

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

*бакалавр*

(назва освітнього ступеня)

на тему: *Програмно-апаратна навчальна платформа  
«Wisdom in Knowledge»*

Виконав: студент IV курсу, групи СІс-44  
спеціальності 123 «Комп'ютерна інженерія»

(шифр і назва спеціальності)

(підпис)

*Цимбалістий В.О.*

(прізвище та ініціали)

Керівник

(підпис)

*Яцишин В.В.*

(прізвище та ініціали)

Нормоконтроль

(підпис)

*Тим С.В.*

(прізвище та ініціали)

Завідувач кафедри

(підпис)

*Осухівська Г.М.*

(прізвище та ініціали)

Рецензент

(підпис)

*Гром'як Р.С.*

(прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Осухівська Г.М.  
(підпис) (прізвище та ініціали)  
« » 2021 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 123 «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

студенту Цимбалістому Віктору Олександровичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Програмно-апаратна навчальна платформа «Wisdom in Knowledge»

Керівник роботи Яцишин Василь Володимирович, к.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «10» лютого 2021 року № 4.7-97

2. Термін подання студентом завершеної роботи 23.06.2021 р.

3. Вихідні дані до роботи Організація стеку Linux Apache MySQL PHP, розміщення веб-сайту на сервері, побудова об'єктно-орієнтованих веб-додатків по патерну MVC.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз вимог та можливих рішень щодо реалізації програмно-апаратної навчальної платформи Wisdom in Knowledge. 2. Розробка концептуальної архітектури та обґрунтування вибору програмного забезпечення проєктованого засобу. 3. Реалізація та тестування навчальної платформи. 4. Безпека життєдіяльності, основи охорони праці. Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. LAMP Client-Server.

2. Мережева взаємодія Client-Server.

3. ER діаграма

4. UML діаграми класів

5. UML діаграми послідовності

6. Каркас MVC

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
<i>Безпека життєдіяльності, основи охорони праці</i>	<i>Пилипець М.І., д.т.н - професор</i>		

7. Дата видачі завдання \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка і затвердження технічного завдання</i>	<i>10.02-16.02.2021</i>	
2	<i>Аналіз технічного завдання</i>	<i>17.02-02.03.2021</i>	
3	<i>Визначення вимог до апаратного та програмного забезпечення навчальної платформи «Wisdom in Knowledge»</i>	<i>03.03-18.03.2021</i>	
4	<i>Проектування схем програмно-апаратної навчальної платформи «Wisdom in Knowledge»</i>	<i>19.03-04.04.2021</i>	
5	<i>Налаштування сервісів платформи</i>	<i>04.04-02.05.2021</i>	
6	<i>Розробка програмного забезпечення веб-сайту</i>	<i>02.05-29.05.2021</i>	
7	<i>Безпека життєдіяльності, основи охорони праці</i>	<i>01.06-08.06.2021</i>	
8	<i>Оформлення кваліфікаційної роботи</i>	<i>09.06-18.06.2021</i>	
9	<i>Попередній захист кваліфікаційної роботи</i>	<i>18.06-22.06.2021</i>	
10	<i>Захист кваліфікаційної роботи</i>	<i>22.06-28.06.2021</i>	

Студент

\_\_\_\_\_ (підпис)

*Цимбалістий Віктор Олександрович*

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

*Яцишин Василь Володимирович*

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Програмно-апаратна платформа навчання Wisdom in Knowledge // Кваліфікаційна робота на здобуття освітнього ступеня бакалавр // Цимбалістий Віктор Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІс-44 // Тернопіль, 2021 // с. – 108, рис. – 34, табл. – 13, аркушів А1. – 6, додат. – 3, бібліогр. – 22.

Ключові слова: Linux, Apache, MySQL, PHP, Front controller, MVC, platform, courses, Front end, Back end, Semantic HTML, БЕМ.

Кваліфікаційна робота бакалавра присвячена розробці програмно-апаратної навчальної платформи «Wisdom in Knowledge», яка реалізується у вигляді веб-орієнтованого засобу, що може бути розгорнутий як на локальному сервері навчального закладу, так і на глобальному хостингу.

Проектування платформи передбачає налаштування сервісів апаратного забезпечення та розробку об'єктно-орієнтованого веб-додатку на основі патерну MVC, що є різновидом архітектури «клієнт-сервер».

Програмна складова апаратно-програмної платформи включає фронтенд та бекенд частини, а також використання реляційної бази даних.

Засоби та елементи керування програмної складової забезпечують взаємодію користувачів з курсами та тестами до них, на основі ролей студент та викладач.

## ABSTRACT

Software and hardware learning platform Wisdom in Knowledge // Qualification work for the bachelor's degree // Tsymbalistyi Viktor Oleksandrovysh // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Systems and Networks Department, Group SIs-44 // Ternopil, 2021 // p. – 108, fig. – 34, tab. – 13, posters A1 – 6, addition. – 3, ref. – 22.

Keywords: Linux, Apache, MySQL, PHP, Front controller, MVC, platform, courses, Front end, Back end, Semantic HTML, BEM.

The qualification work of the bachelor is devoted to the development of software and hardware educational platform "Wisdom in Knowledge", which is implemented in the form of a web-based tool that can be deployed both on the local server of the institution and on global hosting.

The design of the platform involves the configuration of hardware services and the development of object-oriented web application based on the MVC pattern, which is a kind of client-server architecture.

The software component of the hardware and software platform includes the front and backend parts, as well as the use of a relational database.

The tools and elements of software component management provide user interaction with courses and tests for them, based on the roles of student and teacher.

## ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ	8
ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ .....	10
1.1 Аналіз вимог до програмно-апаратної навчальної платформи Wisdom in Knowledge .....	10
1.1.1 Аналіз вимог до компонентів апаратного забезпечення .....	10
1.1.2 Аналіз вимог до програмного забезпечення .....	10
1.2 Аналіз можливих рішень щодо реалізації навчальної платформи Wisdom in Knowledge .....	14
1.2.1 Аналіз технологій реалізації платформи.....	14
1.2.2 Аналіз можливості використання хмарних сервісів.....	17
1.2.3 Аналіз хостингів для публікації платформи .....	18
РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА .....	19
2.1 Розробка концептуальної архітектури програмно-апаратної навчальної платформи Wisdom in Knowledge .....	19
2.2 Обґрунтування вибору апаратного забезпечення проєктованого комп'ютерного засобу.....	21
2.2.1 Розрахунок параметрів серверної частини для функціонування навчальної платформи .....	21
2.2.2 Вибір компонентів апаратного забезпечення для функціонування клієнтської сторони навчальної платформи .....	27
2.3 Обґрунтування вибору програмного забезпечення проєктованого комп'ютерного засобу.....	27
2.3.1 Visual Studio Code.....	27
2.3.2 MySQL .....	28

					<b>КС КРБ 123.184.00.00 ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>				
Розроб.		Цимбалістий В.О.			<b>Програмно-апаратна навчальна платформа «Wisdom in Knowledge»</b>	<b>Літ.</b>	<b>Арк.</b>	<b>Аркуші</b>
Перевір.		Яцишин В.В.				6	108	
Рецензент						<b>ТНТУ, каф. КС, гр. СІс-44</b>		
Н. Контр.		Тиш С.В.						
Затверд.		Осухівська Г.М.						

2.3.3 JavaScript .....	29
2.4 Проектування комп'ютерного засобу .....	29
2.4.1 Побудова use case діаграм.....	29
2.4.2 Проектування БД.....	31
2.4.3 Побудова діаграм класів .....	34
2.4.4 Побудова діаграм послідовностей.....	36
<b>РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА .....</b>	<b>38</b>
3.1 Реалізація навчальної платформи.....	38
3.1.1 Налаштування сервісів .....	38
3.1.2 Розробка алгоритму роботи програми .....	52
3.1.3 Розробка користувацьких інтерфейсів .....	56
3.2 Тестування функціональності платформи .....	57
<b>РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....</b>	<b>59</b>
4.1 Працездатність людини – оператора.....	59
4.2 Загальні вимоги безпеки з охорони праці для користувачів ЕОМ .....	62
4.3 Інструкція з охорони праці для обслуговуючого персоналу серверного обладнання.....	63
<b>ВИСНОВКИ .....</b>	<b>67</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>68</b>
Додаток А Технічне завдання.....	70
Додаток Б Таблиці БД.....	81
Додаток В Лістинг фронтенд, бекенд .....	84

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ,  
СИМВОЛІВ І СКОРОЧЕНЬ

БЕМ (Блок елемент модифікатор)

FTP (File Transport Protocol)

HTTP (Hypertext Transfer Protocol)

HTTPS (Hypertext Transfer Protocol Secure)

IEEE 830 (Institute of Electrical and Electronics Engineers)

LAMP (Linux, Apache, MySQL, PHP)

MVC (Model, View, Controller)

SFTP (Secure File Transport Protocol)

SSL (Secure Sockets Layer)

VDS (Virtual Dedicated Server)

VPS (Virtual Private Server)

					<i>КС КРБ 123.184.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8



## ВСТУП

Сучасні тренди розвитку інформаційних технологій до яких входять такі сфери, як комп'ютерна та програмна інженерія, комп'ютерні науки і кібербезпека, сприяють широкому їхньому впровадженню у сфері освіти та наукових досліджень. Враховуючи важливість таких аспектів як можливість дистанційного та онлайн навчання у період світових пандемій, воєн та інших стихійних лих, на сьогодні актуальним завданням є створення і впровадження неформальних форм навчання, що дозволяють незалежно від місць локації користувачів здобувати якісну освіту. Прикладом цього, є платформи Coursera, Prometheus, Metanit.

Метою даної кваліфікаційної роботи є розробка програмно-апаратної навчальної платформи Wisdom in Knowledge. Дана платформа спрямована на підтримку та забезпечення послуг неформального навчання посередництвом курсів для усіх бажаючих без вікових обмежень.

Проектована платформа повинна базуватись на основі програмно-апаратного комплексу організаційних, інформаційних і технічних рішень, що забезпечують взаємодію викладача з студентом за посередництва веб-сайту розміщеного на сервері. Платформи існують як окремі інтернет-системи (веб-ресурси). Платформа Wisdom in Knowledge повинна реалізовувати комунікацію на основі клієнт-серверної архітектури, де клієнт – це викладач або учень, а сервер – комплекс сервісів з розміщеним веб-сайтом.

Доступ до веб-сайтів повинен забезпечуватись шляхом реалізації звернень до доменного імені, яке прив'язане до публічної статичної IP-адреси сервера.

Обмеженнями на проект програмно-апаратної платформи є використання HTML 5, CSS 3, JS ES6, PHP 7.2, MySQL (MariaDB 10.1), Ubuntu Server 18.04, Apache 2.4.

Розробка та експлуатація платформи Wisdom in Knowledge є актуальною для людей, що бажають здобути нові навички або стати ініціатором засновником курсів неформального навчання.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

## РОЗДІЛ 1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

### 1.1 Аналіз вимог до програмно-апаратної навчальної платформи Wisdom in Knowledge

#### 1.1.1 Аналіз вимог до компонентів апаратного забезпечення

Відповідно до виду хостингу складаються вимоги до апаратного забезпечення, яким повинен відповідати хост-провайдер.

Хостинг провайдер повинен забезпечити організацію DMZ («Demilitarized Zone») з загальнодоступним сервером доступ до якого надається, через міжмережвий екран. Доступ сервера в мережу інтернет здійснюється за допомогою маршрутизатора.

Характеристики сервера, ширина каналу лінії доступу до мережі інтернет забезпечуються відповідно до вимог регламентованих відповідними стандартами.

#### 1.1.2 Аналіз вимог до програмного забезпечення

До програмного забезпечення (ПЗ) входять: операційна система (ОС); веб-сервер; мови та діалекти програмування, які інтерпретуються за допомогою браузера і веб-сервера.

На основі останніх розробляється об'єкт, що представляє собою «Front end» та «Back end» результатом злиття яких є веб-сайт. Сама ОС та веб-сервер здійснюють комунікацію між апаратною частиною та веб-сайтом за посередництва браузера.

					<b>КС КРБ 123.184.00.00 ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>				
Розроб.		Цимбалістий В.О.			<b>Аналіз технічного завдання</b>	<b>Літ.</b>	<b>Арк.</b>	<b>Акрушіє</b>
Перевір.		Яцишин В.В.					10	9
Рецензент						<b>ТНТУ, каф. КС, гр. СІс-44</b>		
Н. Контр.		Тиш С.В.						
Затверд.		Осухівська Г.М.						

Апаратна частина розглядається як один єдиний цільний об'єкт, з точки зору клієнт-серверної архітектури. Для організації аналізу об'єктів застосовуються функціональні та нефункціональні вимоги.

До функціональних вимог, які висуваються до програмно-апаратної платформи належать:

- збереження незалежності і модульності блоків та їх елементів для можливості подальшого використання в інших частинах сайту за допомогою методології БЕМ;

- розробка крос-платформового адаптиву для «Front end» об'єкту за допомогою «Break points», для настільних персональних комп'ютерів (ПК), ноутбуків та планшетів, смартфонів.

Обмеження прав неавторизованого клієнта в користуванні веб-сайтом, у вигляді лише перегляду наявних на платформі курсів. Для зняття обмежень клієнт повинен зареєструватись або авторизуватись в системі з дотриманням валідності форм.

Локальні системні ролі передбачені апаратно-програмною платформою включають в себе: викладач курсу, учень курсу.

Узагальнення ролей:

- учень обмежений в адмініструванні курсами і зберігає за собою право лише їх проходження;

- викладач обмежений адмініструванням лише своїх курсів та зберігає за собою можливість проходження курсів інших викладачів у ролі учня.

Перелік можливостей викладача:

- створення, видалення курсів;
- додавання контенту на створений курс у вигляді навігаційної панелі:
  - а) створення сторінки лекції з додаванням відео, фото, текстового матеріалу, файлів (word, pdf, mp4, тощо);
  - б) створення тестових завдань прив'язаних до курсу;

Перелік можливостей учня:

- передперегляд курсу;

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

- підписка на курс;
- виконання тестових завдань;
- ведення статистики прогресу курсів.

До нефункціональних вимог апаратно-програмної платформи Wisdom in Knowledge належать:

- забезпечення оптимізації розмітки «Front end» об'єкту під зміну контенту, в разі збільшення кількості тексту, що може призвести до виходу за межі блоків.
- усунення конфліктів між блоками, що може спричиняти візуальну колізію зображення.
- дотримання семантичної розмітки «Front end» об'єкту, для покращення просування сайту в пошукових системах, за допомогою пошукових роботів.
- використання сучасних форматів таких як «webp», при розробці веб-сайту для забезпечення оптимізованого використання ресурсів клієнта при відображенні контенту на пристроях, що є критичним для планшетів та смартфонів.

Реалізація «Back end» об'єкту по патерну «MVC». На основі даного шаблону розробляється каркас системи з єдиною точкою входу, яка забезпечує маршрутизацію по сайту.

З урахуванням програмної частини будуються нефункціональні вимоги до апаратної частини, що враховує навантаження ПЗ та трафіку, які будуть опрацьовуватись за допомогою сервера.

Специфікація вимог відповідно стандарту IEEE830 на основі вище перерахованого:

1.1. Метою створення платформи онлайн курсів є забезпечення дистанційного та неформального навчання для всіх осіб, які цікавляться додатковим саморозвитком та удосконаленням у вільний час.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

1.2. Платформа орієнтована на користувача, будь-якого класу чи віку, оскільки є простою та інтуїтивно зрозумілою, що забезпечується даними про рівень складності.

1.3. Визначення, терміни та скорочення.

1.4. Посилання.

1.5. Огляд:

1.5.1. Огляд загальних завдань, які виконують подібні веб-сайти:

1.5.1.1. Можливість самовдосконалюватися через процес навчання в зручній формі інтернет навчання в вигляді курсів;

1.5.1.2. Наявність описової складової структури та наповнення курсу, для ознайомлення студентами;

2. Загальний опис (призначення системи).

2.1. Даний проект забезпечує користувача широкими можливостями менеджменту курсів та ведення статистики.

2.1.1. Користувацькі інтерфейси: наявність на ПК з'єднання Інтернет, наявність браузерів класу Google Chrome, Opera, Mozilla Firefox, Safari.

2.1.2. Апаратні інтерфейси:

2.1.2.1. ПК користувача:

- а) Процесор: Intel Celeron G1840 2.8GHz або Amd A8-6500 3.5GHz;
- б) Графічний прискорювач: Nvidia GeForce 9600gt або Amd hd 4670;
- в) Оперативна пам'ять об'ємом: 1536mb;

2.1.2.2. Смартфон користувача:

- а) Процесор частотою не нижче 1.2GHz;
- б) Графічний прискорювач Mali-400 MP2;
- в) Оперативна пам'ять об'ємом: 768mb.

2.1.2.3. Сервер:

- а) Процесор: Intel Xeon E2697 2.7GHz або AMD Opteron 6176 2.3GHz;
- б) Графічний прискорювач: GeForce 7600gt або аналогічні;
- в) Оперативна пам'ять об'ємом: 32Gb DDR3 1333 в чотирьох каналному режимі.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

г) Накопичувачі: 2TB Gb HDD (RAID 4).

2.1.3. Програмні інтерфейси: Операційна система Ubuntu Server 18.04 у базовій функціональності, PHP 7 або вище, MySQL 8 або вище.

2.1.4. Комутаційні інтерфейси: сервер (наявність з'єднання Інтернет 100Гбіт/с), користувач (наявність з'єднання Інтернет (ADSL, Ethernet і т. д.) 128 Kbit і вище).

2.1.5. Обмеження по пам'яті: розмір сторінки до 190 Кбайт, при наявності з'єднання Інтернет 512 Kbit — час завантаження сторінки 2,9 с.

2.1.6. Операції: можливість розробки курсів, можливість огляду курсів, можливість проходження курсів.

2.1.7. Вимоги до адаптації в середовищі користувача: наявність інтернет браузера типу Google Chrome, Opera, Mozilla Firefox чи Safari.

2.2. Функції продукту:

- а) Реклама платформи для неформального навчання в мережі Інтернет.
- б) Можливість онлайн навчання.

1.2 Аналіз можливих рішень, щодо реалізації навчальної платформи Wisdom in Knowledge

1.2.1 Аналіз технологій реалізації платформи

Веб-сайт – це система електронних документів доступ до яких здійснюється за допомогою унікальної адреси, яка називається доменим ім'ям [3]. Домене ім'я веб-сайту відображається в URL («Uniform Resource Locator») полі браузера.

Даний проект розміщується під адресою «<https://wike.ml>». При переході на сайт браузер завантажує графічний інтерфейс, який розроблений за допомогою «HTML».

HTML – це мова гіпертекстової розмітки, яка надає документу певної структури, після чого даній структурі задається стилізація за допомогою «CSS» [4].

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

CSS – каскадні таблиці стилів [5]. На основі останніх розробляється статичний сайт. Дана процедура вимагає розробки окремих документів для кожної сторінки веб-сайту. Вирішенням даної ситуації є розробка динамічного веб-сайту на основі використання шаблонів, які заповнюються на основі запиту користувача. Однією з мов програмування, яка має нативну підтримку розробки шаблонів є PHP. PHP – це препроцесор гіпертексту, який використовується для генерації динамічного контенту в документі. За допомогою даної мови сценаріїв на сервері створюються сторінки на основі шаблонів, які відправляються користувачу в залежності від потреб контролера.

Створення, видалення, редагування записів здійснюється за допомогою системи керування базою даних (СКБД). Базою даних виступає MariaDB, яка є відгалуженням від СКБД MySQL.

Взаємодія з СКБД MariaDB, реалізується посередством інтерфейсів мови програмування PHP.

Взаємодія адміністратора з базою даних без втручання в код програми здійснюється за допомогою веб-додатку phpMyAdmin. phpMyAdmin – дозволяє адмініструвати СКБД через браузер та надає можливість ініціювати команди SQL.

На основі вище описаних технологій будується додаток за використання шаблону проектування MVC.

MVC передбачає розподіл обов'язків між частинами «Front end» та «Back end», на наступні:

- model (модель) – містить бізнес логіку, яка відповідає за зміну даних бд та вигляду;
- view (вигляд) – відповідає за активацію контролерів та відображення даних отриманих від контролера за допомогою моделі;
- controller (контролер) – реагує на запити вигляду та ініціалізує виконання логічних ланцюгів моделі.

З точки зору хостингів ПЗ відноситься до категорії сервісів, які можуть бути організовані на одній фізичній машині (сервері) у вигляді модульної структури.

До переліку необхідних сервісів для забезпечення реалізації програмно-апаратної платформи входять:

- ос;
- веб-сервер (HTTPS server);
- файловий сервер (FTP server);
- захищена оболонка (Secure Shell – SSH);
- база даних;
- ядро веб-сайту.

Важливим критерієм при виборі технологій є баланс між споживанням ресурсів сервера та складністю реалізації.

Операційна система є фундаментом для інших технологій. Саме за допомогою неї відбувається комунікація між програмним та апаратним забезпеченням.

Windows server – комерційна ОС, з закритим вихідним кодом та своїм стеком технологій такими як Internet Information Server (IIS), ASP.NET, C#, Microsoft SQL Server (MSSQL).

Linux – UNIX подібна ОС з відкритим вихідним кодом, яка є повністю безкоштовною. Базовими веб-серверами для linux є кросплатформові Apache або Nginx, які підтримують різні ОС. За реалізацію веб-додатків часто використовують Hypertext Preprocessor (PHP), JavaScript та Node js, а в ролі бази даних виступає реляційна MySQL.

Відповідно для Windows більш оптимальними по продуктивності виступатимуть технології з його комерційного стеку, а для Linux, технології з вільним доступом.

З вище описаних технологій формуються аббревіатури від назв використовуваних технологій.

LAMP – Linux, Apache, MySQL, PHP.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докum.	Підпис	Дата		16



WIMP – Windows, IIS, MySQL, PHP.

Даний проект будується на основі стеку LAMP.

### 1.2.2 Аналіз можливості використання хмарних сервісів

Хмарні сервіси надають можливість утворити одну логічну систему використовуючи віддалені один від одного фізичні сервери. Іншою перевагою хмарних сервісів є використання гіпервізорів.

Гіпервізор – програмне забезпечення, для моніторингу та керування віртуальними машинами, з прямим доступом до апаратного забезпечення без використання ОС.

Гіпервізор надає змогу виділити необхідну кількість ресурсів для віртуальної машини в залежності від тарифу хостинг провайдера.

Віртуальні машини, які знаходяться на системах з гіпервізорами поділяються на VDS та VPS.

Віртуальний виділений сервер – підтримує апаратну віртуалізацію.

Віртуальний приватний сервер – підтримує віртуалізацію на рівні операційної системи.

Перевагою віртуальних машин над фізичним сервером є масштабованість ресурсів. Перевагою над віртуальним хостингом є повний доступ до налаштувань системи з root правами, що дає змогу встановити LAMP.

Вартість VDS та VPS коливається в залежності від хостинг провайдера та обмежень, які накладаються на обліковий запис замовника.

До переліку функцій, які впливають на вартість належать:

- кількість розміщуваних веб-сайтів;
- простір накопичувача, для збереження файлів;
- пропускна здатність мережі;
- потужність обчислювальних можливостей, які розподілятимуться між веб-сайтами розміщеними на хостингу;

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ док.ум.	Підпис	Дата		17

- сертифікати SSL, та кількість доменів для яких вони можуть бути застосовані;
- можливість розсилки пошти;
- можливість розміщення CMS систем;
- часові вікна підтримки. Для найдорожщих хостингів це 24/7.

Також дані хостинги в залежності від вартості можуть накладати обмеження на розроблюване ПЗ, а саме організацію онлайн чатів, які потребують відкриття сокетів.

Перевагами даних хостингів є можливість отримання безкоштовної публічної статичної IP-адреси та домену. Обмеженням виступає організація піддоменів в залежності до тарифу.

### 1.2.3 Аналіз хостингів для публікації платформи

В межах даного стенду програмно-апаратна платформа будується на основі фізичного сервера з використанням стеку LAMP та сервісів описаних в розділі 1.2.1. Для забезпечення візуального доступу до бази даних встановлюється програмне забезпечення phpMyAdmin.

Дане рішення немає обмежень в використанні апаратних ресурсів програмним забезпеченням.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

## РОЗДІЛ 2 ПРОЕКТНА ЧАСТИНА

### 2.1 Розробка концептуальної архітектури програмно-апаратної навчальної платформи Wisdom in Knowledge

Клієнт-серверна архітектура на якій функціонує стек LAMP комунікує з клієнтом по протоколу HTTP. Протокол гіпертекстової пересилки працює на 80 порті та є незахищеним, що дає зловмисникам змогу перехопити дані. Вирішенням даної проблеми є HTTPS, який є захищеним і працює на 443 порті. Захищеність даного протоколу ґрунтується на основі SSL.

Принцип роботи будь-якого протоколу шифрування базується на використанні ключів. Дані ключі містяться в SSL-сертифікатах, які є персональними. Етапи встановлення безпечного з'єднання між браузером та сервером:

- клієнт вводить в браузері домене ім'я;
- сервер відправляє інформацію про SSL-сертифікат та публічний ключ;
- браузер перевіряє інформацію, генерує сеансовий ключ, шифрує його публічним ключем і відправляє до сервера;
- сервер дешифрує сеансовий ключ;
- безпечне з'єднання встановлено.

Вище описані етапи встановлення безпечного з'єднання ініціюються для кожного переходу на сайт користувачем.

Схему функціонування стеку технологій LAMP наведено в графічній частині LAMP Client-Server.

					<b>КС КРБ 123.184.00.00 ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>				
Розроб.		Цимбалістий В.О.			<b>Проектна частина</b>	<b>Літ.</b>	<b>Арк.</b>	<b>Акрушів</b>
Перевір.		Яцишин В.В.					19	19
Рецензент.						<b>ТНТУ, каф. КС, гр. СІс-44</b>		
Н. Контр.		Тиш С.В.						
Затверд.		Осухівська Г.М.						

Доступ до сервера з мережі інтернет відбувається за допомогою технології NAT, яку реалізовано на маршрутизаторі. Дана технології забезпечує підміну приватної IP-адреси пристрою на публічну. Комунікація клієнт-сервер в площині мережі інтернет наведено в графічній частині мережева взаємодія між клієнтом та сервером.

Узагальнені етапи комунікації клієнт-сервер в площині мережі інтернет:

- клієнт вводить ім'я веб-сайту (домене ім'я) в стрічці адреси (URL) браузера або пошуковій системі. Дана дія призводить до формування запиту в мережу інтернет, а саме DNS сервера;
- dns сервер виконує пошук IP-адреси пристрою та порта сервісу, які є відповідними для доменого імені надісланого клієнтом, щоб надіслати їх клієнту;
- клієнт отримує публічну IP-адресу та порт від DNS сервера, після чого використовує їх для звернення до маршрутизатора на якому встановлена IP-адреса;
- маршрутизатор отримує запит від клієнта на порт, якому відповідає веб-сервер, після чого виконує переадресацію на даний порт за допомогою технології NAT;
- сервер отримує запит на порт, відповідний веб-серверу після чого ініціює HTTPS з'єднання з клієнтом через маршрутизатор з публічною IP-адресою.

Вище описані етапи комунікації справджуються для усіх видів хостингів.

Віддалений доступ до необхідних файлів розміщених в ОС, що використовуються веб-сервером для організації веб-сайту здійснюється за допомогою FTP сервісу. Файловий сервер працює на 21 порту та організовується на тому ж сервері, що й веб-сервер.

Здійснення віддалених конфігурувань ОС на сервері, реалізуються за допомогою SSH, що функціонує на 22 порту.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

## 2.2 Обґрунтування вибору апаратного забезпечення проектного комп'ютерного засобу

### 2.2.1 Розрахунок параметрів серверної частини для функціонування навчальної платформи

Вибір апаратного забезпечення, ґрунтується на основі вимог ПЗ та навантаженням трафіком користувачів, для якого виконується ряд розрахунків.

Необхідно проаналізувати потік клієнтів, які будуть виконувати запити до сервера, тобто переходитимуть на сайт і виконуватимуть ті чи інші дії.

Даний аналіз відноситься до SEO, що відповідає за пошукову оптимізацію, тим самим просуваючи сайт в пошукових системах для збільшення клієнтської бази. Найпростішим, що потрібно виконати, це аналіз запитів по ключових словах. Для слів «онлайн курси», що відповідає тематиці сайту відповідають перші 100 подібних ключових слів з пошукової системи Google, що зображено в табл. 2.1.

Таблиця 2.1 – Частота запитів по ключових словах в Google для України

Ключове слово	Частота
Едера онлайн курси	7200
Едера онлайн курси вхід	2500
Едера онлайн курси увійти	1700
Безкоштовні онлайн курси з сертифікатом	1100
Онлайн курси для вчителів	900
Безкоштовні онлайн курси з сертифікатом для вчителів	800
Усі інші результати	8140
Загалом	22340

Дана частота відповідає 100 ключовим словам за тиждень. При правильно розробленому семантичному ядру, яке відображає зміст сайту для пошукових систем будується рейтингова таблиця. Дана таблиця визначає позиції і вірогідність з якою користувач по ній перейде на сайт. Позиції в пошуковій системі наведено в табл. 2.2.

Таблиця 2.2 – Позиції в пошуковій системі Google

Місце в пошуку	Найменша вірогідність	Найбільша вірогідність
Топ 1	20%	50%
Топ 2	17%	45%
Топ 3	10%	20%
Топ 4	6%	9%
Топ 5, Топ 6	4%	7%
Топ 7	3%	6%
Топ 8, Топ 9, Топ 10	2%	4%

Дані, що наведені в табл. 2.2 відображають вірогідність з якою користувач відкріє сайт відповідно до місця у пошуковій системі. На дану дію також впливають «сніпети», що відображають вміст сайту в пошуковій системі. Якщо сніпет організований таким чином, що повністю відображає дані сайту, які відповідають запиту користувача то вірогідність для «Топ 1» буде прагнути до «50%». Для достовірності обирається середина, тобто 30 – 35. Приклад сніпету зображено на рис 2.1.

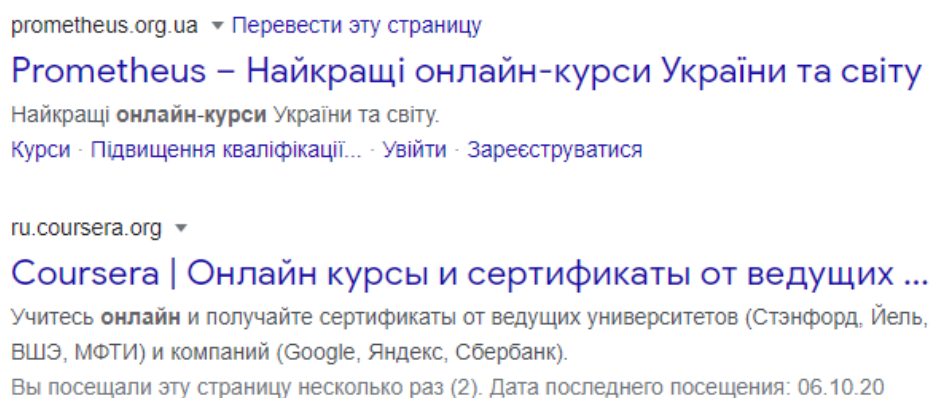


Рисунок 2.1 – Сніпети в пошуковій системі Google

У випадку розроблюваного сайту буде розраховуватись найгірший розклад, коли симулюється наступна ситуація:

- сайт не був опрацьований пошуковими роботами, перший тиждень після публікації, через можливість спаму;
- на даний сайт немає посилась з інших сайтів (реклама).

Згідно табл. 2.1 та 2.2 будується формула розрахунку трафіку користувачів, які переходитимуть на сайт:

Розрахунок кількості відвідувань на сайті за тиждень  $KK_T$  проводиться за такою формулою:

$$KK_T = \frac{ЧЗ_{кк}}{КС_к} * В, \quad (2.1)$$

де  $ЧЗ_{кк}$  – загальна частота запитів в пошуковій системі відповідно кількості ключових слів;

$КС_к$  – Кількість ключових слів;

$В$  – найменша вірогідність переходу на сайт відносно позиції в пошуковій системі.

Відповідно до формули (2.1) задаються наступні дані з табл. 2.1 та 2.2.  $ЧЗ_{кк}=22340$ ,  $КС_к=100$ ,  $В=2$ . В результаті отримано  $KK_T=447$  клієнтів за тиждень, тобто в місяць потрібно буде опрацювати 1920 клієнтів. Середня кількість клієнтів в день становить 64 підключення.

На основі отриманих даних про кількість клієнтів, розраховується характеристики процесора двома параметрами, якого є частота та кількість потоків.

Необхідна частота процесора відштовхується від кількох аспектів до яких входять:

- мінімальні системні вимоги до ОС Linux Ubuntu Server 18.04.4, для якого необхідно 1ГГц;
- кожен процес PHP працює на одному ядрі, у відповідь на запит одного користувача, але паралельно працює кілька підпроцесів Apache (по одному для кожного запиту, до декількох десятків або сотень в залежності від конфігурації). Відповідно чим більша тактова частота тим швидше будуть виконуватись процеси.

Відповідно оптимальним рішенням є процесори з тактовою частотою від 2ГГц.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ док.ум.	Підпис	Дата		23

Кількість ядер процесора впливає на кількість одночасно можливих сесій за секунду та кількість запитів на сторінку за секунду, які може опрацювати веб-сервер використовуючи РНР.

Розрахунок максимальної кількості запитів на сторінку в секунду, які може опрацювати сервер проводиться за наступною формулою:

$$MЗ_c = \frac{K_{яп}}{ЧЗ_c}, \quad (2.2)$$

де  $MЗ_c$  – максимальна кількість запитів на сторінку в секунду;

$K_{яп}$  – кількість ядер процесора;

$ЧЗ_c$  – середній час завантаження для сторінки (секунди).

Під час розрахунку відповідно формул (2.2) та (2.1) будується таблиця з даними середнього часу завантаження (секунд) сторінки та середнього об'єму сторінки (мб). Зібрані дані наведено в табл. 2.3.

Таблиця 2.3 – Середні показники для сторінки

Сторінка	Час завантаження документу (с)	Повний об'єм (мб)
Головна	0.42	6.9
Кабінет користувача	0.3	0.5
Курси	0.48	4.4
Передперегляд курсу	0.45	0.7
Курс	0.37	3.6
Середнє значення	0.4	3.22

Відповідно до формули (2.2) задаються наступні дані з табл. 2.3.  $K_{яп} = 2$ ,  $ЧЗ_c = 0.4$ . В результаті отримано  $MЗ_c = 5$  запитів на сторінку в секунду може опрацювати процесор з двома ядрами.

Дана метрика є важливою для масштабованості веб-сайту, що зазвичай зводиться до межі можливостей центрального процесора. Середній час запиту сторінки фіксує як час, який споживає РНР, так і час, який база даних використовує для запитів.



Методом підбору визначено, що необхідна кількість ядер процесора для сервера становить 12 ядер, що дає змогу опрацювати 30 запитів в секунду.

На основі результатів розрахованих за формулою (2.2) розраховується, максимальна кількість одночасних користувачів на веб-сайті, якщо середня тривалість сеансу для користувача 5 хвилин, а кількість відкритих сторінок 2.82:

$$K_{XB} = (TC_c \div KC_c) \div 60, \quad (2.3)$$

$$КОК = МЗ_c * 60 * K_{XB}, \quad (2.4)$$

де  $K_{XB}$  – кількість кліків здійснених користувачем за хвилину;

$TC_c$  – середня тривалість сеансу в секундах;

$KC_c$  – середня кількість відкритих сторінок за сеанс;

КОК – кількість одночасних користувачів на веб-сайті;

$МЗ_c$  – максимальна кількість запитів в секунду;

60 – часовий проміжок в якому користувачі здійснюють кліки (2.4).

Відповідно до формул (2.4) та (2.5) розрахунок проводиться з наступними даними  $TC_c = 300$ ,  $KC_c = 2.82$ ,  $МЗ_c = 30$ . В результаті КОК = 3186 користувачам в часовому вікні тривалістю 1 хвилина.

Об'єм оперативної пам'яті сервера розраховується за наступною формулою:

$$ОЗП_{мб} = КОК * C_p, \quad (2.5)$$

де  $ОЗП_{мб}$  – об'єм ОЗП (Оперативний запам'ятовуючий пристрій) в мегабайтах;

КОК – кількість одночасних користувачів на веб-сайті;

$C_p$  розмір найбільшої сторінки веб-сайту (мб).

Відповідно до формули (2.5) обчислюється необхідний об'єм оперативної пам'яті з використанням наступних даних КОК = 1062,  $C_p = 6,9$ . В результаті

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

розрахунку  $OЗП_{мб} = 21\ 983$ , що рівне 22 гігабайтам (гб) ОЗП. Також потрібно врахувати ПЗ встановлене на сервері, що в сумі становитиме 26гб оперативної пам'яті.

Об'єм дискового простору поділяється на системні потреби для функціонування сервера та простір, який виділено для функціонування веб-сайту.

Системний розділ розраховується на основі системних характеристик ПЗ, що загалом становитиме до 5гб та 2гб для ОС. В результаті для сервера без навантаження необхідно зарезервувати 10гб для ПЗ з ОС та 26гб для резервного розділу, що рівний об'єму оперативної пам'яті.

Інший простір розраховується за наступною формулою:

$$ДП_{мб} = (1 * ЗКК) + (300 * ЗКК), \quad (2.6)$$

де  $ДП_{мб}$  – об'єм дискового простору для забезпечення функціонування веб-сайту.

$ЗКК$  – загальна кількість користувачів в системі;

1 – розмір запису в базі даних для одного користувача (мб);

300 – простір виділений для файлів одного користувача (мб).

Відповідно до формули (2.6) обчислюється необхідний об'єм дискового простору з такими даними  $ЗКК = 3000$ . В результаті  $ДП_{мб} = 903\ 000$  мегабайт (мб), необхідно для забезпечення функціонування 3000 клієнтів.

Вимоги до провайдера містять ширину каналу інтернет з'єднання, які необхідно забезпечити для коректної роботи веб-сайту.

Ширина каналу обчислюється на основі спожитої кількості даних (мб), за день по такій формулі:

$$ШК = (C_{pc} * KC_c * CC) * 1.25, \quad (2.7)$$

де ШК – ширина каналу;

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

1.25 – зарезервований об’єм необхідний для непередбачених ситуацій;

$C_{pc}$  – середній розмір сторінки;

$KC_c$  – середня кількість відкритих сторінок за сеанс;

$CC$  – середня кількість сесій в день.

Відповідно до формули (2.7) розрахунок проводиться з наступними даними  $C_{pc} = 3.22$ ,  $KC_c = 2.82$ ,  $CC = 64$ . В результаті ШК = 726, що рівне 0.7 гігабайт за день та 22 гігабайти за 30 днів.

## 2.2.2 Вибір компонентів апаратного забезпечення для функціонування клієнтської сторони навчальної платформи

Апаратне забезпечення на стороні клієнта обирається на основі вимог описаних стандартом IEEE 830 в розділі 1.1.2.

## 2.3 Обґрунтування вибору програмного забезпечення проєктованого комп’ютерного засобу

### 2.3.1 Visual Studio Code

Розробка програмної частини, яка відноситься безпосередньо до рівня аплікації, а саме «Front End» та «Back End» об’єктів реалізується за допомогою середовищ розробки до яких відносяться текстові редактори.

За середовище розробки обрано Visual Studio Code, з ряду причин. Даний текстовий редактор позиціонує себе як «легкий» редактор коду для кросплатформової розробки.

До можливостей редактора належать:

- налаштування тем оформлення середовища;
- встановлення розширень (плагінів). Розширення забезпечують гнучкість середовища, не обмежуючи його лише конкретною сферою діяльності;

					КС КРБ 123.184.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

- конфігурування гарячих клавіш за допомогою конфігураційних файлів;
- окреме налаштування, будь-якого встановленого плагіну;
- будь-який плагін можна увімкнути та вимкнути без видалення.

До переваг середовища належать:

- встановлення плагінів в два кліки;
- нативна підтримка мов програмування для розробки веб-додатків;
- вбудована довідка для мов програмування, які відносяться до «Back End» частини, веб додатків;
- нативна підтримка інструментів інтеграції з системою контролю версій GIT та GitHub;
- вбудований плагін «Emmet»;
- перемикання між кількома одночасно активними терміналами.

### 2.3.2 MySQL

Базою даних обрано MariaDB, яка є відгалуженням від СКБД MySQL, з покращеною продуктивністю. MariaDB відноситься до реляційних баз даних, що базуються на наборі даних з зумовленими зв'язками між ними. Ці дані організовані у вигляді набору таблиць, які складаються з стовпців і рядків. У таблицях зберігається інформація про об'єкти, що представлені в базі даних. В стовпцях зберігається тип даних, а в кожній комірці – значення атрибуту.

Кожна стрічка таблиці являє собою набір пов'язаних значень, що відноситься до одного об'єкту або сутності.

Кожен рядок в таблиці може бути позначений унікальним ідентифікатором, який називається первинним ключем, а рядки з декількох таблиць можуть бути пов'язані за допомогою зовнішніх ключів. До цих даних можна отримати доступ багатьма способами, і при цьому реорганізувати таблиці БД не потрібно.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

### 2.3.3 JavaScript

JavaScript – це мова програмування для розробки скриптів, що виконуються стороні клієнта за допомогою вбудованої в браузер «віртуальної машини JavaScript». JavaScript забезпечує:

- опрацювання ініціалізованих клієнтом дій без відправлення запитів на сервер;
- зміну DOM елементів в інтерфейсі користувача «на льоту»;
- виконання обміну даних з сервером без перезавантаження сторінки, що реалізується за допомогою технології «аїах».

Метою застосування JavaScript в даному проєкті є реалізація обробки даних форми реєстрації користувача, за допомогою технології «аїах».

## 2.4 Проєктування комп'ютерного засобу

### 2.4.1 Побудова use case діаграм

Архітектура веб-сайту, що складається з функціональних частин розробляється на основі окремих систем з акторами та прецедентами. Системи розробляються з урахуванням вимог стандарту IEEE830 описаним в розділі 1.1.2.

Веб-сайт складається з наступних систем:

- авторизація (рис 2.2);
- відображення курсів (рис 2.3);
- взаємодія з курсами по моделі студент-курс (рис 2.4);
- взаємодія з тестами по моделі студент-тест (рис 2.5);
- взаємодія з курсами по моделі викладач-курс (рис 2.6).

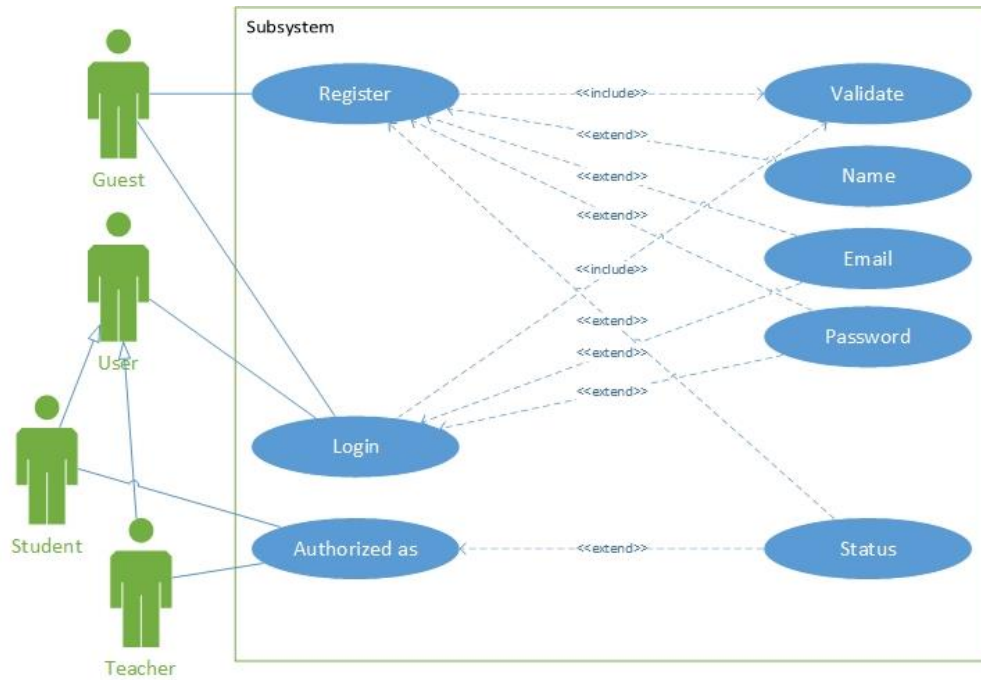


Рисунок 2.2 – Система авторизації

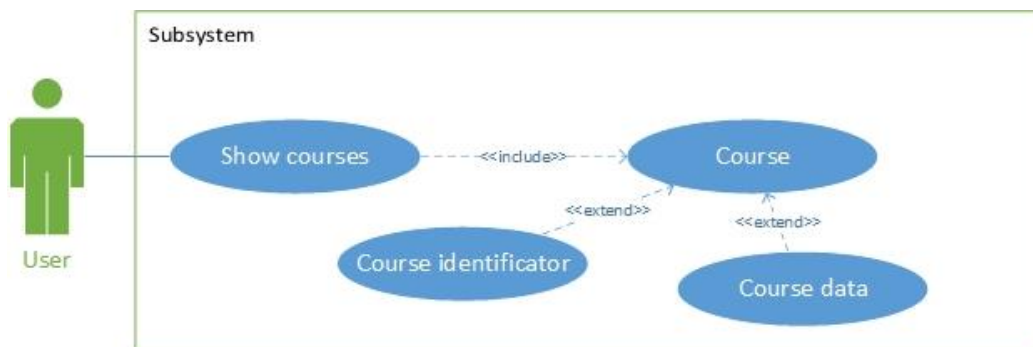


Рисунок 2.3 – Система відображення курсів

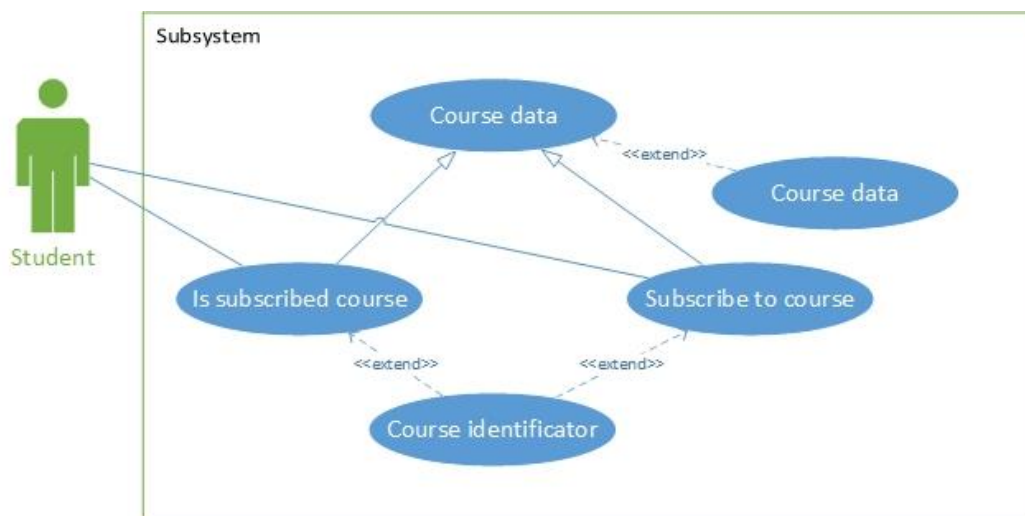


Рисунок 2.4 – Система взаємодії з курсами по моделі студент-курс

Змн.	Арк.	№ докум.	Підпис	Дата

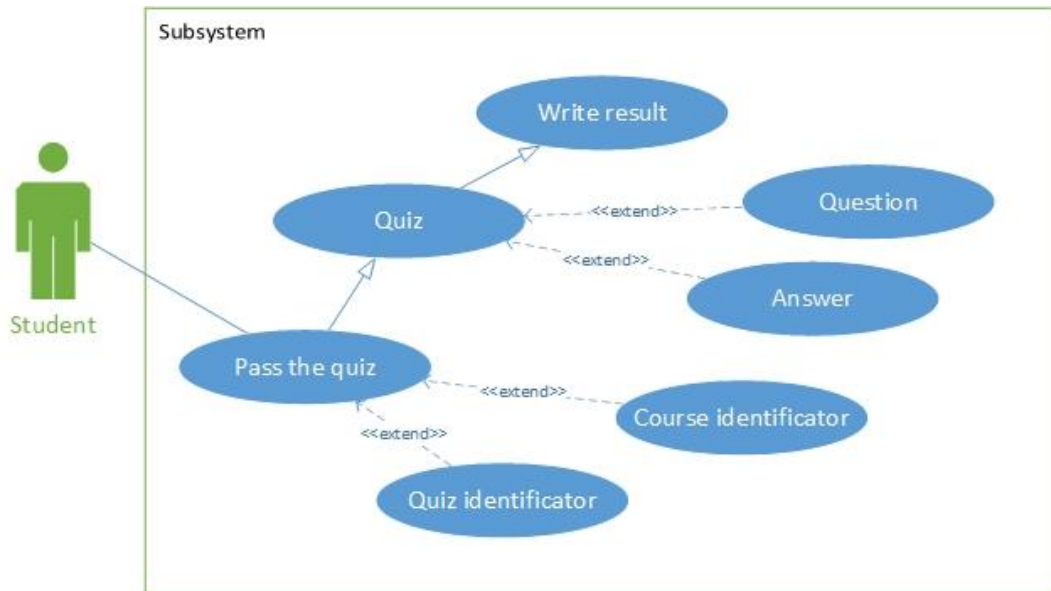


Рисунок 2.5 – Система взаємодії з тестами по моделі студент-тест

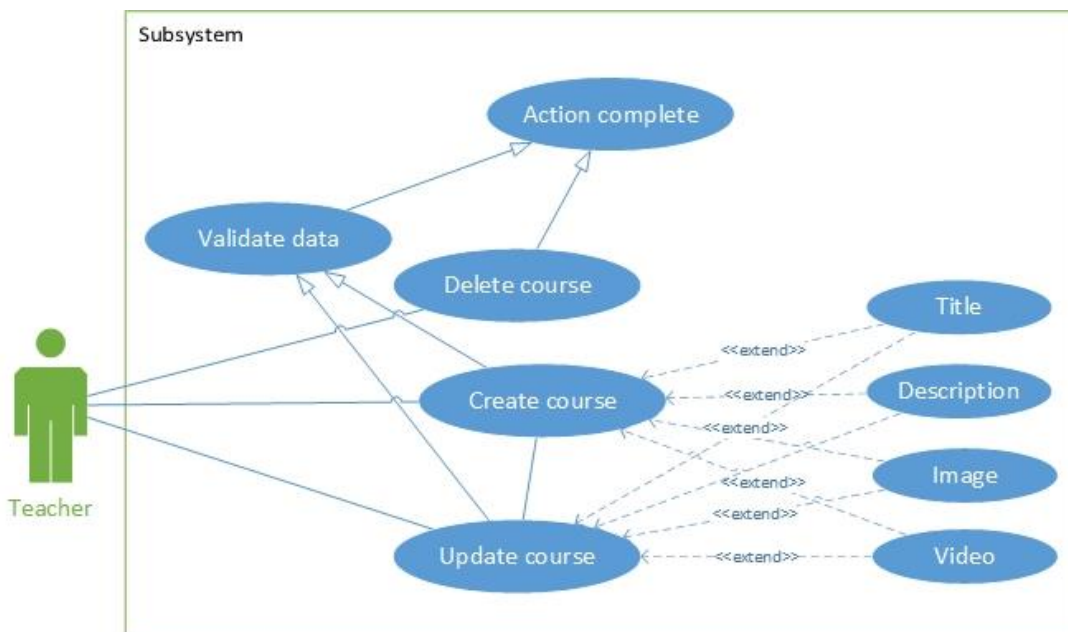


Рисунок 2.6 – Система взаємодії з курсами по моделі викладач-курс

#### 2.4.2 Проектування БД

Важливою складовою веб-сайту для забезпечення його функціонування є менеджмент інформації, який проводиться з використанням БД.

Проектування БД містить наступні етапи: концептуальне проектування; логічне проектування; фізичне проектування.

Змн.	Арк.	№ докум.	Підпис	Дата

Концептуальне проектування – містить в собі моделі (схеми). Концептуальна схема відображає концептуально важливі аспекти ПЗ, виключаючи будь-які аспекти зовнішнього або внутрішнього відображення даних. Ця модель не відображає конкретні потреби окремих користувачів та застосувань, натомість фіксує чим являється ПЗ в цілому, а не з точки зору інтересів або потреб користувачів. Концептуальне моделювання ПЗ здійснюється за допомогою найпопулярнішої графічної мови, яка оперує поняттями «сутність-атрибут-зв’язок» (Entity-Relationship language). Сутності та атрибути зображено в табл. 2.4.

Таблиця 2.4 – Сутності та атрибути

Сутність	Атрибути
User	ID_User ID_Course Name Email Password Status
Course	ID_Course ID_User Title Description Image Video ID_Quiz
Student_Course	ID_User ID_Course
Course_Status	Course_ID ID_User Status
Quiz	ID_Quiz ID_Course ID_Question ID_Answer
Question	ID_Question ID_Answer Text
Answer	ID_Answer ID_Question Text
Quiz_Result	ID_Quiz ID_User Result



На основі сутностей та атрибутів зображених в табл. 2.4, будується послідовність взаємозв'язків між ними.

– Користувач містить дані (ім'я, електронна адреса, пароль, статус), також даний користувач містить свій унікальний ідентифікатор та ідентифікатор курсу. Ідентифікатор курсу вказує на курси які можуть бути в користувача як студента.

– Курс містить дані, які відображають його вміст. Ідентифікатор користувача вказує на викладача, який створив даний курс. Ідентифікатор тесту вказує на чи існує належний курсу тест.

– Сутність «студент курс» відображає курси закріплені за користувачем який являється студентом.

– Сутність «статус курсу» відображає закінчені або активні курси студента.

– Сутність тест відображає належні йому питання з вірною відповіддю, та прив'язкою до курсу, який являється для нього батьківським.

– Сутність питання відображає зміст запитання та належні йому відповіді.

– Сутність відповідь відображає кількість відповідей для сутності запитання.

– Сутність «результат тесту» відображає успішність проходження тесту студентом.

Далі на основі концептуальної моделі, яка відображає сутності та атрибути відбувається логічне проектування, що містить ERD (Entity-Relationship Diagram).

Побудована ER-діаграма відповідно до «сутність відношення» зображена на рис. 2.7.

Фізичне проектування містить побудову таблиць з полями, що містять тип даних та індекс на основі ER-діаграми наведено в додатку Б.

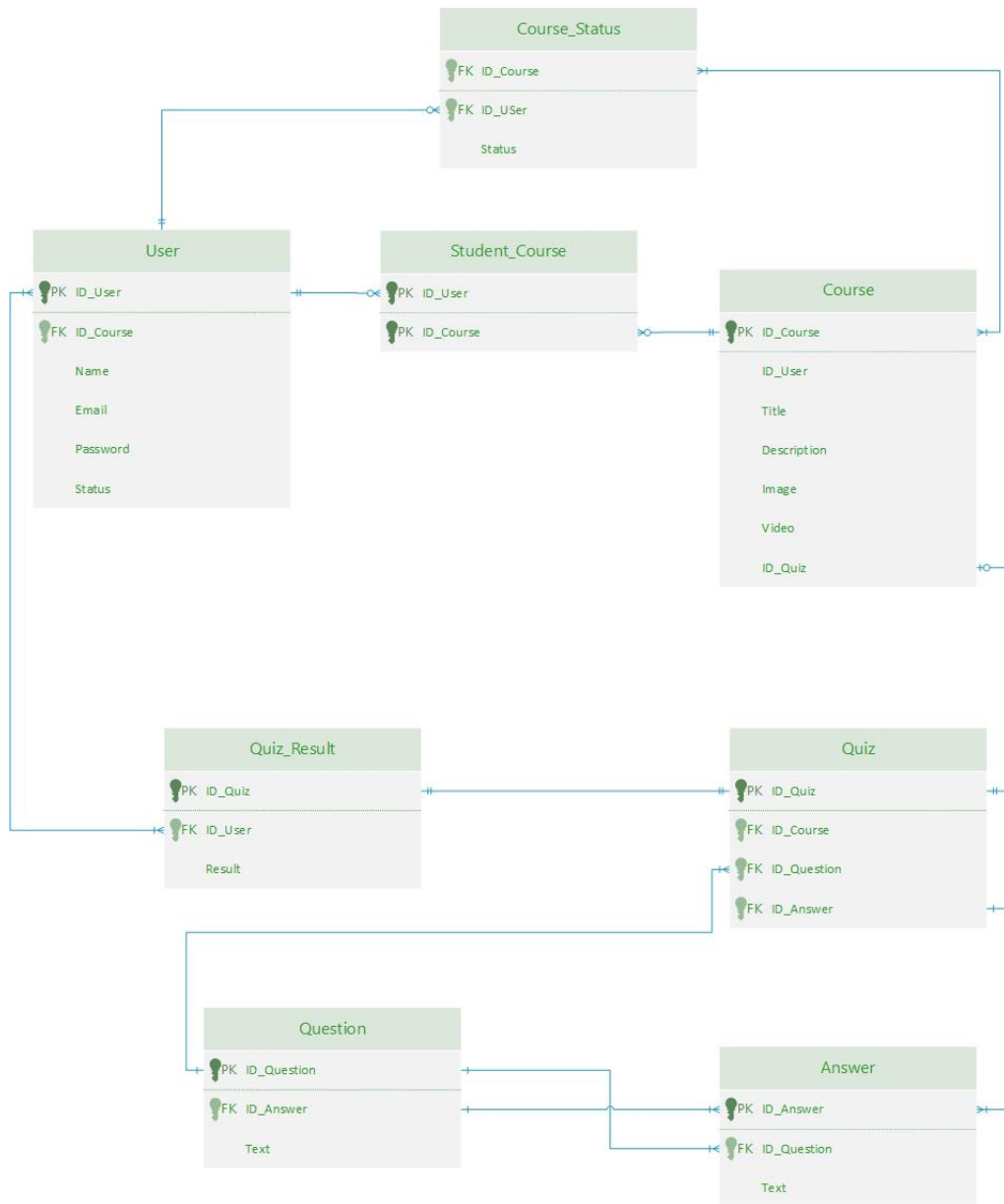


Рисунок 2.7 – ER діаграма

### 2.4.3 Побудова діаграм класів

Платформа навчання «Wisdom in Knowledge» будується на основі патерну MVC, який передбачає використання об'єктно-орієнтованого програмування (ООП) до проектування веб-додатку. ООП розробляється на основі UML діаграм класів, де сутності відображають класи, а атрибути поля та методи з різними типами доступу. На основі побудованих use-case діаграм (див. розділ 2.4.1) розробляються наступні діаграми класів:

- каркас mvc (рис.2.8);

- авторизація (рис.2.9);
- взаємодія не авторизованого користувача з курсами (2.10);
- взаємодія авторизованого користувача з курсами (2.11).

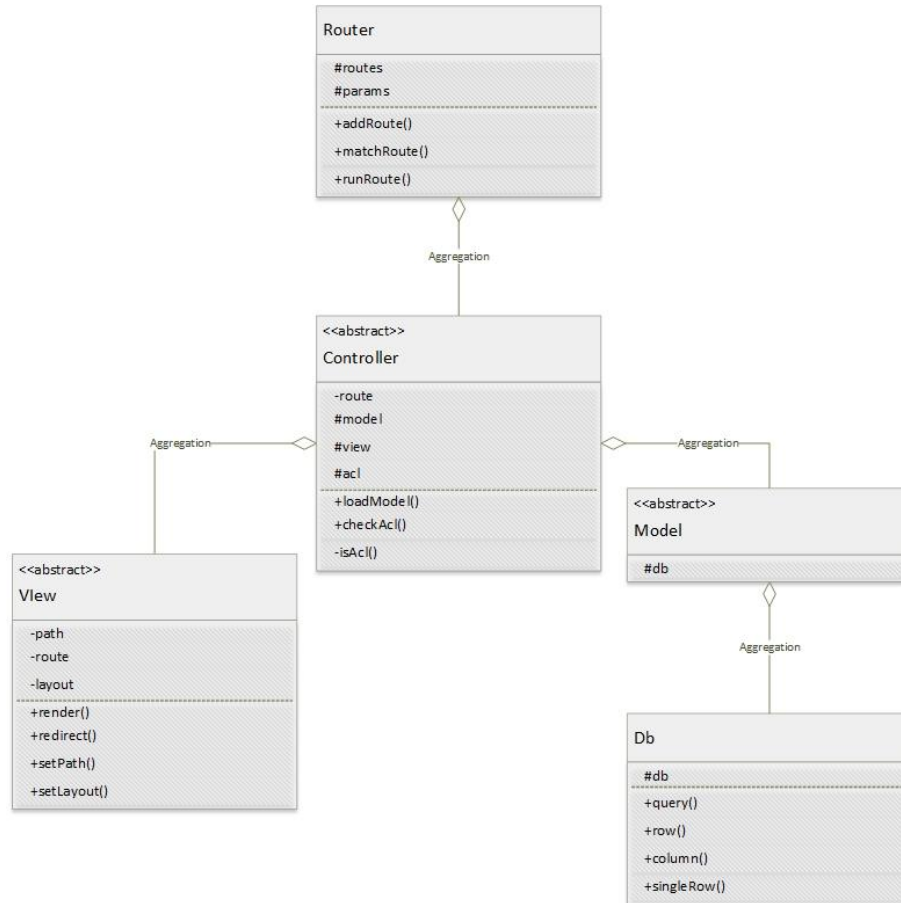


Рисунок 2.8 – Каркас MVC

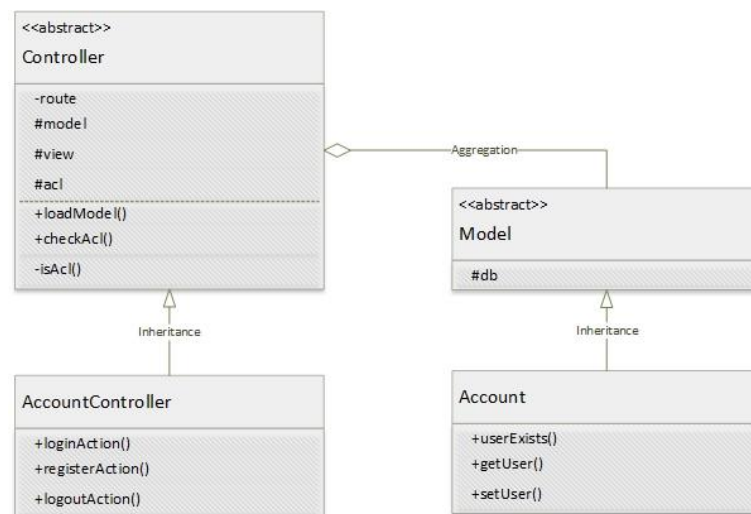


Рисунок 2.9 – Авторизація

Змн.	Арк.	№ докум.	Підпис	Дата

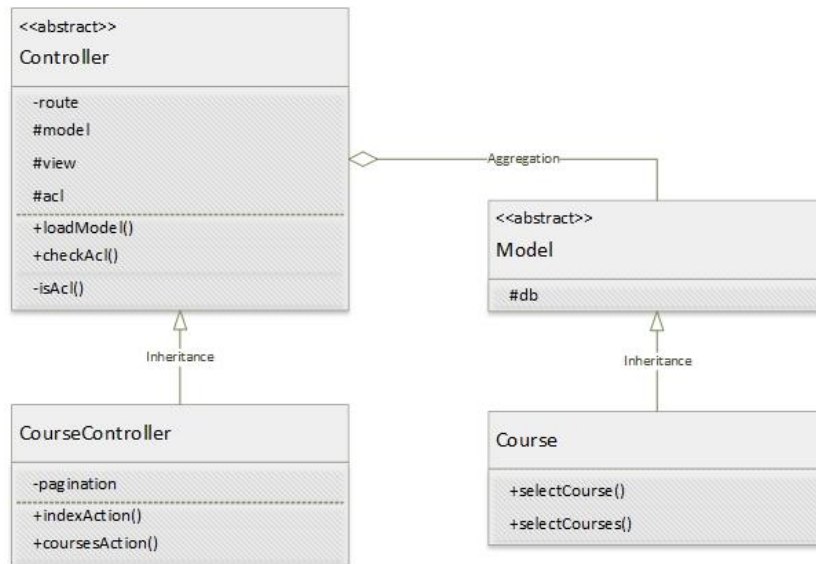


Рисунок 2.10 – Взаємодія не авторизованого користувача з курсами

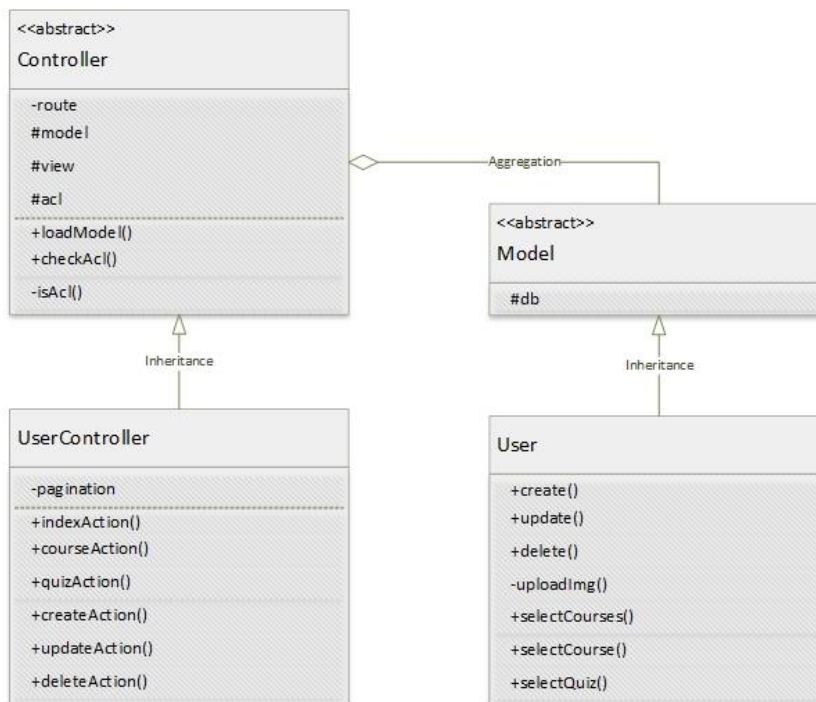


Рисунок 2.11 – Взаємодія авторизованого користувача з курсами

#### 2.4.4 Побудова діаграм послідовностей

На основі діаграм прецедентів розробляються діаграми послідовностей для більш детального опису логіки виконання наступних сценаріїв: авторизація в системі (рис. 2.12); взаємодія з тестами (рис. 2.13); взаємодія з курсами (рис. 2.14).

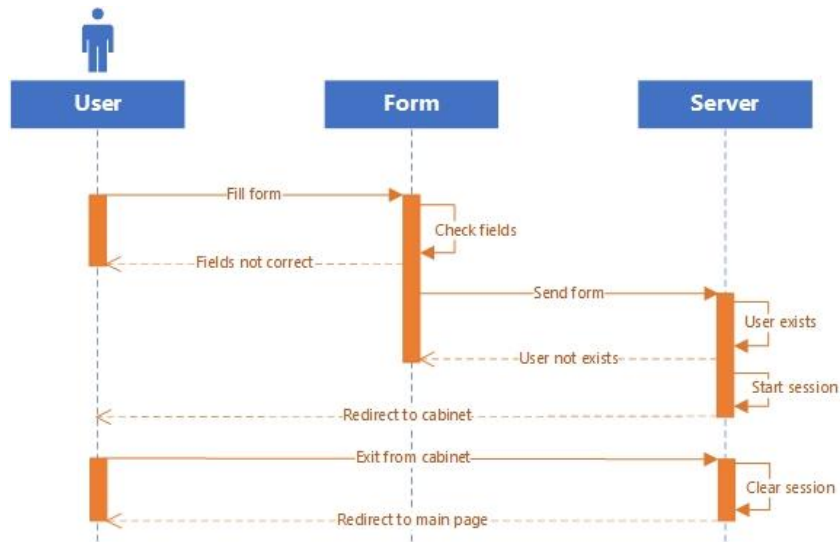


Рисунок 2.12 – Авторизація в системі

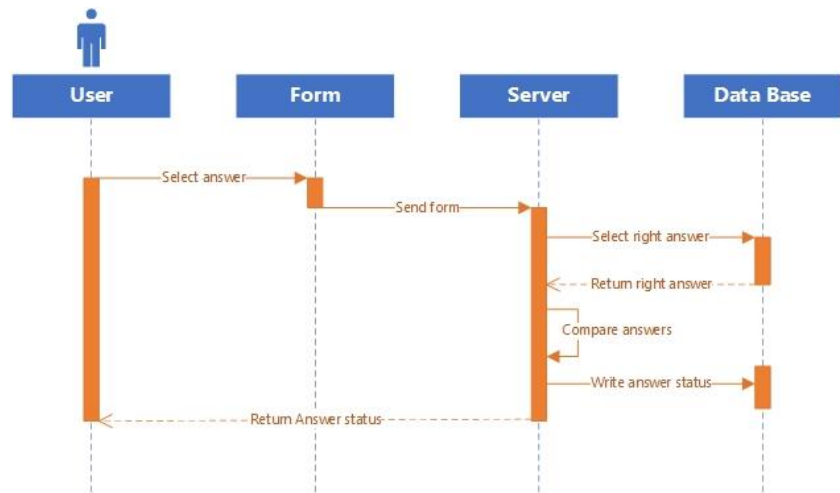


Рисунок 2.13 – Взаємодія з тестами

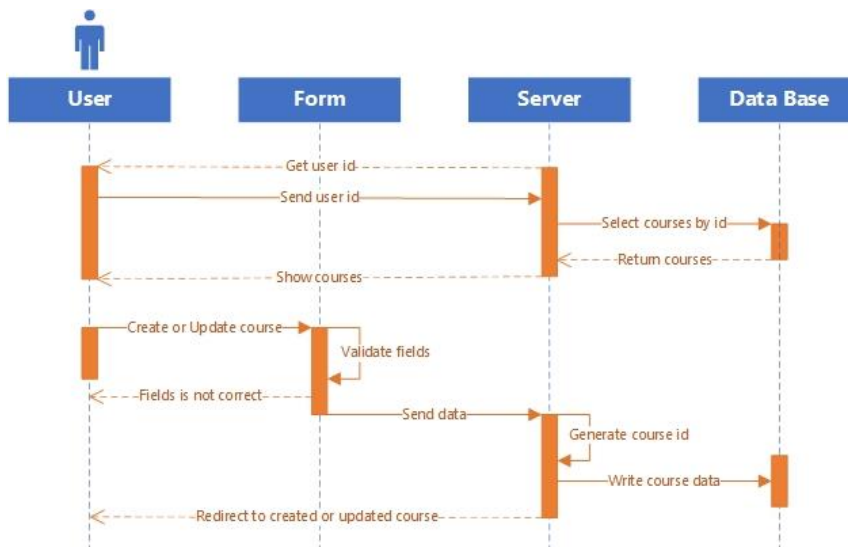


Рисунок 2.14 – Взаємодія з курсами

Змн.	Арк.	№ докум.	Підпис	Дата

## РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Реалізація навчальної платформи

#### 3.1.1 Налаштування сервісів

Забезпечення мережевої взаємодії Client-Server реалізується за допомогою технології NAT та переадресації на приватну IP-адресу сервера та порт на якому функціонує відповідний сервіс.

Встановлення міжмережевої взаємодії сервісів між маршрутизатором та сервером, налаштовується на маршрутизаторі.

В межах розроблюваного стенду задіяно маршрутизатор TP-Link, який передбачає можливість налаштування через графічний інтерфейс. По замовчуванню доступ до даного пристрою здійснюється за приватною IP-адресою «192.168.1.1», або публічною заданою провайдером, яку потрібно вказати в адресній стрічці (URL) вікна браузера. По замовчуванню ім'ям користувача та паролем для входу в панель керування маршрутизатором є «admin».

Налаштування переадресації портів проводиться в декілька етапів:

- відкрити вікно «розширені налаштування», вкладка NAT;
- обрати функцію «віртуальний сервер»;
- у вікні конфігурування віртуальних серверів потрібно задати статичну приватну IP-адресу сервера та порт на якій функціонує сервіс. В якості протоколу потрібно вказати TCP для забезпечення гарантованого транспортування пакетів.

Результат налаштувань зображено на рис. 3.1

					<b>КС КРБ 123.184.00.00 ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>				
Розроб.		Цимбалістий В.О.			<b>Практична частина</b>	<b>Літ.</b>	<b>Арк.</b>	<b>Акрушіє</b>
Перевір.		Яцишин В.В.					38	21
Рецензент						<b>ТНТУ, каф. КС, гр. СІс-44</b>		
Н. Контр.		Тиш Є.В.						
Затверд.		Осухівська Г.М.						

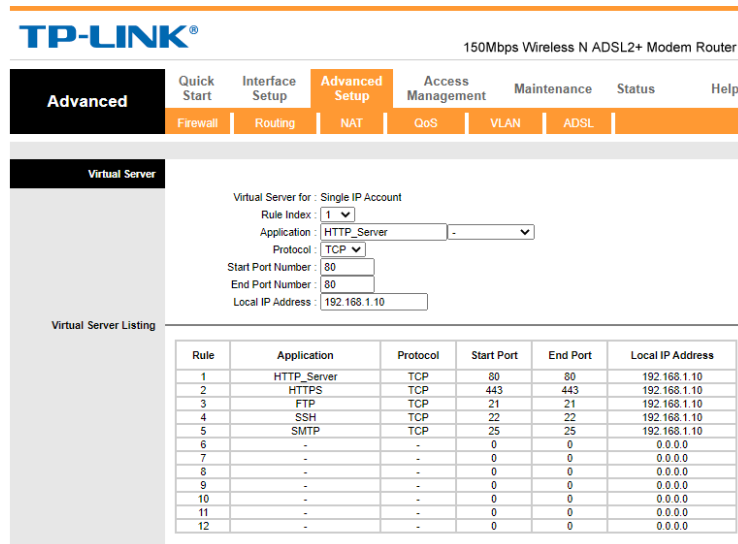


Рисунок 3.1 – Налаштування перенаправлень портів

Вище виконані дії забезпечують перенаправлення усіх пакетів для перелічених портів на відповідні порти сервера, що знаходиться за IP-адресою 192.168.1.10.

Необхідні сервіси для реалізації платформи комунікують з апаратною частиною сервера за посередництва ОС.

Встановлення операційної системи Ubuntu Server «18.04.4», містить такі етапи:

- встановлення приватної статичної IP-адреси;
- розподіл дискового простору;
- створення облікового запису.

По замовчуванню маршрутизатори містять вбудований функціонал DHCP сервера, який динамічно задає IP-адреси вузлам, що підключені до нього. Діапазон даних IP-адрес знаходиться в приватній мережі класу C «192.168.0.0» з маскою «255.255.0.0». Необхідність у статичній IP-адресі пов'язана з налаштуванням перенаправлення усіх зовнішніх запитів на внутрішню приватну адресу. Якщо IP-адреса сервера буде змінюватись, відповідно сервер не отримає запит від клієнта, що приходить з зовні, на раніше встановлену адресу та порт.

До налаштувань IP-адреси входить наступне:

- адреса підмережі та маска: «192.168.1.10»;

- адреса вузла: «192.168.1.10»;
- шлях за замовчуванням: «192.168.1.1»;
- dns сервер: «192.168.1.1».

Розподіл дискового простору здійснюється з врахуванням розрахунків зазначених в підрозділі 2.2.1. Розподіл дискового простору виглядає наступним чином:

- «/»: 10Гб – Розділ для ОС;
- «swp»: 32Гб – Розділ резервного простору;
- «/home»: 1Тб – Розмір домашнього каталогу для збереження усіх даних.

Налаштування облікового запису користувача містить наступні дані:

- ім'я: «dofamine»;
- ім'я сервера: «lamp»;
- ім'я користувача: «dofamine»;
- пароль: «Vitya777Vitya».

Наступним кроком є оновлення пакетів з репозиторіїв та їх встановлення за допомогою команди:

```
sudo apt-get update -y ; sudo apt-get upgrade
```

sudo – команда для виконання дій від імені супер користувача.

apt-get – команда для встановлення, оновлення та видалення програмних пакетів.

update – оновлення списку пакетів. Дана команда використовується разом з apt-get.

Прапорець «-у» призначений для автоматичного підтвердження дій пов'язаних з встановленнями пакетів програм.

upgrade – встановлення оновлених пакетів програм. Використовується аналогічно до update.

Встановлення openSSH сервера. Даний сервіс необхідний для віддаленого адміністрування сервера, по захищеному каналу зв'язку. Даний сервіс функціонує на 22 порту.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40



Перелік команд для встановлення та перевірки статусу openSSH наведено в рис. 3.2.

```
sudo apt-get install openssh-server -y
sudo systemctl status ssh.service
```

Рисунок 3.2 – Лістинг команд встановлення openSSH

Команда «install» в парі з «apt-get» перевіряє залежності пакетів та встановлює їх, після чого відбувається завантаження необхідних пакетів, розпакування, перевірка наявності попередніх версій і необхідності перезапису, якщо такі є.

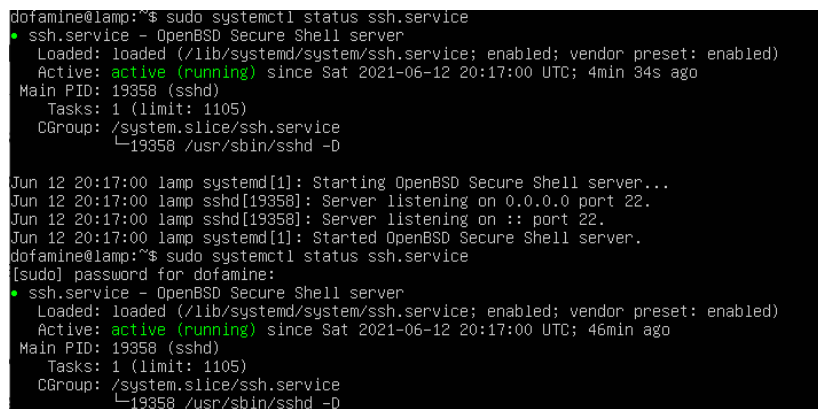
openssh-server – це назва пакету, який потрібно знайти в репозиторії, перевірити наявність залежностей, завантажити необхідні пакети та встановити їх.

systemctl – це набір інструментів для керування сервісами. Даний набір використовує ключові слова для взаємодії з сервісами.

status – команда з набору інструментів «systemctl», яка виводить статус вказаного сервісу.

ssh.service – назва поточного сервісу для «openssh-server».

Результат виконання команди відображення статусу «ssh» сервісу зображено на рис. 3.3.

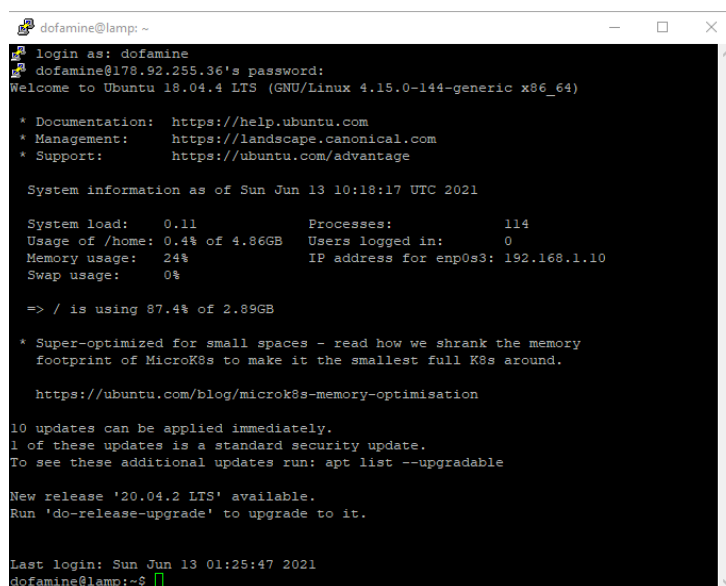


```
dofamine@lamp:~$ sudo systemctl status ssh.service
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2021-06-12 20:17:00 UTC; 4min 34s ago
  Main PID: 19358 (sshd)
  Tasks: 1 (limit: 1105)
  CGroup: /system.slice/ssh.service
          └─19358 /usr/sbin/sshd -D

Jun 12 20:17:00 lamp systemd[1]: Starting OpenBSD Secure Shell server...
Jun 12 20:17:00 lamp sshd[19358]: Server listening on 0.0.0.0 port 22.
Jun 12 20:17:00 lamp sshd[19358]: Server listening on :: port 22.
Jun 12 20:17:00 lamp systemd[1]: Started OpenBSD Secure Shell server.
dofamine@lamp:~$ sudo systemctl status ssh.service
[sudo] password for dofamine:
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2021-06-12 20:17:00 UTC; 46min ago
  Main PID: 19358 (sshd)
  Tasks: 1 (limit: 1105)
  CGroup: /system.slice/ssh.service
          └─19358 /usr/sbin/sshd -D
```

Рисунок 3.3 – Статус ssh сервісу

Віддалений доступ здійснюється за допомогою програми «PuTTY». Віддалений доступ здійснюється за публічною IP-адресою «178.92.255.36» та портом 22. Результат тесування зображено на рис. 3.4.



```
dofamine@lamp: ~
login as: dofamine
dofamine@178.92.255.36's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-144-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Jun 13 10:18:17 UTC 2021

System load:  0.11          Processes:      114
Usage of /home: 0.4% of 4.86GB   Users logged in:  0
Memory usage:  24%          IP address for enp0s3: 192.168.1.10
Swap usage:    0%

=> / is using 87.4% of 2.89GB

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

10 updates can be applied immediately.
1 of these updates is a standard security update.
To see these additional updates run: apt list --upgradable

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Jun 13 01:25:47 2021
dofamine@lamp:~$
```

Рисунок 3.4 – Віддалений доступ по ssh

Перед встановленням інших сервісів потрібно закрити усі порти окрім портів використовуваних сервісами. Дана дія виконується за допомогою міжмережевого екрану, який по замовчуванню вимкнений. Налаштування міжмережевого екрану ОС наведено в рис. 3.5.

```
sudo ufw enable
sudo ufw allow 20/tcp
sudo ufw allow 21/tcp
sudo ufw allow 22/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw allow 3306/tcp
```

Рисунок 3.5 – Лістинг налаштування міжмережевого екрану

Команда «ufw» вказує на використання міжмережевого екрану.

Команда «enable» активує міжмережевий екран.

					КС КРБ 123.184.00.00 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Команда «allow [port]» додає правило, яке дозволяє використання даних портів для протоколів «TCP/UDP».

Перелік портів у відповідності до сервісу: ftp data: «20»; ftp: 21; ssh: «22»; http: «80»; https: «443»; mySQL: «3306».

Встановлення «VsFTPD» сервісу необхідне для організації файлового сервісу, для транспортування великих об'ємів даних з можливістю використання програми FileZilla. Команда для встановлення наведена в лістингу 3.4.

### Лістинг 3.4 – Встановлення FTP сервісу

```
sudo apt-get install vsftpd -y
```

Підключення та вивантаження індексоного файлу в домашній каталог користувача на сервері зображено на рисунку 3.6.

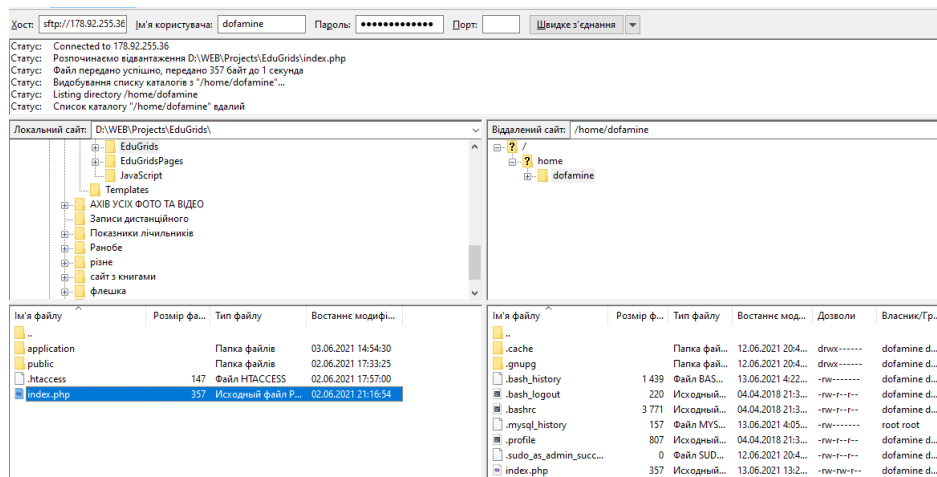


Рисунок 3.6 – Функціонування FTP сервера

По замовчуванню встановлення MariaDB не захищене, тому потрібно виконати сценарій безпеки «mysql\_secure\_installation» див. рис. 3.7, який постачається пакетом. Після чого буде запропоновано встановити пароль користувача, щоб ніхто не зміг увійти до MariaDB.

Далі буде створено обліковий запис для віддаленого доступу, який буде реалізовано за допомогою веб-додатку PhpMyAdmin.

Встановлення СКБД MariaDB наведено в рис. 3.7.

					КС КРБ 123.184.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

```

sudo apt-get install mariadb-server
sudo mysql_secure_installation
Enter current password for root (enter for none): [enter]
Set root password [Y/n]: y
New password: Vitya777Vitya
Re-enter new password: Vitya777Vitya[enter]
Remove anonymous users? (Press y|Y for Yes, any other key for
No): y
Disallow root login remotely? (Press y|Y for Yes, any other key
for No): y
Remove test database and access to it? (Press y|Y for Yes, any
other key for No): y
Reload privilege tables now? (Press y|Y for Yes, any other key
for No): y

```

**Рисунок 3.7 – Встановлення MariaDB**

Встановлення препроцесора гіпертексту PHP наведено в рис. 3.8.

```

sudo apt-get install php php-common php-mysql php-gd php-cli
$ echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

```

**Рисунок 3.8 – Встановлення PHP**

Ланцюг пакетів, які встановлюються див. рис. 3.8 містить наступне:

- пакети з стандартними утилітами «php-common»;
- модуль функціонування з БД «php-mysql»;
- бібліотека для роботи з графікою «php-gd»;
- модуль функціонування PHP в командній стрічці (терміналі) «php-cli».

Наступна команда викликає функцію генерації сторінки документації поточної версії PHP за допомогою останнього встановленого модуля «php-cli».

Команда «tee», що слідує за командою «sudo» див. рис. 3.8, покликана виводити на екран або перенаправляти вихідний матеріал, що є результатом коду

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

PHP. В даному випадку результат перенаправлено за шляхом «/var/www/html» з створенням вихідного матеріалу з назвою «info.php».

Зчитування вихідного матеріалу здійснюється за адресою «wike.ml/info.php». Результат зображено на рис. 3.9.

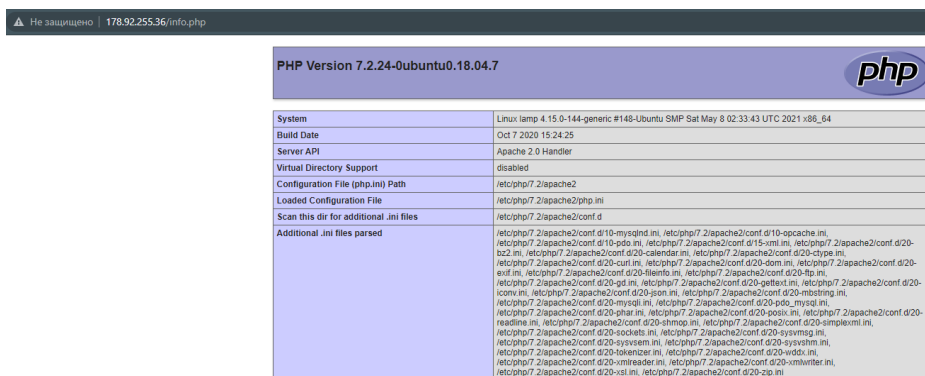


Рисунок 3.9 – Результат встановлення PHP

Встановлення PhpMyAdmin наведено в лістингу 3.5.

Лістинг 3.5 – Встановлення PhpMyAdmin

```
sudo apt-get install phpMyAdmin -y
```

Під час встановлення в діалогових вікнах потрібно вказати наступне:

- веб-сервер «apache2», який буде сконфігуровано для автоматичного запуску PhpMyAdmin;
- створити файл конфігурації «dbconfig-common»;
- задати пароль користувача, що створений під час встановлення MariaDB «Vitya777Vitya».

Після встановлення потрібно копіювати файл конфігурації згенерований PhpMyAdmin для автоматичного запуску за допомогою веб-сервера «apache2». Команда копіювання наведена в лістингу 3.6.

Лістинг 3.6 – Копіювання конфігураційного файлу PhpMyAdmin для веб-сервера «apache2»

```
sudo cp /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf
```

Команда «cp» виконує копіювання файлу з першого каталогу в другий.  
Команди для увімкнення PhpMyAdmin наведено в лістингу 3.7.

Лістинг 3.7 – Активація конфігурації PhpMyAdmin

```
sudo a2enconf phpMyAdmin ; sudo systemctl restart apache2
```

Команда «a2enconf» - це скрипт, який вмикає задану конфігурацію.

Команда «restart» входить до набору інструментів «systemctl» і призначена для перезавантаження сервісів.

Виконавши перезавантаження веб-сервера, веб-додаток PhpMyAdmin автоматично вмикатиметься слідом за веб-сервером. СКБД доступна за адресою «wike.ml/phpmyadmin», проте увійти в систему не вдасться, що було вказано під час встановлення MariaDB в цілях забезпечення безпеки доступу до БД.

СКБД стане доступною після створення облікового запису користувача.  
Створення облікового запису наведено в рис. 3.10.

```
sudo mysql -u root -p
Enter password:[Vitya777Vitya]
MariaDB [(none)]>CREATE USER `dofamine`@'localhost' IDENTIFIED
BY `Vitya777Vitya`;
MariaDB [(none)]>GRANT ALL PRIVILEGES ON *.* TO
`dofamine`@'localhost' WITH GRANT OPTION;
MariaDB [(none)]>FLUSH PRIVILEGES;
exit
```

Рисунок 3.10 – Створення користувача MariaDB

					КС КРБ 123.184.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

Перша стрічка див. рис. 3.10, містить набір прапорів, що відносяться до команди «sudo». Прапор «-u» вказує, що команду потрібно виконувати від імені поточного користувача. Прапор «-r» вказує, що потрібно використовувати заданий запит пароллю для «root».

Наступною стрічкою вводиться пароль доступу до БД.

В стрічці 3 див. рис. 3.10 виконано запит на мові SQL, який створює користувача «dofamine» з паролем «Vitya777Vitya». Параметр «localhost», відповідає за режим доступу до БД. Доступ до БД здійснюється за приватною IP-адресою 192.168.1.10:3306, що є приватною (локальною). Віддалений доступ буде доступний, за допомогою доменого імені «wike.ml» після звернення на який відбувається переадресація на внутрішню приватну мережу, яка являється приватною.

Стрічкою 4 див. рис. 3.10 надаються права доступу для створеного користувача, де список прав перераховано за допомогою регулярного виразу «\*.\*».

Інструкція «FLUSH PRIVILEGES» перезавантажує усі таблиці представлень привілежій, для затвердження змін. Інструкція «exit» здійснює вихід з консолі MariaDB.

На даному етапі звернення до веб-сервера з мережі інтернет здійснюються через публічну статичну IP-адресу виділену провайдером. В цілях реалізації повноти функціонування стенду виконуються наступні дії:

- реєстрація доменого імені «wike.ml» на сайті «freenom.com»;
- реєстрація безкоштовного DNS сервера на сайті «freedns.afraid.org»;
- створення записів на DNS сервері;
- прив'язка дочірніх NS серверів до доменого імені.

Реєстрація доменого імені передбачає реєстрацію на сайті облікового запису користувача за допомогою поштової скриньки. Безкоштовна реєстрація доменого імені затверджується за обліковим записом користувача на один рік.

Аналогічно вище виконаним діям відбувається реєстрація облікового запису на сайті безкоштовного DNS сервера. Після створення облікового запису

потрібно перейти на вкладку «Subdomain» та натиснути «Add a subdomain». У вікні, що відкрилось потрібно створити запис відповідно до сервісів встановлених на сервері:

- type: A;
- subdomain: не вказується;
- domain: wike.ml;
- destination: «178.92.255.36» (Публічна статична IP-адреса виділена провайдером).

Заголовок «Type: [type]» вказує тип запису, де «A» розшифровується як адрес. Дане поле задає відповідність між IP-адресою та ім'ям (доменим ім'ям).

Вище описану дію по створенню запису потрібно повторити, змінивши параметр «Domain» на «www.wike.ml», що являється аліас адресою по відношенню до головного запису.

Прив'язка дочірніх NS серверів, для забезпечення загальнодоступності доменого імені в мережі інтернет виконується в розділі «Management Tools» облікового запису «freenom». Налаштування зображено на рис. 3.11.

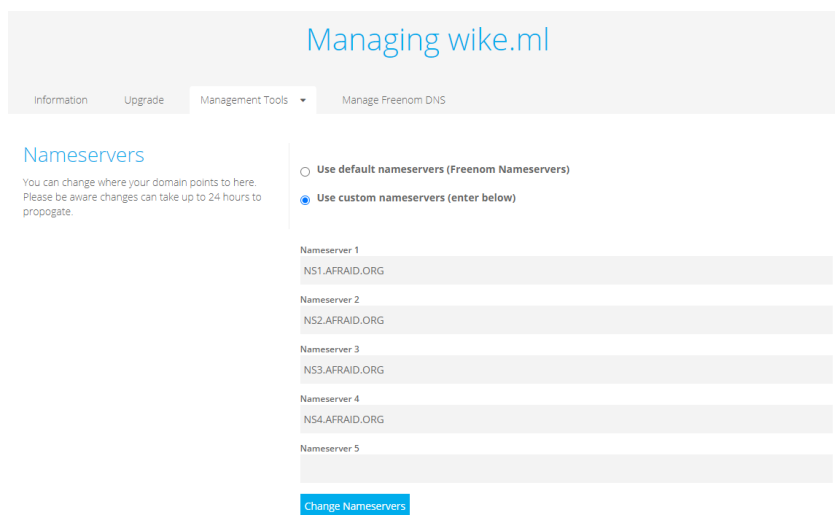


Рисунок 3.11 – Прив'язка дочірніх NS серверів

Звернення до веб-сервера по замовчуванню здійснюється на порт 80, який відповідає незахищеному протоклу HTTP.



Налаштування захищеного протоколу проводиться для веб-сервера apache і містить наступні етапи:

- створення директорії та встановлення прав доступу до створеної директорії;
- завантаження сертифікатів з ключем за допомогою FTP;
- налаштування віртуального хоста, що функціонує на «443» порті;
- увімкнення SSL модуля;
- налаштування переадресації на «443» порт для «wike.ml».

Створення директорії в якій зберігатимуться сертифікати і ключ наведено в лістингу 3.8.

### Лістинг 3.8 – Налаштування директорії sslCA

```
sudo mkdir /etc/sslCA ; sudo chmod -R a+rxw /etc/sslCA
```

Команда «mkdir» створює нову директорію «sslCA» за шляхом «/etc».

Команда «chmod» використовується для зміни прав доступу до програм, файлів, директорій. Прапор «-R» задає рекурсію, що розповсюджує права застосовані до директорії на вкладені файли. Нотація «a+rxw» розшифровується як: «усі можуть читати, записувати та запускати» файл до якого застосовуються дані права.

Вивантаження файлів за допомогою SFTP зображено на рис. 3.12.

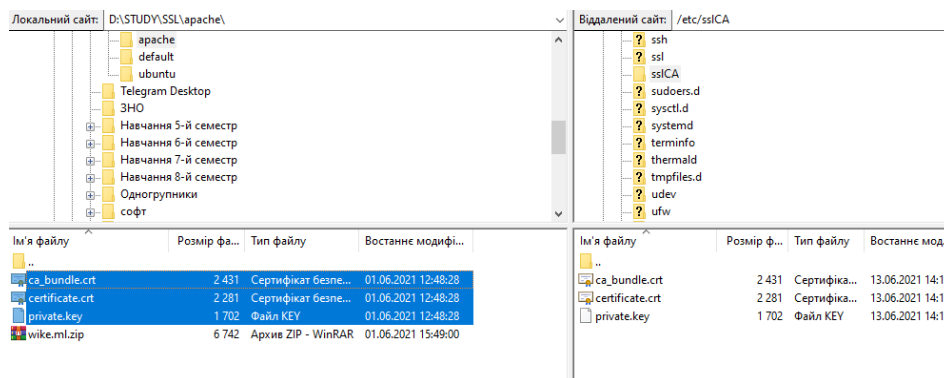


Рисунок 3.12 – Вивантаження файлів сертифікату

Налаштування віртуального хоста проводиться в кінці конфігураційного файлу «ssl.conf» як наведено на рис. 3.13.

Команда «nano» див. рис. 3.13 відкриває файл «ssl.conf» за допомогою вмонтованого текстового редактора «nano».

Увімкнення модуля SSL наведено в лістингу 3.9.

#### Лістинг 3.9 – Активація модуля SSL

```
sudo a2enmod ssl ; sudo systemctl restart apache2
```

Команда «a2enmod» аналогічна «a2enconf».

```
sudo nano /etc/apache2/mods-available/ssl.conf
<VirtualHost _default_:443>
SSLEngine on
SSLCertificateFile /etc/sslCA/certificate.crt
SSLCertificateKeyFile /etc/sslCA/private.key
SSLCertificateChainFile /etc/sslCA/ca_bundle.crt
</VirtualHost>
```

#### Рисунок 3.13 – Створення віртуального хоста в конфігурації модуля SSL

На даному етапі веб-сайт доступний по протоколах «http:wike.ml, https:wike.ml». Налаштування переадресації користувача на захищений протокол по замовчуванню здійснюється редагуванням файлу активної конфігурації «000-default.conf» в якому додатково потрібно розкоментувати стрічку «ServerName» і задати домене ім'я як наведено на рис. 3.14.

```
sudo nano /etc/apache2/sites-available/000-default.conf
ServerName www.wike.ml
Redirect permanent / https://wike.ml
```

#### Рисунок 3.14 – Переадресація на захищений протокол

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		50

Забезпечення роботи веб-сайту на основі патерну MVC, який реалізовується з використанням PHP, досягається наступними кроками:

– налаштування модуля «rewrite», що забезпечує належне функціонування роутерів;

– налаштування пріоритетності індексного файлу з розширенням «.php».

Надання прав перезапису для директорії «/var/www/» та активацію модуля «rewrite» наведено на рис. 3.15. Конфігурація файлу «.htaccess», який містить правила перезапису для адресної стрічки «URL» наведено на рис. 3.16.

```
sudo nano /etc/apache2/apache2.conf
<Directory /var/www/>
Option Indexes FollowSymLinks
AllowOverride ALL
Require all granted
</Directory>
sudo a2enmod rewrite
sudo systemctl restart apache2
```

Рисунок 3.15 – Налаштування роботи модуля «rewrite»

```
AddDefaultCharset utf-8
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} -f
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^(.*)$ index.php
```

Рисунок 3.16 – Конфігураційний файл веб-сервера «.htaccess»

Налаштування пріоритетності індексного файлу наведено на рис. 3.17.

```
sudo nano /etc/apache2/mods-enabled/dir.conf
DirectoryIndex index.php index.html index.cgi index.pl
index.xhtml index.htm
```

Рисунок 3.17 – Налаштування пріоритетності файлів

					КС КРБ 123.184.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

### 3.1.2 Розробка алгоритму роботи програми

Доступ до повного обсягу курсів та тестів потребують наявності акаунту, за допомогою якого користувач авторизує себе в системі як студент або викладач. Дана частина програми розробляється відповідно до діаграми класів (рис. 2.9). та діаграми послідовності (рис 2.12). Код контролера, який реагує на дії користувача сторінки «account/login» наведено в лістингу 3.10.

#### Лістинг 3.10 – Контролер «AccountController»

```
namespace application\controllers;
use application\core\Controller;
class AccountController extends Controller {
    public function loginAction() {
        if (!empty($_POST)) {
            if ($this->model->userExists($_POST['email'])) {
                $user = $this->model->getUser($_POST['email']);
                if ($user['status'] == 'teacher') {
                    $_SESSION['teacher']['id'] = $user['id_user'];
                } else if ($user['status'] == 'student') {
                    $_SESSION['authorize']['id'] = $user['id_user'];
                }
                $this->view->redirect('/user');
            } else {
                $this->view->message('error', 'user not exists');
            }
        }
        $this->view->render('Login page');
    }
    public function registerAction() {
        if (!empty($_POST)) {
            if ($this->model->userExists($_POST['email'])) {
                $this->view->message('error', 'user already exists');
            } else {
                $this->model->setUser($_POST);
            }
        }
    }
}
```

```

        $this->view->render('Register Page');
    }
    public function logoutAction() {
        session_destroy();
        $this->view->redirect('/account/login');
    }
}

```

Метод «loginAction()» виконує перевірки глобальної змінної «\$\_POST» на відсутність значень, після якої відбувається наступна перевірка на існування користувача за допомогою електронної адреси, яка являється унікальною і не повторюється. Результатом повернення методу «userExists» є булева змінна. В результаті наявності акаунту з такою електронною адресою викликається метод для одержання статусу користувача, для подальшого встановлення прав доступу на основі поточної сесії. Результатом успішної авторизації відбувається переадресація на сторінку користувача з його персональними даними.

Метод «render()» відповідає за відображення сторінки. Даний метод наслідується від батьківського контролера в якому створено екземпляр класу «View», за допомогою агрегації. Аналогічно екземпляру класу «View» доступний екземпляр класу «Model».

Окремо винесено модель «Account» для контролера «AccountController». Код моделі наведено в лістингу 3.11.

### Лістинг 3.11 – Модель «Account»

```

namespace application\models;
use application\core\Model;
class Account extends Model {
    public function userExists($email) {
        $params = [
            'email' => $email,
        ];
        if ($this->db->singleRow('SELECT email FROM User
WHERE email = :email', $params)) {
            return true;
        } else {
            return false;
        }
    }
}

```

					КС КРБ 123.184.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

```

public function getUser($email) {
    $params = [
        'email' => $email,
    ];
    return $this->db->singleRow('SELECT * FROM User WHERE
email = :email', $params);
}
public function setUser($user = []) {
    $params = [
        'id_user' => md5(uniqid($user['email'], true)),
        'name' => $user['name'],
        'email' => $user['email'],
        'password' => $user['password'],
        'status' => 'student',
    ];
    $this->db->singleRow('INSERT INTO `User` (`id_user`,
`name`, `email`, `password`, `status`)
VALUES (:id_user, :name, :email, :password,
:status)', $params);
}

```

Модель містить методи, які здійснюють запити на зчитування та запис даних в БД. Метод «userExists()» здійснює пошук запису по ключу «email» і повертає булеве значення в залежності від існування запису. Метод «getUser()» повертає кортеж з всіма атрибутами по ключу «email». Метод «setUser()» здійснює запис нового користувача в таблицю «User», використовуючи отриманий масив даних на вхід методу. Створення, редагування та видалення курсів доступні викладачу здійснюються за допомогою окремих методів. Доступу до даного функціоналу обмежується за допомогою листів доступу. Листи доступу організовано у вигляді окремого асоціативного масиву «ключ – значення». Дані листи існують окремо для кожного контролера. Організацію листів доступу розроблено у контролері, що наведено в лістингу 3.12.

### Лістинг 3.12 – Організація листів доступу

```

namespace application\core;
use application\core\View;
abstract class Controller {
    private $route;
    protected $view;
    protected $model;
    protected $acl;
}

```

					КС КРБ 123.184.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

```

public function __construct($route)
{
    $this->route = $route;
    //$_SESSION['authorize']['id'] = 1;
    if (!$this->checkAcl()) {
        View::errorCode(403);
    }
    $this->view = new View($route);
    $this->model = $this->loadModel($route['controller']);
    //debug($this->model);
}
public function loadModel($name) {
    $path = 'application\models\\' . ucfirst($name);
    if (class_exists($path)) {
        return new $path();
    }
}
public function checkAcl() {
    $this->acl = require 'application/acl/' . $this->route['controller'] . '.php';
    if ($this->isAcl('all')) {
        return true;
    } elseif (isset($_SESSION['authorize']['id']) and $this->isAcl('authorize')) {
        return true;
    } elseif (!isset($_SESSION['authorize']['id']) and $this->isAcl('guest')) {
        return true;
    } elseif (isset($_SESSION['teacher']['id']) and $this->isAcl('teacher')) {
        return true;
    }
    return false;
}
public function isAcl($key) {
    return in_array($this->route['action'], $this->acl[$key]);
}
}

```

Під час переходу між сторінками відбувається автозавантаження контролера у відповідності до даних стрічки адреси. Завантаження контролера ініціює створення екземпляру класу, який наслідується від абстрактного класу «Controller». Виклик конструктора класу «Controller» містить перевірку доступу до сторінки, яку завантажуватиме екземпляр класу «View». Метод «isAcl()» виконує перевірку порівнюючи дані в масиві «routes.php» з даними отриманими на основі поточного маршруту та повертає булеву змінну як результат.

					КС КРБ 123.184.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

Відображення сторінки відбудеться лише у випадку, коли встановлено сесію у відповідності з типом користувача переданого на перевірку в метод «isAcl()».

Вище описані аторизація, листи доступу та інші класи працюють працюють на основі каркасу з єдиною точкою входу розробленого по патерну MVC, який наведено в графічній частині.

Класи контролерів та моделей у відповідності до проєктованих систем в розділі 2 наведено в додатку В.

### 3.1.3 Розробка користувацьких інтерфейсів

Користувацькі інтерфейси розробляються з дотриманням семантики та методології БЕМ як наведено в лістингу 3.13.

Лістинг 3.13 – Розробка фронтенд частини з використання семантичних тегів та дотримання методології БЕМ

```
<section class="courses-grid container">
  <article class="section-title courses-grid__section-
title">
    
    <h2 class="section-title__title">Featured
Courses</h2>
    <p class="section-title__text">There are many
variations of passages of Lorem Ipsum available, but the majority
have suffered alteration in some form.</p>
  </article>
  <div class="courses-grid__course-card">
    <?php foreach ($vars as $course): ?>
      <article class="course-card">
        <span class="course-card__price"><?php echo
$course['price']; ?></span>
        <a class="course-card__link" href="#0">
          
        </a>
        <div class="course-card__desc">
          <a class="course-card__title"
href="#0"><?php echo $course['title']; ?></a>
          <p class="course-card__text"><?php echo
$course['description']; ?></p>
        </div>
      </article>
    </div>
  </div>
</div>
```



```

        <div class="couse-card__tag">
            <ul class="course-card__rating">
                <li class="course-card__star"></li>
                <li class="course-card__star"></li>
                <li class="course-card__star"></li>
                <li class="course-card__star"></li>
                <li class="course-card__star course-
card__star--active"></li>
                <li class="course-
card__reviews"><?php echo $course['reviews'];?></li>
            </ul>
            <a class="tag" href="#0"><?php echo
$course['tag']; ?></a>
        </div>
    </article>
    <?php endforeach; ?>
</div>
</section>

```

Теги: «section» використовується для поділу контенту сайти на окремі змістові блоки; «article» відображає незалежний блок (курс), який є наближений за змістом до даної секції; «div» не зчитується пошуковими роботами і застосовується як обгортка для застосування стилів вирівнювання контенту в середині блоку.

Класи стилів: «section-title», являє собою незалежний блок, який не містить зовнішніх відступів; «courses-grid\_\_section-title» являє собою мікс двох блоків, яких використовується для задання розмірів, відступів та позиціонування даного блоку «section-title» по відношенню до «courses-grid». Розроблені користувацькі інтерфейси наведено в додатку В.

### 3.2 Тестування функціональності платформи

Тестування сторінки з курсами зображено на рис. 3.18. Тестування сторінки авторизації зображено на рис. 3.19.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

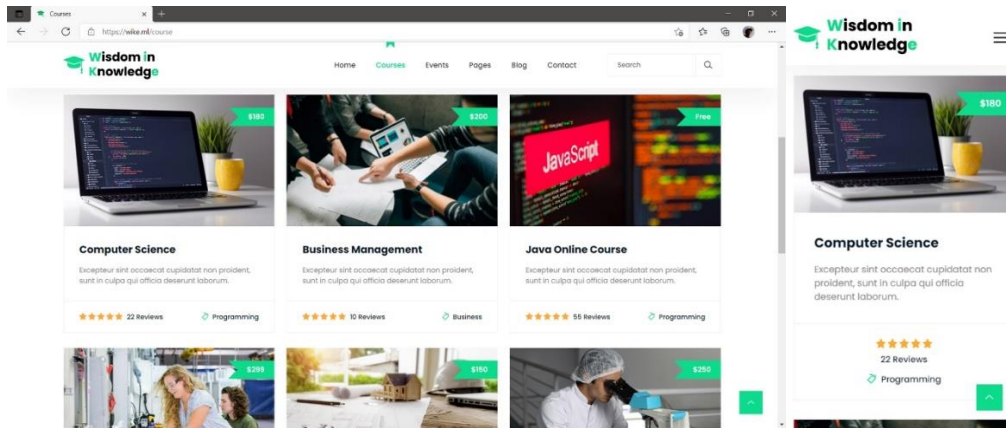


Рисунок 3.18 – Сторінка з курсами

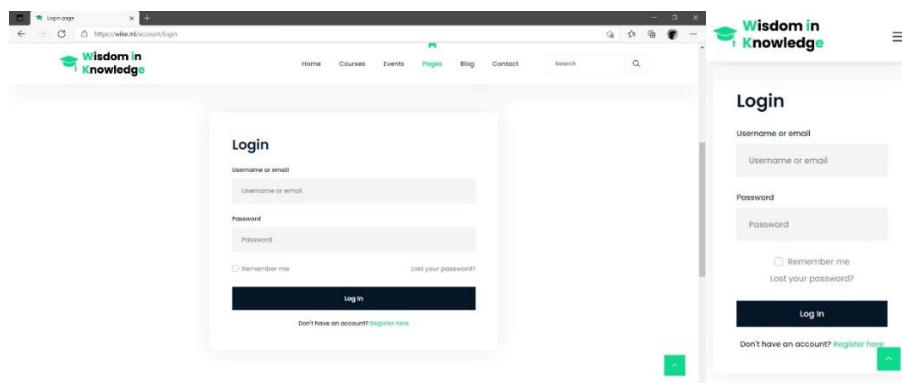


Рисунок 3.19 – Сторінка входу

Змн.	Арк.	№ докум.	Підпис	Дата

КС КРБ 123.184.00.00 ПЗ

Арк.

58

## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

Темою кваліфікаційної роботи є розробка програмно-апаратної навчальної платформи «Wisdom in Knowledge». Відповідно з безпеки життєдіяльності розглядається питання працездатності людини – оператора, яка обслуговуватиме серверну частину програмно-апаратної навчальної платформи, а для основ охорони праці розглядається питання стосовно загальних вимог безпеки з охорони праці для користувачів ЕОМ. В результаті розробляється інструкція з охорони праці для обслуговуючого персоналу серверного обладнання

### 4.1 Працездатність людини – оператора

Під працездатністю людини розуміють можливість її виконувати роботу з необхідною якістю та в установлений час. Працездатність людини залежить як від зовнішніх чинників, так і внутрішнього стану (внутрішні чинники).

До зовнішніх чинників належать: кількість та форма отриманої інформації, зручність робочого місця, характер взаємостосунків в колективі, вплив чинників середовища існування.

До внутрішніх чинників належать: рівень підготовки, тренуваність людини та її емоційна стійкість.

У процесі роботи людина переживає різні функціональні стани, які зумовлюють різні рівні її працездатності.

На рис. 4.1 наведено зміни функціонального стану та якості роботи людини у процесі одного трудового циклу (зміни).

Виділяються 4 фази працездатності: пристосування до праці, стійкої працездатності, субкомпенсації, втоми.

					<b>КС КРБ 123.184.00.00 ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>				
Розроб.		Цимбалістий В.О.			<b>Безпека життєдіяльності, основи охорони праці</b>	<b>Лім.</b>	<b>Арк.</b>	<b>Акрушів</b>
Перевір.		Яцишин В.В.					59	8
Консульт.		Пилипець М.І.				<b>ТНТУ, каф. КС, гр. СІс-44</b>		
Н. Контр.		Тиш С.В.						
Затверд.		Осухівська Г.М.						

Тривалість усіх фаз та усього циклу роботи залежить від рівня підготовки людини до роботи.

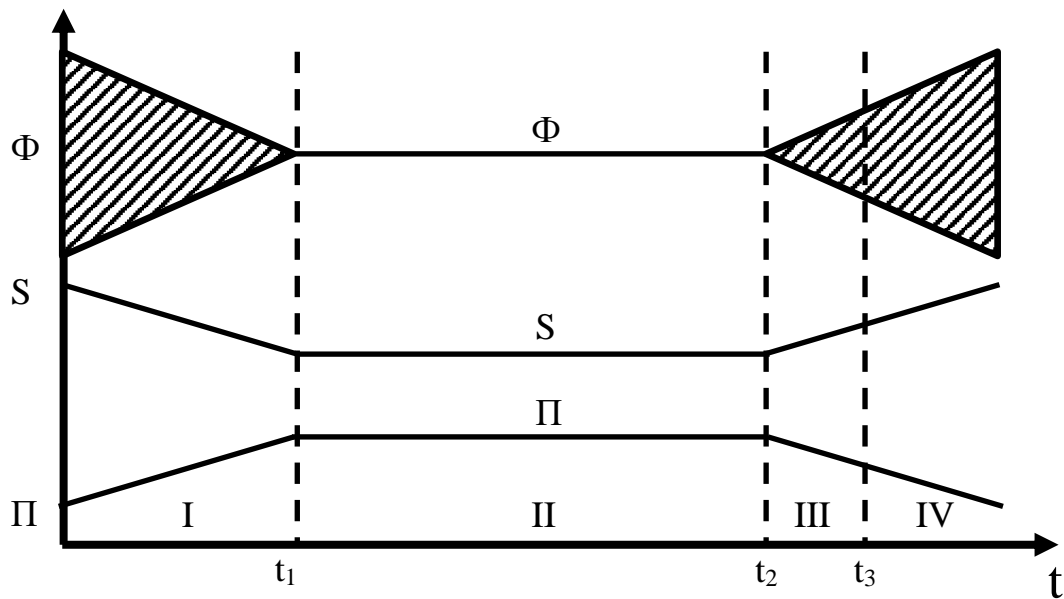


Рисунок 4.1 – Фази працездатності

Ф – показник функціонального стану;

Б – помилки роботи;

П – продуктивність праці.

Фаза пристосування до праці (0 – 1), - це час протягом якого людина адаптується до майбутніх умов праці. Основний показник поступово досягає свого встановленого значення. Тривалість періоду пристосування організму до умов праці залежить від багатьох чинників, серед яких основними є інтенсивність роботи (чим інтенсивніша робота, тим цей період коротший) та рівень готовності людини до майбутньої роботи.

Значного скорочення фази пристосування до праці можна досягти за рахунок попередньої підготовки людини до роботи (виконання фізичних вправ, адаптації зору, слуху та інші) та шляхом посиленого навчального навантаження. Суть останнього полягає в тому, що оператор перед початком роботи проводить короткочасне тренування щодо розв’язання однієї чи кількох задач підвищеної складності.

Фаза стійкої працездатності ( $t_1 - t_2$ ) характеризується найвищою якістю праці при оптимальних рівнях функціонування фізіологічних систем організму. Тривалість цього періоду залежить від інтенсивної роботи. Чим інтенсивніша праця, тим коротший цей період. Найоптимальніша динамічна робота, коли цей період може бути в десятки разів довшим, ніж при статичній діяльності.

На процес стійкої працездатності великий вплив справляють емоції. Негативні (страх, невпевненість, поганий настрій) знижують працездатність. Позитивні (впевненість, спокій, бадьорий настрій) значно продовжують період стійкої працездатності.

Продовження періоду стійкої працездатності можна забезпечити: оптимальним рівнем напруги психофізіологічних функцій; комфортними умовами праці; правильним поєднанням режимів праці та відпочинку; емоційним розвантаженням; використанням тонізуючих напоїв (кава, чай), фармакологічних засобів, зокрема препаратів рослинного походження (вітаміни, препарати, які впливають на енергетичні та метаболічні процеси); інформуванням людини про наслідки її діяльності, наглядом та контролем її роботи.

Практичний досвід свідчить, що вживання легких стимуляторів допомагає знизити сонливість, сприяє підвищенню працездатності на короткий період. Однак активні стимулятори на відповідальних видах робіт здатні викликати негативний ефект – погіршується самопочуття, знижується рухливість та швидкість реакції. Поширене серед населення вживання транквілізаторів, викликаючи заспокоєння та запобігаючи розвитку нервозів, може знизити психічну активність, сповільнити реакції, спричинити апатію та сонливість. [6].

Фаза субкомпенсації ( $t_2 - t_3$ ) розглядається як початок розвитку втоми. В цей період якість праці, ще зберігається на високому рівні, але тільки за рахунок перенапруг і відповідних функцій організму.

Фаза втоми (з моменту у характеризується чітко вираженими зниженнями якості роботи при подальшому погіршенні функціонального стану людини.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

Об'єктивними показниками втоми є зміна частоти пульсу, дихання, зорової та слухової чутливості.

Наступною фазою життєдіяльності людини повинна бути фаза відновлення працездатності (відпочинку), яка може тривати від 3 до 5 хвилин; 60 – 90 хв. і навіть декілька діб. [6].

#### 4.2 Загальні вимоги безпеки з охорони праці для користувачів ЕОМ

ЕОМ та устаткування повинні бути справними і випробуваними відповідно до чинних нормативних актів.

Після закінчення роботи електронно-обчислювальна техніка відключається від електричної мережі. Це ж саме повинно бути здійснено у випадку виникнення аварійної ситуації під час роботи.

Під час експлуатації ЕОМ забороняється здійснювати ремонт та налагодження їх на робочому місці, працювати на зіпсованій техніці, загроможувати робочі місця матеріалами, які не використовуються для поточної роботи.

Обслуговування, ремонт та налагодження ЕОМ, інші операції у цьому плані мають здійснюватись тільки при повному відключенні живлення.

У випадку коли ремонтні та інші операції неможливо здійснити при відключеному живленні, необхідно, щоб устаткування, допоміжна апаратура та прилади були заземлені, роботу виконували два або більше працівників з використанням інструментів з ізольованими ручками, а на підлозі були діелектричні килимки.

Ремонт відеотерміналу без футляра, а також усі види робіт з відкритим кінескопом повинні проводитися в захисних окулярах або масці.

Режим праці та відпочинку працівників електронно-обчислювальної техніки визначається ДСанНіП 3.3.2-007-98 Вимоги до режимів праці і відпочинку при роботі з ВДТ ЕОМ і ПЕОМ. Через кожні 40-50 хв. роботи

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

необхідно робити 3 – 5 хвилинні перерви для відпочинку. Сумарна тривалість роботи на день не повинна перевищувати 4 год., а на тиждень – 20 год.

До роботи з профілактикою обслуговування, налагодження і ремонту ЕОМ допускаються працівники віком старше 18 років, які пройшли попереднє спеціальне навчання, мають відповідне посвідчення, не мають медичних протипоказань, пройшли інструктаж з охорони праці та пожежної безпеки.

Усі працівники підлягають обов'язковому медичному огляду відповідно до Положення про медичний огляд працівників певних категорій.

#### 4.3 Інструкція з охорони праці для обслуговуючого персоналу серверного обладнання

##### 1. Загальні положення

1.1 Дана інструкція, розроблена на підставі ДНАОП 5.2.30-1.08-96 і ВСН 4559-88 "Тимчасові санітарні норми і правила для працівників обчислювальних центрів" і рекомендацій учбово - методичного посібника "Охорона праці користувачів комп'ютерних відео дисплейних терміналів".

1.2. До роботи на персональних ЕОМ допускаються особи від 18 років, що не мають протипоказань за результатами попереднього медичного огляду і пройшли інструктаж, навчання й перевірку знань по охороні праці і мають 1 кваліфікаційну групу з електробезпеки.

1.3. Допуск до роботи на персональних ЕОМ осіб до 18 років (практикантів, учнів) здійснюється під керівництвом досвідчених працівників, що мають кваліфікаційну групу з електробезпеці не нижче 3.

1.4. Праця на робочому місці, оснащеному дисплеєм, супроводжується дією наступних небезпечних і шкідливих факторів: напруження зору; гіподинамія; монотонність праці; підвищений рівень статичної електрики; відбиті блики екрана дисплея; емоційні перенавантаження; можливість поразки електричним струмом;

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

1.5. Для зниження й попередження шкідливого впливу вищевказаних факторів необхідно:

– для зниження рівня статичної електрики розташовувати екран дисплея на відстані не ближче 550 - 700 мм. від очей оператора;

– для зниження бліків екран дисплея повинен розташовуватися перпендикулярно світлового потоку від віконних прорізів або від електросвітильників;

– для зниження втоми очей:

а) освітленість робочого місця повинна бути не менш 300 - 500 люкс; яскравість світіння екрана - не менш 100 кл / кв. м.;

б) мінімальний розмір світної точки - не більш 0,6 мм.;

в) контрастність зображення знака - не менш 0,8;

г) частота регенерації - не менш 72 Гц.;

д) для зниження впливу гіподинамії й емоційних перевантажень варто використовувати технологічні перерви і виконувати комплекс фізичних вправ, зазначених у Додатку 1 і 2, п. 10, Тимчасових санітарних норм.

2. Вимоги безпеки перед початком роботи.

2.1. Залишити в гардеробі вуличний одяг, особисті речі.

2.2. Забрати з робочого місця предмети, що не будуть використовуватися в роботі.

Забороняється класти на блоки ЕОМ папір, книги, документи й інші предмети.

2.3. Забороняється, щоб уникнути перевантаження мережі, підключати ЕОМ через трійники разом з іншими електроприладами.

2.4. Уключити, при необхідності, штучне освітлення, настільний світильник.

2.5. Зовнішнім оглядом переконатися в справності сполучних проводів, штепсельних рознімачів, шин заземлення й вимикачів, у надійності кріплення захисних кожухів і кришок блоків ЕОМ.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64



2.6. Перевірити відсутність пилу на екрані дисплея й правильність установки паперу в прийомний лоток принтера. Не допускати забивання пилом і сторонніми предметами вентиляційних отворів для відводу тепла з блоків ЕОМ.

2.7. Відрегулювати висоту сидіння стільця й підставки для ніг. Відрегулювати положення екрана монітора щодо свого поля зору.

2.8. При виявленні несправностей і інших недоліків, що створюють небезпеку або значні незручності в роботі, заявити про це керівникові відділу, ділянки.

3. Вимоги безпеки під час роботи.

3.1. При включенні персональної ЕОМ і освітлення в електромережу, братися тільки за ізольовані частини штепсельних колодок.

3.2. Дотримувати зазначену в інструкції з експлуатації послідовність включення блоків ЕОМ.

3.3. Щоб уникнути розрядів статичної електрики, забороняється доторкатися до екрана дисплея.

3.4. При введенні даних, редагуванні програм, читанні інформації з екрана, безперервна тривалість роботи перед екраном не повинна перевищувати 1 годину з наступними регламентованими перервами по 10 хвилин для відпочинку та виконувannya комплексу фізичних вправ, релаксаційної гімнастики й аутогенного тренування.

3.5. Забороняється при не відключеному електроживленні ЕОМ : розкривати захисні кожухи й кришки блоків ЕОМ, робити регулювання й чищення внутрішніх деталей, змінювати запобіжники; переключати сполучні шнури блоків ЕОМ; змінювати встановлену конфігурацію робочого місця, переставляти блоки ЕОМ; робити вологе прибирання поверхонь комп'ютера; приймати їжу безпосередньо за клавіатурою комп'ютера.

3.6. Категорично забороняється на робочому місці оператора ЕОМ : курити, користатися відкритим вогнем; зберігати легкозаймисті, вибухонебезпечні і хімічно активні, що руйнують ізоляцію, продукти.

4. Дії персоналу в аварійних ситуаціях.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

4.1. Ознаками аварійної ситуації на робочому місці оператора ЕОМ є:

- поява збоїв у роботі ЕОМ, заїдання паперу в принтері, зникнення зображення на екрані дисплея,
- коротке замикання, іскріння, появи запаху гару, підвищене нагрівання корпусу, штепсельних рознімачів, сполучних проводів, зниження або зникнення напруги в мережі.

4.2. В аварійній ситуації необхідно :

- роботу припинити, ЕОМ відключити від мережі;
- при загорянні використовувати вуглекислотний або порошковий вогнегасники;
- ужити заходів по евакуації людей і наданню першої медичної допомоги постраждалим;
- доповісти про те, що трапилося, керівникові відділу, ділянки;
- при необхідності викликати швидку допомогу, пожежну команду.

5. Вимоги безпеки після закінчення роботи.

5.1. Закінчити працюючі програми, закрити всі каталоги, підготувати комп'ютер до вимикання.

5.2. Відключити ЕОМ і місцеве електроосвітлення від мережі.

5.3. Упорядкувати робоче місце, забрати документи, що використовувалися.

5.4. Переконатися у відсутності пожежної небезпеки.

5.5. Докласти керівникові відділу, ділянки про закінчення роботи і про виявлені під час роботи несправності й інші недоліки.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи бакалавра реалізовано програмно-апаратну навчальну платформу Wisdom in Knowledge з доступом в мережу інтернет використовуючи захищений канал зв'язку.

Під час виконання було проаналізовано вимоги до програмного, апаратного забезпечення, технології реалізації платформи, хмарні сервіси та хостинги публікації платформи. Розроблено структурну схему функціонування стеку LAMP та схему функціонування платформи на основі Client-Server.

Розроблено концептуальну архітектуру програмно-апаратної навчальної платформи Wisdom in Knowledge.

Здійснено обґрунтування вибору апаратного забезпечення проектного засобу на основі розроблених формул. Обґрунтовано вибір програмного забезпечення проектного засобу.

Побудовано діаграми відношень між авторами і прецедентами, діаграми класів та діаграми полідовностей. Виконано проектування бази даних.

Налаштовано функціонування сервісів стеку LAMP. Організовано доступ в мережу інтернет по захищеному протоколу HTTPS.

Розроблено структурну схему каркасу MVC, програму з користувацькими інтерфейсами та здійснено тестування.

					<i>КС КРБ 123.184.00.00 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Александр Кичатов. ЧПУ, роутинг, единая точка входа на PHP. 2020. URL: <https://www.programulin.ru/chpu-routing-edinaya-tochka-vhoda-na-php> (дата звернення 11.04.2021)
2. Бедрий Я.І. 39 Безпека життєдіяльності: Навчальний посібник. Київ: Кондор, 2009. 286 с.
3. Веллинг, Люк, Томпсон, Лора. Разработка веб-приложений с помощью PHP и MySQL, 5-е изд.: Пер. с англ. СПб. : ООО «Альфа-книга», 2017. 768 с.
4. Пьюривал С. Основы разработки веб-приложений. СПб.: Питер, 2015. 272 с.
5. Гоше Х. Д. HTML5. Для профессионалов. СПб.: Питер, 2013. 496 с.
6. Грибан В.Г., Негодченко О.В. Охрана праці. К.: Центр учбової літератури, 2009. 209 с.
7. Грофф, Джеймс Р., Вайнберг, Пол Н., Оппель, Эндрю Дж. SQL: Полное руководство, 3-е изд.: Пер. с англ. М.: ООО «И.Д. Вильямс», 2015. 960 с.
8. Дронов В.А. PHP, MySQL, HTML5 и CSS3. Разработка современных динамических Web-сайтов. СПб.: БХВ-Петербург, 2016. 688 с.
9. Колисниченко Д. Н. Самоучитель системного администратора Linux. СПб.: БХВ-Петербург, 2011 544 с.
10. Макфарланд Д. Большая книга CSS3. 3-е изд. СПб.: Питер, 2014. 608 с.
11. Мартин Фаулер. UML Основы. 3-е изд. СПб.: Питер, 2005. 192 с.
12. Мирошниченко Г.А. Реляционные базы данных: практические приемы оптимальных решений. СПб.: БХВ-Петербург, 2005. 400 с.
13. Мэт Зандстра. PHP: объекты, шаблоны и методики программирования, 5-е изд.: Пер. с англ. СПб. :ООО «Диалектика», 2019. 736 с.
14. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. СПб.: Питер, 2016. 688 с.

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

15. Флэнаган Д. JavaScript. Подробное руководство. Пер. с англ. СПб: Символ-Плюс, 2008. 992 с.
16. Aaron Kili. How to install and configure FTP server in ubuntu. 2017. URL: <https://www.tecmint.com/install-ftp-server-in-ubuntu/> (дата звернення 24.05.2021)
17. Aaron Kili. How to install LAMP stack with phpMyAdmin in ubuntu 18.04. 2018. URL: <https://www.tecmint.com/install-lamp-with-phpmyadmin-in-ubuntu-18-04/> (дата звернення 1.05.2021)
18. A semantic BEM style. 2017. URL: <https://www.tecmint.com/setup-https-ssl-certificates-to-secure-phpmyadmin-login/> (дата звернення 18.04.2021)
19. Free DNS Hosting, Dynamic DNS Hosting, Static DNS Hosting, subdomain and domain hosting. URL: <https://freedns.afraid.org/> (дата звернення 3.06.2021)
20. Freenom A Name for Everyone. URL <https://www.freenom.com/en/index.html> (дата звернення 1.06.2021)
21. Gabriel Canepa. How to setup HTTPS (SSL Certificates) to secure phpMyAdmin login. 2016. URL: <https://www.tecmint.com/setup-https-ssl-certificates-to-secure-phpmyadmin-login/> (дата звернення 14.05.2021)
22. James Kiarie. How to setup free SSL Certificate for apache on Debian. 2019. URL: <https://www.tecmint.com/setup-https-ssl-certificates-to-secure-phpmyadmin-login/> (дата звернення 22.05.2021)

					<b>КС КРБ 123.184.00.00 ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		69

Додаток А  
Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра комп'ютерних систем та мереж

**“Затверджую”**

Завідувач кафедри КС

\_\_\_\_\_ Осухівська Г.М.

“ \_\_\_\_ ” \_\_\_\_\_ 2021 р

КОМП'ЮТЕРНА СИСТЕМА ЗБОРУ ТА АНАЛІЗУ ДАНИХ З МЕТЕОСТАНЦІЙ

**ТЕХНІЧНЕ ЗАВДАННЯ**

на 11 листках

**Вид робіт:**

Кваліфікаційна робота

**На здобуття освітнього ступеня «Бакалавр»**

**Спеціальність 123 «Комп'ютерна інженерія»**

**«УЗГОДЖЕНО»**

Керівник кваліфікаційної роботи

\_\_\_\_\_ к.т.н., доц. Яцишин В.В.

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

**«ВИКОНАВЕЦЬ»**

Студент групи СІс-44

\_\_\_\_\_ Цимбалістий В.О.

« \_\_\_\_ » \_\_\_\_\_ 2021 р.

**Тернопіль 2021**

## 1 Загальні відомості

### 1.1 Повна назва та її умовне позначення

Повна назва теми кваліфікаційної роботи: Програмно-апаратна платформа навчання «Wisdom in Knowledge».

Умовне позначення кваліфікаційної роботи: КС КРБ 123.184.00.00

### 1.2 Виконавець

Студент групи СІс-44, факультету комп'ютерно-інформаційних систем і програмної інженерії, кафедри комп'ютерних систем та мереж, Тернопільського національного технічного університету імені Івана Пулюя, Цимбалістий Віктор Олександрович.

### 1.3 Підстава для виконання роботи

Підставою для виконання кваліфікаційної роботи є наказ по університету (№ 4.7-97 від 10.02.2021 р.)

### 1.4 Планові терміни початку та завершення роботи

Плановий термін початку виконання кваліфікаційної роботи – 10.02.2021 р.

Плановий термін завершення виконання кваліфікаційної роботи – 23.06.2021 р.



## 1.5 Порядок оформлення та пред'явлення результатів роботи

Порядок оформлення пояснювальної записки та графічного матеріалу здійснюється у відповідності до чинних норм та правил ІСО, ГОСТ, ЕСКД, ЕСПД та ДСТУ.

Пред'явлення проміжних результатів роботи з виконання кваліфікаційної роботи здійснюється у відповідності до графіку, затвердженого керівником роботи.

Попередній захист кваліфікаційної роботи відбувається при готовності роботи на 90% , наявності пояснювальної записки та графічного матеріалу.

Пред'явлення результатів кваліфікаційної роботи відбувається шляхом захисту на відповідному засіданні ЕК, ілюстрацією основних досягнень за допомогою графічного матеріалу.

## 2 Призначення і цілі створення системи

### 2.1 Призначення системи

Основне призначення програмно-апаратної платформи навчання «Wisdom in Knowledge», являє собою функціонування веб-сайту на основі клієнт-серверної архітектури. Така платформа функціонує на стеку сервісів, які комунікують між собою, що забезпечує функціонування веб-сайту. Розробка архітектури веб-сайту, яка виконує маніпуляції з даними на основі дій користувачів.

До основних задач, які покликана розв'язати платформа належать:

- Доступ до даних на сервері за захищеним каналом зв'язку.
- Комунікація між клієнтом та сервером по захищеному протоколу.
- Збереження персональних даних користувачів в БД.
- Організація прав доступу до даних веб-сайту.

- Розмежування користувачів за допомогою ролей.
- Забезпечення користувача функціоналом відповідно до ролі.

Така програмно-апаратна платформа надає можливості:

- Розширення нарощування функціоналу веб-сайту побудованого на основі патерну MVC;
- організація кількох веб-сайтів на одному сервері з використанням «alias адрес»;
- організація піддоменів на одній «alias адресі»;
- винесення сервісів на окремі сервери, для забезпечення розподілу навантаження та більш ефективної організації розподілення системних ресурсів.

## 2.2 Мета створення платформи

Метою є розробка стенду програмно-апаратної платформи «Wisdom in Knowledge», який являє собою набір сервісів, що забезпечують функціонування та доступ веб-сайту з мережі інтернет. Веб-сайт розробляється як окремий об'єкт та забезпечує комунікацію користувачів за допомогою розмежування на ролі, які мають різні права доступу до функцій сайту, серед яких:

- перегляд курсу;
- створення курсу;
- редагування курсу;
- видалення курсу;
- виконання тестів.

Користувач, який здійснює адміністрування програмно-апаратної платформи навчання забезпечується технологіями віддаленого менеджменту до яких входять:

- PhpMyAdmin – веб-додаток встановлений на веб-сервері, дозволяє конфігурувати базу даних використовуючи графічний інтерфейс доступ до якої здійснюється за доменою адресою платформи.

- FileZilla – програмне забезпечення для взаємодії з файловим сервером на якому розміщені каталоги з файлами веб-сайту.
- PuTTY – програмне забезпечення для віддаленого доступу до терміналу ОС, на якій базується мульти-сервісна платформа.

## 2.3 Характеристика об'єкту

### 2.3.1 Основні задачі та функції об'єкту

До основних задач, які має виконувати програмно-апаратна навчальна платформа «Wisdom in Knowledge» належить динамічне генерування документів, які являються собою користувацькі інтерфейси. Динамічне генерування документів містить підключення до бази даних для зчитування даних необхідних для заповнення динамічно генерованого документу. Виконання такої дії здійснюється на основі прав доступу користувача, які встановлюються його роллю.

Для досягнення мети роботи необхідно розв'язати ряд задач, основні з яких полягають в обґрунтуванні апаратного та програмного забезпечення, проектуванні структурних схем: стек LAMP; мережева взаємодія Client-Server, каркас MVC. Також передбачено організацію налаштування сервісів та розробку програмного забезпечення.

Розроблюване ПЗ здійснює поділ користувачів на ролі викладача і студента, кожна з яких передбачає свої можливості та права.

Доступ до даних користувачів та сервера повинен бути авторизованим та здійснюватись по захищеному протоколу зв'язку.

### 3 Вимоги до системи

#### 3.1 Вимоги до платформи в цілому

При зверненні до веб-сайту за доменним ім'ям користувач повинен отримати відповідь від сервера в якій міститься документ, що відповідає запиту URL стрічки.

##### 3.1.1 Вимоги до структури та функціонування системи

Структура програмно-апаратної навчальної платформи «Wisdom in Knowledge» містить наступне:

- сервер;
- веб-сервер;
- ftp-сервер;
- ssh-сервер;
- ssl сертифікати;
- програмне забезпечення яке містить користувацькі інтерфейси;
- реляційна база даних для зберігання інформації користувачів та веб-сайту в цілому;

До основних функціональних вимог, які висуваються до програмно-апаратної навчальної платформи «Wisdom in Knowledge» належить:

- створення облікового запису користувача;
- збереження персональних даних на сервері;
- комунікації по захищеному протоколу;
- віддалене адміністрування сервера;
- віддалений доступ до операційної системи та сервісів встановлених на сервері;

– віддалене вивантаження та завантаження даних сервера по захищеному каналу зв'язку.

### 3.1.2 Вимоги до способів та засобів зв'язку між компонентами платформи

Комунікація користувача з веб-сайтом здійснюється за допомогою захищеного протоколу HTTPS шляхом звернення за адресою: <https://wike.ml>. Веб-сайту працює на основі веб-сервера, який виконує підключення та здійснює запити до бази даних. В цілому архітектура програмно-апаратної навчальної платформи «Wisdom in Knowledge», являє собою «клієнт-сервер»

### 3.1.3 Вимоги по діагностуванню програмно-апаратної навчальної платформи «Wisdom in Knowledge»

Діагностування програмно-апаратної навчальної платформи «Wisdom in Knowledge» проводиться адміністратором серверу.

### 3.1.4 Перспективи розвитку, модернізація програмно-апаратної навчальної платформи

До перспектив розвитку програмно-апаратної платформи входить:

- адаптація властивостей веб-сайту у відповідності до функціональних вимог.
- Винесення сервісів в окремі сервери;
- Нарощування кількості веб-сайтів та їх піддоменів з можливістю організації окремого доступу до кожного;
- Збільшення кількості веб-сайтів без змін в апаратному забезпеченні сервера.
- Інтеграція нових сервісів.

### 3.1.5 Вимоги до надійності програмно-апаратної навчальної платформи «Wisdom in Knowledge»

Платформа повинна забезпечувати авторизований віддалений доступ до налаштування ОС та файлів розміщених на сервері.

Веб-сайт повинен містити систему авторизації та поділ користувачів на групи за допомогою ролей на основі яких здійснюється доступ до персональних даних, кожного.

### 3.1.6 Вимоги до функцій та задач, які виконує програмно-апаратна навчальна платформа «Wisdom in Knowledge»

До основних вимог, щодо функцій і задач, які повинна задовільняти і виконувати програмно-апаратна навчальна платформа «Wisdom in Knowledge», належить:

- комунікація клієнта і сервера через мережу інтернет на основі захищеного протоколу;
- відображення документів користувачу у відповідності до його прав;
- створення користувача в базі даних;
- збереження даних користувачів в базі даних;

### 3.1.7 Вимоги до апаратного забезпечення

Мінімальні вимоги до апаратного забезпечення сервера:

- процесор: Intel Xeon E2697 2.7GHz або AMD Opteron 6176 2.3GHz;
- графічний прискорювач: GeForce 7600gt або аналогічні;
- оперативна пам'ять об'ємом: 32Gb DDR3 1333 в чотирьох канальному режимі роботи.
- накопичувачі: 2TB Gb HDD (RAID 4).

Мінімальні вимоги до апаратного забезпечення користувача:

- персональний комп'ютер користувача:
  - а) процесор: Intel Celeron G1840 2.8GHz або Amd A8-6500 3.5GHz;
  - б) графічний прискорювач: Nvidia GeForce 9600gt або Amd hd 4670;
  - в) Оперативна пам'ять об'ємом: 1536mb.
- смартфон користувача:
  - а) процесор з тактовою частотою не нижче 1.2 GHz;
  - б) графічний прискорювач: Mali-400 MP2;
  - в) оперативна пам'ять об'ємом: 768mb.

### 3.1.8 Вимоги до програмного забезпечення

Програмне забезпечення сервера Linux Ubuntu Server, веб-сервер Apache, РСКБД MySQL, JavaScript ES6, SFTP сервер, openSSH сервер, Hypertext Preprocessor та його модулі.

Програмне забезпечення реалізації платформи: Visual studio code

Програмне забезпечення клієнта – веб-браузери: google chrome; opera; mozilla firefox; safari.

## 4 Вимоги до документації

Документація повинна відповідати вимогам ЄСКД та ДСТУ

Комплект документації повинен складатись з:

- пояснювальної записки;
- графічного матеріалу:
  - 1 LAMP Client-Server
  - 2 Мережева взаємодія.
  - 3 Каркас MVC.

\*Примітка: У комплект документації можуть вноситися міни та доповнення в процесі розробки.

## 5 Стадії та етапи проектування

Таблиця 1 – Стадії та етапи виконання кваліфікаційної роботи бакалавра

№ етапу	Назва етапу виконання кваліфікаційної роботи	Термін виконання
1	Розробка і затвердження технічного завдання	10.02-16.02.2021
2	Аналіз технічного завдання	17.02-02.03.2021
3	Визначення вимог до апаратного та програмного забезпечення навчальної платформи «Wisdom in Knowledge»	03.03-18.03.2021
4	Проектування схем програмно-апаратної навчальної платформи «Wisdom in Knowledge»	19.03-04.04.2021
5	Налаштування сервісів платформи	04.04-02.05.2021
6	Розробка програмного забезпечення веб-сайту	02.05-29.05.2021
7	Безпека життєдіяльності, основи охорони праці	01.06-08.06.2021
8	Оформлення кваліфікаційної роботи	09.06-18.06.2021
9	Попередній захист кваліфікаційної роботи	18.06-22.06.2021
10	Захист кваліфікаційної роботи	22.06-28.06.2021

## 6 Додаткові умови виконання кваліфікаційної роботи

Під час виконання кваліфікаційної роботи у дане технічне завдання можуть вноситися зміни та доповнення.



Додаток Б  
Таблиці БД

Розроблена таблиця «User» з відповідними полями до рис. 2.2 наведено в табл. Б.1.

Таблиця Б.1 – User

Індекс	Ім'я поля	Тип
PK	ID_User	Varchar(30)
	Name	Varchar(30)
	Login	Varchar(30)
	Email	Varchar(30)
	Password	Varchar(30)
	Status	Varchar(8)

Розроблена таблиця «Course» з відповідними полями до рис. 2.2 наведено в табл. Б.2.

Таблиця Б.2 – Course

Індекс	Ім'я поля	Тип
PK	ID_Course	Varchar(30)
FK	ID_Quiz	Varchar(30)
FK	ID_User	Varchar(30)
	Title	Varchar(30)
	Description	Text
	Image	Varchar(30)
	Video	Varchar(30)

Розроблена таблиця «Student\_Course» з відповідними полями до рис. 2.2 наведено в табл. Б.3.

Таблиця Б.3 – Student\_Course

Індекс	Ім'я поля	Тип
PK	ID_User	Varchar(30)
PK	ID_Course	Varchar(30)

Розроблена таблиця «Course\_Status» з відповідними полями до рис. 2.2 наведено в табл. Б.4.

Таблиця Б.4 – Course\_Status

Індекс	Ім'я поля	Тип
FK	ID_Course	Varchar(30)
FK	ID_User	Varchar(30)
	Status	Varchar(11)

Розроблена таблиця «Quiz» з відповідними полями до рис. 2.2 наведено в табл. Б.5.

Таблиця Б.5 – Quiz

Індекс	Ім'я поля	Тип
PK	ID_Quiz	Varchar(30)
FK	ID_Course	Varchar(30)
FK	ID_Question	Varchar(30)
FK	ID_Answer	Varchar(30)

Розроблена таблиця «Course» з відповідними полями до рис. 2.2 наведено в табл. Б.6.

Таблиця Б.6 – Question

Індекс	Ім'я поля	Тип
PK	ID_Question	Varchar(30)
FK	ID_Answer	Varchar(30)
	Text	Text

Розроблена таблиця «Answer» з відповідними полями до рис. 2.2 наведено в табл. Б.7.

Таблиця Б.7 – Answer

Індекс	Ім'я поля	Тип
PK	ID_Answer	Varchar(30)
FK	ID_Question	Varchar(30)
	Text	Text

Розроблена таблиця «Course» з відповідними полями до рис. 2.2 наведено в табл. Б.8.

Таблиця Б.8 – Quiz\_Result

Індекс	Ім'я поля	Тип
FK	ID_Quiz	Varchar(30)
FK	ID_User	Varchar(30)
	Result	Varchar(7)

## Додаток В

## Лістинг фронтенд, бекенд

```

//
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-
Compatible"
content="IE=edge">
  <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
  <title>Home</title>
  <link rel="stylesheet"
href="style.css">
  <link rel="preconnect"
href="https://fonts.gstatic.
com">
  <link rel="preconnect"
href="https://fonts.gstatic.
com">
<link
href="https://fonts.googleap
is.com/css2?family=Poppins:i
tal,wght@0,100;0,200;0,300;0
,400;0,500;0,600;0,700;0,800
;0,900;1,100;1,200;1,300;1,4
00;1,500;1,600;1,700;1,800;1
,900&display=swap"
rel="stylesheet">
</head>
<body>
  <section class="toolbar-
section toolbar-section--
hide_on">
    <article
class="toolbar container">
      <div
class="toolbar__left">
        <span
class="toolbar__sm">Follow
Us On: <!-- sm -
social media -->
          <ul
class="toolbar__list">
            <li
class="toolbar__item">
              <a
href="#0"
class="toolbar__link">
                
              </a>
            </li>
            <li
class="toolbar__item">
              <a
href="#0"
class="toolbar__link">
                
              </a>
            </li>
            <li
class="toolbar__item">
              <a
href="#0"
class="toolbar__link">
                
              </a>
            </li>
            <li
class="toolbar__item">
              <a
href="#0"
class="toolbar__link">
                
              </a>
            </li>
            <li
class="toolbar__item">

```

```

                                <a
href="#"#0"
class="toolbar__link">


                                </a>
                                </li>
                                </ul>
                                </div>
                                <div
class="toolbar__right">
                                <a
class="account-link"
href="#"#0">Create an
Account</a>
                                <a
class="btn-link"
href="#"#0">Log In</a>
                                </div>
                                </article>
                                </section>
                                <header class="header">
                                <div
class="container
header__container">
                                
                                <div
class="header__menu">
                                <nav
class="menu">
                                <ul
class="menu__list">
                                <li
class="menu__item">
                                <a href="" class="menu__link
menu__link--active">Home</a>
                                </li>
                                <li
class="menu__item">
                                <a href=""
class="menu__link">Courses</a>
                                </li>
                                <li
class="menu__item">
                                <a href=""
class="menu__link">Events</a>
                                </li>
                                <li
class="menu__item">
                                <a href=""
class="menu__link">Pages</a>
                                </li>
                                <li
class="menu__item">
                                <a href=""
class="menu__link">Blog</a>
                                </li>
                                <li
class="menu__item">
                                <a href=""
class="menu__link">Contact</a>
                                </li>
                                </ul>
                                </nav>
                                <form
class="search-form
header__search-form"
action="#"#0">
                                <input
class="search-form__text"
type="search"
placeholder="Search">
                                <button
class="search-
form__find"></button>
                                </form>
                                </div>
                                </div>
                                </header>
                                <section class="intro
carousel">
                                <section
class="carousel__list">

```

```

        <div
class="carousel__item">
        <article
class="info-box info-box--
center">
            <span
class="info-box__label info-
box__label--center">Start to
learning Today</span>
            <h1
class="info-box__title info-
box__title--center info-
box__title--white">Excellent
And Friendly <br>Faculty
Members</h1>
            <p
class="info-box__text info-
box__text--center info-
box__text--white">Lorem
Ipsum is simply dummy text
of the printing and
typesetting <br>
industry. Lorem Ipsum has
been the industry's standard
<br>
dummy text ever since an to
impression.</p>
            <div
class="info-box__btn info-
box__btn--center">
                <a
href="#0" class="btn
btn__style--def">Learn
More</a>
                <a
href="#0" class="btn
btn__style--alt">Our
Events</a>
            </div>
        </article>
    </div>
    <div
class="carousel__item">
        <article
class="info-box info-box--
center">
            <span
class="info-box__label info-
box__label--center">Start to
learning Today</span>
            <h1
class="info-box__title info-
box__title--center info-
box__title--white">Your
Ideas Will Be <br>Heard &
Supported</h1>
            <p
class="info-box__text info-
box__text--center info-

```

```

        <h1
class="info-box__title info-
box__title--center info-
box__title--
white">Innovation Paradise
<br>For Students</h1>
            <p
class="info-box__text info-
box__text--center info-
box__text--white">Lorem
Ipsum is simply dummy text
of the printing and
typesetting <br>
industry. Lorem Ipsum has
been the industry's standard
<br>
dummy text ever since an to
impression.</p>
            <div
class="info-box__btn info-
box__btn--center">
                <a
href="#0" class="btn
btn__style--def">Learn
More</a>
                <a
href="#0" class="btn
btn__style--alt">Our
Events</a>
            </div>
        </article>
    </div>
    <div
class="carousel__item">
        <article
class="info-box info-box--
center">
            <span
class="info-box__label info-
box__label--center">Start to
learning Today</span>
            <h1
class="info-box__title info-
box__title--center info-
box__title--white">Your
Ideas Will Be <br>Heard &
Supported</h1>
            <p
class="info-box__text info-
box__text--center info-

```

```

box__text--white">Lorem
Ipsum is simply dummy text
of the printing and
typesetting <br>

industry. Lorem Ipsum has
been the industry's standard
<br>

dummy text ever since an to
impression.</p>
      <div
class="info-box__btn info-
box__btn--center">
      <a
href="#" class="btn
btn__style--def">Learn
More</a>
      <a
href="#" class="btn
btn__style--alt">Our
Events</a>
      </div>
    </article>
  </div>
</section>
<div
class="carousel__btns-case">
  <button
class="carousel__btn
carousel__btn--
active"></button>
  <button
class="carousel__btn"></butt
on>
  <button
class="carousel__btn"></butt
on>
  </div>
</section>
<section
class="services">
  <article
class="flat-card">
    <a href="#"
class="flat-
card__title">Trending
Courses</a>
    <p class="flat-
card__text">Lorem ipsum
dolor sit amet, consectetur
adipisicing elit.

```

```

Repellendus, quaerat beatae
nulla debitis vitae
temporibus sed.</p>
      <a href="#"
class="btn btn__style--frame
flat-card__btn">Explore</a>
    </article>
  <article
class="flat-card">
    <a href="#"
class="flat-
card__title">Trending
Courses</a>
    <p class="flat-
card__text">Lorem ipsum
dolor sit amet, consectetur
adipisicing elit.
Repellendus, quaerat beatae
nulla debitis vitae
temporibus sed.</p>
    <a href="#"
class="btn btn__style--frame
flat-card__btn" >Explore</a>
  </article>
<article
class="flat-card">
  <a href="#"
class="flat-
card__title">Trending
Courses</a>
  <p class="flat-
card__text">Lorem ipsum
dolor sit amet, consectetur
adipisicing elit.
Repellendus, quaerat beatae
nulla debitis vitae
temporibus sed.</p>
  <a href="#"
class="btn btn__style--frame
flat-card__btn">Explore</a>
</article>
</section>
<section class="about">
  <div
class="about__container
container">
    <article
class="info-box info-box--
left about__info-box">
      <span
class="info-box__label info-

```

```

box__label--left">Start to
learning Today</span>
      <h1
class="info-box__title info-
box__title--left info-
box__title--black">Welcome
To Our Campus</h1>
      <p
class="info-box__text info-
box__text--left info-
box__text--black">Lorem
Ipsum is simply dummy text
of the printing and
typesetting industry. Lorem
Ipsum has been the
industry's standard dummy
text ever since an to
impression.</p>
      <p
class="info-box__text info-
box__text--left info-
box__text--black">Lorem
Ipsum is simply dummy text
of the printing and
typesetting industry. Lorem
Ipsum has been the
industry's standard dummy
text ever since an to
impression.</p>
      <div
class="info-box__btn info-
box__btn--left">
      <a
href="#0" class="btn
btn__style--black">Read
More</a>
      <a
href="#0" class="btn
btn__style--black
btn__style--play">Play
Video</a>
      </div>
    </article>
    
  </div>
</section>
<section class="courses-
grid container">
      <article
class="section-title
courses-grid__section-
title">
      
      <h2
class="section-
title__title">Featured
Courses</h2>
      <p
class="section-
title__text">There are many
variations of passages of
Lorem Ipsum available, but
the majority have suffered
alteration in some form.</p>
    </article>
    <div class="courses-
grid__course-card">
      <article
class="course-card">
      <span
class="course-
card__price">$180</span>
      <a
class="course-card__link"
href="#0">
      
      </a>
      <div
class="course-card__desc">
      <a
class="course-card__title"
href="#0">Computer
Science</a>
      <p
class="course-
card__text">Excepteur sint
occaecat cupidatat non
proident, sunt in culpa qui
officia deserunt
laborum.</p>
      </div>
      <div
class="course-card__tag">
      <ul
class="course-card__rating">

```



```

class="course-
card__star"></li>
class="course-
card__star"></li>
class="course-
card__star"></li>
class="course-
card__star"></li>
class="course-
card__star"></li>
class="course-card__star
course-card__star--
active"></li>
class="course-
card__reviews">22
Reviews</li>
</ul>
<a
class="tag"
href="#0">Programming</a>
</div>
</article>
<article
class="course-card">
<span
class="course-
card__price">$180</span>
<a
class="course-card__link"
href="#0">

</a>
<div
class="course-card__desc">
<a
class="course-card__title"
href="#0">Computer
Science</a>
<p
class="course-
card__text">Excepteur sint
occaecat cupidatat non
proident, sunt in culpa qui
officia deserunt
laborum.</p>
</div>
</div>
class="couse-card__tag">
<ul
class="course-card__rating">
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-card__star
course-card__star--
active"></li>
<li
class="course-card__star
course-card__star--
active"></li>
<li
class="course-card__star
course-card__star--
active"></li>
</ul>
<a
class="course-
card__reviews">22
Reviews</li>
</ul>
<a
class="tag"
href="#0">Programming</a>
</div>
</article>
<article
class="course-card">
<span
class="course-
card__price">$180</span>
<a
class="course-card__link"
href="#0">

</a>
<div
class="course-card__desc">
<a
class="course-card__title"
href="#0">Computer
Science</a>
<p
class="course-
card__text">Excepteur sint
occaecat cupidatat non
proident, sunt in culpa qui
officia deserunt
laborum.</p>
</div>

```

```

card__text">Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt
laborum.</p>
</div>
<div
class="couse-card__tag">
<ul
class="course-card__rating">
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-card__star
course-card__star--
active"></li>
<li
class="course-
card__reviews">22
Reviews</li>
</ul>
<a
class="tag"
href="#0">Programming</a>
</div>
</article>
<article
class="course-card">
<span
class="course-
card__price">$180</span>
<a
class="course-card__link"
href="#0">

</a>
<div
class="course-card__desc">
<a
class="course-card__title"
href="#0">Computer
Science</a>
<p
class="course-
card__text">Excepteur sint
occaecat cupidatat non
proident, sunt in culpa qui
officia deserunt
laborum.</p>
</div>
<div
class="couse-card__tag">
<ul
class="course-card__rating">
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-
card__star"></li>
<li
class="course-card__star
course-card__star--
active"></li>
<li
class="course-
card__reviews">22
Reviews</li>
</ul>
<a
class="tag"
href="#0">Programming</a>
</div>
</article>
<article
class="course-card">
<span
class="course-
card__price">$180</span>
<a
class="course-card__link"
href="#0">

    </a>
    <div
class="course-card__desc">
    <a
class="course-card__title"
href="#0">Computer
Science</a>
    <p
class="course-
card__text">Excepteur sint
occaecat cupidatat non
proident, sunt in culpa qui
officia deserunt
laborum.</p>
    </div>
    <div
class="couse-card__tag">
    <ul
class="course-card__rating">
    <li
class="course-
card__star"></li>
    <li
class="course-
card__star"></li>
    <li
class="course-
card__star"></li>
    <li
class="course-
card__star"></li>
    <li
class="course-card__star
course-card__star--
active"></li>
    <li
class="course-
card__reviews">22
Reviews</li>
    </ul>
    <a
class="tag"
href="#0">Programming</a>
    </div>
    </article>
    </div>
    </section>
    <section
class="statistic">
    <article
class="simple-text">
    <span
class="simple-
text__head">500+</span>
    <h4
class="simple-
text__desc">Happy
Clients</h4>
    </article>
    <article
class="simple-text">
    <span
class="simple-
text__head">70+</span>
    <h4
class="simple-
text__desc">Online
Courses</h4>
    </article>
    <article
class="simple-text">
    <span
class="simple-
text__head">100%</span>
    <h4
class="simple-
text__desc">Satisfaction</h4
>
    </article>
    <article
class="simple-text">
    <span
class="simple-
text__head">100%</span>
    <h4
class="simple-
text__desc">Support</h4>
    </article>
    </section>
    <section class="events">
    <article
class="section-title
events__section-title">
    
    <h2
class="section-
title__title">Upcoming
Events</h2>

```





```

        <h2
class="newsletter__title">Th
e Newsletter</h2>
        <p
class="newsletter__text">Sub
scribe to us to always stay
in touch with us and get the
latest news about our
company and all of our
activities!</p>
        <form
class="newsletter__form"
action="#0">
                <input
class="newsletter__input"
type="text">
                <button
class="btn btn__style--
green"
type="submit">Subscribe
Now!</a>
        </form>
    </article>
    <ul class="social-
media newsletter-
section__social-media">
        <li
class="social-
media__item"></li>
        <li
class="social-
media__item"></li>
        <li
class="social-
media__item"></li>
        <li
class="social-
media__item"></li>
        <li
class="social-
media__item"></li>
    </ul>
</section>
<section class="self-
course">
    <article
class="info-title self-
course__info-title">
        <span
class="info-
title__about">SiGET INSTANT
ACCESS TO THE FREE</span>

```

```

        <h2 class="info-
title__title">Self
Development Course</h2>
        <p class="info-
title__text">Learn about how
them you went down prying
the wedding ring off his
cold, dead finger. I don't
know what you did, Fry, but
once again, you screwed
up!</p>
                <a href="#0"
class="btn btn__style--
white">Start Learning</a>
    </article>
</section>
<section class="latest-
news">
    <article
class="section-title latest-
news__section-title">
        
        <h2
class="section-
title__title">Latest News &
Blog</h2>
        <p
class="section-
title__text">There are many
variations of passages of
Lorem Ipsum available, but
the majority have suffered
alteration in some form.</p>
    </article>
    <div class="latest-
news__post-card">
        <article
class="post-card">
            
            <ul
class="post-card__tag">
                <li
class="post-card__item">
                    <a
class="tag" href="#0">Online
Courses</a>
                </li>

```

```

        <li
class="post-card__item">
        <a
class="tag-data"
href="#0"><time
datetime="2021-05-25
12:00">June 25,
2021</time></a>
        </li>
    </ul>
    <a
class="post-card__link"
href="#0">How to Study
Online Courses
Effectively</a>
    <p
class="post-
card__text">Lorem Ipsum is
simply dummy text of the
printing and typesetting
industry. Lorem Ipsum has
been the industry's.</p>
        <a href="#0"
class="btn btn__style--
black">Read More</a>
    </article>
</div>
</section>
<section
class="services-list
carousel--auto">
    <article
class="service">
        
    </article>
    <article
class="service">
        
    </article>
    <article
class="service">
        
    </article>
</section>
<section class="footer-
info">
        <div class="footer-
info--flex">
            <section
class="about">
                <div
class="logo about__logo">
                    <img
src="" alt="logo">
                </div>
                <p
class="about__text">Nemo
enim enim voluptatem quia
voluptas sit aspernatur aut
odit aut fugit, sed quia
consequ magni dolores eos
qui ratione.</p>
                <ul
class="social-media
about__social-media">
                    <li
class="social-
media__item"></li>
                    <li
class="social-
media__item"></li>
                    <li
class="social-
media__item"></li>
                    <li
class="social-
media__item"></li>
                    <li
class="social-
media__item"></li>
                </ul>
            </section>
            <section
class="latest-news-min">
                <h3
class="latest-news-
min__title">Latest News</h3>
                <article
class="post-card-min">
                    
                    <ul
class="post-card-min__list">
                        <li
class="post-card-min__item">
                            <a class="post-card-

```

```

min__title" href="#0">Top 10
books you Must read in
2021</a>
</li>
<li
class="post-card-min__item">
<a class="tag-data"
href="#0"><time
datetime="2021-05-25
12:00">June 25,
2021</time></a>
</li>
</ul>
</article>
<article
class="post-card-min">

<ul
class="post-card-min__list">
<li
class="post-card-min__item">
<a class="post-card-
min__title" href="#0">Top 10
books you Must read in
2021</a>
</li>
<li
class="post-card-min__item">
<a class="tag-data"
href="#0"><time
datetime="2021-05-25
12:00">June 25,
2021</time></a>
</li>
</ul>
</article>
<section>
class="course-min-list">
<ul
class="course-min__list">
<li
class="course-min__item">
<a
class="course-min__link"
href="#0"></a>
</li>
<li
class="course-min__item">
<a
class="course-min__link"
href="#0"></a>
</li>
<li
class="course-min__item">
<a
class="course-min__link"
href="#0"></a>
</li>
</ul>
</section>
<section
class="newsletter-min">
<article
class="newsletter
newsletter-min__newsletter">
<h2
class="newsletter__title">Th
e Newsletter</h2>
<p
class="newsletter__text">Sub
scribe to us to always stay
in touch with us and get the
latest news about our
company and all of our
activities!</p>
<form
class="newsletter__form"
action="#0">
<input
class="newsletter__input"
type="text">

```



```

<button class="btn
btn_style--green"
type="submit">Subscribe
Now!</a>
        </form>
    </article>
</section>
</div>
<footer
class="footer">
        <p
class="copyright
footer_copyright">Designed
and Developed by Wisdom in
Knowledge</p>
    </footer>
</section>
<script
src="carousel.js"></script>
</body>
</html>
// -----
<?php
require
'application/lib/Dev.php';

use application\core\Router;
use application\lib\Db;

spl_autoload_register(function ($class)
{
    $Path =
str_replace('\\', '/',
$class . '.php');

    if (file_exists($Path))
    {
        require $Path;
    }
});

session_start();

$router = new Router;
$router->runRoute();
// $db = new Db;
?>
// -----
<?php
namespace application\core;

use application\core\View;

abstract class Controller {

    private $route;
    protected $view;
    protected $model;
    protected $acl;

    public function
__construct($route)
    {
        $this->route =
$route;

        if (!$this->checkAcl()) {
            View::errorCode(403);
        }
        $this->view = new
View($route);
        $this->model =
$this->loadModel($route['controller']);
        //debug($this->model);
    }

    public function
loadModel($name) {
        $path =
'application\models\\' .
ucfirst($name);
    }
}
// $_SESSION['authorize']['id'] = 1;

```

```

        if
(class_exists($path)) {
            return new
$path();
        }

        public function
checkAcl() {
            $this->acl = require
'application/acl/' . $this-
>route['controller'] .
'.php';
            if ($this-
>isAcl('all')) {
                return true;
            } elseif
(isset($_SESSION['authorize'
]['id']) and $this-
>isAcl('authorize')) {
                return true;
            } elseif
(!isset($_SESSION['authorize
?>

// -----
<?php                                     {
                                            $this->db = new
namespace application\core;             Db();
                                            }
use application\lib\Db;
abstract class Model {                   }
    protected $db;
                                            ?>
    public function
__construct()

// -----
<?php                                     public function
namespace application\core;             __construct($route)
class View {                               {
    private $path;                        $this->route =
    private $route;                       $route;
    private $layout =                    $this->path =
'default';                               $route['controller'] . '/' .
                                            $route['action'];
                                            }

```

```

        public function
render($title, $vars = []) {
    extract($vars);
    $path =
'application/views/' .
$this->path . '.php';
    if
(file_exists($path)) {
        ob_start();
        require $path;
        $content =
ob_get_clean();
        require
'application/views/layouts/'
. $this->layout . '.php';
    } else {
        echo 'no exists
file ' . $this->path;
    }
}

    public function
redirect($url) {
        header('location: '
. $url);
        exit;
    }

    public static function
errorCode($code) {

http_response_code($code);
        require
'application/views/errors/'
. $code . '.php';
    }

// -----

<?php
namespace application\core;

class Router {

    protected $routes = [];
    protected $params = [];

    public function
__construct() {
        $arr = require
'application/config/routes.p
hp';

        foreach ($arr as
$key => $val) {
            $this-
>addRoute($key, $val);
        }
        //debug($this-
>routes);
    }

    public function
addRoute($route, $params) {
        $route = '#^' .
$route . '$#';
    }
}
exit;
}

    public function
message($status, $message) {

exit(json_encode(['status'
=> $status, 'message' =>
$message]));
}

    public function
location($url) {

exit(json_encode(['url' =>
$url]));
}

    public function
setPath($path) {
        $this->path = $path;
    }

    public function
setLayout($layout) {
        $this->layout =
$layout;
    }
}

?>

```

```

        $this->
>routes[$route] = $params;
    }

    public function
matchRoute() {
    $url =
trim($_SERVER['REQUEST_URI']
, '/');
    foreach ($this->
>routes as $route =>
$params) {
        if
(preg_match($route, $url,
$matches)) {
            $this->
>params = $params;
            return true;
        }
    }
    return false;
}

    public function
runRoute() {
    if ($this->
>matchRoute()) {
        $path =
'application\controllers\'
. ucfirst($this->
>params['controller']).
'Controller';

```

```

        if
(class_exists($path)) {
            $action =
$this->params['action'] .
'Action';
            if
(method_exists($path,
$action)) {
                $controller = new
$path($this->params);
                $controller->$action();
            } else {
                View::errorCode(404);
            }
        } else {
            View::errorCode(404);
        }
    } else {
        View::errorCode(404);
    }
} else {
    View::errorCode(404);
}
}
}
?>

```