text-danger').remove();
            $('.form-group').removeClass('has-error');

            if (json['redirect']) {
                location = json['redirect'];
            } else if (json['error']) {
                $('#button-register').button('reset');

                if (json['error']['warning']) {
                    $('#collapse-payment-address .panel-
body').prepend('<div class="alert alert-danger alert-dismissible"><i
class="fa fa-exclamation-circle"></i> ' + json['error']['warning'] +
'<button type="button" class="close" data-
dismiss="alert">&times;</button></div>');
                }

                for (i in json['error']) {
                    var element = $('#input-payment-' +
i.replace('_', '-'));

                    if ($(element).parent().hasClass('input-
group')) {
                        $(element).parent().after('<div
class="text-danger">' + json['error'][i] + '</div>');
                    } else {
                        $(element).after('<div class="text-
danger">' + json['error'][i] + '</div>');
                    }
                }

                // Highlight any found errors

```

```

        $('#text-danger').parent().addClass('has-error');
    } else {
        {% if shipping_required %}
        var shipping_address = $('#payment-address
input[name=\'shipping_address\']:checked').prop('value');

        if (shipping_address) {
            $.ajax({
                url: 'index.php?route=checkout/shipping_method',
                dataType: 'html',
                success: function(html) {
                    // Add the shipping address
                    $.ajax({
                        url:
'index.php?route=checkout/shipping_address',
                        dataType: 'html',
                        success: function(html) {
                            $('#collapse-shipping-address .panel-
body').html(html);

                            $('#collapse-
shipping-address').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-address" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_address }} <i class="fa fa-caret-down"></i></a>');
                        },
                        error: function(xhr, ajaxOptions,
thrownError) {
                            alert(thrownError + "\r\n" +
xhr.statusText + "\r\n" + xhr.responseText);
                        }
                    });
                }
            });

            $('#collapse-shipping-method
.panel-body').html(html);

            $('#collapse-shipping-
method').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-method" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_method }} <i class="fa fa-caret-down"></i></a>');

            $('a[href=\'#collapse-shipping-
method\"]').trigger('click');

            $('#collapse-shipping-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_shipping_method }}');

            $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_payment_method }}');

            $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
        },
        error: function(xhr, ajaxOptions, thrownError) {
            alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
        }
    });
    } else {
        $.ajax({
            url: 'index.php?route=checkout/shipping_address',
            dataType: 'html',
            success: function(html) {
                $('#collapse-shipping-address .panel-
body').html(html);

```

```

                                $('#collapse-shipping-
address').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-address" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_address }} <i class="fa fa-caret-down"></i></a>');

                                $('#a[href=\'#collapse-shipping-
address\']').trigger('click');

                                $('#collapse-shipping-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_shipping_method }}');

                                $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_payment_method }}');

                                $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
                                },
                                error: function(xhr, ajaxOptions, errorThrown) {
                                    alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
                                }
                                });
                                }
                                {% else %}
                                $.ajax({
                                    url: 'index.php?route=checkout/payment_method',
                                    dataType: 'html',
                                    success: function(html) {
                                        $('#collapse-payment-method .panel-
body').html(html);

                                        $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-payment-method" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_payment_method }} <i class="fa fa-caret-down"></i></a>');

                                        $('#a[href=\'#collapse-payment-
method\']').trigger('click');

                                        $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
                                        },
                                        error: function(xhr, ajaxOptions, errorThrown) {
                                            alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
                                        }
                                        });
                                {% endif %}

                                $.ajax({
                                    url: 'index.php?route=checkout/payment_address',
                                    dataType: 'html',
                                    complete: function() {
                                        $('#button-register').button('reset');
                                    },
                                    success: function(html) {
                                        $('#collapse-payment-address .panel-
body').html(html);

                                        $('#collapse-payment-
address').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-payment-address" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_payment_address }} <i class="fa fa-caret-down"></i></a>');
                                        },

```

```

        error: function(xhr, ajaxOptions, thrownError) {
            alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
        }
    });
},
error: function(xhr, ajaxOptions, thrownError) {
    alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
}
});
});

// Payment Address
$(document).delegate('#button-payment-address', 'click', function() {
    $.ajax({
        url: 'index.php?route=checkout/payment_address/save',
        type: 'post',
        data: $('#collapse-payment-address input[type=\'text\'],
#collapse-payment-address input[type=\'date\'], #collapse-payment-address
input[type=\'datetime-local\'], #collapse-payment-address
input[type=\'time\'], #collapse-payment-address input[type=\'password\'],
#collapse-payment-address input[type=\'checkbox\']:checked, #collapse-
payment-address input[type=\'radio\']:checked, #collapse-payment-address
input[type=\'hidden\'], #collapse-payment-address textarea, #collapse-
payment-address select'),
        dataType: 'json',
        beforeSend: function() {
            $('#button-payment-address').button('loading');
        },
        complete: function() {
            $('#button-payment-address').button('reset');
        },
        success: function(json) {
            $('.alert-dismissable, .text-danger').remove();
            $('.form-group').removeClass('has-error');

            if (json['redirect']) {
                location = json['redirect'];
            } else if (json['error']) {
                if (json['error']['warning']) {
                    $('#collapse-payment-address .panel-
body').prepend('<div class="alert alert-warning alert-dismissable">' +
json['error']['warning'] + '<button type="button" class="close" data-
dismiss="alert">&times;</button></div>');
                }

                for (i in json['error']) {
                    var element = $('#input-payment-' +
i.replace('_', '-'));

                    if ($(element).parent().hasClass('input-
group')) {
                        $(element).parent().after('<div
class="text-danger">' + json['error'][i] + '</div>');
                    } else {
                        $(element).after('<div class="text-
danger">' + json['error'][i] + '</div>');
                    }
                }

                // Highlight any found errors
                $('.text-danger').parent().parent().addClass('has-
error');
            } else {
                {% if shipping_required %}
                $.ajax({
                    url: 'index.php?route=checkout/shipping_address',

```

```

        dataType: 'html',
        success: function(html) {
            $('#collapse-shipping-address .panel-
body').html(html);

                $('#collapse-shipping-
address').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-address" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_address }} <i class="fa fa-caret-down"></i></a>');

                $('a[href=\'#collapse-shipping-
address\']').trigger('click');

                $('#collapse-shipping-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_shipping_method }}');
                $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_payment_method }}');
                $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
            },
            error: function(xhr, ajaxOptions, errorThrown) {
                alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
            }
        }).done(function() {
            $.ajax({
                url:
'index.php?route=checkout/payment_address',
                dataType: 'html',
                success: function(html) {
                    $('#collapse-payment-address
.panel-body').html(html);
                },
                error: function(xhr, ajaxOptions,
thrownError) {
                    alert(thrownError + "\r\n" +
xhr.statusText + "\r\n" + xhr.responseText);
                }
            });
        });
        {% else %}
        $.ajax({
            url: 'index.php?route=checkout/payment_method',
            dataType: 'html',
            success: function(html) {
                $('#collapse-payment-method .panel-
body').html(html);

                    $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-payment-method" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_payment_method }} <i class="fa fa-caret-down"></i></a>');

                    $('a[href=\'#collapse-payment-
method\']').trigger('click');

                    $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
            },
            error: function(xhr, ajaxOptions, errorThrown) {
                alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
            }
        });
    
```

```

    }).done(function() {
        $.ajax({
            url:
'index.php?route=checkout/payment_address',
            dataType: 'html',
            success: function(html) {
                $('#collapse-payment-address
.panel-body').html(html);
            },
            error: function(xhr, ajaxOptions,
thrownError) {
                alert(thrownError + "\r\n" +
xhr.statusText + "\r\n" + xhr.responseText);
            }
        });
    });
    {% endif %}
}
},
error: function(xhr, ajaxOptions, thrownError) {
    alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
}
});
});

// Shipping Address
$(document).delegate('#button-shipping-address', 'click', function() {
    $.ajax({
        url: 'index.php?route=checkout/shipping_address/save',
        type: 'post',
        data: $('#collapse-shipping-address input[type=\'text\'],
#collapse-shipping-address input[type=\'date\'], #collapse-shipping-
address input[type=\'datetime-local\'], #collapse-shipping-address
input[type=\'time\'], #collapse-shipping-address input[type=\'password\'],
#collapse-shipping-address input[type=\'checkbox\']:checked, #collapse-
shipping-address input[type=\'radio\']:checked, #collapse-shipping-address
textarea, #collapse-shipping-address select'),
        dataType: 'json',
        beforeSend: function() {
            $('#button-shipping-address').button('loading');
        },
        success: function(json) {
            $('#alert-dismissible, .text-danger').remove();
            $('#form-group').removeClass('has-error');

            if (json['redirect']) {
                location = json['redirect'];
            } else if (json['error']) {
                $('#button-shipping-address').button('reset');

                if (json['error']['warning']) {
                    $('#collapse-shipping-address .panel-
body').prepend('<div class="alert alert-warning alert-dismissible">' +
json['error']['warning'] + '<button type="button" class="close" data-
dismiss="alert">&times;</button></div>');
                }

                for (i in json['error']) {
                    var element = $('#input-shipping-' +
i.replace('_', '-'));

                    if ($(element).parent().hasClass('input-
group')) {
                        $(element).parent().after('<div
class="text-danger">' + json['error'][i] + '</div>');
                    } else {
                        $(element).after('<div class="text-
danger">' + json['error'][i] + '</div>');
                    }
                }
            }
        }
    });
}

```

```

    }
}

// Highlight any found errors
$('.text-danger').parent().parent().addClass('has-
error');
} else {
$.ajax({
url: 'index.php?route=checkout/shipping_method',
dataType: 'html',
complete: function() {
$('#button-shipping-address').button('reset');
},
success: function(html) {
$('#collapse-shipping-method .panel-
body').html(html);

$('#collapse-shipping-
method').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-method" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_method }} <i class="fa fa-caret-down"></i></a>');

$('#a[href=\'#collapse-shipping-
method\']').trigger('click');

$('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_payment_method }}');

$('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');

$.ajax({
url:
'index.php?route=checkout/shipping_address',
dataType: 'html',
success: function(html) {
$('#collapse-shipping-address .panel-
body').html(html);
},
error: function(xhr, ajaxOptions, thrownError)
{
alert(thrownError + "\r\n" +
xhr.statusText + "\r\n" + xhr.responseText);
}
});
},
error: function(xhr, ajaxOptions, thrownError) {
alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
}
}).done(function() {
$.ajax({
url:
'index.php?route=checkout/payment_address',
dataType: 'html',
success: function(html) {
$('#collapse-payment-address
.panel-body').html(html);
},
error: function(xhr, ajaxOptions,
thrownError) {
alert(thrownError + "\r\n" +
xhr.statusText + "\r\n" + xhr.responseText);
}
});
});
}
}

```

```

    },
    error: function(xhr, ajaxOptions, thrownError) {
        alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
            xhr.responseText);
    }
});
});

// Guest
$(document).delegate('#button-guest', 'click', function() {
    $.ajax({
        url: 'index.php?route=checkout/guest/save',
        type: 'post',
        data: $('#collapse-payment-address input[type=\'text\'],
            #collapse-payment-address input[type=\'date\'], #collapse-payment-address
            input[type=\'datetime-local\'], #collapse-payment-address
            input[type=\'time\'], #collapse-payment-address
            input[type=\'checkbox\']:checked, #collapse-payment-address
            input[type=\'radio\']:checked, #collapse-payment-address
            input[type=\'hidden\'], #collapse-payment-address textarea, #collapse-
            payment-address select'),
        dataType: 'json',
        beforeSend: function() {
            $('#button-guest').button('loading');
        },
        success: function(json) {
            $('.alert-dismissible, .text-danger').remove();
            $('.form-group').removeClass('has-error');

            if (json['redirect']) {
                location = json['redirect'];
            } else if (json['error']) {
                $('#button-guest').button('reset');

                if (json['error']['warning']) {
                    $('#collapse-payment-address .panel-
                    body').prepend('<div class="alert alert-warning alert-dismissible">' +
                    json['error']['warning'] + '<button type="button" class="close" data-
                    dismiss="alert">&times;</button></div>');
                }

                for (i in json['error']) {
                    var element = $('#input-payment-' +
                    i.replace('_', '-'));

                    if ($(element).parent().hasClass('input-
                    group')) {
                        $(element).parent().after('<div
                    class="text-danger">' + json['error'][i] + '</div>');
                    } else {
                        $(element).after('<div class="text-
                    danger">' + json['error'][i] + '</div>');
                    }
                }

                // Highlight any found errors
                $('.text-danger').parent().addClass('has-error');
            } else {
                {% if shipping_required %}
                var shipping_address = $('#collapse-payment-address
                input[name=\'shipping_address\']:checked').prop('value');

                if (shipping_address) {
                    $.ajax({
                        url: 'index.php?route=checkout/shipping_method',
                        dataType: 'html',
                        complete: function() {
                            $('#button-guest').button('reset');
                        },

```

```

        success: function(html) {
            // Add the shipping address
            $.ajax({
                url:
'index.php?route=checkout/guest_shipping',
                dataType: 'html',
                success: function(html) {
                    $('#collapse-shipping-address .panel-
body').html(html);

                    $('#collapse-
shipping-address').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-address" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_address }} <i class="fa fa-caret-down"></i></a>');
                },
                error: function(xhr, ajaxOptions,
thrownError) {
                    alert(thrownError + "\r\n" +
xhr.statusText + "\r\n" + xhr.responseText);
                }
            });

            $('#collapse-shipping-method
.panel-body').html(html);

            $('#collapse-shipping-
method').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-method" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_method }} <i class="fa fa-caret-down"></i></a>');

            $('a[href=\'#collapse-shipping-
method\']').trigger('click');

            $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_payment_method }}');

            $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
        },
        error: function(xhr, ajaxOptions, thrownError) {
            alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
        }
    });
} else {
    $.ajax({
        url: 'index.php?route=checkout/guest_shipping',
        dataType: 'html',
        complete: function() {
            $('#button-guest').button('reset');
        },
        success: function(html) {
            $('#collapse-shipping-address .panel-
body').html(html);

            $('#collapse-shipping-
address').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-address" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_address }} <i class="fa fa-caret-down"></i></a>');

            $('a[href=\'#collapse-shipping-
address\']').trigger('click');

```

```

                $('#collapse-shipping-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_shipping_method }}');
                $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_payment_method }}');
                $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
            },
            error: function(xhr, ajaxOptions, thrownError) {
                alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
            }
        });
    }
    {% else %}
    $.ajax({
        url: 'index.php?route=checkout/payment_method',
        dataType: 'html',
        complete: function() {
            $('#button-guest').button('reset');
        },
        success: function(html) {
            $('#collapse-payment-method .panel-
body').html(html);

                $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-payment-method" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_payment_method }} <i class="fa fa-caret-down"></i></a>');

                $('a[href=\'#collapse-payment-
method\']').trigger('click');

                $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
            },
            error: function(xhr, ajaxOptions, thrownError) {
                alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
            }
        });
    }
    {% endif %}
}
},
error: function(xhr, ajaxOptions, thrownError) {
    alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
}
});
});
});

// Guest Shipping
$(document).delegate('#button-guest-shipping', 'click', function() {
    $.ajax({
        url: 'index.php?route=checkout/guest_shipping/save',
        type: 'post',
        data: $('#collapse-shipping-address input[type=\'text\'],
#collapse-shipping-address input[type=\'date\'], #collapse-shipping-
address input[type=\'datetime-local\'], #collapse-shipping-address
input[type=\'time\'], #collapse-shipping-address input[type=\'password\'],
#collapse-shipping-address input[type=\'checkbox\']:checked, #collapse-
shipping-address input[type=\'radio\']:checked, #collapse-shipping-address
textarea, #collapse-shipping-address select'),
        dataType: 'json',
        beforeSend: function() {

```

```

        $('#button-guest-shipping').button('loading');
    },
    success: function(json) {
        $('.alert-dismissible, .text-danger').remove();
        $('.form-group').removeClass('has-error');

        if (json['redirect']) {
            location = json['redirect'];
        } else if (json['error']) {
            $('#button-guest-shipping').button('reset');

            if (json['error']['warning']) {
                $('#collapse-shipping-address .panel-
body').prepend('<div class="alert alert-danger alert-dismissible">' +
json['error']['warning'] + '<button type="button" class="close" data-
dismiss="alert">&times;</button></div>');
            }

            for (i in json['error']) {
                var element = $('#input-shipping-' +
i.replace('_', '-'));

                if ($(element).parent().hasClass('input-
group')) {
                    $(element).parent().after('<div
class="text-danger">' + json['error'][i] + '</div>');
                } else {
                    $(element).after('<div class="text-
danger">' + json['error'][i] + '</div>');
                }
            }

            // Highlight any found errors
            $('.text-danger').parent().addClass('has-error');
        } else {
            $.ajax({
                url: 'index.php?route=checkout/shipping_method',
                dataType: 'html',
                complete: function() {
                    $('#button-guest-shipping').button('reset');
                },
                success: function(html) {
                    $('#collapse-shipping-method .panel-
body').html(html);

                    $('#collapse-shipping-
method').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-shipping-method" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_shipping_method }} <i class="fa fa-caret-down"></i>');

                    $('a[href=\'#collapse-shipping-
method\']').trigger('click');

                    $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('{{
text_checkout_payment_method }}');

                    $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
                },
                error: function(xhr, ajaxOptions, thrownError) {
                    alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
                }
            });
        }
    },
    error: function(xhr, ajaxOptions, thrownError) {

```

```

                alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
            }
        });
    });

$(document).delegate('#button-shipping-method', 'click', function() {
    $.ajax({
        url: 'index.php?route=checkout/shipping_method/save',
        type: 'post',
        data: $('#collapse-shipping-method input[type=\'radio\']:checked,
#collapse-shipping-method textarea'),
        dataType: 'json',
        beforeSend: function() {
            $('#button-shipping-method').button('loading');
        },
        success: function(json) {
            $('#alert-dismissible, .text-danger').remove();

            if (json['redirect']) {
                location = json['redirect'];
            } else if (json['error']) {
                $('#button-shipping-method').button('reset');

                if (json['error']['warning']) {
                    $('#collapse-shipping-method .panel-
body').prepend('<div class="alert alert-danger alert-dismissible">' +
json['error']['warning'] + '<button type="button" class="close" data-
dismiss="alert">&times;</button></div>');
                }
            } else {
                $.ajax({
                    url: 'index.php?route=checkout/payment_method',
                    dataType: 'html',
                    complete: function() {
                        $('#button-shipping-method').button('reset');
                    },
                    success: function(html) {
                        $('#collapse-payment-method .panel-
body').html(html);

                        $('#collapse-payment-
method').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-payment-method" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{
text_checkout_payment_method }} <i class="fa fa-caret-down"></i></a>');

                        $('a[href=\'#collapse-payment-
method\']').trigger('click');

                        $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('{{
text_checkout_confirm }}');
                    },
                    error: function(xhr, ajaxOptions, thrownError) {
                        alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
                    }
                });
            }
        },
        error: function(xhr, ajaxOptions, thrownError) {
            alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
        }
    });
});

$(document).delegate('#button-payment-method', 'click', function() {

```

```

$.ajax({
    url: 'index.php?route=checkout/payment_method/save',
    type: 'post',
    data: $('#collapse-payment-method input[type=\'radio\']:checked,
#collapse-payment-method input[type=\'checkbox\']:checked, #collapse-
payment-method textarea'),
    dataType: 'json',
    beforeSend: function() {
        $('#button-payment-method').button('loading');
    },
    success: function(json) {
        $('#alert-dismissible, .text-danger').remove();

        if (json['redirect']) {
            location = json['redirect'];
        } else if (json['error']) {
            $('#button-payment-method').button('reset');

            if (json['error']['warning']) {
                $('#collapse-payment-method .panel-
body').prepend('<div class="alert alert-danger alert-dismissible">' +
json['error']['warning'] + '<button type="button" class="close" data-
dismiss="alert">&times;</button></div>');
            }
        } else {
            $.ajax({
                url: 'index.php?route=checkout/confirm',
                dataType: 'html',
                complete: function() {
                    $('#button-payment-method').button('reset');
                },
                success: function(html) {
                    $('#collapse-checkout-confirm .panel-
body').html(html);

                    $('#collapse-checkout-
confirm').parent().find('.panel-heading .panel-title').html('<a
href="#collapse-checkout-confirm" data-toggle="collapse" data-
parent="#accordion" class="accordion-toggle">{{ text_checkout_confirm }}
<i class="fa fa-caret-down"></i></a>');

                    $('a[href=\'#collapse-checkout-
confirm\']').trigger('click');
                },
                error: function(xhr, ajaxOptions, thrownError) {
                    alert(thrownError + "\r\n" + xhr.statusText +
"\r\n" + xhr.responseText);
                }
            });
        }
    },
    error: function(xhr, ajaxOptions, thrownError) {
        alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
    }
});
});
//--></script>
{{ footer }}

```