

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

“ _____ ” _____ 2022 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“Дослідження та програмна реалізація хмарної системи
моніторингу бази даних спортивних результатів з аналізом та
консолідацією статистичної інформації”**

Виконав здобувач вищої освіти

II курсу, групи _____ КН-21М1,4

ОПП «Комп’ютерні науки»

спеціальності 122 «Комп’ютерні науки»

_____ Тертичний А. Г.

« _____ » _____ 2022 р.

Керівник проекту

кандидат фіз.-мат. наук, доцент

_____ Якименко Н. М.

« _____ » _____ 2022 р.

Рецензент _____ Босько В. В. _____

м. Кропивницький

Центральноукраїнський національний технічний університет

Факультет Механіко-технологічний

Кафедра Кібербезпеки та програмного забезпечення

Рівень вищої освіти бакалавр

Галузь знань . 12 “Інформаційні технології”

Спеціальність 122 “Комп’ютерні науки”

Освітньо-професійна програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

“ ” 2022 року

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

Тертичного Антона Геннадійовича

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації

2. Керівник роботи Якименко Наталія Миколаївна, канд. фіз.-мат. наук, доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “17” 08 2022 року № 18-13

3. Строк подання роботи до захисту 22.12.2022 р.

4. Мета та завдання випускної кваліфікаційної роботи Метою роботи є дослідження та програмна реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації

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.

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	25.10.2022	12.11.2022
Охорона праці	Оришак О.В., к.т.н., доцент	04.11.2022	20.11.2022

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	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	17.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання
«__»____20 р.

Підпис керівника

(прізвище та ініціали)

Завдання прийнято до виконання

«__»____20 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Тертичний А. Г. Дослідження та програмна реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації.

Метою розробки є створення хмарної бази даних спортивних результатів, автоматизація аналізу статистичної інформації.

Об'єктом дослідження є процес моніторингу хмарної бази даних з аналізом статистичної інформації.

Предметом дослідження є алгоритми та методи реалізації інтерактивного хмарного комплексу моніторингу та аналізу.

Методи дослідження базуються на методах непараметричної оцінки точності прогнозу і довірчої границі для нього. Метод ковзних середніх, метод експоненціального згладжування.

Результат роботи – програмна реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації.

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.

Програмне забезпечення може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10 і Linux 3.16 – 5.2-rc1.

Програмне забезпечення розроблено в середовищі OpenServer та XAMPP.

Ключові слова: комп'ютерні науки, хмарні технології.

ABSTRACT

Tertychnyi A. H. Research and software implementation of a cloud system for monitoring sports results database with the analysis and consolidation of statistical information. 122 Computer science. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this final qualification work for the second (master's) level of higher education, software was developed, which is intended for a cloud system for monitoring the database of sports results with the analysis and consolidation of statistical information.

The purpose of the development is to create a cloud database of sports results, to automate the analysis of statistical information.

The object of the research is the process of monitoring the cloud database with the analysis of statistical information.

The subject of research is the algorithms and methods of implementing an interactive cloud monitoring and analysis complex.

Research methods are based on methods of non-parametric assessment of forecast accuracy and confidence limits for it. Method of moving averages, method of exponential smoothing.

The result of the work is the software implementation of a cloud system for monitoring the database of sports results with the analysis and consolidation of statistical information.

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.

The software can be used on personal computers of IBM PC architecture with Windows XP/Vista/7/8/10 and Linux 3.16 - 5.2-rc1.

The software is developed in the OpenServer and XAMPP environment.

Keywords: computer science, cloud technologies.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	6
1.1 Призначення системи	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ.....	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання.....	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ.....	20
3.1 Опис функціонування системи.....	20
3.2 Розробка структурної схеми	25
3.3 Розробка функціональної схеми.....	28
3.4 Розробка діаграми процесів.....	30
4 РЕАЛІЗАЦІЯ РОБОТИ.РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ,ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ І ПРОГРАМНИХ РІШЕНЬ	33
4.1 Розробка блок–схем та опис алгоритмів функціонування системи	33
4.2 Захист розробленого програмного забезпечення	40
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	43
5.1 Впровадження системи в експлуатацію	43
5.2 Інструкція щодо експлуатації системи.....	43
6 НАУКОВА НОВИЗНА.....	47

						ВКРМ-122.22.0015.00.00.ПЗ		
Зм.	Арк.	№ докум.	Підпис	Дата	Дослідження та програмна реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації	Літ	Аркуш	Аркушів
Розробив		Тертичний А. Г.				М	1	84
Перевір		Якименко Н. М.						
Н. Контр.		Гермак В. С.				ЦНТУ зр. КН-21м		
Затвердив		Смірнов О. А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ	48
7.1 Техніко-економічне обґрунтування теми дипломного проекту	48
7.2 Розрахунок трудомісткості розробки програмної продукції	50
7.3 Визначення чисельності виконавців і планового фонду зарплати	52
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	56
7.5 Визначення собівартості розробки та ціни програмної продукції	61
7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції	64
7.7 Визначення експлуатаційних витрат	65
7.8 Визначення економічної ефективності програмної продукції.....	66
7.9 Висновки	68
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ.....	69
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	69
8.3 Аналіз умов праці на робочому місці	70
8.4 Розробка заходів з умов поліпшення умов праці	73
8.5 Вимоги до захисного занулення.....	74
8.6 Розрахункова частина. Розрахувати захисне занулення.....	75
8.7 Висновки до розділу	78
9 ОСНОВНІ ВИСНОВКИ	79
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80

Зм.	Арк.	№ докум.	Підпис	Дата

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

SQL	-	Мова Структурованих Запитів
LOM	-	Локальна обчислювальна мережа
CSP	-	Cloud Service Providers, провайдери хмарних сервісів
IaaS	-	Інфраструктура як послуга
IOT	-	Інтернет речей
ITSM	-	Інтеграція управління ІТ-послугами
PaaS	-	Платформа як послуга
SaaS	-	Програмне забезпечення як послуга
QoS	-	Якість обслуговування
VNF	-	Віртуальні мережеві функції
WAN	-	Глобальна мережа, всесвітня комп'ютерна мережа

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

3

ВСТУП

Одним із ключових завдань удосконалення інформаційних комунікацій є завдання дослідження (моніторингу) та аналізу деякої інформації. Кожна людина відправляє запит, сподіваючись швидко отримати деякі достовірні дані, які вона потім може використати у власних цілях і з певною метою. Звідси випливає необхідність розширювати об'єми інформації задля вдоволення потреб кожного користувача. Для реалізації поставленої задачі використовуються хмарні бази даних, що містять великі об'єми інформації. Використання таких баз дозволяє легко, а головне швидко поповнювати обсяги даних, які можна одразу використовувати в режимі віддаленого доступу.

Головною перевагою використання саме хмарних баз даних є висока швидкодія роботи з ними. Іншою задачею є розробка зрозумілого та зручного інтерфейсу користувача, який може легко та без зусиль отримати необхідну йому інформацію, не застосовуючи при цьому введення специфічних команд або термінів. Глобалізація Інтернету серед населення сприяє тому, що кінцевий програмний продукт матиме вигляд Інтернет-сайту зі зрозумілим оформленням, з яким буде легко працювати.

Для реалізації поставлених задач розроблено програмне забезпечення моніторингу хмарної БД спортивних результатів, які будуть відображатися на Internet-сторінці. Для зацікавлення користувача сайт має містити щось принципово нове, незвідане йому раніше, а саме - аналіз статистичної інформації, яку можна використовувати для широких потреб. Це значно полегшує роботу аналітикам та селекціонерам, яким раніше доводилося самотійно і власноруч аналізувати та обробляти дані задля подальшого прогнозування певних результатів або для висновку по проведеній роботі.

Мета дослідження. Метою магістерської роботи є створення хмарної бази даних спортивних результатів, автоматизація аналізу статистичної інформації.

Зм.	Арк.	№ докум.	Підпис	Дата

У відповідності з темою та метою роботи було сформульовано основні наукові задачі:

- вибрати та обґрунтувати технологію обробки даних;
- вибрати та обґрунтувати методику аналізу статистичної інформації;
- розробити програмне забезпечення моніторингу хмарної бази даних спортивних результатів з аналізом статистичної інформації.

Об'єктом дослідження є процес моніторингу хмарної бази даних з аналізом статистичної інформації.

Предметом дослідження є алгоритми та методи реалізації інтерактивного хмарного комплексу моніторингу та аналізу.

Методи дослідження. Поставлені задачі розв'язуються за допомогою моделі нормальних помилок, непараметричної оцінки точності прогнозу і довірчої границі для нього. Застосовуються також евристичні прийоми, не засновані на ймовірсно-статистичній теорії: метод ковзних середніх, метод експоненціального згладжування.

Наукова новизна. Наукове значення роботи полягає у вдосконаленні процесу аналізу статистичної інформації. Було отримано такі наукові результати:

– вперше в Україні була автоматизована методика аналізу статистичної інформації за допомогою засобів статистики, що виводить результати через інтерактивну мережу;

– вперше реалізовано можливість аналітичного прогнозування статистичної інформації без урахування людського фактору.

Практичне значення одержаних результатів дослідження полягає в тому, що програмне забезпечення моніторингу хмарної бази даних спортивних результатів з аналізом статистичної інформації може

використовуватися не лише у сфері спорту, але і (з деякою модернізацією) в економічній, метеорологічній, фінансовій галузях, у господарстві чи освіті тощо.

Зм.	Арк.	№ докум.	Підпис	Дата

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ СИСТЕМИ

1.1 Призначення системи

Розвиток сучасних інформаційних та комп'ютерних технологій і комп'ютеризація всіх сфер суспільно-економічної діяльності сприяє знаходженню нових рішень, здатних ще більше спростити і зробити більш ефективними будь-які процеси, що виконує певна людина чи група осіб.

Особливого значення набуває прогнозування певних результатів, спираючись на які, користувач має змогу більш чітко і виважено прийняти важливе для нього рішення, яке принесе для нього користь.

Програмне забезпечення моніторингу хмарної бази даних з аналізом статистичної інформації являтиме собою Інтернет-сайт зі зрозумілим для користувача інтерфейсом та зручною адміністраторською частиною. Сайт призначений для прогнозування спортивних результатів та динаміки розвитку спортсменів і може використовуватися:

- як засіб для значного полегшення роботи аналітикам та селекціонерам, яким раніше доводилося самотійно і власноруч аналізувати та обробляти дані задля подальшого прогнозування певних результатів або для висновку по проведеній роботі;
- функціонерами задля успішної реклами певної спортивної події з метою залучення більшої кількості глядачів;
- реалізаторами спортивних сувенірів, які за допомогою прогнозування наперед намагатимуться постачати товар, що стосується певного цікавого матчу або спортсмена, що скоро набуде популярності;
- букмекерами, які на основі даних будуть вносити корективи у коефіцієнти на тоталізаторах.

Зм.	Арк.	№ докум.	Підпис	Дата

1.2 Область застосування системи

Призначення кінцевого програмного продукту – спрощувати роботу працівникам спортивної сфери задля подальших успішних дій, пов'язаних з позитивними результатами під час матчів або для детального спостереження за певним гравцем.

Система повинна забезпечувати виконання таких завдань:

- а) реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації;
- б) надавати користувачам цілодобовий і простий доступ до достовірної і свіжої інформації. При цьому повинен забезпечуватися зручний пошук необхідних документів;
- в) забезпечувати двосторонній зв'язок з відвідувачами;
- г) привертати увагу клієнтів і партнерів до вашої компанії.

Головна особливість програмного забезпечення моніторингу хмарної бази даних спортивних результатів з аналізом статистичної інформації в тому, що дане ПЗ може використовуватися не лише у сфері спорту, але і (з деякою модернізацією) в економічній, метеорологічній, фінансовій галузях, у господарстві чи освіті тощо. З цього можна зробити висновок, що застосування кінцевого програмного продукту може принести користь багатьом користувачам і сприятиме оптимізації багатьох галузей та специфічних напрямів господарства, культури, спорту тощо.

Зм.	Арк.	№ докум.	Підпис	Дата

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Потреби сучасного українського суспільства, глобалізаційні виклики часу, зміни, які відбуваються в ході інтеграції України у Європу, дедалі сильніше впливають на таку важливу сферу, як прогнозування. За таких умов виникає потреба глибокого і всебічного аналізу даних, і особливо, аналізу за допомогою використання хмарних технологій.

Вирішення вищезазначеного класу проблем можливе лише на основі цілеспрямованого, виваженого і спланованого використання інформаційних технологій, одна з переваг яких полягає у можливості контролю усіх поширених галузей та діяльності людства.

Як показує аналіз, найважливішим аспектом будь-якого процесу прогнозування є використання статистичної обробки інформації. Традиційним методам статистики властиві: недостатні обсяги даних, відсутність об'єктивних вимірjuвальних показників, неякісна однобічність і суб'єктивний вплив на результат прогнозування.

Широкое використання людством хмарних технологій створило передумови до застосування систем моніторингу баз даних та аналізу статистичної інформації, що стосується будь-якої діяльності особи або суспільства. Актуальність таких систем очевидна не лише для визначення достовірності прогнозованих результатів, але й для проведення моніторингу будь-якого об'єму інформації через інтерактивну мережу.

Хмарна технологія — це технологія розподіленої обробки цифрових даних, яка використовує ці технології для надання обчислювальних ресурсів як онлайн-сервісів користувачам Інтернету. Програма запускається у вікні веб-

Зм.	Арк.	№ докum.	Підпис	Дата

браузера на локальному ПК та відображає результати. При цьому всі необхідні програми та їх дані знаходяться на віддаленому інтернет-сервері та тимчасово кешуються на клієнті: ПК тощо. Перевага технології полягає в тому, що користувач може отримати доступ до власних даних, не турбуючись про інфраструктуру, операційну систему та програмне забезпечення, які він використовує.

На сьогоднішній день у світі існує велика кількість систем моніторингу хмарних баз даних, що зумовлює необхідність певної їх класифікації. Можна виділити такі типи системи моніторингу хмарної бази даних з аналізом статистичної інформації:

- за можливістю поповнення бази даних та внесення змін до програмної оболонки: відкриті і закриті;
- за схемою взаємодії з користувачем: гнучкі та формалізовані;
- за середовищем застосування: локальні (паритетні, “клієнт-сервер”), глобальні;
- за предметно-галузевим напрямком: від економіки до спорту.

Складові частини типової системи моніторингу хмарної бази даних з аналізом статистичної інформації:

- підсистема моніторингу;
- підсистема створення, видалення і редагування інформації;
- підсистема статистичних даних;
- підсистема аналізу та обробки статистичних даних;
- підсистема виведення даних, які прогнозуються;
- підсистема керування користувачами;
- база даних певних результатів і пов'язаної з ними статистичної інформації;
- база даних користувачів.

Для того, щоб використання системи моніторингу хмарної бази даних з аналізом статистичної інформації було доцільним, необхідне виконання декількох умов:

Зм.	Арк.	№ докум.	Підпис	Дата

- процес моніторингу повинен бути не гіршим від існуючих аналогів;
- результат прогнозованих статистичних даних повинен бути достовірним та виправдовувати очікування користувачів;
- для того, щоб одержати додаткові переваги перед рішенням важливих питань, прогнозування повинне максимально абстрагуватися від людського фактору при аналізі статистичних даних на основі моніторингу хмарних баз даних.

На жаль, цей момент практично недосяжний при використанні неавтоматизованого моніторингу, оскільки існує ймовірність необ'єктивної обробки статистичних даних.

Для уникнення цього необхідно використовувати в базі даних тільки найпростіші типи даних, а це, у свою чергу, сильно спрощує прогнозування і знижує його якість.

Для виходу з такої ситуації необхідно чітко визначити основні характеристики і вимоги, які ставляться до систем моніторингу хмарної бази даних з аналізом статистичної інформації:

- отримані результати повинні відповідати результатам усного прогнозування експертів;
- результати прогнозування повинні бути об'єктивними.

У системах моніторингу хмарних БД з аналізом статистичної інформації має бути передбачено:

- точна і здатна до адаптації оцінка прогнозованих даних;
- можливість використання як для внутрішнього так і відкритого використання;
- аналіз статистичних даних з метою оцінки якості прогнозування;
- легкість створення і модифікації баз даних;
- збір, збереження і представлення у зручній формі різноманітної статистичної інформації щодо процесу моніторингу;
- відсутність твердої прив'язки до конкретного апаратного і програмного забезпечення;

Зм.	Арк.	№ докум.	Підпис	Дата

- висока безпека, захищеність і стабільність;
- зручний і ефективний імпорт і експорт тестів;
- підтримка безпечного й універсального механізму керування користувачами з розподілом прав доступу.

Серед переваг комп'ютерних систем моніторингу хмарних баз даних з аналізом статистичної інформації можна виділити:

- автоматизація обробки результатів;
- забезпечення об'єктивності прогнозованих даних;
- підвищення оперативності прогнозування;
- зменшення витрат на аналіз статистичної інформації провідними спеціалістами.

Нижче представлено інтернет-сайти на спортивну тематику, схожі за своїм призначенням з програмним забезпеченням, що розроблено.

Розглянемо Інтернет-ресурс football.ua, який вміщує в собі велику кількість оглядів чемпіонатів, новин, фото та статей на футбольну та навколофутбольну тематику.

База даних цього сайту достатньо розширена та охоплює як позиції, команд, так і їх склади, рейтинг, інформацію про клуб.

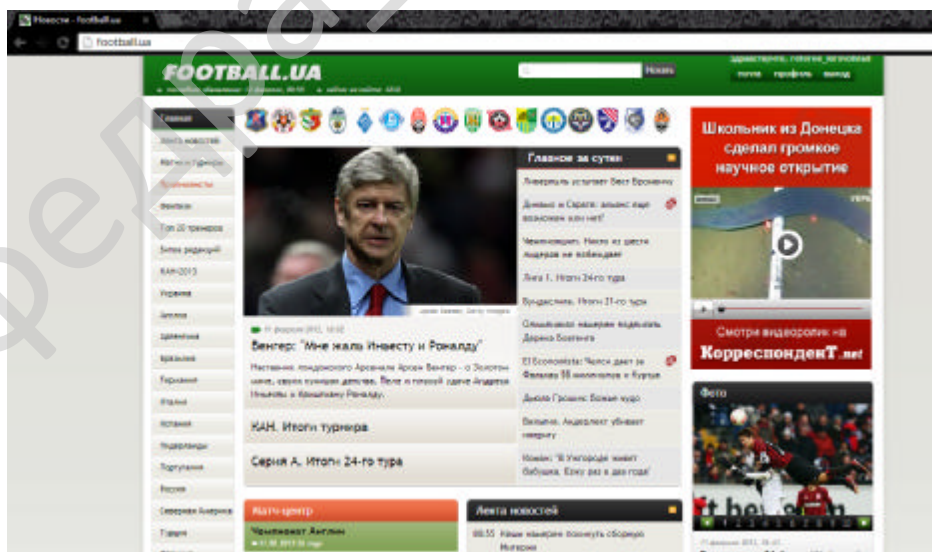


Рисунок 2.1 – Зовнішній вигляд інтернет-сайту football.ua

Зм.	Арк.	№ докум.	Підпис	Дата

Перевагою даного веб-сайту є різноманітність напрямків, орієнтованих на футбольну тематику, проте основним його недоліком є те, що усі результати (наприклад, результати Ліги Чемпіонів УЄФА 2012, що зображені на рис 2.2) вводяться власноруч адміністрацією сайту, а не генеруються автоматично після введення результату певного матчу.

The screenshot shows the website fcobol.ua/champions_league.html. The main content area displays the 'Турнірна таблиця' (Tournament Table) for the 'Ліга чемпіонів, 2012-2013' (Champions League, 2012-2013) as of 20 October 2012. The table is organized into five groups (A through E), each with four teams. The columns represent the team's position in the group, the team's name, and the number of points accumulated. The website also features a sidebar with various news articles and a navigation menu on the left.

Рисунок 2.2 – Турнірна таблиця Ліги Чемпіонів УЄФА 2012

На офіційному сайті організації УЄФА (uefa.com) користувач має можливість прослідкувати за рейтингами країн, які накопичують бали в разі успішних ігор клубів, підпорядкованих цій державі. На рисунку показаний список держав у файлі формату PDF, який відображає поточне положення країн.

National Team Coefficients Overview			Ranking to be used for the UEFA EURO 2012 Qualifying Draw	
WC 2006: 20% (GF)	EURO 2008: 40% (GF)	WC 2010: 40% (GF)		
1) Spain	39364	28)	Austria	24381
2) Germany	38294	29)	Bosnia-Herzegovina	24365
3) Netherlands	37821	30)	Slovenia	24221
4) Italy	35838	31)	Latvia	23303
5) England	34819	32)	Hungary	23048
6) Croatia	33677	33)	Lithuania	22071
7) Portugal	33226	34)	Belarus	21515
8) France	32551	35)	Belgium	21426
9) Russia	32477	36)	Wales	21274
10) Greece	31268	37)	F.Y.R. Macedonia	19409
11) Czech Republic	30871	38)	Cyprus	18791
12) Sweden	30895	39)	Montenegro	18791
13) Switzerland	30396	40)	Albania	18319
14) Serbia	29811	41)	Estonia	17792
15) Turkey	29447	42)	Georgia	16819
16) Denmark	29222	43)	Moldova	16734
17) Slovakia	28228	44)	Iceland	15404
18) Romania	28145	45)	Armenia	15164
19) Ukraine	28133	46)	Kazakhstan	14730
20) Israel	28052	47)	Liechtenstein	13581

Рисунок 2.3 – Рейтинг країн від офіційного веб-сайту УЄФА

Але алгоритм розрахунку і прогнозування рейтингів недосконалий, так як виводить лише кінцевий результат за наступними значеннями, наприклад, на сайті УЄФА це буде мати такий вигляд: «якщо три український клуби пройдуть в наступний раунд, рейтинг України збільшиться на 4 пункти». Мета ПЗ, що розробляється, – спроектувати прогноз на кшталт «рейтинг України підвищиться на 3,5-4 пункти, так як шанси трьох українських клубів на прохід у наступний раунд надзвичайно високий» або навпаки.

Перспективою даного дослідження є можливість розробки оптимального програмного забезпечення системи моніторингу хмарної бази даних з аналізом статистичної інформації, формування базових засад щодо його програмної реалізації і забезпечення ефективного впровадження і функціонування цієї системи.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Зараз у світі хмарних технологій найбільший вплив мають три основні компанії - AWS, Azure, Google Cloud. Ці компанії займають найбільшу частку

ринку в світі (за винятком Китаю та Alibaba Cloud), є технологічними лідерами та очолюють тенденції розвитку хмарних сервісів IaaS. Наприклад, AWS має понад 100 сервісів (IaaS, SaaS, PaaS).

Технології AWS засновані на серверних кластерах (фермах), що розташовані по всьому світу. Спектр сервісів AWS включає обчислення та зберігання даних, їх передачу по мережі, аналітику, мобільні застосунки, інструменти для розробників і т. д. Плата за користування базується на комбінації використання апаратних засобів/ОС/програмного забезпечення/мережевих функцій, вибраних користувачем, а також вимог до доступності, надлишковості, безпеки та додаткових параметрів.

Хмара Microsoft Azure зберігає дані у захищених центрах обробки даних у 26 регіонах світу, включаючи країни Євросоюзу. Центри обробки даних Azure відповідають вимогам конфіденційності даних ЄС та міжнародному стандарту ISO 27018. Щоб працювати з ресурсами та програмами Microsoft Azure, потрібен лише ПК з доступом до Інтернету. Час на розгортання сервісів у хмарі AZURE обчислюється годинами, тому при необхідності оперативного збільшення та зменшення потужностей або переїзду робота компанії не переривається. Переваги Microsoft Azure полягають в наданні доступу до додатків де завгодно і будь-коли, можливості роботи з мобільних пристроїв, достатній безпеці зберігання конфіденційних даних, достатньо великих обсягах зберігання даних, оплаті лише за фактично використані ресурси, економії в порівнянні з покупкою серверів.

Google Cloud Platform — набір хмарних служб від компанії Google, де крім інструментів для керування надається ряд модульних хмарних служб: обчислення, зберігання даних, аналіз даних та машинне навчання.

Види хмарних технологій

Хмарні технології поділяються на такі категорії:

а) загальнодоступна хмара – багато користувачів, які мають доступ до ІТ-інфраструктури одночасно. Але користувачі не мають можливості керувати та

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

1

обслуговувати цю хмару, вся відповідальність лежить на її власнику. Абонентом наданих послуг може стати будь-яка компанія або фізична особа.

б) приватна хмара — це обчислювальна інфраструктура, якою керує один користувач для власної вигоди. Інфраструктура керування приватною хмарою може бути розташована в приміщеннях користувача, у приміщеннях зовнішнього оператора або частково в приміщеннях користувача та оператора.

в) гібридна хмара – це IT-інфраструктура, яка поєднує в собі найкращі якості публічних і приватних хмар. Вона складається з окремих об'єктів, з'єднаних між собою стандартизованими або запатентованими технологіями, які дозволяють передавати дані або програми між компонентами.

Хмарні обчислення мають кілька рівнів:

Низькорівнева «Інфраструктура як послуга» (IaaS, Infrastructure as a Service). Користувачі отримують основні обчислювальні ресурси - процесори та пристрої зберігання даних - і використовують їх для створення власних операційних систем і програм. Споживачі не керують основною хмарною інфраструктурою, а контролюють операційну систему, системи зберігання та розгорнуті програми. Можливий обмежений контроль над вибором мережевих компонентів, наприклад хоста з мережевим екраном.

Наступний рівень — «Платформа як послуга» (PaaS, Platform as a Service). Користувачі можуть встановити власні програми на платформі, наданій постачальником послуг. Користувачі не керують основною хмарною інфраструктурою: мережами, серверами, операційними системами та системами зберігання даних, але можуть контролювати розгорнуті додатки та певні параметри конфігурації середовища розміщення.

Найвищим рівнем хмарних обчислень є «програмне забезпечення як послуга» (SaaS, програмне забезпечення як послуга). Вони зберігаються у вигляді даних у «хмарі», а також пов'язаних програм. Користувачам для роботи потрібен лише веб-браузер. Споживач використовує програму провайдера, яка працює в хмарній інфраструктурі. При цьому користувач не керує основною хмарною інфраструктурою – мережею, сервером, операційною системою,

Зм.	Арк.	№ докum.	Підпис	Дата

системою зберігання даних, індивідуальними налаштуваннями програми, за винятком окремих налаштувань конфігурації програми.

Для створення програмного забезпечення зручно буде використати мову PHP. Це є скриптова мова з відкритим вихідним кодом, створена для написання веб-додатків (скриптів і сценаріїв), які виконуються на веб-серверах. Деякі популярні системи управління контентом, такі як WordPress, Drupal та Joomla також створені на базі PHP.

PHP має відкритий вихідний код і поширюється по GPL ліцензії. Він дозволяє досить швидко створювати прототипи і готові сайти. Застосування PHP фреймворків дозволяє створювати веб-додатки в найкоротші терміни. PHP також надає можливість додатки масштабувати і легко обслуговувати, дозволяє їм розвиватися в часі.

PHP є кросплатформовим, сумісним з більшістю операційних систем, таких як Windows, Linux, Solaris, Mac OSX.

PHP має велику кількість розширень і бібліотек, представлених в інтернеті у вільному доступі. PHP містить ряд готових бібліотек для роботи із популярними базами даних.

Синтаксис і конструкції PHP включають багато елементів, які є у мовах програмування C, Perl, Pascal. PHP є мовою з універсальним синтаксисом і водночас пристосована до веб-програмування.

PHP можна налаштувати так, щоб вона забезпечувала максимальну свободу дій і безпеку. PHP може працювати в безпечному режимі (safe mode), який обмежує можливості застосування PHP користувачами. Наприклад: максимальний час виконання та використання пам'яті.

PHP включає надійні механізми шифрування. PHP також сумісний з багатьма додатками інших розробників, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції. Вихідний код PHP не можна переглянути у браузері, оскільки він виконується на сервері.

PHP використовується не лише у поєднанні з HTML, але й із JavaScript, WML, XML та іншими мовами програмування. PHP-код може передаватися

Зм.	Арк.	№ докum.	Підпис	Дата

будь-яким браузером і пристроям, в тому числі стільниковим телефонам, портативним комп'ютерам. Також PHP-код можна виконувати в режимі командного рядка.

PHP працює на різних web-серверах (Apache, Netscape Enterprise Server, Microsoft IIS, Stronghold, Zeus) і платформах (UNIX, Solaris, FreeBSD, Windows 95/98/NT/2000/XP/2003).

Сервер MariaDB є однією з найпопулярніших реляційних баз даних з відкритим кодом. Він створений розробниками MySQL і є відкритим кодом. Він часто присутній в хмарних пропозиціях і є стандартним у більшості дистрибутивів Linux.

З особливостей MariaDB, в порівнянні з MySQL, можна відзначити включення до складу наступних сховищ:

Aria — засноване на MyISAM високонадійне сховище, що відрізняється підвищеною стійкістю і збереженням цілісності даних після атаки і катастроф, при повній сумісності з MyISAM. Завдяки веденню логу операцій, у разі атаки проводиться відкат результатів виконання поточної операції. Також підтримується можливість відновлення стану з будь-якої точки в послідовності операцій (включаючи підтримку CREATE / DROP / RENAME / TRUNCATE).

У доповненні до стандартних механізмів зберігання даних, - MyISAM, Blackhole, CSV, Memory, і Archive, - СУБД MariaDB містить наступні способи зберігання даних:

- а) Aria
- б) FederatedX (прозора заміна Federated)
- в) OQGRAPH - з версії 5.2
- г) SphinxSE - з версії 5.2
- д) TokuDB
- е) Cassandra (MariaDB 10.0)
- ж) CONNECT (MariaDB 10.0)
- з) SEQUENCE (MariaDB 10.0)

Зм.	Арк.	№ докум.	Підпис	Дата

Покращена підсистема асинхронного введення / виводу для механізму зберігання даних InnoDB в операційних системах сімейства Windows.

Використання механізму зберігання даних Aria (раніше Maria) дозволяє виконувати комплексні запити без використання тимчасових таблиць, створюваних на пристроях зберігання (HDD / SSD / SAS / FC) за рахунок агресивних методів кешування в оперативній пам'яті.

За замовчуванням цей механізм зберігання включений до складу MariaDB починаючи з версії 5.1 і є продуктивним аналогом MyISAM.

Весь вихідний код MariaDB поширюється під ліцензіями GPL, LGPL або BSD. MariaDB не містить закритих модулів або компонентів, на кшталт тих, що містяться в Mysql Enterprise Edition. Однак, це не впливає на доступний функціонал MariaDB.

Всі технології, що існують в закритій версії MySQL 5.5 Enterprise Edition, в повному обсязі представлені і в MariaDB.

MariaDB включає тестові пакети для всіх виправлених помилок у вихідному коді.

MariaDB розробляється open source спільнотою.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну магістерську роботу, реалізації підлягає дослідження та програмна реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації на сайті.

В процесі розробки кваліфікаційної 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;

Зм.	Арк.	№ докум.	Підпис	Дата

б) вибрати та обґрунтувати методику побудови системи. Розробити функціональну та структурну схеми системи;

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

г) організувати інтерфейс користувача з метою формування та виводу повідомлень про некоректні дії користувача та нестандартні ситуації;

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

ж) сформуванати висновки про виконаний обсяг робіт та одержані результати.

Кафедра КБПЗ – 2022 рік

Зм.	Арк.	№ докум.	Підпис	Дата

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ ЩОДО ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Опис і обґрунтування проектних рішень щодо методів, технологій використаних у проектуванні системи

Вся користувацька частина системи реалізована без перезавантаження сторінки, що було досягнуто використанням технології AJAX.

AJAX (Asynchronous JavaScript And XML) - підхід до побудови користувацьких інтерфейсів веб-застосунків, де веб-сторінка не перезавантажується, а у фоновому режимі відправляє запити на сервер і довантажує необхідні користувачу дані.

Вперше термін AJAX був публічно використаний 18 лютого 2005 у статті Джесі Джеймса Гарретта (Jesse James Garrett) «Новий підхід до веб-застосувань». Гарретт придумав термін, коли йому довелося якось назвати новий набір технологій, пропонований ним клієнту.

Проте, в тій чи іншій формі, багато технологій були доступні і використовувалися набагато раніше, наприклад в підході «Remote Scripting», запропонованим компанією Microsoft в 1998, або з використанням HTML елемента IFRAME, що з'явився в Internet Explorer 3 в 1996.

AJAX став особливо популярним після використання його компанією Google в сервісах Gmail, Google Maps і Google Suggest. Схема роботи AJAX-технології показана на рисунку 3.1.

Зм.	Арк.	№ докум.	Підпис	Дата

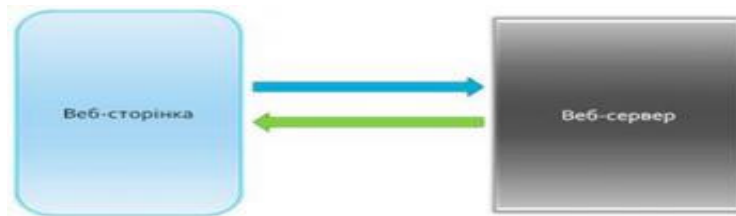


Рисунок 3.1 – Схема роботи AJAX-технології

Термін Ажах представляє широку групу веб-технологій, які можуть бути використані для реалізації веб-додатків, які взаємодіють із сервером у фоновому режимі, не заважаючи поточному стану на цій сторінці.

Ключовою перевагою є інтерактивність веб-інтерфейсу. Інші переваги, такі як зменшення трафіку, навантаження на сервер будуть залежати від безпосередньої реалізації проекту.

Недоліки технології Ажах:

- відсутність інтеграції із стандартними користувацькими інструментами браузера;
- проблема індексації пошуковими системами;
- ускладнення проекту.

Всі дані в AJAX передаються методами POST або GET, у відповідь на запит приходять дані у форматі JSON.

JSON - універсальні структури даних. Теоретично, всі сучасні мови програмування підтримують їх в тій чи іншій формі. Так як JSON використовується для обміну даними між різними мовами програмування, то має сенс будувати його на цих структурах.

Розглянемо модель оцінки рейтингу команд з використанням кваліметричного методу. Кваліметрія - наукова дисципліна, в рамках якої вивчаються методологія і проблематика комплексної, кількісної оцінки якості об'єктів будь-якої природи. В даному випадку проводиться рейтингова оцінка двох футбольних клубів, що беруть участь у матчі.

Зм.	Арк.	№ докум.	Підпис	Дата

Етапи виконання роботи:

1. Вибір показників, що формують рейтинг футбольної команди.
2. Аналіз показників:
 - виділення стимуляторів/дестимуляторів;
 - формування вагових коефіцієнтів.
3. Побудова математичної моделі:
 - приведення показників до порівнянного вигляду;
 - розрахунок інтегрального показника рейтингу команд.
4. Формування лінгвістичної інтервальної шкали.
5. Оцінка точності моделі (ітеративний етап):
 - підготовка статистичних даних;
 - розрахунок показників точності моделі.
6. Формування висновків про доцільність застосування моделі на практиці.

Отже, в першу чергу перед нами постає завдання вибору показників, що формують рейтинг спортивної команди. Доцільно вибрати для оцінки такі параметри, збір статистичної інформації про які не був би утрудненим. В той же час дані показники повинні володіти високим ступенем важливості для рейтингу команди. Врахувавши вищенаведені фактори, обрано наступні параметри:

- місце команди в турнірній таблиці;
- скільки останніх ігор команда не пропускала жодного м'яча (підряд);
- скільки останніх ігор команда забивала м'ячі (підряд);
- скільки очок набрано в останніх п'яти іграх;
- скільки м'ячів команда забила в останніх п'яти іграх;
- скільки м'ячів команда пропустила в останніх п'яти іграх;
- важливість матчу для команди (0-9 балів);
- відпочинок (скільки днів команда не грала);
- кількість гравців основного складу, які пропускають матч;
- домашній / виїзний матч (1/0).

Єдине зауваження: параметр «відпочинок» не враховується, якщо команди не грали більше двох тижнів.

Зм.	Арк.	№ докум.	Підпис	Дата

Серед вищенаведених показників є як стимулятори, так і дестимулятори. Стимуляторами називаються показники, збільшення яких призводить до зростання рейтингу команди, а дестимулятори - це показники, зменшення яких призводить до зростання рейтингу футбольного клубу. Серед виділених показників стимуляторами є наступні:

- скільки останніх ігор команда не пропускала жодного м'яча (підряд);
- скільки останніх ігор команда забивала м'ячі (підряд);
- скільки очок набрано в останніх п'яти іграх;
- скільки м'ячів команда забила в останніх п'яти іграх
- важливість матчу для команди;
- відпочинок;
- домашній / виїзний матч.

Інші показники є дестимуляторами.

Кожен з перерахованих вище показників впливає на рейтинг команди в певній мірі, причому її вага різна. Для того, щоб відобразити це розходження в моделі, було введено вагові коефіцієнти (Таблиця 3.3).

Таблиця 3.3 – Вагові коефіцієнти

Показник	Ваговий коефіцієнт
Місце команди в турнірній таблиці	9
Скільки останніх ігор команда не пропускала жодного м'яча (підряд)	4
Скільки останніх ігор команда забивала м'ячі (підряд)	6
Скільки очків набрано в останніх п'яти іграх	8
Скільки м'ячів команда забила в останніх п'яти іграх	7
Скільки м'ячів команда пропустила в останніх п'яти іграх	7
Важливість матчу для команди (0-9 балів)	8
Відпочинок (скільки днів команда не грала)	6
Кількість гравців основного складу, що пропускають матч	6
Домашній / виїзний матч	7

Емпіричні дослідження показали, що дані коефіцієнти доцільно корегувати залежно від команд, які беруть участь у матчі. Наприклад, якщо у команди в запасі є висококваліфіковані футболісти, то для неї знижується вага показника «Кількість гравців основного складу, які пропускають матч». Крім того, є команди, які грають домашні зустрічі значно краще, ніж виїзні.

Визначившись із показниками і ваговими коефіцієнтами, можемо приступати до розрахунку відносного рейтингу команд. Для цього нам необхідно виконати ряд дій:

1. Розрахувати відносну вагу показника для кожної команди. Для цього треба використати формулу (1) для показників-стимуляторів і формулу (2) для показників-дестимуляторів

$$Brel_i = \frac{B_i}{\sum_{j=1}^2 B_j}, \quad (1)$$

де B – вихідне значення показника, i – команда, $Brel$ – відносне значення показника.

$$Brel_i = 1 - \frac{B_i}{\sum_{j=1}^2 B_j} \quad (2)$$

2. Розрахувати рейтинг команд за формулою (3).

$$Ra_i = Mrel_i^T \cdot W, \quad (3)$$

де W – матриця вагових коефіцієнтів, $Mrel$ – матриця, що містить відносні значення показників $Brel$ для кожної команди, i – команда, Ra – рейтинг команди.

3. Для зручності порівняння нормується рейтинг команд на одиницю за формулою (4).

$$Rrel_i = \frac{Ra_i}{\sum_{j=1}^2 Ra_j} \quad (4)$$

де Ra – абсолютний рейтинг команди, i – команда, $Rrel$ – рейтинг команди, нормований на одиницю.

Зм.	Арк.	№ докум.	Підпис	Дата

Для інтерпретації отриманого рейтингу команд необхідно ввести так звану лінгвістичну інтервальну шкалу. У кваліметрії шкала вимірювань є засобом адекватного зіставлення і визначення чисельних значень окремих властивостей і якостей відмінності об'єктів. У даній моделі використовується п'ятирівнева лінгвістична шкала, наведена в таблиці 3.4.

Таблиця 3.4 – Рівні шкали виведення результатів

Результат	Рейтинг команди
Перемога	$X > 75\%$
Перемога\нічия	$55\% < X \leq 75\%$
Нічия	$45\% \leq X < 55\%$
Поразка\нічия	$25\% \leq X < 45\%$
Поразка	$X < 25\%$

3.2 Розробка структурної схеми

На рисунку 3.2 зображена структурна схема розробленої системи. Основними компонентами розробленого, у ході виконання кваліфікаційної роботи, комплексу є:

- модуль моніторингу хмарної БД;
- модуль розрахунку результатів команд;
- модуль збору статистичної інформації;
- модуль аналізу статистичної інформації;
- обробка і обчислення результатів для прогнозування;
- модуль виведення результатів команди та прогнозу.

Модуль моніторингу хмарної БД

В даному модулі обробляється інформація, введена адміністратором, стосовно певного зіграного матчу. Зчитуються дані забитих голів першої та

другої команди і загальний результат виводиться перед користувачем та записується в хмарну БД.

Модуль розрахунку результатів команд

Завдання цього модулю-формування турнірної таблиці на основі зчитаних даних з БД та зміна пунктів зіграних ігор, турнірних балів, кількості забитих та пропущених м'ячів, які також записуються в окрему БД з послідуною зміною турнірної таблиці у вікні сайту.

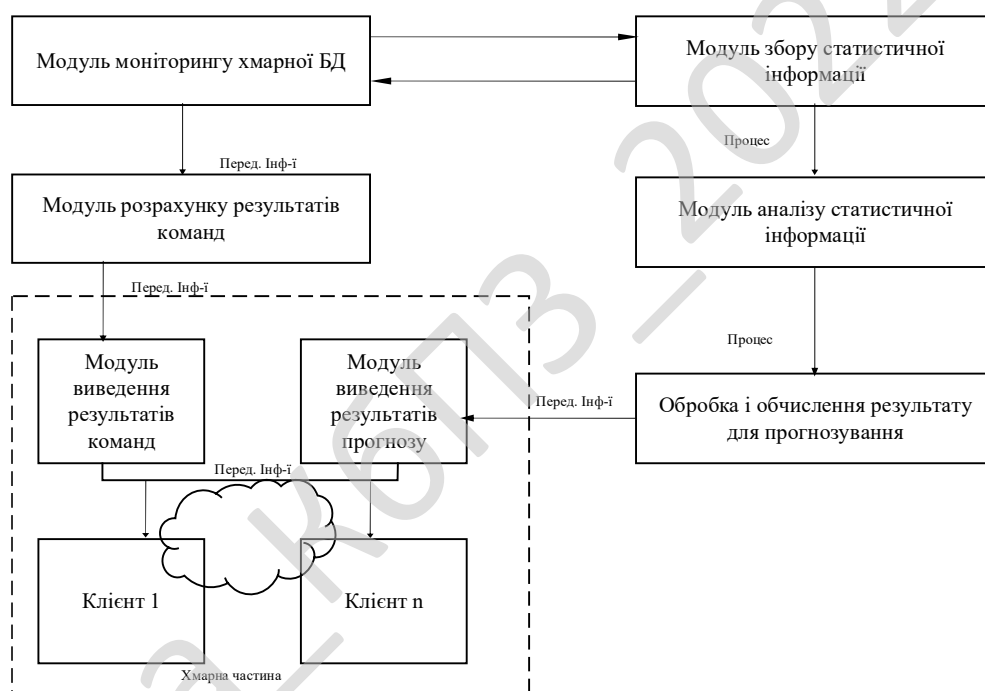


Рисунок 3.2 – Структурна схема системи

Модуль збору статистичної інформації

Даний модуль призначений для отримання інформації про турнірне положення команд, кількість зіграних ігор, кількість забитих та пропущених м'ячів. Щойно система отримала необхідні значення, вони одразу передаються до модулю аналізу статистичної інформації.

Зм.	Арк.	№ докум.	Підпис	Дата

Модуль аналізу статистичної інформації

У даному модулі на основі БД турнірної таблиці формуються коефіцієнти стимуляторів та дестимуляторів.

Наприклад, якщо «Динамо» (Київ) займає друге місце перед матчем з командою «Шахтар» (яка знаходиться на першому місці), то команді присвоюється коефіцієнт з наступним алгоритмом: якщо команда посідає місце з 1 по 8, застосовувати показники коефіцієнтів 6-9. Турнірне положення з 9 по 16 місце задіює коефіцієнти 2-5. Для першого та другого місця коефіцієнт=9, для третього та четвертого коефіцієнт=8 і т.д.

Після визначення стимулятора турнірного положення треба визначити стимулятор забитих м'ячів. Звернувшись до БД і отримавши кількість забитих м'ячів, проводиться аналіз для виведення коефіцієнту. Кількість забитих голів ділиться на кількість зіграних матчів. Отримане значення обробляється за наступними критеріями:

- не забито голів-коефіцієнт=0;
- середній показник забитих м'ячів у іграх=0,2 і менше-присвоїти коефіцієнт 1;
- при показникові 0,21-0,4-присвоїти коефіцієнт 2;
- при значенні 0,41-0,6 коефіцієнт=3;
- при 0,61-0,75-коефіцієнт=4;
- значенню 0,76-0,9 відповідає коефіцієнт 5;
- при показникові 0,91-2 коефіцієнт=6;
- при значенні 2,1 і більше коефіцієнт=7.

Далі визначається показник-дестимулятор, що характеризує пропущені м'ячі. Визначення коефіцієнту для цього показника аналогічне до визначення коефіцієнту забитих м'ячів.

Обробка і обчислення результатів для прогнозування

На основі коефіцієнтів далі формуються показники загального положення команд. Використовуючи коефіцієнт турнірного розміщення

Зм.	Арк.	№ докум.	Підпис	Дата

команди, використовується наступна формула: $V = \text{коефіцієнт} - [\text{номер команди в турнірній таблиці}]$. Отже, для «Динамо» $V = 9 - 2 = 7$, а для «Шахтаря» $V = 9 - 1 = 8$.

Після цього формується рейтинг команди, який змінить значення рейтингу у відповідній таблиці та внесе зміни до БД. Рейтинг формується за наступною формулою:

-при попередній перемозі $R = 0,4 + ((V + \text{коеф. забитих м'ячів}) / \text{коеф. пропущених м'ячів})$;

-при попередній нічій $R = 0,1 + ((V + \text{коеф. забитих м'ячів}) / \text{коеф. пропущених м'ячів})$;

-при попередній поразці $R = (V + \text{коеф. забитих м'ячів}) / \text{коеф. пропущених м'ячів}$.

Після запису рейтингу до БД програма виконує обчислення для прогнозу результату. Для цього використана наступна формула: $R_{rel} = R1 - R2$. Після цього значення R_{rel} порівнюються з тими, які представлені у таблиці 2 та формується певний прогноз на наступний матч зі значеннями «поразка \ поразка або нічия \ нічия \ перемога або нічия \ перемога». Але цього недостатньо для професійних аналітиків, так як важливу роль грає безпосередньо рахунок.

Даний пункт розраховується за формулою $G = (\text{забиті м'ячі} - \text{пропущені м'ячі})^R / 10$ з округленням до цілого числа. Достовірність даного результату приблизно рівна 92%.

Модуль виведення результатів команди та прогнозу

Даний модуль вже є безпосередньо хмарною частиною (клієнтським сайтом), який виводить для користувача інформацію у відповідності з запитом, яку користувач може застосовувати у своїх цілях, не порушуючи чинне законодавство та норми етики.

3.3 Розробка функціональної схеми

На рисунку 3.3 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Зм.	Арк.	№ докум.	Підпис	Дата



Рисунок 3.3 – Функціональна схема системи

Архітектура кінцевого програмного продукту- «клієнт-сервер». Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- хмара, яка забезпечує взаємодію між клієнтами та серверами.

Хмара на схемі зображена як канал передачі даних. До серверу входять наступні частини:

- введення даних;
- моніторинг даних;
- обчислення результатів команд;
- отримання статистичної інформації;
- аналіз і обробка статистичної інформації, формування прогнозу.

Обробка статистичної інформації здійснюється за рахунок застосування кваліметричного підходу обчислення статистичної інформації. Методи кваліметричної оцінки поділяються на евристичні та кількісні. В основу евристичних методів покладено принцип формалізації безпосереднього досвіду фахівців, а кількісні методи ґрунтуються на кількісних залежностях і характеристиках архітектурних об'єктів. До евристичних методів належать

інтуїтивний метод, оснований на принципі вибору та урахування такої сукупності чинників, яка для проєктувальників у даній ситуації здається вирішальною. Метод експертних оцінок базується на використанні досвіду групи фахівців і на законах теорії імовірності.

Кількісні методи поділяються на однокритеріальні та багатокритеріальні. Сутність однокритеріальних методів полягає у виборі найбільш важливої, значущої властивості об'єкта, яка формує його якість, тобто у виборі домінуючої мети. До цих методів відносяться: метод коефіцієнтів, метод мінімізації транзитної площі, метод мінімізації переміщень, метод числення вартості по кошторисах, метод числення вартості за укрупненими показниками. Сутність багатокритеріальних методів полягає у виборі і порівнянні по множині критеріїв з урахуванням їх важливості і доцільності. До цих методів відносяться метод «теоретичного еталону», метод оцінки по групі критеріїв, метод інтегральної оцінки; метод комплексної оцінки.

У розділі 3.1 наведено формули, які сприяють реалізації кваліметричного методу аналізу для поставленої задачі.

Частина клієнта складається з наступних частин:

- вибір результатів для перегляду;
- виведення результатів команд;
- виведення прогнозу.

Дані, що виводяться після запиту користувача, розміщуються у хмарній БД, яка використовується Інтернет-сайтом.

3.4 Розробка діаграми процесів системи

Діаграма взаємодії процесів системи, розробленої у результаті виконання магістерської роботи, наведена на рисунку 3.4. З нього видно, що процеси взаємодіють наступним чином.

Спершу завантажується процес виведення головного вікна програми. Він взаємодіє з наступними процесами:

Зм.	Арк.	№ докум.	Підпис	Дата

- запит клієнта на результати команд;
- запит клієнта на результати прогнозу;
- введення інформації адміністратором.

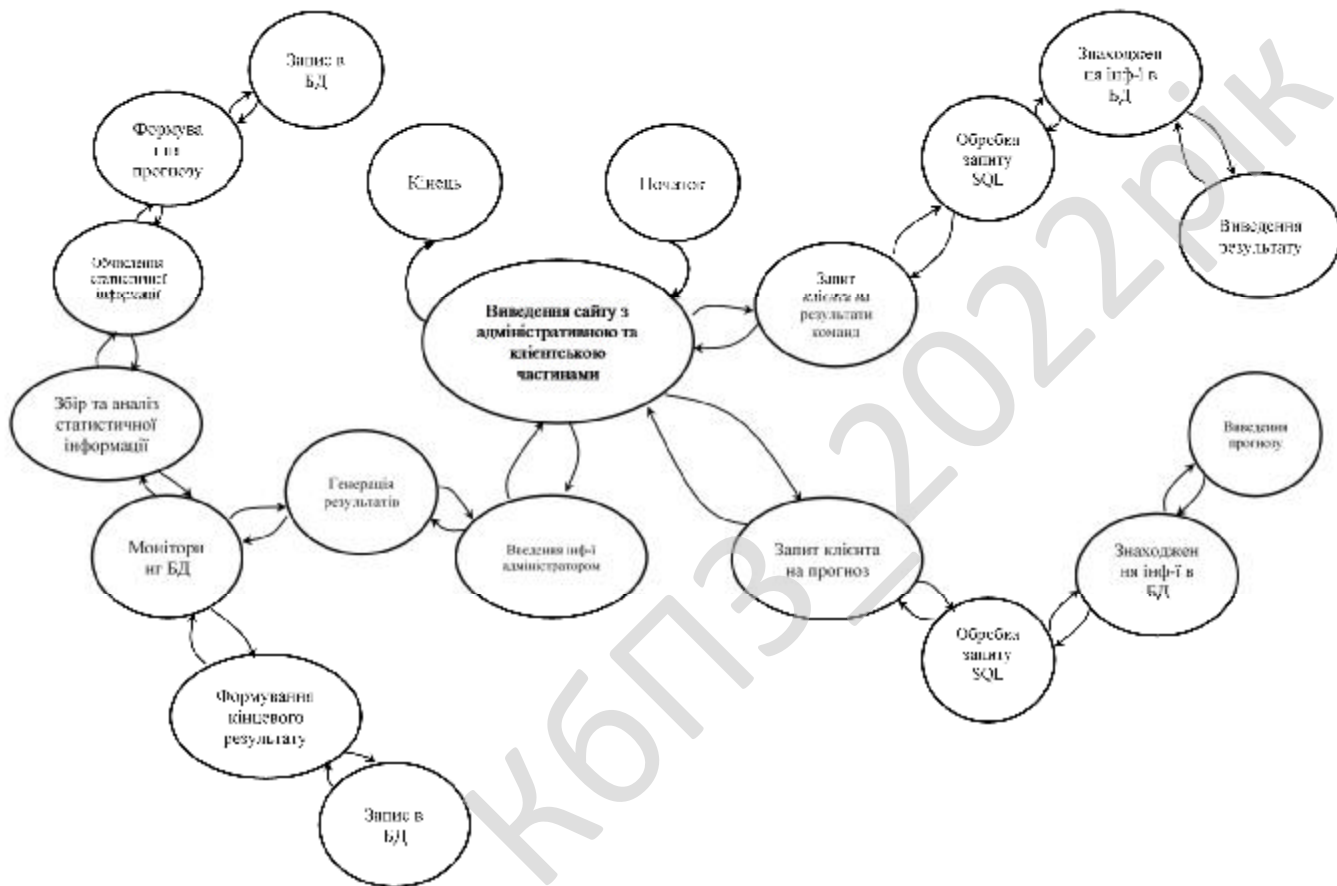


Рисунок 3.4 – Діаграма процесів системи

Процес запити клієнта на перегляд результатів команд відбувається наступним чином:

- запит обробляється;
- відбувається пошук даних в БД у відповідності із запитом;
- виводиться результат команд.

Процес запити клієнта на перегляд прогнозу аналогічний попередньому.

У випадку, якщо дані не було знайдено, користувачеві буде виведена помилка.

Процес введення даних адміністратором обробляє дані і генерує їх наступним чином:

Зм.	Арк.	№ докум.	Підпис	Дата

-відбувається моніторинг хмарної БД, під час якого вносяться зміни до таблиць та файлів БД у відповідності із введеним результатом адміністратора;

-відбувається збір та аналіз статистичної інформації після генерації результатів команд;

-відбувається процес обчислення статистичних даних;

-формується прогноз;

-вводяться зміни до БД.

Після успішного виконання усіх операцій адміністратор побачить вже згенеровану таблицю та прогнозовані дані, які потім можуть переглядати клієнти.

Кафедра КБПЗ – 2022 рік

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

3

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

4.1 Опис блок-схеми алгоритму роботи пристрою, системи

На рисунку 4.1 наведено блок-схему основної програми.

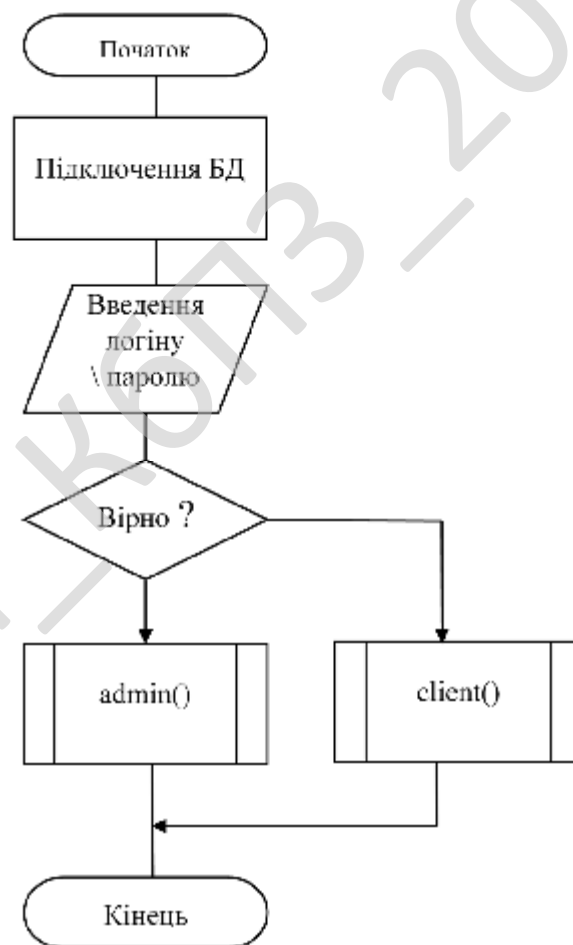


Рисунок 4.1 – Блок-схема головної програми

Після відкриття інтернет-сайту користувачеві пропонується ввести ім'я користувача (логін) та пароль для отримання прав адміністратора з метою

внесення певних даних. Якщо комбінація логіну та паролю невірна (дані зберігаються в БД аутентифікації), то користувач має змогу використовувати лише клієнтську частину програми, тобто лише переглядати необхідну йому інформацію. При отриманні прав адміністратора, можна виконувати введення даних, які програма самостійно обробляє шляхом моніторингу хмарної БД та розрахунку необхідних результатів.

Розрахунок для формування турнірної таблиці показаний на рисунку 4.2.

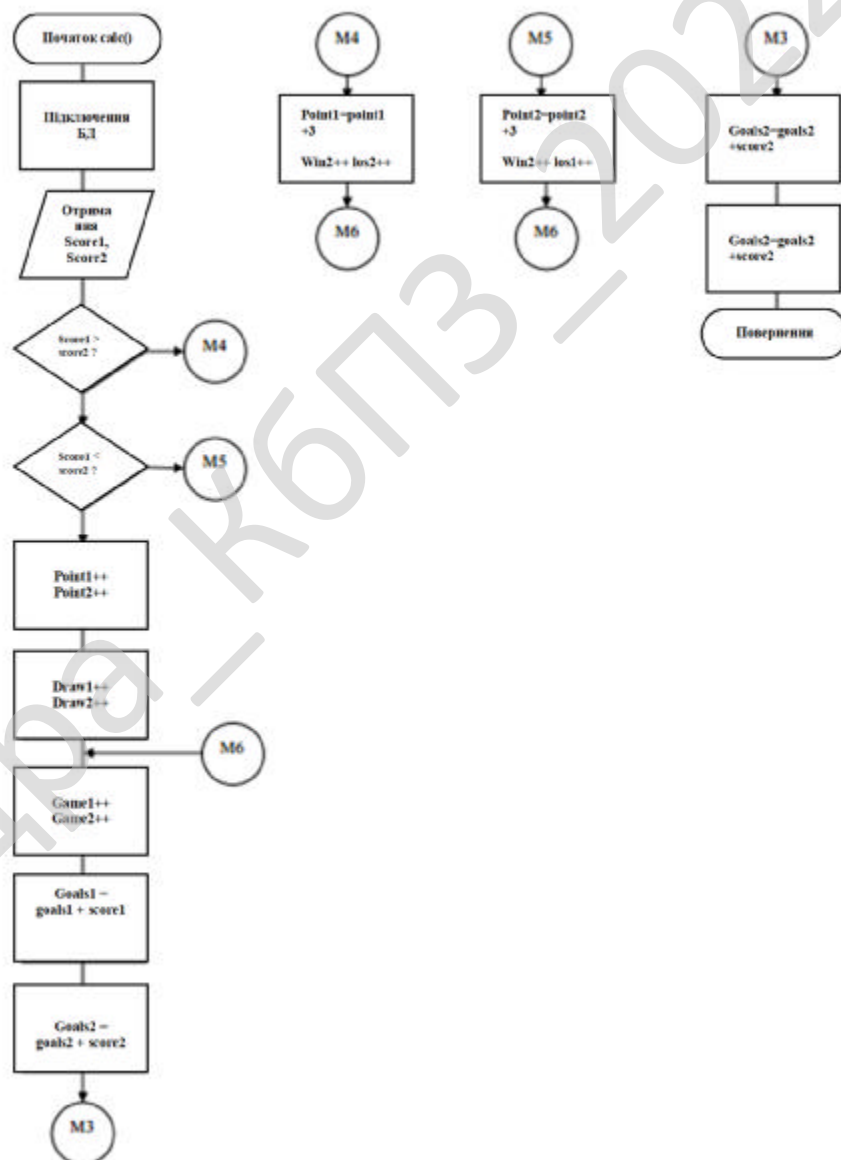


Рисунок 4.2 - Блок-схема розрахунку результатів для формування турнірної таблиці.

Отримавши рахунок матчу, точніше сказати, забиті голи однією та іншою командою (score1, score2), відбувається порівняння кількості м'ячів. Якщо перша команда забила більше другої, їй до кількості очок (point1) додаються ще три. В той же час показник поразки іншої команди інкрементується. Після цього показник обох команд в графі зіграних ігор (game1, game2) збільшується на одиницю та підраховуються забиті та пропущені м'ячі. Відповідно, до забитих м'ячів (goals1) додається власний показник за матч (score1), а до пропущених (goalm1) - показник забитих м'ячів суперника (score2).

Якщо друга команда забила більше, ніж перша, виконуються ті ж дії, але для протилежної команди.

У випадку нічиєї змінюється лише кількість турнірних балів (для кожної з команд по 1) та інкрементується графа нічийного матчу.

Після розрахунку даного матчу зміни у турнірній таблиці записуються до бази даних, і ці дані вже може переглядати клієнт.

Після генерації кількості турнірних балів програма отримує коефіцієнти та виконується процес обчислення статистичної інформації для виведення рейтингу команди.

Процес формування прогнозу виглядає наступним чином та наведений в програмному кодї:

```
<?
$X=R1-R2;
$Y=R1/R2*100;
if $X==0 {
echo "Матч закінчиться вничью";
}
elseif $X > 0
{
if $Y > 75 {
echo "Перемога команд1";}
else
echo "Перемога команди 1 або нічия";
}
else {
```

Зм.	Арк.	№ докум.	Підпис	Дата

```

if $Y > 75{
echo "Перемога команди 2";
}
else {
echo "Перемога команди 2 або нічия";
}
}

include("table.inc.php");
mysql_connect(localhost,$username,$password);
@mysql_select_db($table) or die( "Unable to select database");
$query="SELECT goals FROM table";
$query1="SELECT goalm from table";
$goals=mysql_query($query);
$goalm=mysql_query($query1);

mysql_close();

$G1 = (goals1 - goalm1)^R1 / 10;
$G2 = (goals2 - goalm2)^R2 / 10;

$G1 = ceil($G1);
$G2 = ceil($G2);

echo "Рахунок матчу $G1:$G2";

?>

```

Після обчислення рейтингу команд та занесення цих значень до бази даних відбувається розрахунок значень для прогнозування результату наступного матчу за наступним алгоритмом:

- розраховується різниця рейтингів та їх відношення;
- відбувається порівняння різниць рейтингів. Якщо різниця додатня, то перевагу має перша команда, якщо від'ємна - то друга. За нульового показника виводиться повідомлення про майбутній нічийний результат;
- після визначення різниці йде порівняння відношення рейтингів для виявлення, впевненою буде перемога чи з імовірністю нічийного результату;
- після визначення майбутнього результату матчу відбувається

Зм.	Арк.	№ докум.	Підпис	Дата

підрахунок забитих м'ячів у наступній грі, враховуючи поточний показник забитих\пропущених м'ячів та рейтинг команд.

Частина клієнта зображена на рисунку 4.4.

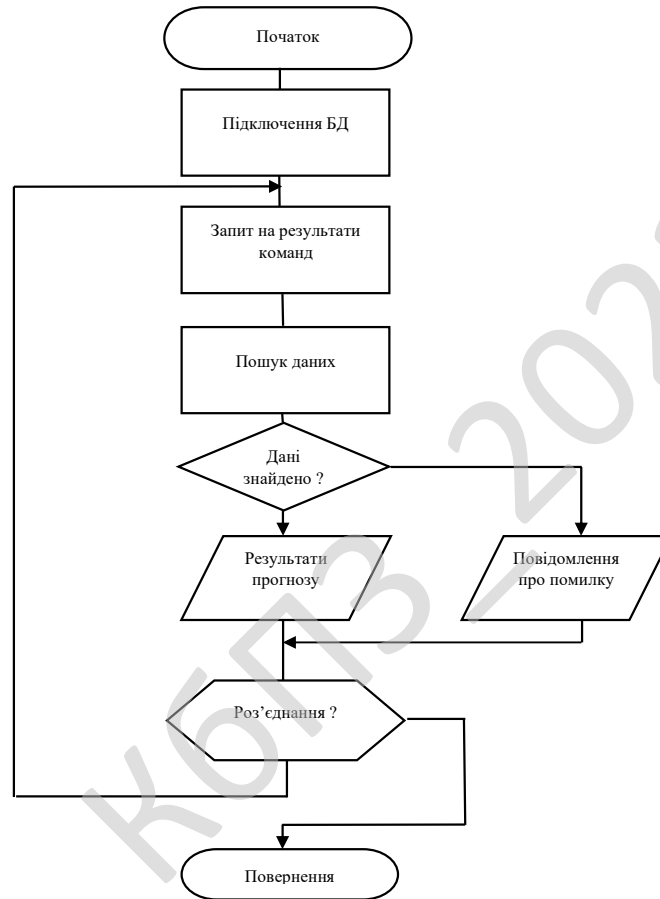


Рисунок 4.4 – Блок-схема частини клієнта

Після виведення вікна сайту та пропустивши процес аутентифікації, який необхідний для набуття адміністративних прав, користувач опиняється у клієнтському середовищі сайту, де він може лише переглядати результати, що його цікавлять, без внесення корективів у дані хмарної БД.

Схема відправки запиту та отримання інформації наступна:

- користувач відправляє запит на перегляд результатів команд або на інформацію стосовно прогнозування;
- обробник запитів SQL сервера виконує пошук відповідної інформації

шляхом моніторингу мережної БД;

- після знаходження необхідної інформації обробник запитів посилає дані для відображення клієнту, які виводяться перед ним на екрані;

- у випадку, якщо запитана інформація не була знайдена, перед користувачем виводиться помилка.

Процес введення даних адміністратором та перегляд інформації користувачем триває до тих пір, поки сервер або клієнт не отримає інформацію про розірвання з'єднання. Розриви з'єднання поділяються на такі типи:

- lost service;
- lost carrier;
- user request;
- admin reset;
- auto reset;
- session timeout.

Lost-Service – подібна помилка втрати зв'язку характерна для типів з'єднання rppoe та може виникати у випадках невірно отриманої ір-адреси чи фізичного пошкодження мережних апаратно-технічних засобів.

Щодо lost carrier - то це помилка, що найчастіше виникає внаслідок збою всередині мережного протоколу. В PPPoE / PPTP існує протокол перевірки "життєздатності" rppr з'єднання, котре через певні інтервали часу (lcp-echo-interval) посилає пакети, схожі за своєю метою з lcp ping, але реалізовані спеціально для тунельних з'єднань і звані LCP-пакетами. Якщо dbybrkb проблеми в обміні подібними пакетами на певному інтервалі часу, наприклад, за 60 секунд жоден пакет не повернувся, то система завершить сесію з Lost-Carrier.

В IPoE / IPN / HotSpot - схожий алгоритм, його реалізація залежить від авторів NAS, і там не використовується LCP, так як це спеціальний протокол для rppr-тунелів, але суть даного завершення залишається тією ж, і вказує на проблему в транспорті між абонентом і сервером доступу.

User request вказує на те, що сервер або робоча станція «добровільно» розірвала з'єднання. Якщо користувач не завершував Інтернет-сесію, то

Зм.	Арк.	№ докум.	Підпис	Дата

головною причиною такого типу розірвання з'єднання може бути шкідлива програма-«вірус».

Admin reset розрив може виникнути або у профілактичних цілях, або викликаний тим, що робоча станція надсилає шкідливу інформацію («спам»), або на ній виявлено «DHCP кільця». У цьому випадку, як і в попередньому, необхідно застосовувати антивірусну програму.

Auto reset найчастіше виникає внаслідок проблем на сервері. З'єднання повертається після усунення несправностей.

Session timeout виникає, якщо користувачеві хмари необхідно поновити пакет Інтернет-послуг (іншими словами - скінчилася оплата трафіку).

Динамічне оновлення даних перед користувачем можливе завдяки наступному фрагменту коду:

```
include("../blocks/connectbd.php"); //приєднуємося до Б.Д.
$table=$_POST['table']; //приєднання таблиці
$res = mysql_query ("SELECT match FROM table where
id='".$Динамо-Кривбас"'"); // «витаємо» певний матч
$kol = mysql_fetch_array($res);
$res1=$score1; //кількість забитих м'ячів першою командою
$i=0; //лічильник масиву
Do {
    $array_id[$key]=iconv("windows-1251","utf-8",$val);
}
$kol_questions=iconv("windows-1251","utf-8", $kol_questions);
$match=array('score1'=>$array_id, 'score2'=>$array1_id);
// повертаємо результат
echo json_encode($match);
```

Користувач без оновлення сторінки і обравши запит на результат матчу «Динамо»-«Кривбас», отримав результат матчу за секунди після запиту (рисунок 4.5).

Динамо К **1:1** Кривбас 3.03.2013 19:30

Рисунок 4.5 – Виведений результат

Зм.	Арк.	№ докум.	Підпис	Дата

Через те що формат JSON, в якому передаються необхідні результати роботи скрипту працює лише з кодуванням utf-8, у циклі виконується перекодування кожного елементу масиву, після чого формується результат вже перекодованих даних.

4.2 Захист розробленого програмного забезпечення

Захист інформації програмного забезпечення реалізовується провайдером, на базі якого буде розміщена наша система, оскільки система має хмарну реалізацію.

Тобто On-Cloud – програми, які вже розгорнуті на віддаленому сервері або розроблені з урахуванням такої можливості.

Основними перевагами такого рішення є те, що не потрібно мати власний сервер. Усіма технічними питанням буде займатися провайдер.

Також хмарна реалізація дає можливість швидкого автоматизованого розширення або скорочення інфраструктури, що дозволить динамічно реагувати в залежності від навантаженості та потреб системи.

Безперебійна робота інфраструктури і забезпечення аварійного відновлення це безумовна перевага такого рішення.

Вибір провайдера – це один із основних етапів перенесення проекту в хмару. Розглянемо фактори на які потрібно звернути увагу при виборі провайдера:

1) відмовостійкість – це безперебійний доступ до інфраструктури та збереження наших даних у разі несправностей. Надійність дата-центру визначається міжнародною класифікацією ANSI/TIA-942. Стандарт ранжує дата-центри за 4-ма класами – Tier I, II, III та IV. Чим вищий клас, тим більше систем продубльовано на випадок відмов обладнання. Рекомендуються Tier III та IV.

2) захист даних – це критично важливий фактор. Тому при виборі провайдера звертаємо увагу на внутрішні протоколи. Звертаємо увагу на наявність сертифікату ISO/IES 27001, який регулює політику безпеки, контроль

Зм.	Арк.	№ докум.	Підпис	Дата

за кваліфікацією співробітників, системи моніторингу, регулярність внутрішніх аудитів та інше.

3) додаткові послуги. Гарний хмарний провайдер полегшує роботу. Прикладом таких послуг є налаштування захисту від DDoS-атак, налаштування VPN тунелів для безпечних підключень та самостійне керування DNS записами.

Заходи безпеки

Ніколи не з'єднуватися з базою даних, використовуючи обліковий запис власника бази даних або суперкористувача. Завжди намагайтеся використовувати спеціально створених користувачів з максимально обмеженими правами.

Використовувати підготовлені вирази з прив'язаними змінними. Ця можливість надається розширеннями PDO , MySQLi і іншими бібліотеками.

Завжди перевіряйте введені дані на відповідність очікуваному типу. У PHP є безліч функцій для перевірки даних: починаючи від найпростіших функцій для роботи зі змінними і функцій визначення типу символів (таких як `is_numeric ()` і `ctype_digit ()` відповідно) і закінчуючи Perl-сумісними регулярними виразами .

У разі, якщо додаток очікує введення, застосуєте функцію `ctype_digit ()` для перевірки введених даних, або примусово встановлені тип за допомогою `settype ()`, або просто використовуйте числове уявлення за допомогою функції `sprintf ()`.

Безпечніша реалізація посторінкової навігації

```
<?php
    settype($offset, 'integer');
    $query = "SELECT id, name FROM products ORDER BY name LIMIT 20 OFFSET
    $offset;";

    $query = sprintf("SELECT id, name FROM products ORDER BY name LIMIT 20
    OFFSET %d;",
                    $offset);
```

Зм.	Арк.	№ докум.	Підпис	Дата

Якщо на рівні бази даних не підтримуються прив'язані змінні, то завжди екрануйте нечислові дані, що використовуються в запитах до БД за допомогою спеціальних екрануючих функцій, специфічних для використовуваної бази даних (наприклад, `mysql_real_escape_string ()` , `sqlite_escape_string ()` і т.д.) . Загальні функції такі як `addslashes ()` корисні тільки в певних випадках, тому краще уникати їх використання.

Кафедра _ КБПЗ _ 2022 рік

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

4

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ЕКСПЛУАТАЦІЮ

5.1 Впровадження системи в експлуатацію

Після розробки програмного продукту його потрібно впровадити в експлуатацію. При цьому виправляються виявлені помилки, система налаштовується на відповідний режим роботи.

Розроблена система моніторингу хмарної бази даних з аналізом статистичної інформації рекомендується для впровадження в спортивних сайтах та виданнях, які насамперед намагаються спрогнозувати ту чи іншу подію у сфері спорту.

Для впровадження системи були проведені деякі роботи, такі як:

- об'єднання Б.Д. сайта та Б.Д. системи в одну;
- реалізація єдиного адміністрування системи та сайту;
- розробка інструкції по використанню програмного забезпечення;
- розробка інструкції адміністратора.

5.2 Інструкція щодо експлуатації системи

При переході на даний Інтернет-сайт перед користувачем виводиться інформація про останній завершений або поточний тур чемпіонату України з результатами матчів та датою їх проведення (якщо адміністратор вже вніс корективи до БД).

Зм.	Арк.	№ докum.	Підпис	Дата

Чемпіонат України

Сезон 2012/2013

Тур: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 **19**

Матч		Дата проведення	
Шахтер	4:1	Вольнь	1.03.2013 19:00
Ильичевец	1:0	Металлург З	2.03.2013 13:30
Днепр	3:0	Арсенал К	2.03.2013 17:30
Металлист	3:2	Металлург Д	2.03.2013 19:30
Заря	2:1	Говерла	3.03.2013 13:30
Таврия	-:-	Карпати	3.03.2013 16:00
Чорноморець	-:-	Ворскла	3.03.2013 18:30
Динамо К	-:-	Кривбасс	3.03.2013 19:30

Рисунок 5.1 – Початкове вікно програми

Наступна вкладка сайту для користувача-турнірна таблиця, згенерована на основі зіграних матчів:

	Команда	Ігри	Победи	Ничьи	Поражения	Забито	Пропущено	Разница	Очки
1	Шахтер	19	18	0	1	56	10	46	54
2	Днепр	19	12	5	2	34	13	21	41
3	Металлист	19	12	3	4	38	17	21	39
4	Динамо К	18	12	1	5	34	15	19	37
5	Чорноморець	18	9	3	6	21	19	2	30
6	Металлург Д	19	7	5	7	26	25	1	26
7	Ильичевец	19	7	4	8	21	17	4	25
8	Заря	19	7	4	8	21	23	-2	25
9	Вольнь	19	6	5	8	19	27	-8	23
10	Арсенал К	19	6	5	8	18	29	-11	23
11	Таврия	18	6	3	9	12	21	-9	21
12	Кривбасс	18	6	3	9	18	29	-11	21
13	Ворскла	18	5	5	8	18	22	-4	20
14	Карпати	18	5	4	9	23	28	-5	19
15	Говерла	19	2	5	12	19	41	-22	11
16	Металлург З	19	0	3	16	5	47	-42	3

Рисунок 5.2 – Згенерована турнірна таблиця

Зм.	Арк.	№ докум.	Підпис	Дата	

БКРМ-122.22.0015.00.00.ПЗ

Арк.

4

Наступною вкладкою йдуть матчі наступного туру та можливість прогнозування цих матчів:

Чемпионат Украины 
Сезон 2012/2013

Матч	Дата проведення	
Черноморец - Ворскла	3.03.2013 18:30	Строгнозировать
Динамо К - Кривбасс	3.03.2013 19:30	Строгнозировать
Металлург Д - Ильячевец	8.03.2013 19:00	Строгнозировать
Металлург З - Таврия	9.03.2013 13:30	Строгнозировать
Кривбасс - Заря	9.03.2013 15:30	Строгнозировать
Карпаты - Черноморец	9.03.2013 17:30	Строгнозировать
Говерла - Днепр	10.03.2013 13:30	Строгнозировать
Ворскла - Шахтер	10.03.2013 16:00	Строгнозировать

Рисунок 5.3 – Наступні матчі туру

Натиснувши рядок «Прогнозувати», перед користувачем відкривається вікно з коефіцієнтами, де йому залишається лише внести інформацію про те, скільки днів команда не грала та скільки ключових гравців пропускають гру:

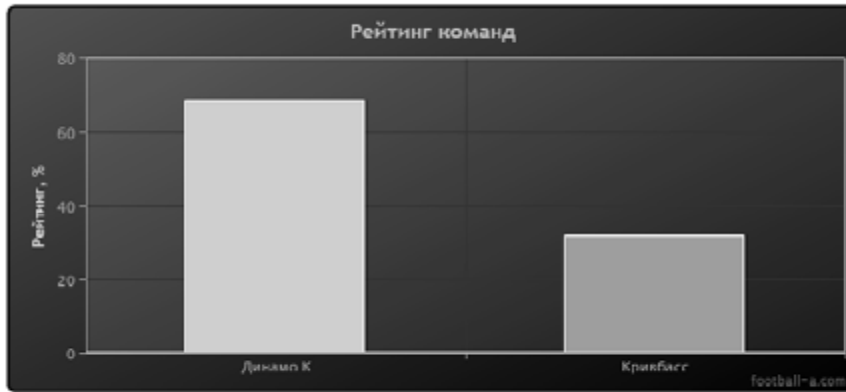
Матч: Динамо К - Кривбасс состоится 3.03.2013 в 19:30

Показатель	Динамо К	Кривбасс	Важность показателя
Отдых (сколько дней команда не играла):	<input type="text" value="17"/>	<input type="text" value="55"/>	<input type="range" value="55"/>
Количество игроков основного состава, пропускающих матч:	<input type="text" value="2"/>	<input type="text" value="4"/>	<input type="range" value="4"/>
Важность матча для команды (0-9):	<input type="text" value="Средняя"/>	<input type="text" value="Средняя"/>	<input type="range" value="5"/>
Место команды в турнирной таблице:	<input type="text" value="4"/>	<input type="text" value="12"/>	<input type="range" value="12"/>
Очков набрано в последних пяти играх:	<input type="text" value="12"/>	<input type="text" value="6"/>	<input type="range" value="6"/>
Сколько мячей команда забила в последних пяти играх:	<input type="text" value="15"/>	<input type="text" value="4"/>	<input type="range" value="4"/>
Сколько мячей команда пропустила в последних пяти играх:	<input type="text" value="3"/>	<input type="text" value="12"/>	<input type="range" value="12"/>
Домашний выездной матч:	<input type="text" value="Ѡ"/>	<input type="text" value="ѡ"/>	<input type="range" value="5"/>

Рисунок 5.4 – Вікно прогнозування

Натиснувши «порівняти», перед користувачем виведеться графік рейтингів команди (у %), текстовий прогноз та прогнозований результат матчу:

Результаты вычислений



Прогноз на матч: Победа команды "Динамо К" или ничья.
Счет: 3-1

Рисунок 5.5 – Результат прогнозування

Спираючись на дані рисунку 4.1.5, було продемонстровано, що програма надає більш-менш достовірний прогноз без втручання людського фактору, виводячи всі результати (окрім днів пропуску та відсутніх гравців, що не можна передбачити) автоматично.

Зм.	Арк.	№ докум.	Підпис	Дата

6 НАУКОВА НОВИЗНА

У ході виконання магістерської роботи було розроблено програмне забезпечення, яке призначено для хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації.

Метою роботи є програмне забезпечення хмарної системи для моніторингу баз даних спортивних результатів, аналізу та інтеграції статистичної інформації.

У відповідності з темою та метою роботи було сформульовано основні наукові задачі:

- вибрати та обґрунтувати технологію обробки даних;
- вибрати та обґрунтувати методику аналізу статистичної інформації;
- розробити програмне забезпечення моніторингу хмарної бази даних спортивних результатів з аналізом статистичної інформації.

Об'єктом дослідження є процес моніторингу хмарної бази даних з аналізом статистичної інформації.

Предметом дослідження є алгоритми та методи реалізації інтерактивного хмарного комплексу моніторингу та аналізу.

Поставлені задачі розв'язуються за допомогою моделі нормальних помилок, непараметричної оцінки точності прогнозу і довірчої границі для нього. Застосовуються евристичні прийоми, не засновані на ймовірностно-статистичній теорії: метод ковзних середніх, експоненціальне згладжування.

Наукове значення роботи полягає у вдосконаленні процесу аналізу статистичної інформації. Було отримано такі наукові результати:

- вперше в Україні була автоматизована методика аналізу статистичної інформації за допомогою кваліметрії, що виводить результати через інтерактивну мережу;
- вперше реалізовано можливість аналітичного прогнозування статистичної інформації без урахування людського фактору.

Зм.	Арк.	№ докум.	Підпис	Дата

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми дипломного проекту

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Проаналізувавши ринок програм аналогів було визначено ймовірну кількість програмних рішень. Також було підраховано необхідний час для розробки та впровадження програми і визначено орієнтовну вартість програмного забезпечення розробника. Час на розробку склав 60 днів (три місяці). В магістерській роботі були проведені дослідження та програмна реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації.

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

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

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	15
3. Запланований термін розробки, днів	Fpq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	3
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Зм.	Арк.	№ докум.	Підпис	Дата

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	15000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20
39. Об'єм програмного коду тис. рядків	Size	2,45

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 як одна з найбільш тривалих і трудомістких робіт значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

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.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026.$$

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

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

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

де: PV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

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

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)%.

$$T_{РП} = 0,3 \cdot 3,23 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 82 = 168 \text{ люд/день.}$$

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

Зм.	Арк.	№ докум.	Підпис	Дата

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:

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

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

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

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

Чисельність інженерів-електронників 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.

Зм.	Арк.	№ докум.	Підпис	Дата

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

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

Виходячи з цього фонд робочого часу інженерів-електронників складає:

$$\Phi_{op}^c = \frac{Z_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

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

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

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

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

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 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 (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2019, серверу доступу ADSL (ОС 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	

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

БКРМ-122.22.0015.00.00.ПЗ

Арк.

5

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	

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

5

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	7875	23625
Продакт-менеджер	0,25	6500	4875
Інженер-програміст	3,8	6500	74100
Інженер-електронщик	1,2	6500	23400
Адміністратор мережі	0,5	6500	9750
Дизайнер WEB	0,25	6500	4875
Інженер-верстальник	0,25	6500	4875
Бухгалтер-економіст	0,5	7000	10500
Всього за період розробки	$R_{cn} = 7,75$	-	$\Phi_{роб} = 156000$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{156000}{8,25 \cdot 60} = 315 \text{ грн.}$$

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:

$$B_{уд} = R_{cn}^1 S_{y} \Pi_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 13 робочих місць;

Зм.	Арк.	№ докум.	Підпис	Дата

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) ціна одного квадратного метра площі новобудови, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 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:

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

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

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

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 по прайсу Інтернет-магазину Компбест за 09.11.22 – джерело <https://compbest.com.ua>.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок	Hyundai iTMC Pentino Tower	7347

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Процесор	Intel Core i5-2400 (4 ядра по 3.1 - 3.4 GHz); Cache Memory 6 MB	-
Системна плата	Материнська плата MSI B75MA-P45 (s1155, Intel B75, PCI-Ex16, 2x USB 3.0, 6x USB 2.0, 5x Audio Ports, LAN (RJ-45), 2x PS/2; порти дискретної графіки - DVI, DisplayPort	-
Відеокарта	ATI Radeon HD 8490, 1 GB GDDR3, 64-bit; Integrated Intel HD Graphics 2000	-
Жорсткий диск	120 GB SSD, 500 GB HDD	-
Оперативна пам'ять	16 GB DDR3	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	ATX Pentino Tower, 3GTLA-489, PSU 350W(FSP Brand: ATX-350PNR, 12cm), black, (front bezel – black+light silver; body material – 0.6mm), 80mm fan (rear), 2xUSB2.0/AUDIO/MIC, Air Duct, Tool-less chassis design,Thermally Advantaged Chassis	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	-

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

БКРМ-122.22.0015.00.00.ПЗ

Арк.

5

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
інше	Клавіатура, мишка, система охолодження процесора	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	12	10947	13136,4	144500,4
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	163051,9

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			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	163051,9	-	-
Всього по групі	163051,9	50	81525,95
Група 5, 6			
4. Вимірювальні пристрої	9031	25	2257,75
5. Транспортні засоби	143000	20	28600
6. Господарський інвентар	28000	25	7000
Всього по групі 5,6	180031	-	37857,75
Нематеріальні активи			
7. Нематеріальні активи	20000	10	2000
Разом	$K_p = 1771082,9$		$A_p = 189783,7$

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description>, складає 143000 грн

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

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

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 315 \cdot 209 / 15 = 4389 \text{ грн.}$$

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, %.

$$Z_d = 4389 \cdot 10 \cdot 0,01 = 439 \text{ грн.}$$

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

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{оц} = 0,01 \cdot 22(4389 + 439) = 1043 \text{ грн.}$$

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

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

де: H_z – загальногосподарські витрати, %.

$$G_{осн} = 4389 \cdot 15 \cdot 0,01 = 659 \text{ грн.}$$

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.;

N_e – кількість екземплярів програм, шт.

Кількість паперу визначаємо за потребою. Приймаємо одну пачку паперу

Зм.	Арк.	№ докум.	Підпис	Дата

(250 арк.) на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 105$ грн., визначаємо вартість паперу:

$$Z_{M1} = C_n \cdot N_m \cdot n. \quad (7.16)$$

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

Згідно прийнятих норм по комплектації до вартості запам'ятовуваних пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо одну):

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 23 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 43 грн./шт.

$$Z_{M2} = 43 \text{ грн.}$$

Згідно методики розрахунку, одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – 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.

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

$$Z_M = (105 + 43 + 1702) / 15 = 124 \text{ грн.}$$

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

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

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

$$O_n = 4389 \cdot 15 \cdot 0,01 = 659 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 15$ прим.):

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

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

Зм.	Арк.	№ докум.	Підпис	Дата

$$A_m = 189783,7 \cdot 3 / (15 \cdot 12) = 3164 \text{ грн.}$$

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

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

$$C_n = 4389 + 349 + 1043 + 659 + 124 + 659 + 3164 = 10387 \text{ грн.}$$

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, %.

$$P_p = 0,01 \cdot 50 \cdot 10387 = 5194 \text{ грн.}$$

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	2	3
1. Основна зарплата виконавців	Z_o	4389
2. Додаткова зарплата виконавців	Z_d	349
3. Відрахування на соціальні потреби	C_{oc}	1043
4. Загальногосподарські витрати	Γ_{ocn}	659
5. Витрати на матеріали	Z_m	124
6. Освоєння нових операційних систем, мов програмування	O_n	659

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	3164
8. Повна собівартість програмного забезпечення	C_n	10387
9. Плановий прибуток	P_p	5194
10. Ціна підприємства $C_n = C_n + P_p$	C_n	15581
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	3116,2
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	18697,2

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	–	14281
Всього капітальних витрат	–	14281

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 технічне обслуговування	Z_p	26840	13420
2. Витрати на електроенергію	$Z_{ел}$	2268	2016
3. Витрати на амортизацію	$Z_{ам}$	0	3570
Всього витрат за рік	I	29108	19006

Витрати на обслуговування системи:

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

де: T_p – кількість годин обслуговування системи за рік, год.;

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 200 годин на рік до 100 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 200 \cdot 100 \cdot 1,1 \cdot 1,22 = 26840 \text{ грн.}$$

до:

$$Z_{p \text{ нов}} = 100 \cdot 100 \cdot 1,1 \cdot 1,22 = 13420 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

$$Z_{ел баз} = 0,45 \cdot 2400 \cdot 2,1 = 2268 \text{ грн.}$$

$$Z_{ел нов} = 0,40 \cdot 2400 \cdot 2,1 = 2016 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

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	–	14281	–	3570,25
Всього відрахувань	-	–	14281	–	3570,25

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:

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

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

$$E_e = (15581 - 10387) \cdot 15 - (0,05 \cdot 1408000 + 0,5 \cdot 163051,9 + 0,2 \cdot 143000 + 0,25 \cdot 37031 + 0,1 \cdot 20000) \cdot 3/12 = 29964 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p^* – балансова вартість основних фондів розробника без врахування

Зм.	Арк.	№ докум.	Підпис	Дата

вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_{\bar{o}} = \frac{363083}{(15581 - 10387) \cdot 15 \cdot 12 / 3} = 1,16 \text{ року.}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n(K_n - K_{\bar{o}}), \quad (7.27)$$

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

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

$$E_{cn} = (29108 - 19006) - 0,2 \cdot 14281 = 7246 \text{ грн.}$$

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

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{14281}{29108 - 19006} = 1,4 \text{ роки.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	15
2. Повна собівартість розробленої програми	Грн.	10387
3. Ціна розробленої програми	Грн.	15581
4. Плановий прибуток від реалізації розробленої програми	Грн.	5194
5. Рентабельність програмної продукції	%	50

Продовження таблиці 7.13

Найменування показників	Одиниця виміру	Величина
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1771083
7. Загальний прибуток від реалізації програмної продукції	Грн.	79340
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	31394
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,165
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	14281
11. Величина економічного ефекту у користувача програмної продукції	Грн.	7246
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	1,4

7.9 Висновки

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

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

6

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Електронна насиченість сучасного виробництва створює електричні небезпеки, які можуть виникати через електромережі, обладнання та інструменти під напругою, комп'ютери та організаційне обладнання, функціонування якого залежить від електрики. Це якраз і визначило актуальність проблеми електробезпеки – усунення ураження електричним струмом.

Електробезпека - це система організаційно-технічних заходів і засобів захисту людей від шкідливої і небезпечної дії електричного струму, дуги, електромагнітних полів і статичної електрики.

Електричні травми становлять невеликий відсоток порівняно з іншими видами виробничих травм, але вони очолили список за кількістю важких і особливо смертельних травм. Аналіз виробничих травм показує, що в середньому близько 18% усіх серйозних і смертельних нещасних випадків спричинені ураженням електричним струмом. Найбільша кількість уражень електричним струмом (60-70%) припадає на електрообладнання напругою до 1000 В. Це пояснюється широким розповсюдженням такого обладнання та відносно низьким рівнем підготовки операторів., Електроустановки що працюють понад 1000 В, обслуговується спеціально навченим персоналом, що знижує травматизм при їхній роботі.

Основна мета занулення - попередити можливе ураження електричним струмом.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Працівники повинні бути обізнані з плануванням робочого місця з персональним комп'ютером, нормами виробничого середовища, організацією та

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

6

обладнанням робочого місця, способами праці та відпочинку при використанні візуальних дисплейних терміналів (ВДТ), проходженням медичних оглядів. ДСанПіН 3.3.2.007 -98 «Гігієнічні вимоги до організації роботи на електронно-обчислювальних візуальних терміналах», 10 грудня 1998 р. (далі ДСанПіН 3.3.2.007-98)[51].

За пунктом 2.3. [51] площа на одне робоче місце має становити не менше ніж 6,0 м², а об'єм не менше ніж 20,0 м³.

За пунктом 4.4. [51] при розміщенні робочих столів з ВДТ слід дотримувати такі відстані між бічними поверхнями ВДТ 1,2 м, відстань від тильної поверхні одного ВДТ до екрана іншого ВДТ - 2,5 м.

Шкідливими факторами при використанні персонального комп'ютера є неіонізуючі випромінювання промислових частот, підвищене нервово-емоційне навантаження на працюючого, підвищене навантаження на органи зору і дрібні рухи очей.

Ці фактори можуть спричинити певні розлади здоров'я працівників, у тому числі підвищений артеріальний тиск, кон'юнктивіт, тендиніт та інші захворювання.

Як і будь-яке інше електричне обладнання, комп'ютери можуть бути джерелом ураження електричним струмом для операторів, особливо якщо вони підключені неправильно. Тому всі працівники, які працюють з персональним комп'ютером, повинні мати допуск 1 (або 2) групи з електробезпеки.

У зв'язку із зазначеними вище факторами працівники, які працюють з персональними комп'ютерами, зобов'язані проходити первинні та періодичні медичні огляди згідно з пунктом 6.2.3 додатку 4 до наказу МОЗ України. Порядок". 246 21 травня 2007 р [52].

8.3 Аналіз умов праці на робочому місці фахівця

Розглянемо умови праці в робочій кімнаті програміста. Геометричні розміри кімнати наведені в таблиці:

Зм.	Арк.	№ докум.	Підпис	Дата

БКРМ-122.22.0015.00.00.ПЗ

Арк.

7

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	5,38
Довжина	5,95
Висота	2,8

В цьому приміщенні працюють чотири працівника.

Таблиця 8.2 – Площа та об'єм приміщень на одного працівника

Геометрична характеристика	Одиниця виміру	Нормативне значення	Фактичне значення
Площа, S	м ²	не менше 6.0	8.002
Об'єм, V	м ³	не менше 20.0	22.407

За даними, наведеними в табл. 8.1 та таблиці 8.2 можна зробити висновок, що площа та об'єм приміщення, розрахований для одного робочого місця програміста, відповідає нормативним вимогам пункт 2.3. [51]. та наказу Міністерства соціальної політики України від 14.02.2018 № 207 «Про безпеку та охорону здоров'я працівників при дозволі на використання екранних пристроїв» розділ III [53].

З цього можна зробити висновок, що умови праці та чистота відповідають вимогам роботи програміста.

Температура повітря в приміщенні залежить від температури зовнішнього повітря і впливу теплової енергії на приміщення. Джерелами тепла в цьому приміщенні є електрообладнання, освітлення та люди. У денний час джерелом надлишкового тепла є сонячна радіація. Відповідно до постанови Головного державного лікаря України від 01.12.1999 р. № 42 [54] роботи, які проводяться в цьому приміщенні, віднесені до Іа категорії. При цьому людина споживає до 120 калорій на годину. На вологість повітря в приміщенні впливає багато факторів,

серед яких: вологість атмосферного повітря, волога, що виділяється людиною (пара при диханні та на поверхні шкіри).

Мікроклімат повітряного середовища приміщень характеризується запиленістю повітря. Мікроклімат приміщення визначається такими факторами, як вологість, температура, швидкість повітряного потоку та інтенсивність теплового випромінювання, що діє на організм людини. Аналіз мікроклімату передбачає визначення вищезазначених факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведені оптимальні і фактичні значення параметрів мікроклімату для експлуатаційної вагової категорії і пов'язаного з нею приміщення. У приміщеннях, де встановлені комп'ютери, рекомендується використовувати тільки оптимальні значення показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, ь,%	Швидкість повітря, м/с	Температу ра,°С	Вологіст ь%	Швидкість повітря, м/с
Холодна	21-24	41-65	0,2	25-27	41-55	0,4
Тепла	27-25	54-72	0,3	24-27	55-65	0,11

Проведений аналіз показав, що показники мікроклімату в приміщенні відповідають встановленим нормам. У холодну пору року використовують штучне опалення.

Для запобігання пилу регулярно провітрюйте та робіть вологе прибирання приміщення.

У приміщенні є наступні джерела шуму: принтер HP 1100, електродвигун вентилятор комп'ютера.

Одним із найважливіших факторів, що впливають на продуктивність праці та запобігають травматизму та професійним захворюванням програмістів, є освітленість робочого місця.

З 2019 року набули чинності Національні будівельні норми України «Природне та штучне освітлення» - ДБН В.2.5-28:2018 [55], які конкретизують вимоги до використання всього освітлювального обладнання, в тому числі світлодіодного.

Відповідно до таблиці 5.1 [55] роботу працівників, які часто працюють за комп'ютером, можна віднести до V ступеня зорової роботи, що відрізняє об'єкти (символи на екрані дисплея) від темного фону (під зоровою роботою). ступінь V) є відносно великим. Приміщення можна віднести до першої групи приміщень, в яких об'єкти зорової праці виділяються фіксованим напрямком лінії зору при роботі на робочу поверхню. Для таких місць і рівнів зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (комбіноване, загальне освітлення) не повинно перевищувати 1,5 %, а освітленість штучного освітлення — 300 лк.

Крім того, все поле зору має бути достатньо рівномірно освітлене – важлива гігієнічна вимога. Оскільки яскраве світло в області периферичного зору може істотно підвищити напругу очей, що може викликати швидку втоми очей, освітлення кімнати і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

На основі аналізу умов праці на розглянутих ділянках приходимо до наступних результатів:

- Розмір кімнати, з розрахунку на одного працівника, до стандарту;
- Мікроклімат відповідає нормативним значенням;
- Акустичні умови роботи не перевищують нормативних значень;

Зм.	Арк.	№ докум.	Підпис	Дата

Отже, можна припустити, що основною причиною можливого зниження працездатності програмістів є психофізіологічні фактори, тому основні рекомендації такі: підтримувати в колективі позитивну психологічну атмосферу та стандартний розклад, організовувати робоче місце з урахуванням ергономічні вимоги.

Рекомендовані дії: відповідно до плану евакуації (який повинен бути розміщений на видному місці в приміщенні), регулярні візуальні перевірки персоналом способу евакуації персоналу з приміщень включені до колективного договору з якомога меншою кількістю перших - повинні бути оснащені клапанами маски для обличчя або аптечкою іншого обладнання для штучного дихання. Регулярно перевіряйте параметри занулення.

Періодично візуально ознайомлювати працівників із способом евакуації людей з приміщень згідно з планом евакуації, передбачати розподільний щиток спеціальними розетками із заземлюючими контактами, організувати заземлення всіх електроприладів і обладнання з робочою напругою вище 36 В.

Оскільки у людей, вражених електричним струмом, може розвинутися фібриляція шлуночків серця, бажано мати в організації дефібрилятор і персонал, навчений ним користуватися.

8.5 Вимоги до захисного занулення

Захисне занулення

Захисне занулення, призначене для захисту людей від ураження електричним струмом під час дотику до непровідних металевих частин, які можуть потрапити під напругу через порушення ізоляції. Визначає ці терміни.

Занулення – це з'єднання металевих частин, що нормально не перебувають під напругою, з багаторазово заземленим нульовим проводом. Занулення виконують в системах із заземленою нейтраллю, воно забезпечує надійне відключення установки при замиканні на корпус.

Зм.	Арк.	№ докум.	Підпис	Дата

Металеві частини важкодоступних електроконструкцій, що не мають інших видів захисту з електробезпеки.

Металеві частини електроагрегату необхідно занулити шляхом електричного підключення нульового захисного провідника до точки заземлення джерела живлення.

8.6 Розрахункова частина. Розрахувати захисне занулення

Розрахунок захисного занулення.

Вихідні дані:

$P_{\text{НОМ}}$ – 60.968 кВт номінальна потужність навантаження;

$U_{\text{НОМ}}$ – 220 В номінальна напруга;

$\cos \phi$ – 0.8 коефіцієнт потужності навантаження;

Площа перерізу фазного провoda 95мм²;

$I_{\text{доп}}$ – довгочасний допустимий струм 215А;

Вибір захисного апарату. Для вибору захисного апарату визначають величину номінального струму плавкої вставки $I_{\text{ном.вст.}}$, виходячи з максимального робочого струму в ланцюзі $I_{\text{мах}}$ з урахуванням пускового струму електродвигуна $I_{\text{пуск}}$:

$$I_{\text{НОМ.ВСТ.}} \geq I_{\text{мах}} \quad (8.1)$$

Максимальний робочий струм у ланцюзі:

$$I \frac{P_{\text{НОМ}}}{\sqrt{3}U_{\text{НОМ}} \cos \phi_{\text{мах}}} = \frac{60.968 \cdot 10^3}{\sqrt{3} \cdot 220 \cdot 0.8} \approx 200 \text{ А} \quad (8.2)$$

де $P_{\text{НОМ}}$ - номінальна потужність навантаження, кВт;

$U_{\text{НОМ}}$ - номінальна напруга, В;

$\cos \phi$ - коефіцієнт потужності навантаження.

За розрахованим значенням струму $I_{\text{ном.вст.}}$ вибираємо плавку вставку – тип запобіжника ПН2-250.

Зм.	Арк.	№ докum.	Підпис	Дата

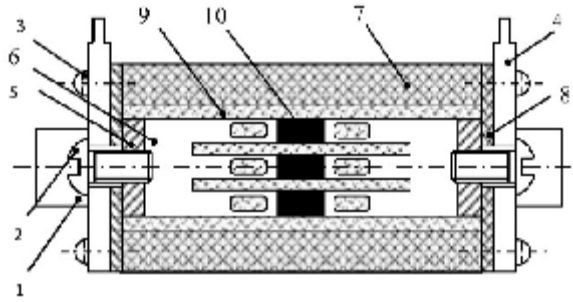


Рисунок 8.1 – Конструкція плавкого пластинчатого запобіжника серії ПН-2

1 – контакт; 2, 3 – гвинти; 4 – пластини кріплення; 5 – диски; 6 – плавка вставка;

7 – порцелянова трубка; 8 – азбестова прокладка; 9 – кварцовий пісок; 10 – свинцево-олов'яна напайка

Вибір перерізу фазного провідника. Переріз фазного провідника вибирається з умови максимально допустимого нагріву:

$$I_{\text{доп}} \geq I_{\text{max}} \quad (8.3)$$

де $I_{\text{доп}}$ – довгочасний допустимий з умови нагріву струм навантаження провідника, А.

Вибираємо переріз фазного провода. Так при виконанні умови (8.3) $200\text{A} < 215\text{A}$, для $I_{\text{доп}} = 215\text{A}$ для двох одножильних алюмінієвих проводів, прокладених в одній трубці, площу перерізу фазного провода рівною 95mm^2 .

Вибір перерізу нульового захисного провідника. При умові виготовлення нульового та фазного провідників з одного металу (алюмінію) запишемо:

$$S_{\text{нз}} \geq 0,5S_{\text{ф}}; S_{\text{ф}} = S_{\text{нз}}/2 = 95/2 = 47,5\text{mm}^2 \quad (8.4)$$

Розрахунок величини струму короткого замикання $I_{\text{кз}}$. Значення $I_{\text{кз}}$ залежить від фазної напруги мережі $U_{\text{ф}}$, і опорів мережі, у тому числі від повних опорів джерела струму Z_{T} і петлі (контуру) фазного провідника – нульового захисного провідника (петлі фаза – нуль, $Z_{\text{н}}$), і може бути розраховане за формулою для обчислення дійсного значення (модуля) струму короткого замикання:

Зм.	Арк.	№ докум.	Підпис	Дата

$$I_K = \frac{U_\phi}{\frac{Z_T}{3} + Z_n} \quad (8.5)$$

Повний опір петлі фаза – нуль, Z_n у дійсній формі (за модулем) дорівнює:

$$Z_n = \sqrt{(R_\phi + R_{нз})^2 + X_n^2}, \quad (8.6)$$

де R_ϕ , $R_{нз}$ – активні опори відповідно фазного та нульового захисних провідників;

$X_{вз}$ – зовнішній індуктивний опір петлі фаза-нуль (опір взаємоіндукції між фазним і нульовим проводами).

Для алюмінієвих провідників X_ϕ , $X_{нз}$ порівняно малі (біля 0,0156 Ом/км), тому ними можна знехтувати.

Зовнішній індуктивний опір $X_{вз}$ при малих значеннях відстані між проводами, порівняно з їх діаметром незначний (не більше 0,1 Ом/км) і ним можна також знехтувати. Тоді розрахункова формула (8.5) набирає вигляду:

$$I_K = \frac{U_\phi}{\frac{Z_T}{3} + R_\phi + R_{нз}} \quad (8.7)$$

Активний опір ланцюга фаза-нуль для провідника з кольорових металів визначається за формулою:

$$R_\phi + R_{нз} = \sum_{i=1}^n \frac{\rho_i l_i}{S_i} = 0,028 \cdot 255 \left(\frac{1}{95} + \frac{1}{47,5} \right) = 0,20 \text{ Ом} \quad (8.8)$$

де ρ_i - питомий опір i -ї ділянки провідника (для алюмінію – 0,028 Ом мм²/м);

l_i - довжина i -ї ділянки провідника, який має однаковий переріз, м;

S_i - площа перерізу провідника.

n – 2 кількість провідників.

Для потужності трансформатора 160кВА при захисному з'єднанні обмоток У/У_н модуль повного опору $Z_T=0,5$ Ом.

Перевірка надійності вимикання ушкодженого електроапарата. Перевіримо, чи справдиться нерівність:

$$I_K \geq k_{\text{ном.вст.}} \quad (8.9)$$

Зм.	Арк.	№ докум.	Підпис	Дата

де $k=1,25$ – коефіцієнт кратності струму стосовно номінально струму захисного апарата (при захисті автоматичними вимикачами з електромагнітними розчеплювачами і номінальним струмом, більшим за 100А).

$$k_{\text{ном.вст.}} 1,25 \cdot 200 = 250\text{А,}$$

Оскільки $250\text{А} < 550\text{А}$, то можемо зробити висновок, що нерівність справдилась, а, отже, нульовий захисний провідник вибраний правильно, тобто здатність системи, що вимикає, забезпечена.

8.7 Висновки до розділу

Виконання вимог нормативних документів, таких як ДНАОП 0.00-1.31-99 «Правила охорони праці при роботі з комп'ютером» [56], значно підвищує безпеку використання електроенергії, але не може на 100% гарантувати відсутність ураження електричним струмом, навіть якщо принципи застосування ВДТ та іншої оргтехніки формуються при прийнятті заходів безпеки праці.

Дотримання всіх необхідних умов праці не тільки сприяє збереженню здоров'я працівників, але й підвищує загальну продуктивність.

Зм.	Арк.	№ докум.	Підпис	Дата

9 ОСНОВНІ ВИСНОВКИ

Метою випускної кваліфікаційної роботи було розробити програмне забезпечення для хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації. В процесі виконання роботи був даний аналіз сучасного стану і можливостей хмарної системи моніторингу бази даних.

Після проведення всіх досліджень в якості мови програмування була обрана мова PHP.

У результаті проведеної роботи було розроблено:

- технічне завдання згідно з темою випускної кваліфікаційної роботи;
- структурна схема програмної системи;
- функціональна схема програмної системи;
- діаграма процесів розробленої системи.

Згідно до розроблених схем розроблено блок схеми основної програми та підпрограми, за якими розроблено програму моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації.

Програмна система оснащена відповідним користувацьким інтерфейсом, який забезпечує відповідний рівень взаємодії користувача, який має відповідну кваліфікацію та знання.

Робота системи є універсальною і її можна використовувати і для інших продуктів, які не були розглянуті в цій реалізації. Метод дає системний підхід для вирішення подібних задач, але для кожної потрібні будуть лише деякі логічні зміни, щоб система підходила до нової бази та методів пошуку її компонентів.

Зм.	Арк.	№ докум.	Підпис	Дата

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. База данных SQL Azure [Электронный ресурс]. – Режим доступа: <https://azure.microsoft.com/ru-ru/services/sql-database/>
2. Microsoft SQL Server Management Studio Express [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/ru-ru/download/details.aspx?id=8961>
3. Download MySQL Workbench [Электронный ресурс]. – Режим доступа: <https://dev.mysql.com/downloads/workbench/>
4. Download HeidiSQL 10.1 [Электронный ресурс]. – Режим доступа: <https://www.heidisql.com/download.php>
5. PL/SQL Developer [Электронный ресурс]. – Режим доступа: https://www.allroundautomations.com/download.html?gclid=CjwKCAjw27jnBRBuEiwAdjQXDKEev6BxtoUkKR00rcto5UImCkVoPBKxe1v-qaxFnSvItp4ntvLLCRoC85oQAvD_BwE
6. Database .NET [Электронный ресурс]. – Режим доступа: <https://marketplace.visualstudio.com/items?itemName=fishcode.lib.DatabaseNET>
7. Visual Studio Data Generator [Электронный ресурс]. – Режим доступа: <https://www.dutchdatadude.com/generating-perfect-test-data-using-visual-studio-and-custom-data-generators/>
8. ApexSQL Generate [Электронный ресурс]. – Режим доступа: <https://www.apexsql.com/sql-tools-generate.aspx>
9. Upscene Advanced Data Generator [Электронный ресурс]. – Режим доступа: <https://www.sqlmanager.net/ru/products/mysql/datagenerator>
10. EMS Data Generator for MySQL [Электронный ресурс]. – Режим доступа: <https://www.sqlmanager.net/ru/products/mysql/datagenerator>
<https://www.gsapps.com/products/datagenerator/>

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

8

11. Azure Monitor SQL insights [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/en-us/updates/azure-monitor-sql-insights-for-azure-sql-in-public-preview/>
12. Azure Monitor [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/ru-ru/services/monitor/#overview>
13. Choose between the vCore and DTU [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/azure/azure-sql/database/purchasing-models>
14. DTU and vCore based models for Azure SQL Databases [Электронный ресурс] – Режим доступа: <https://www.sqlshack.com/dtu-and-vcore-based-models-for-azure-sql-databases/>
15. Resource limits for single databases using the DTU-based purchasing model [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-dtu-resource-limits-single-databases>
16. Azure SQL Database DTU Calculator [Электронный ресурс]. – Режим доступа: <http://dtucalculator.azurewebsites.net/>
17. Intelligent Insights using AI to monitor [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/azure/azure-sql/database/intelligent-insights-overview>
18. Azure Activity Log [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/activity-log>
19. Metrics supported by Azure Event Grid [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/azure/event-grid/metrics>
20. Автоматичне масштабування Azure [Электронный ресурс] – Режим доступа: <https://azure.microsoft.com/ru-ru/features/autoscale/>
21. Э. Фримен, Э. Фримен. Изучаем HTML, XHTML и CSS = Head First HTML with CSS & XHTML. — П.: «Питер», 2010.
22. Дэвид Сойер Макфарланд. Новая большая книга CSS = CSS: The Missing Manual. — Санкт-Петербург: Питер, 2017.

Зм.	Арк.	№ докум.	Підпис	Дата

23. Дмитрий Котеров, Алексей Костарев. PHP. — СПб.: «БХВ-Петербург», 2005
24. Костарев А. Ф. PHP 5. — СПб.: «БХВ-Петербург», 2008.
25. Мэтт Зандстра. PHP: объекты, шаблоны и методики программирования = PHP Objects, Patterns and Practice, Third Edition. — 3-е издание. — М.: «Вильямс», 2010.
26. Когаловский М.Р. Энциклопедия технологий баз данных. — М.: Финансы и статистика, 2002.
27. Кузнецов С. Д. Основы баз данных. — 2-е изд. — М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2007.
28. Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. SQL: полное руководство, 3-е издание = SQL: The Complete Reference, Third Edition. — М.: «Вильямс», 2014.
29. Bartholomew, Daniel. Getting Started with MariaDB. — 2013. — ISBN 9781782168096.
30. Якоб Нильсен, Хоа Лоранжер. Web-дизайн: удобство использования Web-сайтов = Prioritizing Web Usability. — М.: «Вильямс», 2007.
31. A.B. Bagula Online Traffic Engineering: The Least Interference Optimization Algorithm / A.B. Bagula, M. Botha, and A.E Krzesinski. // IEEE Communications Society – 2004, P. 1232-1236.
32. Anees. Shaikh Evaluating the Impact of Stale Link State on Quality-of- Service Routing / Anees Shaikh, Jennifer Rexford, and Kang G. Shin.// IEEE/ACM Transactions on Networking. – 2001. – №9(2), P. 162-176.
33. Chakraborty Basabi Simultaneous Search for Multiple Routes using Genetic Algorithm / Basabi Chakraborty // IEEE International Conference on Computational Intelligence for Measurement System and Applications Boston, MA, USA, 14-16, July 2004, P. 77-80.
34. C. Barakat On TCP performance in a heterogeneous network: a survey Barakat, E. Altman, and W. Dabbous. // IEEE Communications Magazine. – 2000. –
35. Столлингс В. Современные компьютерные сети / Вильям

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

8

Столлинс.– СПб.: Питер, 2003. – 778 с.

36. Таненбаум Э. Компьютерные сети / Эндрю Таненбаум; пер. с англ. А. Леонтьев. – СПб.: Питер, 2002. – 848 с.

37. Закон України "Про Основні засади розвитку інформаційного суспільства в Україні на 2007-2015 роки" від 09.01.2007 р., № 537-V. [Електронний ресурс]. – Доступний з <http://zakon1.rada.gov.ua/laws/>.

38. Кристофер Эберт Эскурс в историю программных технологий / Открытые системы. – 2008. – № 10. – С. 32-36.

39. Кучук Г.А. Синтез топології гібридного хмарного сховища даних у гетерогенному середовищі / Г.А. Кучук // Авиационно-космическая техника и технология : сб. науч.-техн. тр. – 2013. – № 9. – С. 280-284.

40. Мельник А. Організація хмарних обчислень на базі масиву програмованих комірок логіки / А. Мельник, Н. Козак // Вісник Національного університету "Львівська політехніка". – Сер.: Комп'ютерні системи проектування. Теорія і практика. – Львів : Вид-во НУ "Львівська політехніка". – 2010. – № 672. – С. 45-48.

41. Федонюк С.В. "Хмарні" технології в електронному врядуванні / С.В. Федонюк // Науковий вісник Волинського НУ ім. Лесі Українки. – 2011. – № 20. – С. 13-19.

43. Определение Облачных Вычислений (Драфт) Рекомендации Национального Ин-та Стандартов и Технологий (США). [Электронный ресурс]. – Доступный с <http://cloud.sorlik.ru/definition.html>.

44. Джонс М. Тим. Анатомия облачной инфраструктуры хранения данных / М. Тим Джонс. [Электронный ресурс]. – Доступный с <http://www.ibm.com/developerworks/ru/library/cl- cloudstorage/>.

45. Introducing Metal as a Service: provisioning for the hyperscale era. [Electronic resource]. – Mode of access <http://www.markshuttleworth.com/archives/1103>.

Зм.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.22.0015.00.00.ПЗ

Арк.

8

46. Vivek Kundra. Federal cloud computing strategy / Kundra Vivek. February 8, 2011. [Electronic resource]. – Mode of access http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/federal-cloud-computing-strategy.pdf.

47. Сайт Державної фіскальної служби. [Електронний ресурс]. – Доступний з <http://sfs.gov.ua/modernizatsiya-dps-ukraini/arkchiv/proekt--modernizatsiya-derj/elektronna-kartka/>.

48. Бюджетний кодекс України. [Електронний ресурс]. – Доступний з <http://zakon1.rada.gov.ua/laws/>

49. Витер М.Б., Засадна Х.Е. Использование облачных технологий в системе информационного взаимодействия государственных органов

50. Робсон Э., Фримен Э. Изучаем HTML, XHTML и CSS 2-е издание 720 с.

51. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин – <https://zakon.rada.gov.ua/rada/show/v0007282-98#Text>

52. Про затвердження Порядку проведення медичних оглядів працівників певних категорій – <https://zakon.rada.gov.ua/laws/show/z0846-07#Text>

53. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями – <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>

54. Постанова Головного державного санітарного лікаря від 1.12.1999 р. № 42 – <https://www.umj.com.ua/article/3892/postanova-golovnogo-derzhavnogo-sanitarnogo-likarya-vid-1-12-1999-r-42>

55. Природне і штучне освітлення – https://zakon.isu.net.ua/sites/default/files/normdocs/dbn_v_2.5-28_2018.pdf

56. Днаоп 0.00-1.31-99 правила охорони праці під час експлуатації електронно-обчислювальних машин – <https://budinfo.org.ua/doc/1811550/DNAOP-0-00-1-31-99-PRAVILA-OKhORONI-PRATsI-PID-ChAS-EKSPLUATATsIYi-ELEKTRONNO-OBChISLUVALNIKh-MASHIN>

Зм.	Арк.	№ докум.	Підпис	Дата

БКРМ-122.22.0015.00.00.ПЗ

Арк.

8

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	3
5.1 Вміст проекту.....	3
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик	3
5.4 Вимоги до архітектури.....	4
5.5 Вимоги до надійності	4
5.6 Умови експлуатації	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	5
5.8.1 Обладнання	5
5.8.2 Мова програмування.....	5
5.8.3 Вхідні дані	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації	5
7 Економічні вимоги	6
8 Вимоги до охорони праці	6
9 Перелік документів, що розробляються	6
10 Етапи розробки	6
11 Порядок контролю та приймання.....	7

					БКРМ-122.22.0015.00.00.ТЗ			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Тертичний А. Г.</i>			Дослідження та програмна реалізація хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації	<i>Лім</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевір</i>		<i>Якименко Н. М.</i>				М	1	7
<i>Н. Контр.</i>		<i>Гермак В. С.</i>				<i>ЦНТУ гр. КН-21м</i>		
<i>Затвердив</i>		<i>Смірнов О. А.</i>						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію хмарної системи моніторингу бази даних спортивних результатів з аналізом та консолідацією статистичної інформації.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (наказ № 18-13 від 17. 08. 2022 р.).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є програмна реалізація хмарної системи моніторингу бази даних спортивних результатів на основі впровадження нових інформаційних технологій і застосування сучасних засобів програмування.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є розробки, які ведуться кафедрою кібербезпеки та програмного забезпечення і стосовні до теми бібліографічні джерела.

					<i>ВКРМ-122.22.0015.00.00.ТЗ</i>	Арк.
						2
Зм.	Арк.	№ докум.	Підпис	Дата		

5 Технічні вимоги

5.1 Вміст проєкту

Складовими розробки є:

- аналіз існуючих комп'ютеризованих хмарних систем на предмет їхньої відповідності сучасним вимогам;
- вибір і обґрунтування методики побудови додатків і засобів їхньої реалізації;
- розробка структур даних і механізму їхньої взаємодії, робочих форм, засобів і правил бізнесу;
- техніко-економічне обґрунтування доцільності прийнятого до розробки компоненту автоматизованої хмарної системи;
- аналіз умов праці програміста в лабораторії K505;
- розробка програми, яка реалізує алгоритми роботи компоненту, а також розрахунок його часових характеристик в рамках додаткового завдання.

5.2 Показники призначення

Система повинна забезпечувати:

- ведення моніторинг документів в формі їхнього природного подання;
- простий, інтуїтивно зрозумілий інтерфейс з користувачем;
- цілісність даних в таблицях БД.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно містити обмежень на

					<i>ВКРМ-122.22.0015.00.00.ТЗ</i>	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

розширення структур БД, форм, засобів, звітів і запитів SQL.

5.4 Вимоги до архітектури

Компонент, що розробляється, повинен використовувати архітектуру клієнт/сервер, реляційні БД і об'єктно-орієнтовані засоби розробки.

5.5 Вимоги до надійності

Компонент повинен використати існуючі угоди по стандартним викликам процедур, функцій, засобів і форм, визначених технічною документацією на середовище розробки.

5.6 Умови експлуатації

Автоматизовані робочі місця користувачів системи повинні задовольняти наступним умовам експлуатації:

- температура повітря: 18-22⁰;
- відносна вологість повітря при 20⁰ С до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу і параметрів технічних засобів

Компонент повинен бути реалізований на ЕОМ типу IBM PC в операційному середовищі Windows 10 і орієнтований на сумісні з цією платформою зовнішні пристрої, мережне обладнання і прикладне програмне забезпечення.

					<i>ВКРМ-122.22.0015.00.00.ТЗ</i>	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

5.8 Вимоги до інформаційної та програмної сумісності

Сумісність програмного забезпечення повинна бути забезпечена за рахунок його реалізації засобами об'єктно-орієнтованої СУБД, працюючої під управлінням ОС Windows 10 (рекомендується СУБД Mysql).

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

HTML, CSS, PHP, JS, SQL

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 в вигляді опису структури даних, схем і описів алгоритмів, інструкції користувача, а також текстів вхідних модулів програмного забезпечення в відповідності з ЄСПД.

					<i>ВКРМ-122.22.0015.00.00.ТЗ</i>	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

7 Економічні вимоги

7.1 Для програми ядра виконавчої системи моделі необхідно виробити функціонально-вартісний аналіз варіантів розробки

7.2 Виконати розрахунок кошторису витрат на реалізацію прийнятого варіанту з урахуванням розцінок на 1 жовтня 2022 р.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути наведений аналіз умов праці програміста в лабораторії К505 ЦНТУ і вироблений розрахунок категорії важкості праці програміста.

9 Перелік документів, які необхідно розробити

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	– 84 аркуша.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної

					<i>ВКРМ-122.22.0015.00.00.ТЗ</i>	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

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

10.4 Побудова схем взаємодії структур даних.

10.5 Створення прототипу компоненту. Створення програмного продукту.

10.6 Відлагодження компоненту, аналіз отриманих результатів.

10.7 Робота над питаннями охорони праці і техніки безпеки.

10.8 Розрахунки по техніко-економічному обґрунтуванню (остаточні).

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю і приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12. 2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2022 р.

					<i>ВКРМ-122.22.0015.00.00.ТЗ</i>	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ
Керівник випускної кваліфікаційної роботи
за другим (магістерським) рівнем вищої освіти
_____ Якименко Н. М.

*Дослідження та програмна реалізація хмарної системи моніторингу бази
даних спортивних результатів з аналізом та консолідацією статистичної
інформації*

Лістинг програми

Код документу 12

Носій: DVD-диск

Загальна кількість аркушів: 35

Літера: РП

Кропивницький – 2022 року

about.php

```

<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Про нас</title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class = "body1" align = "center" style="font-family: sans-serif;">
      <h1 align="center">Про нас</h1>
      <p>
        Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas
        lobortis volutpat metus id ultricies. Integer molestie gravida sem, ut
        scelerisque augue porttitor at. In iaculis tempor enim nec pellentesque. Cras
        maximus, mauris fringilla pellentesque malesuada, turpis lorem venenatis libero,
        eu fermentum dolor enim vulputate magna. Nam sagittis lorem nec nibh venenatis
        porta. Cras sem quam, tincidunt at mi eget, sagittis sollicitudin dui. Nulla ac
        interdum diam. Suspendisse potenti. Cras tellus metus, blandit aliquet libero
        ac, congue porta augue.

        Nunc in neque sit amet est finibus cursus. Sed euismod magna et urna porta
        ultricies. Duis at commodo est. Duis malesuada diam in dolor euismod aliquam.
        Pellentesque venenatis ex sit amet molestie interdum. Maecenas ut ligula quis
        erat porta accumsan. Morbi ac dolor id dui luctus maximus ac eget libero. Proin
        bibendum libero et sapien lacinia, sit amet mollis odio ullamcorper. Etiam
        dapibus iaculis felis, sed ullamcorper orci placerat in. In tortor eros,
        suscipit a lectus quis, ornare malesuada ex. Curabitur justo risus, lobortis in
        lacinia eget, cursus quis tortor. Fusce vehicula at diam dictum congue.
        Praesent.
      </p>
    </div>
    <?php include 'footer.php'; ?>
    <script type="text/javascript">
      console.log(1235);
      $("#page2").css('color', 'var(--main-color)');
      $("#page2").css('background', 'white');
    </script>
  </body>
</html>

```

accessDenied.php

```
<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class = "body1" align = "center">
      <h3 id = "subTitle">Доступ заборонено</h3>
    </div>
    <?php include 'footer.php'; ?>
    <script type="text/javascript">
      console.log(1235);
      $("#page1").css('color', 'var(--main-color)');
      $("#page1").css('background', 'white');
    </script>
  </body>
</html>
```

Кафедра _ КБПЗ _ 2022 рік

addComment.php

```
<?php
require_once 'setting.php';

if (isset($_POST['submit'])) {
    if ($_POST['commentText'] != '') {
        $userId = $_SESSION['currentUser']['id'];
        $pubId = $_GET['id'];
        $commentText = $_POST['commentText'];

        $q = 'INSERT INTO `comments` (`userId`, `pubId`, `commentText`, `date`)
VALUES';
        $q = $q . ' (' . $userId . ', ' . $pubId . ', ' . $commentText . ', ' . 'Now()' . ')';

        mysqli_query($db, $q);
    }
    header('Location: pubPage.php?id=' . $_GET['id']);
}
?>
```

Кафедра КБПЗ – 2022 рік

addPublic.php

```
<?php
    require_once('setting.php');

    if (isset($_POST['submit'])) {
        if ($_POST['name'] != '') {
            mysqli_query($db, 'INSERT INTO `publics` (`Title`,`Text`)
VALUES ("'. $_POST['name'] . '", "' . $_POST['text'] . '");');
            // echo 'INSERT INTO `publics` (`Title`,`Text`) VALUES ("'.
$_POST['name'] . '", "' . $_POST['text'] . '");';
            header('Location: blog.php');
        }
    }

?>
```

Кафедра _ КБПЗ _ 2022 рік

addService.php

```
<?php
    require_once('setting.php');

    if (isset($_POST['submit'])) {
        if ($_POST['serviceName'] != '') {
            mysqli_query($db, 'INSERT INTO `services`
(`Title`,`ServiceTypeId`) VALUES ("'. $_POST['serviceName'] . '", ' .
$_POST['serviceType'] . ')');
            header('Location: services.php?serviceTypeId=' .
$_POST['serviceType']);
        }
    }

?>
```

Кафедра _ КБПЗ _ 2022 рік

adminPage.php

```
<?php
    require_once 'setting.php';
    if (!($_SESSION['currentUser']['isAdmin'])) {
        header('Location: accessDenied.php');
    }
?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <title>Сторінка адміністратора</title>
        <link rel="stylesheet" href="main.css">
        <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
        <script src="body1.js" charset="utf-8"></script>
    </head>
    <body>
        <?php include 'header.php'; ?>
        <div class = "body1" align = "center">
            <a id="adminB" href="adminPageUserList.php" class =
"myButton">Користувачі</a>
            <a id="adminB" href="adminPageAddService.php" class = "myButton">Додати
послугу</a>
            <a id="adminB" href="adminPageAddPublic.php" class = "myButton">Додати
публікацію</a>
        </div>
        <?php include 'footer.php'; ?>
    </body>
</html>
```

adminPageAddPublic.php

```
<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Гандикап</title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class = "body1" align = "center">
      <form class="addComment" method="post" action="addPublic.php">
        <h1>Додати публікацію</h1>
        <input type="text" name="name" placeholder="Назва">
        <br>
        <br>
        <textarea name="text"></textarea>
        <br>
        <br>
        <input class="myButton" type="submit" name="submit" value="Надіслати">
      </form>
    </div>
    <?php include 'footer.php'; ?>
  </body>
</html>
```

adminPageAddService.php

```
<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Гандикап</title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class = "body1" align = "center">
      <form class="addComment" method="post" action="addService.php">
        <h1>Додати послугу</h1>
        <input type="text" name="serviceName" placeholder="Назва">
        <span>Тип послуги: </span>
        <select name="serviceType" required>
          <option value="1">Наші зали</option>
          <option value="2">Наші тренери</option>
          <option value="3">Наші курси</option>
        </select>
        <br>
        <br>
        <input class="myButton" type="submit" name="submit" value="Надіслати">
      </form>
    </div>
    <?php include 'footer.php'; ?>
  </body>
</html>
```

adminPageUserList.php

```

<?php
    require_once 'setting.php';
    if (!($_SESSION['currentUser']['isAdmin'])) {
        header('Location: accessDenied.php');
    }
?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <title>Сторінка адміністратора</title>
        <link rel="stylesheet" href="main.css">
        <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
        <script src="body1.js" charset="utf-8"></script>
    </head>
    <body>
        <?php include 'header.php'; ?>
        <div class = "body1">

            <?php
                $_user = mysqli_query($db, 'SELECT * FROM `users`');

                while ($user = mysqli_fetch_assoc($_user)) {
                    ?>
                    <?php if (!$user['isAdmin']): ?>
                        <div style="margin-bottom: 20px" class="currentUserInfo">
                            <img width="100px" src=<?=$user['ava'] ?>>
                            <div style="display: inline-block;">
                                <?=$user['FirstName'] ?><br>
                                <?=$user['Name'] ?><br>
                                <?=$user['LastName'] ?><br>
                                <?=$user['E - mail'] ?><br>
                                <?=$user['Login'] ?><br>
                                <?=$user['VIP'] ? "Так" : "Hi" ?><br>
                                <?=$user['isAdmin'] ? "Так" : "Hi" ?><br>
                            </div>
                            <a style="background: red;" id="adminB"
href=<?='adminScript.php?script=0&userID=' . $user['id']; ?>
class="myButton">Видалити</a>
                                <?php if (!$user['VIP']): ?>
                                    <a id="adminB" href=<?='
adminScript.php?script=1&vip=0&userID=' . $user['id']; ?>
class="myButton">Надати VIP</a>
                                <?php else: ?>
                                    <a id="adminB" href=<?='
adminScript.php?script=1&vip=1&userID=' . $user['id']; ?>
class="myButton">Забрати VIP</a>
                                <?php endif; ?>
                                <?php if (!$user['isAdmin']): ?>
                                    <a id="adminB" href=<?='
adminScript.php?script=2&admin=0&userID=' . $user['id']; ?>
class="myButton">Надати адмін. права</a>
                                <?php else: ?>
                                    <a id="adminB" href=<?='
adminScript.php?script=2&admin=1&userID=' . $user['id']; ?>
class="myButton">Забрати адмін. права</a>
                                <?php endif; ?>
                            </div>
                        <?php endif; ?>
                    <?php
                }
            ?>

```

```
        </div>  
        <?php include 'footer.php'; ?>  
    </body>  
</html>
```

Кафедра _ КБПЗ _ 2022 рік

adminScript.php

```
<?php
    require_once 'setting.php';

    switch ($_GET['script']) {
        case 0:
            mysqli_query($db, 'DELETE FROM `users` WHERE `users`.`id` = '
. $_GET['userID']);
            break;
        case 1:
            if (!$_GET['vip']) {
                mysqli_query($db, 'UPDATE `users` SET `VIP` = 1 WHERE
`id` = ' . $_GET['userID']);
            } else if ($_GET['vip']){
                mysqli_query($db, 'UPDATE `users` SET `VIP` = 0 WHERE
`id` = ' . $_GET['userID']);
            }
            break;
        case 2:
            if (!$_GET['admin']) {
                mysqli_query($db, 'UPDATE `users` SET `isAdmin` = 1
WHERE `id` = ' . $_GET['userID']);
            } else if ($_GET['admin']){
                mysqli_query($db, 'UPDATE `users` SET `isAdmin` = 0
WHERE `id` = ' . $_GET['userID']);
            }
            break;
        default:
            break;
    }

    header('Location: adminPageUserList.php');
?>
```

blog.php

```

<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="main.css">
    <title></title>
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class="body1">
      <?php
        $publicPreviems = mysqli_query($db, 'SELECT * FROM `publics`');

        while($publicPreviem = mysqli_fetch_assoc($publicPreviems)){
          ?>

            <div style="display: inline-block;">
              <a class="blogPrev" href=<?= 'pubPage.php?id=' .
$publicPreviem['id'] ?>>
                <img width="150px" src=<?= $publicPreviem['Image']; ?> alt="">
                <br>
                <p><?php echo $publicPreviem['Title'] ?></p>
              </a>
            </div>

            <?php
          }
        ?>

      </div>
      <?php include 'footer.php'; ?>
      <script type="text/javascript">
        console.log(1235);
        $("#page4").css('color', 'var(--main-color)');
        $("#page4").css('background', 'white');
      </script>
    </body>
  </html>

```

body1.js

```
$(document).ready(function() {  
    var header = $('header').height();  
    var footer = $('footer').height();  
    var body1 = $('.body1');  
    body1.css('min-height', ($(window).height() - (header + footer)) - 20);  
    $(window).resize(function() {  
        var header = $('header').height();  
        var footer = $('footer').height();  
        var body1 = $('.body1');  
        body1.css('min-height', ($(window).height() - (header + footer)) - 20);  
    });  
});
```

Кафедра _ КБПЗ _ 2022 рік

footer.php

```
<footer>
  <div id = "f1">
    <a href="index.php" class = "pagef"><b>Головна</b></a>
    <a href="services.php?serviceTypeId=1" class = "pagef"><b>Наши
послуги</b></a>
    <a href="blog.php" class = "pagef"><b>Блог</b></a>
  </div>
  <div id = "f2">
    <h2 class = "headerSiteTitle">&copy ГАНДИКАП</h2>
    <br>
    <h4 class = "headerSiteTitle">+380123456789</h4>
  </div>
</footer>
```

Кафедра _ КБПЗ _ 2022 рік

header.php

```

<header>
  <h1 class = "headerSiteTitle"><a href="index.php">ГАНДИКАП</a></h1>
  <div id = "headerButtonsBlock" align = "end">
    <?php if (!isset($_SESSION['currentUser'])): ?>
      <a class="myButton" href="signup.php">
        <b>Зареєструватись</b>
      </a>
      <a class="myButton" href="signin.php">
        <b>Увійти</b>
      </a>
    <?php else: ?>
      <a class="myButton" href="userPage.php">
        <b>Обліковий запис <?= $_SESSION['currentUser']['isAdmin'] ?
'адміністратора' : 'користувача'; ?></b>
      </a>
      <a class="myButton" href="signout.php">
        <b>Вийти</b>
      </a>
    <?php endif; ?>
  </div>
  <div id = "pages" align = "center">
    <a id = "page1" href="index.php" class = "page"><b>Головна</b></a>
    <a id = "page2" href="about.php" class = "page"><b>Про нас</b></a>
    <a id = "page3" href="services.php?serviceTypeId=1" class = "page"><b>Наші
послуги</b></a>
    <a id = "page4" href="blog.php" class = "page"><b>Блог</b></a>
  </div>
</header>

```

index.php

```
<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Гандикап</title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class = "body1" align = "center">
      <h1 id = "siteTitle">ГАНДИКАП</h1>
      <br>
      <h3 id = "subTitle">СПОРТИВНИЙ КЛУБ</h3>
    </div>
    <?php include 'footer.php'; ?>
    <script type="text/javascript">
      console.log(1235);
      $("#page1").css('color', 'var(--main-color)');
      $("#page1").css('background', 'white');
    </script>
  </body>
</html>
```

Кафедра _ КБПЗ _ 2022 рік

main.css

```
:root {
  --main-color: rgb(20, 79, 150);
  --buttons-color: rgb(66, 238, 72);
}
body {
  margin: 0;
  padding: 0;
  background: #F1F1F1;
}
header {
  background: var(--main-color);
  height: 70px auto;
}
footer {
  width: 100%;
  background: black;
  height: 50px;
}
footer div {
  display: inline-block;
}
#f1 {
  margin-top: 10px;
}
#f2 {
  float: right;
}
.headerSiteTitle {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
  color: white;
  display: inline;
  margin: 10px;
}
.headerSiteTitle a {
  text-decoration: none;
  color: white;
}
#siteTitle {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
  color: var(--main-color);
  display: inline;
  font-size: 150px;
}
#subTitle {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
  color: var(--main-color);
  display: inline;
  font-size: 50px;
}
#headerButtonsBlock {
  margin: 20px;
  display: inline;
  position: absolute;
  right: 0;
```

```
}
#headerButtonsBlock a{
  margin: 5px;
}
#mainInfo {
  margin: 10px;
  background: white;
}

#servicesTypes {
  padding: 10px;
}
#pages {
  padding-bottom: 10px;
}

#adminB {
  display: block;
  margin: 5px;
}

.body1 {
  background: white;
  /* margin: 10px; */
  margin-top: 0px;
  padding: 10px;
}
.services {
  background: white;
  margin: 10px;
  padding: 10px;
}
.button {
  background: white;
  text-decoration: none;
  display: inline-block;
  font-family: sans-serif;
  color: var(--main-color);
}
.button2 {
  background: red;
  width: 150px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-family: sans-serif;
  color: white
}
.button div{
  margin: 10px;
}
.button2 div{
  margin: 10px;
}
.page {
  color: white;
  text-decoration: none;
  font-family: sans-serif;
  margin: 10px;
  padding: 15px;
  /* background: green; */
}
.pagef {
```

```
    color: white;
    text-decoration: none;
    font-family: sans-serif;
    margin: 10px;
}
.serviceList {
    margin: 10;
    padding: 10;
    background: white;
    display: inline-block;
}
.serviceItem {
    margin: 10px;
    padding: 10px;
    /* width: 250px; */
    height: 100px;
    background: var(--main-color);
    color: white;
    font-family: sans-serif;
    display: inline-block;
    box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.serviceItem span {
    vertical-align: top;
    margin-left: 10px;
}
.loginForm {
    padding: 10px;
}
.loginForm input {
    border-style: none;
    text-align: center;
}
.loginForm form {
    padding: 10px;
    width: 200px;
    background: var(--main-color);
    box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.signupForm form {
    padding: 10px;
    width: 300px;
    background: var(--main-color);
    box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.signupForm input {
    border-style: none;
    text-align: center;
}
.button3 {
    background: var(--main-color);
    color: white;
    text-decoration: none;
    font-family: sans-serif;
    padding: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 10);
    margin: 20px;
}
```

```
}
.blogPrev::before {
  display: inline-block;
}
.blogPrev {
  background: var(--main-color);
  text-align: center;
  color: white;
  margin-left: 0px;
  margin-right: 5px;
  margin-bottom: 10px;
  padding: 10px;
  font-family: sans-serif;
  display: inline-block;
  box-shadow: 0 0 10px rgba(0, 0, 0, 10);
  text-decoration: none;
}
.blogPrev span {
  text-decoration: none;
}
.publicationText {
  background: var(--main-color);
  font-family: sans-serif;
  margin: 5px;
  padding: 5px;
  color: white;
  box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.publicationText img {
  position: absolute;
  margin-top: 0;
}
.addComment {
  margin: 10px;
  padding: 10px;
  background: var(--main-color);
  color: white;
  width: 700px;
  font-family: sans-serif;
  box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.addComment h3 {
  margin-left: 0;
  margin-right: 0;
  margin-top: 0;
  margin-bottom: 10;
}
.comment {
  margin: 10px;
  padding: 10px;
  background: var(--main-color);
  color: white;
  width: 700px;
  font-family: sans-serif;
  box-shadow: 0 0 10px rgba(0, 0, 0, 10);
}
.comment * {
  display: inline;
}
.comment img {
```

```
width: 50px;
height: 50px;
}
.delete {
padding: 5px;
background: red;
font-family: sans-serif;
color: white;
text-decoration: none;
float: right;
}
.currentUserInfo {
color: white;
text-align: left;
font-family: sans-serif;
background-color: var(--main-color);
display: inline-block;
padding: 10px;
font-size: 11px;
}
.myButton {
background: var(--buttons-color);
padding: 10px;
border-radius: 20px;
color: white;
text-decoration: none;
font-family: sans-serif;
border-style: none;
}
}
```

Кафедра _ КБПЗ _ 2022 рік

pubPage.php

```

<?php
    require_once 'setting.php';
    $id = $_GET['id'];
    $_pub = mysqli_query($db, 'SELECT * FROM `publics` WHERE `id` = ' . $id);
    $pub = mysqli_fetch_assoc($_pub);
?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
    <head>
        <meta charset="utf-8">
        <title><?php echo $pub['Title']; ?></title>
        <link rel="stylesheet" href="main.css">
        <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
        <script src="body1.js" charset="utf-8"></script>
    </head>
    <body>
        <?php include 'header.php'; ?>
        <div class="body1">
            <div class="publicationText">
                <h1><?php echo $pub['Title']; ?></h1>
                <p>
                    <?php echo $pub['Text']; ?>
                </p>
                <span><?php echo $pub['PubDate']; ?></span>
            </div>
            <?php
                $comments = mysqli_query($db, 'SELECT * FROM `comments` WHERE `pubId`
= ' . $id);

                while ($comment = mysqli_fetch_assoc($comments)) {
                    $_user = mysqli_query($db, 'SELECT * FROM `users` WHERE `id` = ' .
$comment['UserId']);
                    $_user = mysqli_fetch_assoc($_user);
                    ?>
                    <div class="comment">
                        <span>
                            <img align="left" hspace="5" src=<?php echo $_user['ava']; ?>
alt="">
                            <?php
                                echo $_user['FirstName'] . ' ' . $_user['Name'] . ' ' .
$_user['LastName'];
                                ?>
                            </span>
                            <br>
                            <br>
                            <p style="margin-left: 60px; margin-top: 10px; text-align:
center;">
                                <?php
                                    echo $comment['commentText'];
                                    ?>
                                </p>
                                <br>
                                <span>
                                    <?php
                                        echo $comment['date'];
                                        ?>
                                    </span>
                                    <?php
                                        if (($_SESSION['currentUser']['isAdmin'])): ?>
                                        <a class = "delete" href=
                                        <?php
                                            echo 'pubPage.php?removeCommentId=' . $comment['id'] .
'&id=' . $id;
                                        ?>
                                    </p>
                                </div>
                }
            </div>
        </div>
    </body>
</html>

```

```
        >Видайти</a>
    <?php endif; ?>
</div>
<?php
}
?>
<?php if (isset($_SESSION['currentUser'])): ?>
    <form class="addComment" action=?php echo 'addComment.php?id=' . $id;
?> method="post">
    <h3>Коментувати</h3>
    <textarea name="commentText" rows="8" cols="80"></textarea>
    <input class="myButton" type="submit" name="submit" value="Надіслати">
    </form>
    <?php endif; ?>
</div>

<?php include 'footer.php'; ?>
</body>
</html>
```

Кафедра КБПЗ – 2022 рік

services.php

```

<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="main.css">
    <title></title>
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class="body1">
      <br>
      <div id = "servicesTypes" align = "center">
        <?php
          $servicesTypes = mysqli_query($db, 'SELECT * FROM `services_types`');
          while ($serviceType = mysqli_fetch_assoc($servicesTypes)) {
            ?>
            <a id = "serviceTypeButton" class = "button3" href=<?php echo
'services.php?serviceTypeId=' . $serviceType['id']; ?>><?php echo
$serviceType['Name']; ?></a>
            <?php
              }
            ?>
          </div>
          <br>
          <div id = "serviceList" align="center">
            <?php
              $serviceTypeId = $_GET['serviceTypeId'];
              $services = mysqli_query($db, 'SELECT * FROM `services` WHERE
`ServiceTypeId` = ' . $serviceTypeId);

              while ($service = mysqli_fetch_assoc($services)) {
                ?>
                <div width="120px" class="serviceItem">
                  <img width="100" src=<?php echo $service['Image']; ?> alt="">
                  <span ><?php echo $service['Title']; ?></span>
                </div>
              <?php
                }
              ?>
            </div>
          </div>
          <?php include 'footer.php'; ?>
          <script type="text/javascript">
            $("#page3").css('color', 'var(--main-color)');
            $("#page3").css('background', 'white');
          </script>
        </body>
      </html>

```

setting.php

```
<?php
session_start();
$db = mysqli_connect('localhost', 'root', '123', 'gandikap');
mysqli_query($db, "SET NAMES utf8");
if (!$db) {
    echo mysqli_error();
    die();
}
?>
```

Кафедра КБПЗ – 2022 рік

signin.php

```

<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Увійти</title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class="body1">
      <div class="loginForm" align = "center">
        <form id = "loginForm" class="" action="signin.php" method="post">
          <h1 class = "headerSiteTitle">Увійти</h1>
          <br><br>
          <input type="text" name="Login" value="<?=@$_POST['Login']?>"
placeholder="Ім'я користувача">
          <br><br>
          <input type="password" name="Password" value="" placeholder="Пароль">
          <br><br>
          <button class="myButton" type="submit" name="submit"
><b>Надіслати</b></button>
        </form>
      </div>
      <?php
        $data = $_POST;
        if (isset($data['submit'])){
          if ($data['Login'] == ''){
            $errors[] = 'Ви не ввели логін!!!';
          }
          if ($data['Password'] == ''){
            $errors[] = 'Ви не ввели пароль!!!';
          }
          $login = $data['Login'];
          $user = mysqli_query($db, 'SELECT * FROM `users` WHERE `Login` = "' .
$login . '"');
          if (mysqli_num_rows($user) == 0) {
            $errors[] = 'Користувача з таким логіном не знайдено!!!';
          } else {
            $pass = $data['Password'];
            $pass_hash = mysqli_fetch_assoc($user)['Password'];
            if (!password_verify($pass, $pass_hash)){
              $errors[] = 'Невірний пароль!!!';
            }
          }
          if (empty($errors)){
            $user = mysqli_query($db, 'SELECT * FROM `users` WHERE `Login` = "'
. $login . '"');
            $_SESSION['currentUser'] = mysqli_fetch_assoc($user);
          }
        }
        <script type="text/javascript">
          document.location = 'index.php';
        </script>
      <?php
    } else {
      echo '<hr>';
      echo '<span>' . $errors[0] . '</span>';
    }
  }
}

```

```
    }  
  }  
  ?>  
</div>  
<?php include 'footer.php'; ?>  
</body>  
</html>
```

Кафедра КБПЗ – 2022 рік

signinout.php

```
<?php
    require_once 'setting.php';

    unset($_SESSION['currentUser']);
    header('Location: index.php');
?>
```

Кафедра _ КБПЗ _ 2022 рік

signup.php

```

<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Зареєструватись</title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class="body1">
      <div class="signupForm" align = "center">
        <form id = "loginForm" class="" action="signup.php" method="post">
          <h1 class = "headerSiteTitle">Зареєструватись</h1>
          <br><br>
          <input type="text" name="Login" value="<?=@$_POST['Login']?>"
placeholder="Логін">
          <br><br>
          <input type="text" name="FirstName" value="<?=@$_POST['FirstName']?>"
placeholder="Прізвище">
          <br><br>
          <input type="text" name="Name" value="<?=@$_POST['Name']?>"
placeholder="Ім'я">
          <br><br>
          <input type="text" name="LastName" value="<?=@$_POST['LastName']?>"
placeholder="По батькові">
          <br><br>
          <input type="text" name="Mail" value="<?=@$_POST['Mail']?>"
placeholder="e - mail">
          <br><br>
          <input type="password" name="Password1" value="" placeholder="Пароль">
          <br><br>
          <input type="password" name="Password2" value=""
placeholder="Підтвердження пароля">
          <br><br>
          <button class="myButton" type="submit" name="submit"
><b>Надіслати</b></button>
        </form>
      </div>
    <?php
      $data = $_POST;
      if (isset($data['submit'])){
        if ($data['Login'] == ''){
          $errors[] = 'Ви не ввели логін!!!';
        }
        $loginCheck = mysqli_query($db, 'SELECT * FROM `users` WHERE `Login` =
"' . $data['Login'] . "'");
        if (mysqli_num_rows($loginCheck) !=0){
          $errors[] = 'Користувач з таким логіном вже існує!!!';
        }
        if ($data['FirstName'] == ''){
          $errors[] = 'Ви не ввели Прізвище!!!';
        } else if ($data['Name'] == ''){
          $errors[] = 'Ви не ввели Ім'я!!!';
        } else if ($data['LastName'] == ''){
          $errors[] = 'Ви не ввели По батькові!!!';
        } else if ($data['Mail'] == ''){
          $errors[] = 'Ви не ввели e - mail!!!';
        }
        $mailCheck = mysqli_query($db, 'SELECT * FROM `users` WHERE `E - mail`
= "' . $data['Mail'] . "'");

```

```

if (mysqli_num_rows($mailCheck) !=0){
    $errors[] = 'Користувач з таким е - mail вже існує!!!';
}
if ($data['Password1'] == ''){
    $errors[] = 'Ви не ввели пароль!!!';
} else if ($data['Password2'] == ''){
    $errors[] = 'Ви не ввели підтвердження пароля!!!';
} else if ($data['Password1'] != $data['Password2']){
    $errors[] = 'Підтвердження пароля не відповідає паролю!!!';
}

if (empty($errors)){
    $login =      '' . $data['Login'] . '';
    $firstName =  '' . $data['FirstName'] . '';
    $name =       '' . $data['Name'] . '';
    $lastName =   '' . $data['LastName'] . '';
    $mail =       '' . $data['Mail'] . '';
    $password =   '' . password_hash($data['Password1'],
PASSWORD_DEFAULT) . '';

    mysqli_query($db, "INSERT INTO
`users`(`Login`,`FirstName`,`Name`,`LastName`,`E - mail`,`Password`) VALUES (
        $login,
        $firstName,
        $name,
        $lastName,
        $mail,
        $password)");

    $user = mysqli_query($db, 'SELECT * FROM `users` WHERE `Login` =
' . $login . ');
    $_SESSION['currentUser'] = mysqli_fetch_assoc($user);

    ?>
    <script type="text/javascript">
        document.location = 'userPage.php';
    </script>
    <?php
} else {
    echo '<hr>';
    echo '<span>' . $errors[0] . '</span>';
}
}
?>
</div>
<?php include 'footer.php'; ?>
</body>
</html>

```

userPage.php

```

<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Гандикап</title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class = "body1">
      <div class="currentUserInf">
        <?php
          $user = $_SESSION['currentUser'];
          ?>
          <div style="margin-bottom: 20px" class="currentUserInfo">
            <img width="100px" src=<?=$user['ava'] ?>>
            <div style="display: inline-block;">
              <?=$user['FirstName'] ?><br>
              <?=$user['Name'] ?><br>
              <?=$user['LastName'] ?><br>
              <?=$user['E - mail'] ?><br>
              <?=$user['Login'] ?><br>
              <?=$user['VIP'] ? "Так" : "Hi" ?><br>
              <?=$user['isAdmin'] ? "Так" : "Hi" ?><br>
            </div>
          </div>
          <?php if (($SESSION['currentUser']['isAdmin'])): ?>
            <br>
            <a href="adminPage.php" class="myButton">Сторінка
адміністратора</a>
          <?php endif; ?>
        <?php
          ?>
        </div>
      </div>
      <?php include 'footer.php'; ?>
    </body>
  </html>

```

azureCon.php

```
$host = 'mydemosever.mysql.database.azure.com';
$username = 'myadmin@mydemosever';
$password = 'your_password';
$db_name = 'your_database';

$_SESSION['azur5e'] = $db_name;

$conn = mysqli_init();

mysqli_ssl_set($conn, NULL, NULL, "/var/www/html/DigiCertGlobalRootG2.crt.pem",
NULL, NULL);

mysqli_real_connect($conn, 'mydemosever.mysql.database.azure.com',
'myadmin@mydemosever', 'yourpassword', 'quickstartdb', 3306, NULL,
MYSQLI_CLIENT_SSL);

if (mysqli_connect_errno())
{
    die('Failed to connect to MySQL: ' . mysqli_connect_error());
}
```

Кафедра КБПЗ – 2022 рік

datPage.php

```

<?php require_once 'setting.php'; ?>
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Дани</title>
    <link rel="stylesheet" href="main.css">
    <script src="jquery-3.3.1.min.js" charset="utf-8"></script>
    <script src="body1.js" charset="utf-8"></script>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <div class = "body1" align = "center" style="font-family: sans-serif;">
      $product_name = 'BrandNewProduct';
      $product_color = 'Blue';
      $product_price = 15.5;
      if ($stmt = mysqli_prepare($conn, "INSERT INTO Products
      (ProductName, Color, Price) VALUES (?, ?, ?)"))
      {
        mysqli_stmt_bind_param($stmt, 'ssd', $product_name,
        $product_color, $product_price);
        mysqli_stmt_execute($stmt);
        printf("Insert: Affected %d rows\n",
        mysqli_stmt_affected_rows($stmt));
        mysqli_stmt_close($stmt);
      }
      $product_name = 'BrandNewProduct';
      $product_color = 'Blue';
      $product_price = 15.5;
      if ($stmt = mysqli_prepare($conn, "INSERT INTO Products
      (ProductName, Color, Price) VALUES (?, ?, ?)"))
      {
        mysqli_stmt_bind_param($stmt, 'ssd', $product_name,
        $product_color, $product_price);
        mysqli_stmt_execute($stmt);
        printf("Insert: Affected %d rows\n",
        mysqli_stmt_affected_rows($stmt));
        mysqli_stmt_close($stmt);
      }
      printf("Reading data from table: \n");
      $res = mysqli_query($conn, 'SELECT * FROM Products');
      while ($row = mysqli_fetch_assoc($res))
      {
        var_dump($row);
      }
      $product_name = 'BrandNewProduct';
      if ($stmt = mysqli_prepare($conn, "DELETE FROM Products WHERE
      ProductName = ?")) {
        mysqli_stmt_bind_param($stmt, 's', $product_name);
        mysqli_stmt_execute($stmt);
        printf("Delete: Affected %d rows\n",
        mysqli_stmt_affected_rows($stmt));
        mysqli_stmt_close($stmt);
      }
    </div>
    <?php include 'footer.php'; ?>
    <script type="text/javascript">
      console.log(1235);
      $("#page2").css('color', 'var(--main-color)');
      $("#page2").css('background', 'white');
    </script>
  </body>
</html>

```

```
az group delete \  
  --name $AZ_RESOURCE_GROUP \  
  --yes
```

azure.sh

Кафедра _ КБПЗ _ 2022 рік