

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук та технологій
(повне найменування факультету)

Кафедра комп'ютерних систем та мереж
(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістра
(ступінь вищої освіти)

на тему **ІНФОРМАЦІЙНА СИСТЕМА АВТОШКОЛИ З ФУНКЦІЄЮ
ПІДВИЩЕННЯ РІВНЯ ЗНАЇНЬ СТУДЕНТІВ**
(назва теми)

Виконав: студент 2 курсу, групи КНТ-513м

Спеціальності 123 Комп'ютерна інженерія
(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні системи та мережі

БИЦЮТА К.В.
(ПРИЗВИЩЕ та ініціали)

Керівник КУДЕРМЕТОВ Р.К.
(ПРИЗВИЩЕ та ініціали)

Рецензент ЄФИМЕНКО М.В.
(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
Кафедра «Комп'ютерні системи та мережі»
Ступінь вищої освіти магістерський
Спеціальність 123 Комп'ютерна інженерія
(код і найменування)
Освітня програма (спеціалізація) «Комп'ютерні системи та мережі»
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Зав. кафедри Кудерметов Р.К.

“___” _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

БИЦЮТИ КИРИЛА ВІКТОРОВИЧА

(ПРИЗВИЩЕ, ім'я, по батькові)

- Тема проєкту (роботи) Інформаційна система автошколи з функцією підвищення рівня знань студентів
керівник проєкту (роботи) к. т. н., доцент, КУДЕРМЕТОВ Равіль Камілович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від « 18 » жовтня 2024 року № 149
- Строк подання студентом проєкту (роботи) 16 грудня 2024 року
- Вихідні дані до проєкту (роботи) перелік питань до тестування знань з ПДР, перелік тем і їх змісту, необхідних для складання іспиту з ПДР
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)) 1) Аналіз предметної області;
2) Моделювання програмного забезпечення;
3) Проєктування системи;
4) Реалізація вебресурсу.
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Презентація

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	КУДЕРМЕТОВ Р.К., доцент		
нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання « 9 » _____ вересня _____ 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз технічного завдання: вивчення предметної області, збір вимог до системи	10.10.2024 р.	
2	Моделювання програмного забезпечення: створення діаграм прецедентів і послідовності	20.10.2024 р.	
3	Проектування архітектури системи: вибір клієнт-серверної моделі, створення бази даних	01.11.2024 р.	
4	Розробка вебресурсу: створення інтерфейсу, програмування функціональних модулів	11.11.2024 р.	
5	Оформлення отриманих результатів та висновків у пояснювальній записці	20.11.2024 р.	
6	Оформлення графічного матеріалу	27.11.2024 р.	
7	Оформлення допоміжних матеріалів	12.12.2024 р.	

Студент

_____ **Кирило БИЦЮТА**
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ **Равіль КУДЕРМЕТОВ**
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 84 стор., 43 рис , 1 додаток, 19 джерел.

CSS, HTML, JAVASCRIPT, MYSQL, PHP, WORDPRESS, АВТОШКОЛА,
ВЕБРЕСУРС, КЛІЄНТ-СЕРВЕР, СУБД, ЧАТ-БОТ

Об'єктом дослідження є інтегрована інформаційна система для автошколи, яка автоматизує ключові аспекти її діяльності.

Метою роботи є підвищення якості обслуговування клієнтів та спрощення адміністративних процесів автошколи за рахунок розробки сучасної інформаційної системи. Основним завданням є створення багатофункціонального вебресурсу, що дозволяє учням зручно реєструватися, отримувати доступ до навчальних матеріалів, проходити автоматизовані тести, взаємодіяти з викладачами, а також отримувати рекомендації щодо покращення знань через інтегрований чат-бот.

Структура роботи включає чотири основні частини. Перша частина присвячена аналізу предметної області, де виконано порівняння існуючих аналогів і сформульовано вимоги до інформаційної системи.

У другій частині здійснено моделювання програмного забезпечення, зокрема розроблено діаграми прецедентів і послідовності.

Третя частина охоплює проєктування архітектури, що передбачає вибір клієнт-серверної моделі, відповідних технологій і бази даних для реалізації.

Завершує проєкт четверта частина з реалізацією вебресурсу, де створено інтерфейс, функціональні модулі системи та інтегровано чат-бот для підтримки навчального процесу.

ABSTRACT

Explanatory note to the master's work: 84 p., 43 figures, 1 supplement, 19 sources.

DRIVING SCHOOL, WEB RESOURCE, CLIENT-SERVER, DBMS, CHATBOT, HTML, CSS, JAVASCRIPT, MYSQL, PHP, WORDPRESS

The object of the study is an integrated information system for a driving school, which automates key aspects of its activities.

The subject of the study is computer-based testing systems in driving schools.

The aim of the work is to improve the quality of customer service and simplify administrative processes in the driving school through the development of a modern information system. The main task is to create a multifunctional web resource that allows students to easily register, access learning materials, take automated tests, interact with instructors, and receive recommendations for improving their knowledge via an integrated chatbot.

The structure of the work includes four main parts. The first part is dedicated to the analysis of the subject area, where a comparison of existing analogs is made and requirements for the information system are formulated.

The second part involves software modeling, including the development of use case and sequence diagrams.

The third part covers system architecture design, which includes choosing a client-server model, appropriate technologies, and a database for implementation.

The final part of the project is the implementation of the web resource, where the interface is created, the system's functional modules are developed, and a chatbot is integrated to support the learning process.

ЗМІСТ

Перелік скорочень	
Вступ	8
1 Аналіз предметної області	10
1.1 Опис предметної області	10
1.2 Вимоги до вебресурсу	13
1.2.1 Зовнішній інтерфейс	18
1.2.2 Функціональні вимоги	20
1.2.3 Вимоги до робочих характеристик	21
1.2.4 Атрибути якості програмного забезпечення	21
1.2.5 Функції вебсистеми	22
2 Моделювання програмного забезпечення	23
2.1 Моделювання прецедентів	23
2.2 Моделювання діаграм послідовності	27
3 Проєктування системи	31
3.1 Архітектура програмного забезпечення	31
3.3 Проєктування бази даних	39
4 Реалізація вебресурсу	47
4.1 Вибір інструментів розробки	47
4.2 Створення бази даних	48
4.3 Програмна реалізація вебсервісу	54
4.3.1 Розробка загальної частини системи	54
4.3.2 Розробка особистого кабінету учня	57
4.3.4 Розробка телеграм-бота для перевірки тестування	67
4.4 Інтерфейс користувача	71
Висновки	78
Перелік джерел посилання	79
Додаток а	81

ПЕРЕЛІК СКОРОЧЕНЬ

API – Application Programming Interface;
NTFS – New Technology File System;
PHP – Personal Home Page Tools;
IDE – Integrated Development Environment;
SQL – Structured Query Language;
CSS – Cascading Style Sheets;
HTML – Hyper Text Markup Language;
СУБД – Система управління базами даних;
UML – Unified Modeling Language;
MVC – Model-View-Controller;
ER – Entity-Relationship;
CMS – Content Management System;
SEO – Search Engine Optimization.

ВСТУП

З розвитком сучасних інтернет-технологій вебсайти та онлайн-системи стали не лише інструментом представлення інформації, але й важливими засобами автоматизації бізнес-процесів. Сьогодні для успішного функціонування будь-якої компанії, зокрема автошкіл, необхідно забезпечити клієнтам можливість взаємодії з послугами через вебінтерфейс, що включає запис на заняття, отримання зворотнього зв'язку, доступ до навчальних матеріалів і тестування.

Інформаційна система для автошколи призначена спростити процес управління навчанням, зменшити адміністративне навантаження та підвищити зручність для учнів. Окрім стандартних функцій, таких як реєстрація на практичні та теоретичні заняття, система також впроваджує модуль тестування. Цей модуль дозволяє учням проходити внутрішні іспити з теоретичних аспектів водіння, зокрема правил дорожнього руху, надання першої медичної допомоги та основ безпеки руху, що забезпечує оперативний контроль знань та підготовку до державних іспитів.

Автоматизоване тестування є важливим компонентом навчального процесу в автошколі. Для учнів – це можливість зручно скласти теоретичні тести, а інструкторам – отримувати точні результати та аналізувати прогрес кожного учня. Завдяки цьому, адміністрація автошколи має змогу своєчасно коригувати навчальний процес, покращуючи підготовку майбутніх водіїв.

Статистика показує, що майже 40% користувачів по всіх частинах світу, якщо є вибір взаємодії з чат-ботом або реальною людиною, надають перевагу спілкування з віртуальним помічником. На сьогоднішній день, чат боти активно використовуються у бізнесі з цілю проведення консультацій та представлення затребуваної інформації користувачам.

Боти сприяють автоматизації спілкування з користувачами, дозволяючи їм більш точно та персоналізовано надавати потрібну інформацію. Крім того, використання чат-ботів дає змогу оптимізувати витрати, надавати цілодобову

підтримку та одночасно обслуговувати велику кількість користувачів без зниження якості обслуговування.

До того ж, застосування чат-ботів дозволяє знизити витрати, забезпечити цілодобову підтримку та надати свої послуги водночас для великої кількості клієнтів без втрат у якості. На них можна перекласти типові задачі, такі як ставити заздалегідь вказані питання та відповідати на них, шукати інформацію по запиту та виконувати інші базові доручення [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Автошкола – це навчальний заклад, який спеціалізується на підготовці водіїв різних категорій. Основне завдання автошколи – навчити майбутніх водіїв правилам дорожнього руху, технічним моментам експлуатації транспортних засобів, надання домедичної допомоги, та практичним навичкам керування автомобілем [2]. Крім того, автошколи проводять перепідготовку водіїв для отримання нових категорій прав.

Процес навчання в автошколі включає два основних компоненти: теоретичний і практичний курси. Теоретичний курс охоплює вивчення правил дорожнього руху, основ дорожньої безпеки та технічні характеристики транспортних засобів. Практичний курс зосереджений на розвитку навичок водіння як на автодромі, так і в умовах реального дорожнього руху.

Учні автошколи поділяються на групи, які проходять навчання за встановленим графіком, що включає теоретичні та практичні заняття. Для успішного завершення навчання кожен учень має скласти внутрішні іспити, після чого допускається до державних іспитів для отримання посвідчення водія.

В автошколі працює кілька ключових співробітників, кожен з яких виконує певні функції, необхідні для організації та проведення навчального процесу.

Директор автошколи відповідає за загально керівництво діяльністю автошколи, організовує документообіг, затверджує навчальні програми та графіки занять, контролює фінансові питання та визначає ціни на навчання, моніторить успішність учнів а загальну ефективність навчального процесу.

Секретар директора організовує прийом документів нових учнів, реєструє їх у системі, відповідає за розклад занять, його формування та оновлення, забезпечує зворотній зв'язок із учнями, адмініструє сайт автошколи, виконує доручення директор, допомагає в управлінні навчальним процесом.

Бухгалтер веде фінансовий облік та складає звітність, контролює оплату навчання учнями та обробляє платежі, проводить розрахунки з працівниками автошколи та іншими підрядниками, відповідає за фінансову прозорість та правильність ведення бухгалтерських операцій.

Інструктор з теоретичного навчання навчає учнів правилам дорожнього руху, основам безпеки та будови автомобіля, проводить теоретичні заняття, вебінари та заліки, організовує та проводить тестування знань учнів, аналізує їх успішність, розробляє та вдосконалює навчальні матеріали.

Інструктор з практичного навчання проводить практичні заняття з водіння на автодрому та в умовах реального міського руху, навчає учнів основам керування та маневрування, відповідає за моніторинг практичних навичок учнів, веде облік успішності, може проводити вебінари або надавати консультації з практичних аспектів водіння.

У сучасних умовах, автошколи впроваджують нові інформаційні системи для автоматизації ключових процесів, таких як запис на заняття, моніторинг успішності учнів та проведення тестування. Ці системи не лише спрощують адміністративні процеси, але й надають учням зручний доступ до навчальних матеріалів та графіків занять. Сучасна автошкола може також пропонувати інтерактивні вебінари, тренінги та відеоуроки, що дозволяє учням навчатись дистанційно. Це особливо актуально для теоретичних частини курсу, де учні можуть проходити навчання онлайн, а тестування проводити за допомогою інтегрованих цифрових систем.

У процесі навчання автошколи недостатньо просто складати тести з теоретичної частини – важливо також аналізувати результати та звертати вагу на ті питання, де були допущені помилки. Тестування само по собі є лише засобом контролю знань, проте без аналізу результатів та подальшого повторення складних тем, деякі незрозумілі для них моменти можуть в подальшому перетворитись на проблему як для них, так і для учасників дорожнього руху. Для підвищення якості навчання необхідно зосередитись на роботі з помилками,

систематично повертаючись до проблемних тем і доповнюючи знання там, де учень допустить неточності.

З цією метою пропонується створити чат-бот, який буде автоматично аналізувати результати тестування учнів та надавати рекомендації для подальшого вивчення матеріалу. Чат-бот може надавати пояснення до кожної неправильної відповіді, пропонуючи додаткові матеріали для вивчення конкретних тем. Такий підхід дозволить кожному учневі отримати індивідуальну підтримку під час підготовки, що значно підвищить ефективність навчального процесу та забезпечить краще засвоєння теоретичних знань.

Отже, на основі опису предметної області виділено наступні ключові задачі:

- реєстрація та управління обліковими записами користувачів (створення особистих кабінетів для учнів, викладачів та адміністрації, можливість реєстрації нових учнів та управління їхніми даними, надання доступу до навчальних матеріалів та записів на заняття через особисті кабінети);

- система онлайн тестування (автоматизоване тестування теоретичних знань з можливістю перевірки результатів у реальному часі, збереження результатів тестів для подальшого аналізу, можливість повторного тестування з різними варіантами питань);

- аналіз результатів тестування (автоматичний аналіз помилок після кожного тесту, надання учням індивідуальних рекомендацій на основі зроблених помилок, відображення прогресу учнів та підказки щодо того, які теми потребують додаткового вивчення);

- інтеграція чат-бота (створення чат-бота, який надаватиме учням пояснення до неправильно виконаних завдань, функція автоматичної рекомендації навчальних матеріалів на основі результатів тестів, можливість задавати питання чат-боту з отриманням швидких відповідей з теоретичних тем);

- модуль для запису на практичні та теоретичні заняття (можливість для учнів записуватись на заняття через веб-ресурс, автоматизоване формування розкладу занять з урахуванням кількості учнів та доступних інструкторів, сповіщення учнів про заплановані заняття та можливість перенесення;

- моніторинг успішності учнів (створення інструментів для відстеження успішності учнів за допомогою інструкторів та адміністрації, аналіз прогресу учнів на основі теоретичних та практичних занять;
- система навчальних матеріалів (доступ до теоретичних матеріалів, таких як відео уроки, лекції, інтерактивні тренінги, можливість самостійного вивчення та повторення матеріалів, інтеграція з чат-ботом для надання додаткових пояснень щодо матеріалів);
- система сповіщень та нагадувань (надсилання повідомлень учням про майбутні заняття, результати тестів та важливі події, нагадування про тести або повторення певних тем);
- інтерфейс для інструкторів (можливість перегляду результатів тестування кожного учня, оцінка прогресу учнів на практичних заняттях, налаштування графіків занять та контроль за їх виконанням).

1.2 Вимоги до вебресурсу

Вебресурс – це будь-який ідентифікований ресурс (цифровий, фізичний або абстрактний), наявний у мережі Інтернет або підключений до неї. Вони є невід’ємною частиною повсякденного життя для навчання, роботи, спілкування та розваг.

Вебресурси можуть бути представлені у вигляді вебсторінок, блогів, форумів, інтернет магазинів, освітніх платформ тощо, в залежності від типу і змісту. До них можна отримати доступ за допомогою веббраузера через URL-адресу.

Для того, щоб визначитись з вимогами до вебресурсу автошколи було проведено аналіз функціональності та інтерфейсу наступних сайтів автошкіл:

- автошкола «СТАРТ АВТО» [3];
- автошкола «GREEN-WAY» [4].

Автошкола «СТАРТ АВТО» – це онлайн-платформа автошколи, що надає послуги з навчання водінню в Запоріжжі. На цьому вебресурсі користувачі можуть:

- записатися на курси водіння подавши заявку на навчання в автошколі на різні категорії (А, В, С та інші);
- переглянути доступні дати та час проведення занять, а також обрати зручний для себе графік навчання;
- дізнатися про актуальні ціни на курси водіння;
- пройти онлайн-консультацію за допомогою форми зворотного зв'язку;
- отримати інформацію про теоретичні іспити;
- зареєструватися на практичні заняття;
- переглянути додаткові послуги, які надає автошкола, наприклад, підтримка під час складання державних іспитів або можливі акції та знижки.

На початковій сторінці сайту цієї автошколи можна відмітити наступні елементи:

- блок навігації по сайту;
- інформація про категорії водійських прав та ціни навчання;
- контактні телефони філіалів автошколи;
- графік груп для онлайн та офлайн навчання;
- розділи про філії автошколи в різних районах міста;
- спеціальні пропозиції та акції;
- блок з інформацією про викладачів та інструкторів.

Загальний вигляд головної сторінки автошколи «СТАРТ АВТО» представлено на рисунку 1.1.

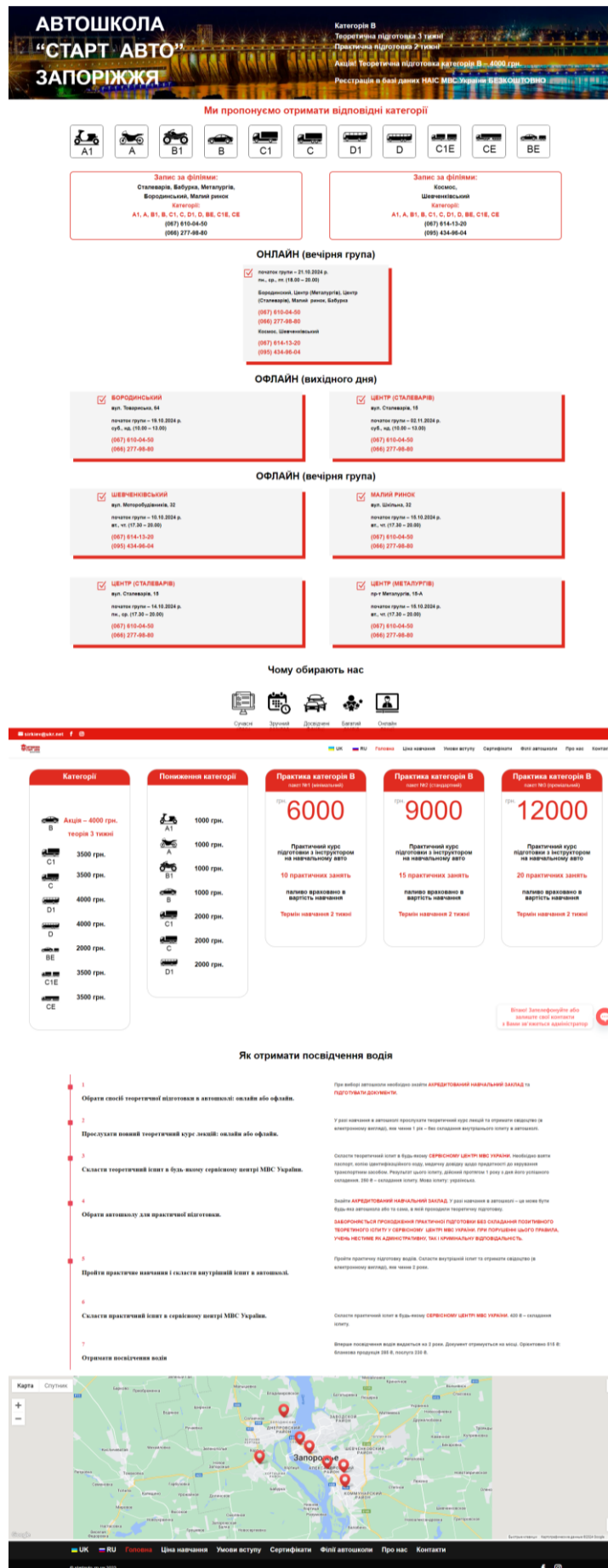


Рисунок 1.1 – Головна сторінка сайту автошколи «СТАРТ АВТО»

Щоб надіслати запит на зворотний зв'язок, потрібно натиснути на незакріплений елемент в нижньому правому куті сторінки, як показано на рисунку 1.2, після чого можна вибрати один з перелічених там месенджерів.

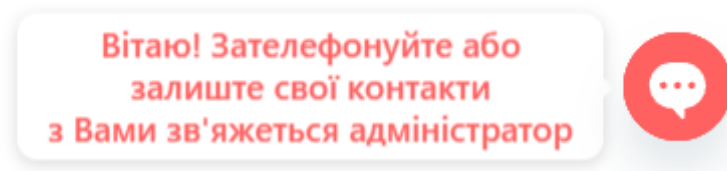


Рисунок 1.2 – Незакріплений елемент для зворотного зв'язку автошколи «СТАРТ АВТО»

Сайт автошколи «GREEN-WAY» має майже ідентичну структуру головної сторінки, як і попередньо розглянутий, але з деякими відмінностями:

- вебресурс фокусується саме на онлайн-формат навчання з акцентом на самостійну підготовку і тести;
- головна сторінка містить інтерактивні елементи: відеоуроки, тести, доступ до онлайн-матеріалів та можливість зареєструватися для доступу до додаткових функцій;
- сайт є більш інтерактивним, пропонуючи користувачам пройти тести з ПДР і брати участь у курсах онлайн без необхідності відвідувати фізичну автошколу;
- є можливість створити особистий акаунт для отримання розширених можливостей, таких як індивідуальна статистика навчання та коментарі;
- вся інформація постійно оновлюється, включаючи актуальні зміни у ПДР, що робить навчання більш динамічним і сучасним.

Головна сторінка сайту автошколи «GREEN-WAY» представлена на рисунку 1.3.

Щоб зареєструватись на сайті, треба натиснути на кнопку «Особистий кабінет» зверху у правому куті та заповнити форму, вигляд якої можна побачити нижче на рисунку 1.4.

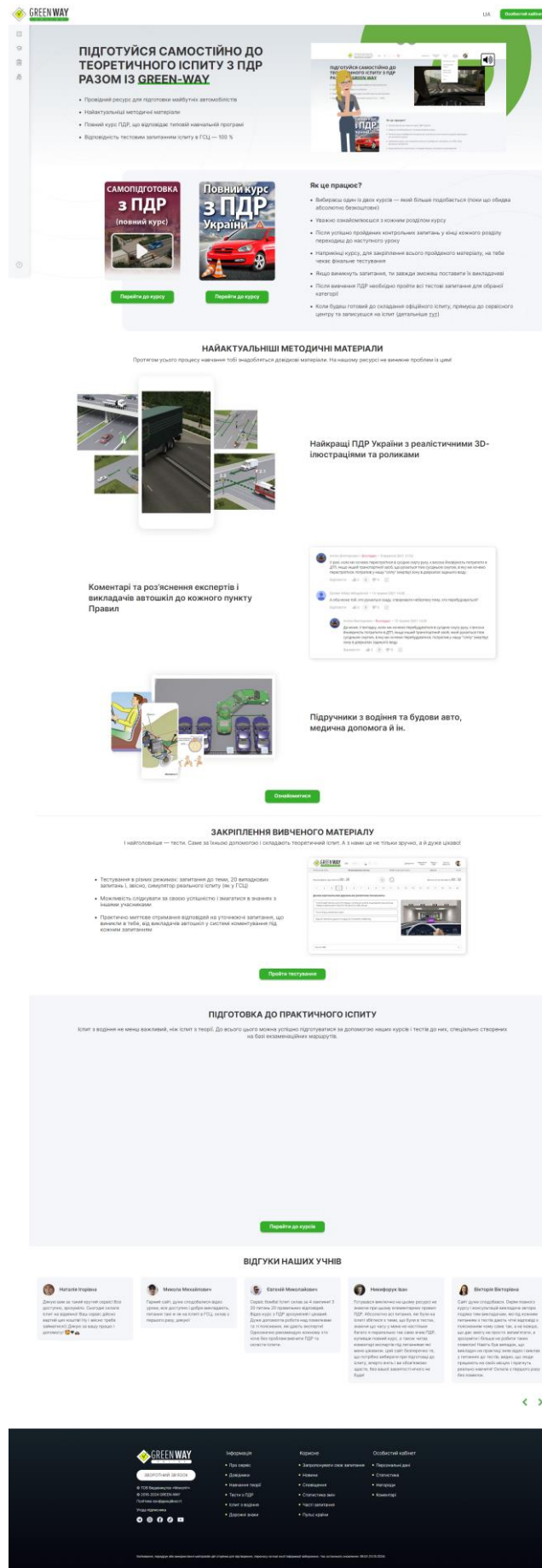


Рисунок 1.3 – Головна сторінка сайту автошколи «GREEN WAY»

Увійдіть, щоб використати повний функціонал сервісу

Статистика успішності Режим іспиту з ПДР Нагороди Турнірна таблиця Онлайн-консультація з викладачем Навчальні курси Підготовка до практичного іспиту

Email*

Email

Пароль*

Пароль

Увійти

[Реєстрація](#)
[Нагадати пароль](#)

Увійти за допомогою соціальних мереж

Facebook Google

Рисунок 1.4. – Форма реєстрації особистого кабінету на сайті автошколи «GREEN WAY»

Отже, після проведення аналізу існуючих вебресурсів автошкіл, можемо зробити підсумок про функціонал, який вони можуть запропонувати для користувача системи. На сайті «СТАРТ АВТО» можна записатися на офлайн-курси водіння, дізнатися про філії автошколи, ціни та розклад занять, тоді як на сайті «GREEN WAY» можна пройти онлайн-курси з ПДР, виконати тести та підготуватися до іспиту самостійно.

Розроблюваний вебресурс повинен враховувати всі переваги існуючих на ринку продуктів та усунути їхні недоліки, якщо такі є.

1.2.1 Зовнішній інтерфейс

Цей пункт описує користувацькі інтерфейси системи, які всі реалізовані через вебінтерфейс.

Інтерфейс складається з головної сторінки та сторінки з відгуками. Зареєстровані користувачі мають доступ до особистого кабінету, розкладу занять і відеоуроків.

Головна сторінка, на яку потрапляє користувач після входу в систему, забезпечує повну навігацію та містить загальну інформацію про автошколу, а також форму для подачі запиту на реєстрацію нового учня.

У реєстраційній формі майбутній учень вказує особисті дані, які використовуються для зв'язку та подальшої реєстрації в автошколі.

Розділ з розкладом практичних та теоретичних занять доступна лише для авторизованих користувачів та адміністраторів сайту. Розклад складається відповідно до правил, встановлених автошколою. Адміністратор сайту відповідає за заповнення інформації про теоретичні заняття. Для внесення даних про практичні заняття використовується спеціальна форма запису.

Форма реєстрації на практичне заняття включає поля, заповнення яких передає інформацію в розклад, якщо відповідні слоти в ньому ще не зайняті.

Сторінка для організації онлайн-зустрічей, семінарів, тренінгів та інших заходів доступна тільки для авторизованих користувачів. Вона містить відео вебінару в режимі реального часу, а також блок коментарів, через який учні можуть взаємодіяти з обраним викладачем.

Сторінка відгуків дозволяє користувачам залишати та переглядати відгуки про автошколу.

До сторінки інтерфейсу адміністратора має доступ виключно адміністратор сайту, яка призначена для внесення потрібних даних на вебсайт автошколи. На цій сторінці адміністратор має змогу реєструвати нових учнів або викладачів у системі, заповнювати та коригувати розклад занять, а також відображати результати успішності учнів. Крім того, адміністратор може підтверджувати або відхиляти дані, отримані від користувачів системи.

Приведено набір засобів і методів для забезпечення взаємодії між різними компонентами програмного забезпечення:

- система реєстрації зобов'язана поширювати вірні дані та зберігати їх на визначений час для можливого наступного використання;

- система відеоконференцій має забезпечити якісну передачу аудіо та відео з платформи, яку використовує викладач автошколи під час уроків у режимі онлайн;
- система запису на заняття з практики приймає унікальні дані на запит користувача і формує нове заняття після підтвердження необхідного запису в розкладі;
- система реєстрації зобов'язана відправляти користувачу електронний лист з підтвердженням заявки та дозволу до інтерфейсу системи для учнів, які вже авторизовані на ресурсі;
- система з розкладом повинна повідомляти користувачу про заплановане практичне заняття та його час;
- система онлайн-вебінарів повинна надавати можливість спілкування учнів з викладачем через коментарі, які будуть доступні для перегляду викладачу та іншим учасникам протягом семінару.

1.2.2 Функціональні вимоги

Інформаційна система автошколи повинна забезпечувати належну обробку даних учнів на всіх етапах їхньої взаємодії з системою. Форма реєстрації повинна приймати та обробляти дані нового користувача. Після підтвердження реєстрації система зобов'язана надати доступ до навчальних матеріалів і послуг. Окрім цього, система має перевіряти достовірність введених даних, наприклад, унікальність e-mail або телефону, а адміністратор повинен мати можливість перевірити правильність введених даних. Після успішної реєстрації учня дані зберігаються у базі, з можливістю їх оновлення адміністратором під час навчання. У разі відмови від навчання або його завершення система надає функцію видалення тимчасових даних користувача, щоб забезпечити зберігання лише необхідної інформації.

Система також повинна обробляти дані для запису на практичні заняття. Після введення даних для запису система перевіряє їх унікальність у реальному часі, зокрема перевіряючи доступність обраного часу в розкладі. Якщо такий запис уже існує, користувачу виводиться повідомлення про помилку і

рекомендація змінити введені параметри. У разі успіху, дані про нове заняття додаються до бази даних і автоматично відображаються у загальному розкладі.

Адміністратор повинен мати доступ до спеціальної панелі для управління системою, де він може вносити зміни до даних учнів, додавати нові навчальні матеріали, редагувати розклад занять і слідкувати за прогресом учнів. Крім того, система повинна відслідковувати прогрес студентів на основі тестування та участі у заняттях. Якщо результати виявляються недостатніми, система повинна пропонувати студентам додаткові матеріали для підвищення рівня знань. Усі дані про успішність учнів зберігаються у базі для подальшого аналізу та поліпшення навчального процесу.

1.2.3 Вимоги до робочих характеристик

У цій системі відсутні специфічні вимоги до продуктивності, оскільки час обробки надісланих запитів та відповідей залежить від якості з'єднання користувача. Кількість синхронних підключень залежить від потужності та можливостей сервера. Система повинна гарантувати високу швидкість виконання запитів і демонструвати відмінні результати у пошукових системах.

1.2.4 Атрибути якості програмного забезпечення

Система повинна гарантувати високу надійність і безпеку. Усі операції, що стосуються конфіденційної інформації клієнтів, повинні виконуватись з використанням протоколу SSL для забезпечення захищеного з'єднання. Крім того, система повинна автоматично виходити з облікових записів користувачів після завершення сесії, щоб запобігти несанкціонованому доступу. Важливі дані повинні бути зашифровані перед їхньою передачею через незахищені з'єднання мережі Інтернет. Доступ до серверних систем має бути наданий виключно тільки аутентифікованим адміністраторам. Рекомендовано реалізувати двофакторну аутентифікацію для підвищення рівня безпеки облікових записів. Регулярні аудити безпеки та оновлення системи забезпечать захист від нових загроз та вразливостей.

Система повинна забезпечити збереження усіх баз даних з акцентом на надійність через регулярне резервне копіювання, яке постійно оновлюється для

відображення останніх змін. Стабільність бази даних та операційної системи є критично важливою для загальної стійкості системи. Доступність системи повинна бути забезпечена у будь-який час, що, в свою чергу, дозволяє користувачам підключатись через веббраузер, за винятком періодів простою сервера. Крім того, необхідно реалізувати механізми моніторингу, як виявлятимуть та сповіщатимуть про можливі проблеми з доступом, що дозволить своєчасно реагувати на збої і підтримувати безперервність обслуговування. Важливо провадити резервне копіювання даних для запобігання їх втрат у разі непередбачених ситуацій.

У разі аварійного завершення роботи або пошкодження бази даних система відображатиме сторінку з повідомленням про помилку. Адміністратор повинен якнайшвидше завантажити резервну копію бази даних і звітність на сервер, після чого система буде перезавантажена. Це забезпечить безперервний доступ до сервісу.

Вебресурс є повністю портативним, що дозволяє користувачам системи автошколи отримувати доступ до його функцій з будь-якого пристрою, підключеного до Інтернету і оснащеного веббраузером. Кінцеві користувачі можуть використовувати будь-яку операційну систему, включаючи Windows, Mac OS, Linux, а також мобільні ОС. Вебресурс має бути адаптивним, забезпечуючи коректне відображення інформації та стабільну роботу на персональних комп'ютерах, ноутбуках та мобільних пристроях.

1.2.5 Функції вебсистеми

Розробка інформаційної системи для автошколи передбачає впровадження функціоналу, який сприятиме покращенню процесу навчання та аналітики, що дозволить студентам досягти вищого рівня знань. Згідно з проведеним аналізом існуючих рішень та виявленими вимогами, новий веб-ресурс вирішить такі ключові завдання:

- система забезпечить доступ до всебічної інформації, що допоможе майбутнім водіям зробити свідомий вибір при обранні автошколи;

- вебресурс буде створено у форматі розподілених додатків, що дозволить користувачам працювати з системою з різних пристроїв у мережі, зберігаючи дані на серверах. Це також забезпечить можливість оновлення інформації в реальному часі;
- система передбачатиме різні рівні доступу для авторизованих і неавторизованих користувачів, що дозволить забезпечити конфіденційність налаштувань;
- адаптація функції миттєвого надсилання електронних сповіщень забезпечить високий рівень комунікації між системою та користувачами;
- лише адміністратор зможе здійснювати повну реєстрацію нового клієнта, при цьому користувачі можуть подавати запити на реєстрацію, що підвищить зручність процесу;
- можливість перегляду та оновлення персональної інформації;
- користувачі матимуть змогу залишати відгуки щодо своїх вражень, які будуть переглядатися та фільтруватися адміністратором для подальшого покращення якості послуг;
- система передбачає проведення відео уроків, що дозволить студентам поглибити свої знання в зручному форматі;
- інтеграція з чат-ботами для швидких відповідей на поширені запитання;
- користувачі зможуть записуватися на практичні заняття через інтернет, з урахуванням специфіки роботи автошколи, днів виходу і вказаних адміністратором обмежень.

2 МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання прецедентів

Мета діаграми прецедентів полягає в документуванні акторів (усіх зовнішніх елементів поза системою) і варіантів використання (усіх внутрішніх

елементів системи) разом із їхніми взаємозв'язками. Актори – це всі особи або системи, які взаємодіють із даною системою, включаючи користувачів, адміністраторів та інші зовнішні ресурси. Ця діаграма надає нагальне представлення, як користувачі взаємодіють із системою, а також визначити основні функції, які вона повинна виконувати [203-204, 5].

Актора «клієнт» представлено на рисунку 2.1.



Рисунок 2.1 – Актор клієнт автошколи

Прецедент «Реєстрація на навчання» включає процес заповнення інформації про користувача та початкову реєстрацію в системі. Прецедент «Отримання квитанції про оплату навчання» надає інформацію про здійснений платіж за курс навчання. Прецедент «Запис на заняття» дозволяє учням та клієнтам автошколи обирати зручний для них час для запису на практичні заняття. Прецедент «Перегляд розкладу на сайті» відображає розклад, за яким на даний момент проходить навчання в автошколі. «Проходження теоретичних занять» дає можливість користувачам отримувати знання від інструкторів з теоретичного навчання через систему з онлайн-вебінарами.

Актора «бухгалтер» зображено на рисунку 2.2. Прецедент «Складання бухгалтерського звіту» надає можливість бухгалтеру застосовувати дані про учнів і персонал автошколи для підготовки узагальнених показників, які відображають господарсько-фінансову діяльність підприємства за минулий період. Прецедент

«Складання квитанцій про оплату навчання» забезпечує створення квитанцій з детальною інформацією про здійснені клієнтами платежі за навчання.



Рисунок 2.2 – Актор бухгалтер

Актор «директор» зображений на рисунку 2.3. Прецедент «Керування автошколою» дає директору можливість видавати накази, вказівки та розпорядження секретарю відповідно до роботи автошколи.



Рисунок 2.3 – Актор директор

Актора «секретар» відображено на рисунку 2.4. Прецедент «Виконання доручень директора» передбачає повну залежність від директора, оскільки секретар виконує всі його накази і розпорядження, які насамперед пов'язані з роботою в системі. Прецедент «Адміністрування сайту» забезпечує секретарю повний доступ до керування вебресурсом. Прецедент «Заповнення інформації на сайті» передбачає внесення та оновлення контенту на сайті. «Ведення розкладу на сайті» дозволяє додавати дані й редагувати інформацію про заняття у розкладі. Прецедент «Реєстрація учнів» дозволяє реєструвати користувачів у системі та надавати їм права авторизованих користувачів.



Рисунок 2.4 – Актор секретар

Актор «інструктор теорії або практики» зображений на рисунку 2.5. Прецедент «Проведення теоретичних занять» дає змогу інструктору організовувати та проводити вебінари онлайн для учнів автошколи через спеціалізовану систему. Прецедент «Формування розкладу» має на увазі встановлення днів навчання, які будуть відображатись в системі як доступні для учнів записи на заняття.

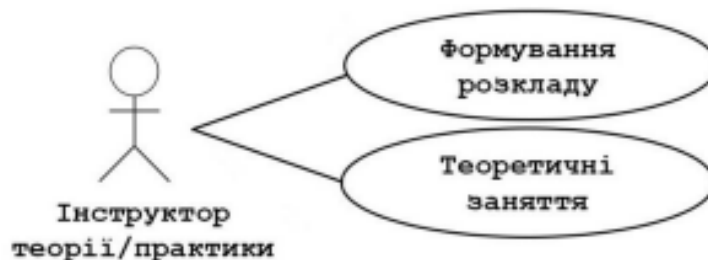


Рисунок 2.5 – Актор інструктор теорії або практики

Діаграма прецедентів (use case diagram) являє собою графічне представлення взаємодій між акторами, які виступають зовнішніми користувачами або системами, та прецедентами, що описують функції і процеси всередині системи (рис. 2.6).

Діаграма допомагає зрозуміти ключові елементи системи, показуючи не лише користувачів і їхню роль, а й спосіб, у який вони впливають на роботу системи, взаємодіючи з різними процесами або функціональними можливостями.

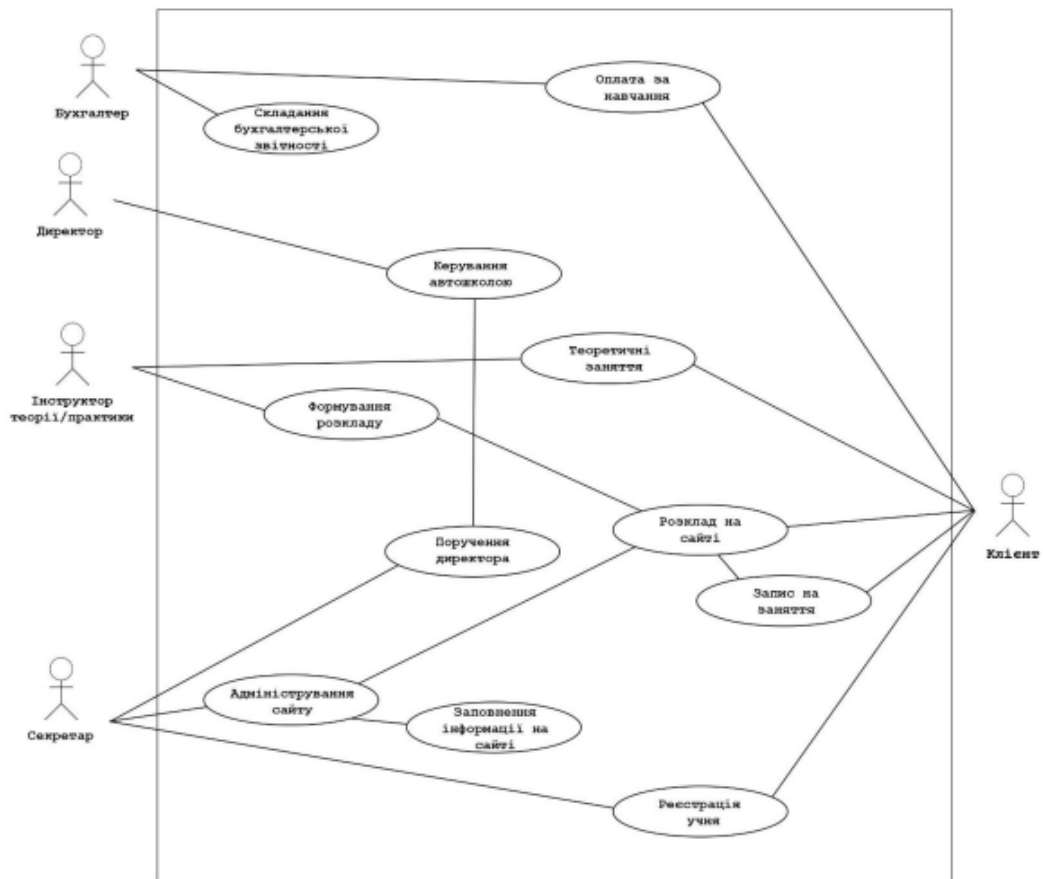


Рисунок 2.6 – Загальна діаграма прецедентів

2.2 Моделювання діаграм послідовності

Для моделювання відношення об'єктів у мові UML використовуються діаграми взаємодії, які відображають, як об'єкти взаємодіють між собою в рамках певного сценарію. Одним із ключових аспектів цієї взаємодії є час, що має важливе значення для коректного розуміння порядку виконання дії. Задля відображення тимчасових характеристик передачі та прийому повідомлень між об'єктами застосовується діаграма послідовності, яка чітко показує, як об'єкти взаємодіють між собою, зокрема в якій послідовності надсилають та отримують повідомлення, що є критично важливим для розробки та аналізу складних систем.

Діаграма прецедентів (діаграма варіантів використання) в UML ілюструє взаємозв'язок між акторами та прецедентами в системі, показуючи, як різні

користувачі взаємодіють з функціональністю системи. Прецедент «Реєстрація на навчання» представлений на діаграмі послідовності, відображений на рисунку 2.7.

У цьому процесі основним виконавцем є адміністратор сайту, який відповідає за обробку реєстраційних запитів. Діаграма послідовності детально відображає кроки, які проходять учасники, а також порядок і часові інтервали, протягом яких відбуваються різні дії, що забезпечує чітке уявлення про процес реєстрації.

Потреби сторін, які є учасниками процесу:

- відвідувач сайту бажає отримати доступ до системи через авторизацію, що дозволить йому розпочати навчання в автошколі;
- адміністратор сайту прагне зареєструвати нового учня в системі, забезпечуючи ефективний процес обробки реєстраційних запитів;
- автошкола зацікавлена в збільшенні кількості учнів, оскільки це сприяє розвитку її бізнесу та покращенню фінансових показників.

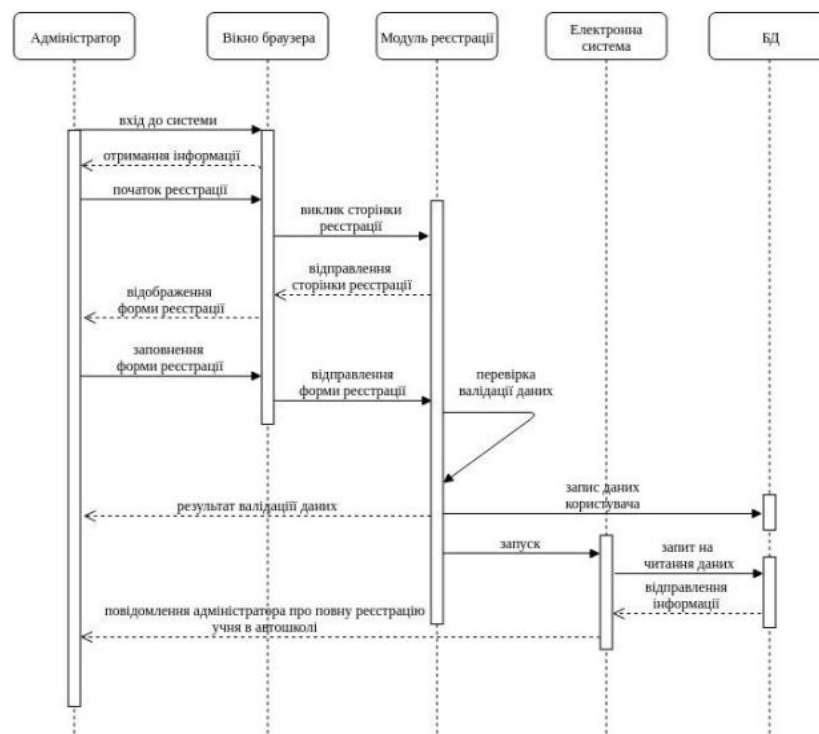


Рисунок 2.7 – Діаграма послідовності прецеденту «Реєстрація навчання»

Діаграма послідовності прецеденту «Запис на заняття» наведена далі на рисунку 2.8.

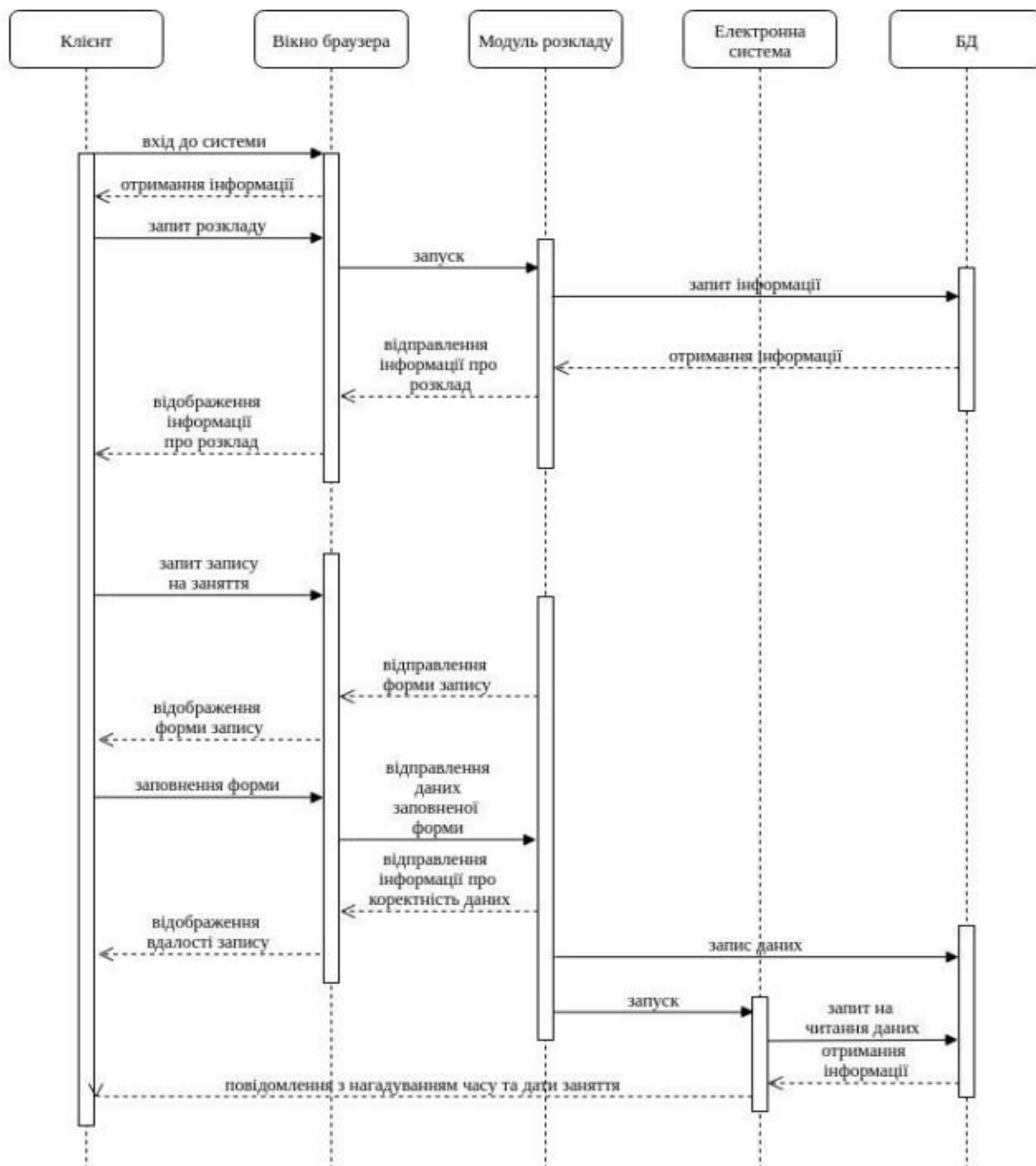


Рисунок 2.8 – Діаграма послідовності прецеденту «Запис на заняття»

Результат (постумови): користувача успішно зареєстровано в системі.

Основний потік (основний успішний сценарій):

- адміністратор натискає кнопку «зареєструвати нового учня»;
- система відкриває сторінку для реєстрації;
- адміністратор заповнює необхідну інформацію для реєстрації;

- система перевіряє правильність введених даних (валідація);
- адміністратор натискає кнопку «zareestruvati», і система зберігає введені дані користувача в базі даних;
- система реєструє користувача;
- система відправляє користувачеві електронний лист із повідомленням про успішну реєстрацію та дані для авторизації в системі.

Альтернативні потоки для прецеденту «Реєстрація на навчання»:

а) вказані некоректні дані:

1) система інформує адміністратора сайту про некоректність введених даних і вказує поля, які потрібно виправити;

2) адміністратор виправляє дані у зазначених полях;

3) система повторно виконує валідацію виправлених полів.

б) адміністратор не натиснув кнопку «zareestruvati» і перейшов на іншу сторінку сайту:

1) система видаляє всі заповнені дані з форми.

Прецедент «Запис на заняття».

Основний виконавець: учень автошколи.

Сторони інтересу та їхні потреби:

- відвідувач сайту: бажає записатися на практичне заняття;
- автошкола: зацікавлена у складанні розкладу занять.

Результати (постумови): учень записується на практичне заняття, а система нагадує йому про час і дату заняття.

Основний потік (основний успішний сценарій):

- учень переходить на сторінку запису на практичне заняття на сайті;
- система відображає розклад та форму для запису на заняття;
- користувач заповнює необхідну інформацію для запису;
- учень натискає кнопку «записатися»;
- система виконує перевірку введених даних та перевіряє їх на унікальність;
- система показує результат перевірки;

- система зберігає дані про запис у базі даних та відображає на сайті дату і час практичного заняття учня;
- система відправляє учню нагадування про заняття за 6 годин і за годину до початку через електронну пошту.

Альтернативні сценарії прецеденту «Запис на заняття»:

а) вказано некоректні дані:

- 1) система інтерактивно вказує на поля з помилками;
- 2) користувач коригує вказані дані;
- 3) користувач ініціює збереження даних.

б) спроба перейти на іншу сторінку без збереження даних:

- 1) система виводить повідомлення про втрату незбережених даних;
- 2) система скидає форму до початкового стану.

3 ПРОЄКТУВАННЯ СИСТЕМИ

3.1 Архітектура програмного забезпечення

Вебресурс реалізує архітектуру клієнт-сервер [19-20, 6]. Клієнт-сервер – це обчислювальна або мережна архітектура, що передбачає розподіл завдань або навантаження між постачальниками послуг (серверами) та їх замовниками (клієнтами). Ця архітектура є оптимальною для розподілених мережних додатків і забезпечує взаємодію та обмін даними між ними.

Клієнти працюють паралельно та незалежно один від одного. Один сервер здатний одночасно обробляти запити від кількох клієнтів. Взаємодія в моделі клієнт-сервер визначається, перш за все, розподілом обов'язків між клієнтом і сервером.

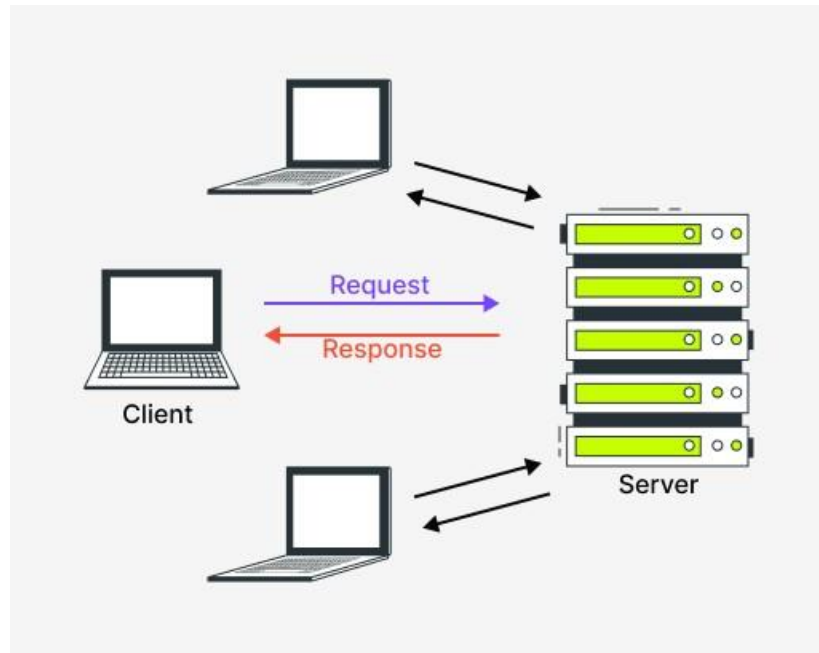


Рисунок 3.1 – Клієнт-серверна архітектура

Існує три рівні операцій:

- рівень представлення даних, що фактично є інтерфейсом користувача і відповідає за відображення даних та введення команд управління;
- прикладний рівень, на якому реалізується основна логіка додатків та обробка інформації;
- рівень управління даними, що забезпечує зберігання даних і доступ до них.

На вебресурсі використовується модель, в якій вся логіка управління даними зосереджена на сервері. Клієнтська програма виконує лише функції рівня представлення.

Основна концепція архітектури «клієнт-сервер» полягає в розподілі мережевого додатка на кілька компонентів, кожен з яких виконує певний набір сервісів. Компоненти такого додатку можуть працювати на різних комп'ютерах, виконуючи функції як сервера, так і клієнта. Це підвищує надійність, безпеку та продуктивність мережевих додатків і самої мережі.

3.2 Аналіз та вибір засобів розробки

Розробку вебсистеми можна умовно поділити на дві частини:

- сторона клієнта, яка також називається *front end*, охоплює розробку всіх елементів, що відображаються на екрані користувача;
- сторона сервера, яка включає в себе додаток, що управляє всією системою, базу даних і сам сервер [7].

На стороні клієнта використовуються основні компоненти, такі як HTML і CSS. HTML (мова гіпертекстової розмітки) інструктує веббраузер, як відображати вміст вебсторінок, тоді як CSS (каскадні таблиці стилів) відповідає за візуальне оформлення цього вмісту. Для спрощення роботи з цими технологіями часто використовують Bootstrap як корисну основу.

JavaScript (JS) надає вебсторінкам інтерактивність, додаючи анімацію та динамічно оновлюваний контент, а також керуючи мультимедійними елементами. Існує безліч бібліотек JS, таких як jQuery, React.js і Zepto.js, а також фреймворків, таких як Angular, Vue, Backbone і Ember, які значно спрощують і прискорюють процес веброзробки.

Компоненти на стороні сервера є невидимими для користувачів, але вони відповідають за управління вмістом, що відображається на стороні клієнта. Серверні мови програмування використовуються для створення логіки вебсайтів і додатків, а фреймворки для цих мов пропонують безліч інструментів, що спрощують і пришвидшують процес кодування.

Серед найпопулярніших мов програмування та їх основних фреймворків можна виділити:

- Ruby (Ruby on Rails);
- Python (Django, Flask, Pylons);
- Java (Spring) та PHP (Laravel).

PHP (PHP: Hypertext Preprocessor) є широко використовуваною мовою сценаріїв загального призначення з відкритим кодом, яка може виконувати

динамічні операції та бути вбудованою в HTML. PHP є основною мовою для WordPress, оскільки саме ядро WordPress і майже всі теми та плагіни розроблені на цій мові [8].

Вебресурсу необхідно місце для зберігання своїх даних, для чого використовуються бази даних. Існують два основні типи баз даних: реляційні та нереляційні. Серед найбільш поширених баз даних, що використовуються у веброзробці, можна виділити MySQL та PostgreSQL. Обидві ці системи пропонують надійне зберігання даних та забезпечують високу продуктивність для вебдодатків.

MySQL – це кросплатформна реляційна система керування базами даних (СКБД), яка має програмні інтерфейси для мов програмування, таких як C, Java, PHP і Python, що робить її популярною для створення власних додатків. У MySQL немає обмежень на максимальний розмір таблиці, проте розмір таблиці залежить від її типу. Наприклад, тип MyISAM обмежений максимальним розміром файлу, що визначається файловою системою операційної системи. У NTFS цей розмір теоретично може досягати 32 ексабайт. Для InnoDB одна таблиця може зберігатися у кількох файлах, що формують єдиний табличний простір, який може досягати 64 терабайт.

MySQL підтримує більшість функцій SQL, а робота з нею здійснюється в консольному режимі. Проте існують також графічні оболонки, такі як phpMyAdmin, HeidiSQL і Workbench, які забезпечують більш зручний інтерфейс для користувачів. Крім того, MySQL дозволяє оптимізувати створення і налаштування бази даних за допомогою виконання команд на SQL. Серед недоліків можна зазначити, що MySQL має платну підтримку [9].

PostgreSQL – це безкоштовна об'єктно-реляційна система керування базами даних (СКБД), яка доступна в різних версіях для численних платформ, включаючи Unix-системи, Linux, MS Windows і MacOS. У PostgreSQL розмір бази даних може досягати практично необмежених обсягів, а кількість записів і індексів у таблицях не має обмежень. Максимальний розмір таблиці може становити 32 гігабайти, тоді як записи можуть мати обсяг до 1,6 терабайта.

СКБД підтримує стандартні, часткові та функціональні індекси, має планувальник запитів, систему блокування, контроль буферної і кеш-пам'яті, а також табличні простори. Ці функції забезпечують високу продуктивність PostgreSQL завдяки її масштабованості та низьким вимогам до апаратних ресурсів. PostgreSQL також ефективно стискає і розтискає дані, що дозволяє зберігати більше інформації на дисковому просторі. Проте серед недоліків можна відзначити, що PostgreSQL не має графічних оболонок для полегшення використання [10].

Системи управління контентом (CMS) відповідають за створення, зберігання, пошук, опис (анотування) та публікацію (відображення) різних типів вмісту. CMS забезпечують виконання стандартних завдань, що зазвичай пов'язані з редагуванням і публікацією, а також виконують додаткові дії, такі як схвалення та позначка контенту для подальшого редагування або перегляду. Консоль WordPress надає інструменти для управління цими завданнями і контролю редакційного процесу.

CMS використовуються в різних сферах, від блогів до складних веб-сайтів. Вбудована система тем і плагінів, разом з продуманою архітектурою, дозволяє створювати практично будь-які вебпроекти на основі WordPress, що робить цю платформу універсальним інструментом для розробки.

Система керування контентом WordPress, яка є основою для розроблюваного вебресурсу, працює з мовами програмування HTML, CSS, PHP та JavaScript, використовуючи базу даних MySQL. Саме ці мови програмування будуть застосовані для розробки системи.

Телеграм-боти – це автоматизовані програми, що працюють всередині месенджера Telegram і виконують певні дії у відповідь на команди або запити користувачів. Боти можуть виконувати різні функції, наприклад, відповідати на запити, надсилати сповіщення, виконувати пошук, допомагати з організацією подій тощо. Чат-боти спрощують взаємодію з різними сервісами, дозволяючи користувачам виконувати безпосередньо в чаті, що зручно та швидко [11].

Телеграм-ботів можна застосовувати у таких сферах:

- інформаційні боти – надають інформацію за запитом користувача, наприклад, новини, погоду, курс валют, тощо;
- сервіси підтримки – автоматизовані системи підтримки клієнтів, що відповідають на поширені питання або переадресовують на оператора;
- боти для покупок – допомагають здійснювати замовлення товарів або послуг через;
- розважальні боти – надають ігри, вікторини або інший інтерактивний контент для розваг;
- організаційні боти – нагадування про важливі події, організація завдань або запису в календарі;
- боти для автоматизації – інтеграція з іншими сервісами для автоматизації процесів.

Серед середовищ розробки можна виділити два інструменти – PyCharm та Visual Studio Code.

PyCharm – це інтегроване середовище розробки (IDE), спеціалізоване на написанні програм на мові Python. Воно надає розробникам інструменти для зручного створення, тестування та налагодження Python-додатків. Цей інструмент значно полегшує роботу з мовою Python, починаючи від автодоповнення коду і, закінчуючи потужним відгадчиком та інструментами рефакторингу [12].

Visual Studio Code (VS Code) – це безкоштовний, легкий, але потужний текстовий редактор з підтримкою багатьох мов програмування, розроблений компанією Microsoft. Популярний інструмент серед розробників завдяки своїй гнучкості, швидкості та можливості розширення за допомогою плагінів [13].

При розробці даного проєкту, було вирішено використати мову Python в IDE PyCharm.

Для створення Телеграм-бота мовою Python існує декілька бібліотек:

- python-telegram-bot;
- aiogram;
- pyTelegramBotAPI.

python-telegram-bot – офіційна бібліотека від Telegram для розробки ботів на Python. Надає зручний інтерфейс для взаємодії з Telegram Bot API, що надає змогу створювати автоматизовані програми, які можуть відповідати на повідомлення, реагувати на команди користувачів і виконувати інші дії, містить детальну документацію та активну спільноту на GitHub [14].

Основні можливості цього боту:

- бібліотека має інтуїтивний інтерфейс для роботи з основними функціями Telegram Bot API, такими як надсилання та отримання повідомлень, відповідей, зображень, файлів;
- дозволяє легко налаштовувати обробники команд, що відповідають на певні текстові повідомлення або команди від користувачів;
- підтримка вебхуків (webhook) та опитувань (polling) (боти можуть працювати як у режимі постійного опитування серверів Telegram, так і за допомогою вебхуків для отримання оновлень);
- бібліотека підтримує надсилання зображень, аудіо, відео, документів та інших типів медіа;
- підтримка інтерактивних інтерфейсів, таких як інлайн-кнопки, спеціальні клавіатури або команди, що дозволяють створювати складніші взаємодії з користувачами;
- можливість асинхронного програмування дозволяє створювати ефективні боти для обробки великої кількості запитів;
- бібліотека добре документована і підтримується великою спільнотою, що робить її однією з найпопулярніших для створення Телеграм-ботів на Python.

Aiogram – це асинхронна Python-бібліотека для створення Telegram-ботів, яка використовує Telegram Bot API. Вона орієнтована на простоту та ефективність, дозволяючи створювати ботів з мінімальними зусиллями та використовувати переваги асинхронного програмування для більш швидкої і масштабованої роботи з великим потоком [15].

Основний функціонал схожий на python-telegram-bot, але варто звернути увагу на такі можливості:

- всі основні функції `aiogram` працюють асинхронно на основі `asuncio`, що дозволяє одночасно обробляти багато запитів і забезпечує високу продуктивність;
- дозволяє легко розділяти логіку бота на різні модулі, що сприяє кращій організації коду та підтримці проєкту;
- підтримує фільтри для управління тим, які повідомлення чи події обробляти;
- можна створювати власні клавіатури, інлайн-кнопки та інші інтерфейсні елементи;
- має добре документовану API та активну спільноту розробників;
- легко інтегрується з іншими бібліотеками та фреймворками для додаткових можливостей.

`pyTelegramBotApi` (відома як `Telebot`) – це Python-бібліотека для роботи з Telegram Bot Api, яка дозволяє легко створювати та керувати Телеграм-ботами. Ця бібліотека надає простий інтерфейс для взаємодії з Telegram, що робить її популярною серед початківців і розробників, яким потрібно швидко створити функціональних ботів [16].

Його основні його функції не відрізняються від вище розглянутих інструментів, але є деякі зміни:

- становий механізм (FSM) – дає можливість використовувати механізм керування станами для складних сценаріїв роботи з ботом, наприклад, для чат-ботів кількома етапами взаємодії;
- має велику документацію та багато прикладів робіт, що дозволяє швидко почати роботу навіть тим, хто зіткнувся з цим вперше.

Отже, враховуючи проведений аналіз існуючого інструментарію для розробки Телеграм-бота на мові Python, для даного проєкту підходить `aiogram`.

3.3 Проєктування бази даних

Найважливішим компонентом будь-якої інформаційної системи є база даних. База даних представляє собою структурований і організований набір даних, що об'єднані відповідно до певної обраної моделі, яка описує характеристики як фізичних, так і віртуальних систем.

ER-модель (модель «сутність – зв'язок») є моделлю даних, що дозволяє описувати концептуальні схеми предметної області. Вона використовується при високорівневому (концептуальному) проєктуванні баз даних, дозволяючи виділити ключові сутності та позначити зв'язки, які можуть виникати між ними. Сутність має один або декілька атрибутів, які або їй належать, або успадковуються через відношення з іншими сутностями.

WordPress автоматично створює таблиці в базі даних з необхідною інформацією [17]. Платформа використовує кілька взаємопов'язаних таблиць, між якими встановлені зв'язки типу «один до багатьох». Більшість таблиць пов'язані з однією або кількома іншими через одне поле, яке є унікальним ідентифікатором для кожного запису (наприклад, `post_id`).

База даних WordPress складається з набору таблиць, кожна з яких має унікальну назву, що починається з префікса бази даних. За замовчуванням цей префікс – `wp`. Проте на етапі установки, з метою підвищення безпеки, префікс `wp` змінюють на довільні 4 або більше латинських буквено-цифрових символів.

Додаток `phpMyAdmin` дозволяє редагувати таблиці бази даних за допомогою зрозумілого графічного інтерфейсу або через мову запитів SQL.

Вміст основних таблиць бази даних WordPress включає такі елементи:

- `wp_posts` – у цій таблиці зберігаються відомості про автора статті, дату і час публікації, дату і час останньої зміни, тексти, статус (опубліковано, чернетка, на затвердження) та інша інформація. Зв'язки з іншими таблицями встановлені через поля `post_id` (з таблицею `wp_postmeta` та `wp_term_relationships`);

– `wp_postmeta` – ця таблиця містить велику кількість даних про записи та сторінки сайту, включаючи інформацію про прикріплені файли (зображення, документи, відео) та дані, заповнені при створенні або редагуванні записів. Зв'язки з іншими таблицями здійснюються через `post_id` (з таблицею `wp_posts`);

– `wp_comments` – тут зберігаються коментарі, залишені до записів. Крім тексту коментаря, таблиця містить додаткову інформацію, включаючи ім'я, електронну пошту та сайт автора, IP-адресу, з якої надійшов коментар, дату, час та інше. Зв'язки з іншими таблицями встановлені через `post_id` (з таблицею `wp_posts`);

– `wp_users` – таблиця містить список зареєстрованих користувачів, їхній логін та пароль у зашифрованому вигляді, електронну адресу, дату реєстрації та інші відомості. Зв'язки з іншими таблицями встановлені через `post_author` (з таблицею `wp_posts`);

– `wp_options` – в цій таблиці зберігаються опції та налаштування WordPress. Зв'язки з іншими таблицями відсутні.

Виходячи з описаної предметної області та вимог до системи, окрім базових сутностей WordPress, створюємо власні сутності:

– `schedule_exercise` – містить розклад занять. У кожному занятті є інформація про студента, інструктора, дату та час. Дата та час є унікальними, але можуть повторюватися залежно від кількості викладачів;

– `exercise_confirmed_day` – містить дати підтверджених днів для практичних занять кожного викладача;

– `student` – містить інформацію про зареєстрованого учня автошколи, а саме повне ім'я учня, групу навчання, дату початку та закінчення навчання в автошколі, а також кількість витрачених грошей на навчання.

З наповненням сайту база даних WordPress буде поступово розростатися, і кількість таблиць значно збільшиться, особливо з додаванням нових плагінів, кожен з яких створюватиме власні таблиці. Ключовою сутністю в WordPress є таблиця `wp_users`, яка відповідає за зберігання інформації про зареєстрованих користувачів.

Сутність `wp_users` складається з таких атрибутів:

- `ID` – унікальний ідентифікатор користувача;
- `user_login` – логін користувача;
- `user_pass` – зашифрований пароль користувача;
- `user_nicename` – очищена URL-адреса на основі логіна користувача;
- `user_email` – електронна адреса користувача;
- `user_url` – URL-адреса вебсайту користувача (якщо є);
- `user_registered` – дата реєстрації користувача;
- `user_activation_key` – ключ, що використовується для скидання пароля;
- `user_status` – статус користувача;
- `display_name` – ім'я користувача, яке відображається на сайті.

Ця структура дозволяє системі WordPress ефективно управляти інформацією про користувачів і забезпечує гнучкість при подальшому розширенні функціональності через плагіни.

Атрибути сутності «`schedule_exercise`» включають:

- `ID` – унікальний ідентифікатор заняття;
- `Student_id` – ідентифікатор студента, пов'язаний з користувачем;
- `Teacher_id` – ідентифікатор викладача;
- `Date_id` – ідентифікатор доступної дати заняття;
- `time` – час проведення заняття.

Сутність «`exercise_confirmed_day`» містить такі атрибути:

- `ID` – унікальний ідентифікатор підтвердженого дня;
- `Data` – дата заняття;
- `Teacher_id` – ідентифікатор викладача, який проводить заняття.

Атрибути сутності «`student`» включають:

- `ID` – унікальний ідентифікатор студента;
- `group` – група навчання студента;
- `data_start` – дата початку навчання;
- `data_end` – дата закінчення навчання;

- spend_money – кількість витрачених грошей на навчання;
- name – повне ім'я студента.

Між сутностями «wp_users» та «student» існує зв'язок 1:1, оскільки кожен учень автошколи має унікальну пов'язану з ним інформацію в базі даних, і навпаки. Це означає, що кожен користувач системи має відповідний запис в таблиці «student», який містить додаткові відомості про його навчання.

Така структура забезпечує чітку організацію даних та їх зручне використання в системі автошколи.

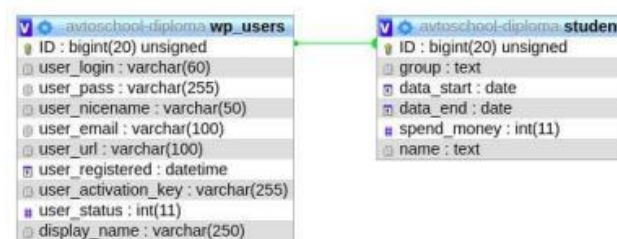


Рисунок 3.2 – Зв'язок між сутностями «wp_users» та «student»

Між сутностями «wp_users» та «schedule_exercise» існує зв'язок 1:M, оскільки один учень може брати участь у кількох заняттях, а один викладач може проводити безліч занять. Ця взаємозалежність відображає можливість кожного користувача системи (учня або викладача) бути залученим до кількох розкладів занять. Таким чином, один запис у таблиці «wp_users» може відповідати багатьом записам у таблиці «schedule_exercise», що забезпечує гнучкість у веденні розкладу.

Ця структура проілюстрована на рисунку 3.3, де показаний зв'язок між користувачами та їх розкладом занять.

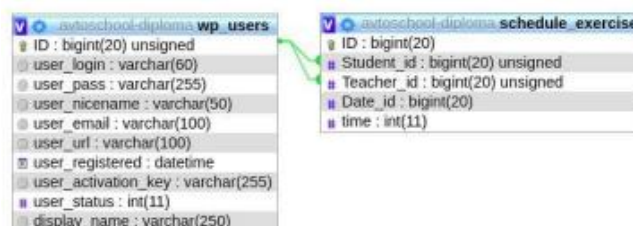


Рисунок 3.3 – Зв'язок між сутностями «wp_users» та «schedule_exercise»

Між сутностями «schedule_exercise» та «exercise_confirmed_day» існує зв'язок 1:M, оскільки одна підтверджена дата може бути пов'язана з багатьма заняттями. Це означає, що кілька занять можуть проводитися в одну й ту саму дату, що відображає реальну ситуацію, коли в одному дні може бути багато занять, як для різних студентів, так і з різними викладачами.

Ця взаємозалежність між сутностями «schedule_exercise» та «exercise_confirmed_day» показана на рисунку 3.4, де наглядно демонструється їх зв'язок.



Рисунок 3.4 – Зв'язок між сутностями «schedule_exercise» та «exercise_confirmed_day»

ER-діаграма бази даних автошколи представлена на рисунку 3.5.

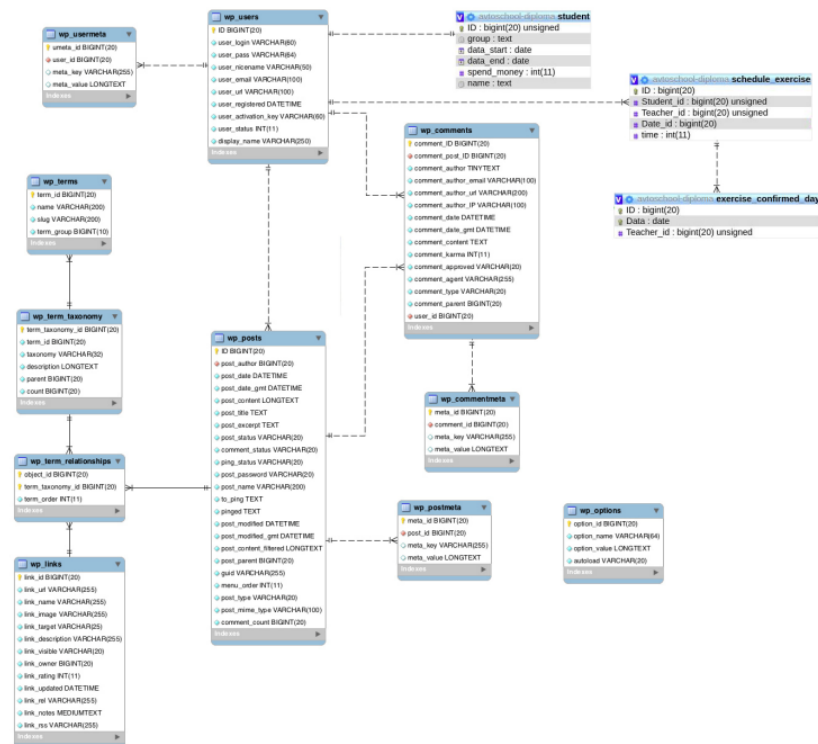
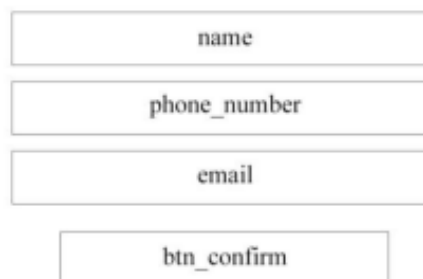


Рисунок 3.5 – ER-діаграма бази даних

3.4 Проєктування інтерфейсу користувача

У цьому розділі описані інтерфейси користувача для системи. Всі вони реалізовані через вебінтерфейс. Під інтерфейсом користувача розуміється візуальне оформлення сторінок проєкту. Для демонстрації зовнішнього вигляду сторінок створюються їх макети.

На головній сторінці відображено логотип автошколи, а також додаткову контактну інформацію, яка розміщена в нижній частині. Крім того, на сторінці розміщена форма для реєстрації та зворотного зв'язку, як показано на рисунку 3.6, де користувач повинен вказати свої точні персональні дані для зворотного зв'язку та повної реєстрації в системі. Основна інформація про автошколу знаходиться в центрі уваги та представлена у вигляді контенту на макеті. Головна сторінка представлена на рисунку 3.7.



Макет форми реєстрації нового учня, що складається з чотирьох елементів, розташованих вертикально:

- name
- phone_number
- email
- btn_confirm

Рисунок 3.6 – Макет форми реєстрації нового учня



Макет головної сторінки, що складається з шести елементів, розташованих вертикально:

- nav_menu
- img_logo
- content
- register form
- footer

Рисунок 3.7 – Макет головної сторінки

Сторінка відгуків про автошколу має схожу структуру з головною сторінкою, проте з деякими відмінностями (рис. 3.8). На ній відображаються підтверджені відгуки, залишені користувачами. Також присутня форма для написання нових відгуків та побажань для автошколи.

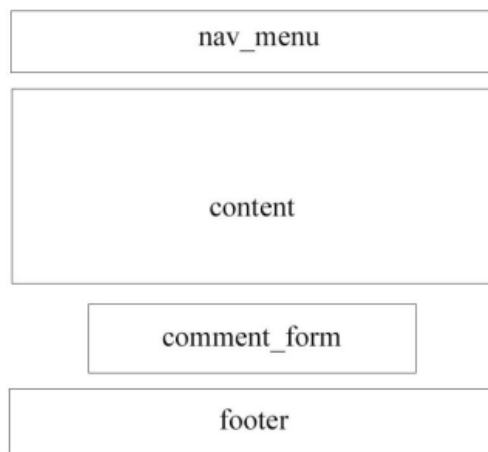


Рисунок 3.8 – Макет сторінки відгуків

Сторінка розкладу занять побудована аналогічно до головної сторінки, але з певними змінами (рис. 3.9). Вгорі розміщена форма для запису на практичне заняття, що забезпечує зручний доступ до неї. У контентній частині відображається розклад занять. Ліворуч знаходиться навігаційна панель, яка дозволяє легко переміщатися по особистому кабінету учня автошколи.

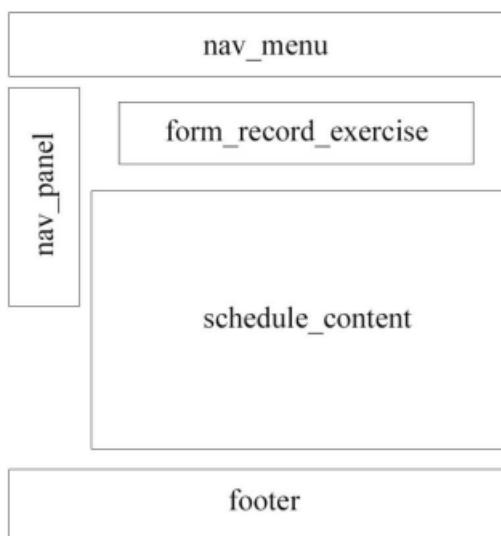


Рисунок 3.9 – Макет сторінки відображення розкладу занять

Форма запису на практичне заняття містить поля для введення інформації, яка буде використана для реєстрації бажання учня вибрати зручний і бажаний час для навчання згідно з розкладом (рис. 3.10).

The diagram shows a rectangular form with a light gray border. Inside, there are four input fields arranged in a 2x2 grid. The top-left field is labeled 'name', the top-right is 'instructor', the bottom-left is 'date', and the bottom-right is 'time'. Below this grid, centered horizontally, is a button labeled 'btn confirm'.

Рисунок 3.10 – Макет форми для запису на практичне заняття

Сторінка відеоконференції має аналогічну структуру, як і сторінка розкладу занять, з певними змінами (рис. 3.11). Форма для запису на практичне заняття відсутня. У контексті розташовані відеопотік заняття та чат для спілкування в режимі реального часу. Зліва залишається навігаційна панель кабінету учня автошколи.

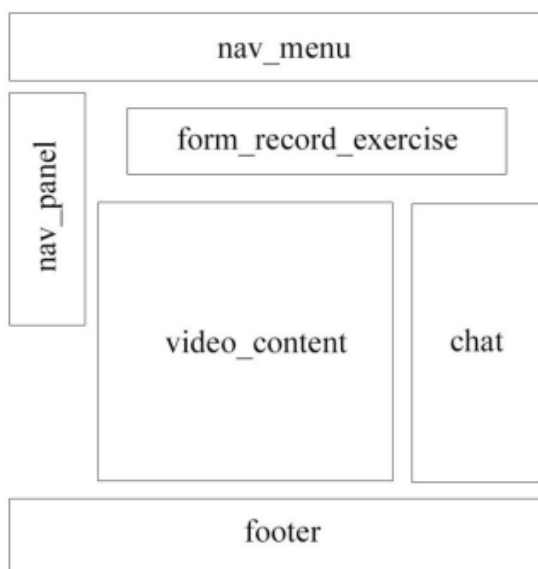


Рисунок 3.11 – Макет сторінки відеоконференції

4 РЕАЛІЗАЦІЯ ВЕБРЕСУРСУ

4.1 Вибір інструментів розробки

WordPress використовується як основа для побудови веб-системи завдяки своїй гнучкості, широким можливостям кастомізації та багатій екосистемі плагінів. Платформа забезпечує зручність у розробці та адмініструванні, що дозволяє легко додавати новий контент і керувати сайтом. Завдяки інтеграції з плагінами, такими як Quiz And Survey Master (QSM), система дозволяє організовувати інтерактивне тестування для учнів, а підтримка мультимедійного контенту сприяє покращенню процесу навчання через використання відео, графіки та схем.

MySQL використовується для зберігання та обробки даних у системі завдяки своїй надійності, швидкодії та простоті інтеграції з веб-додатками. Ця система управління базами даних дозволяє ефективно організовувати й обробляти великі обсяги інформації, зокрема дані про учнів автошколи, результати тестування, навчальні матеріали та розклади занять. MySQL забезпечує швидкий доступ до даних і підтримує складні SQL-запити, що дозволяє здійснювати аналіз успішності учнів та генерувати звіти. Завдяки своїй сумісності з WordPress і багатьма іншими платформами, MySQL є ідеальним рішенням для зберігання структурованих даних, необхідних для стабільної роботи інформаційної системи автошколи.

Бібліотека aiogram використовується для створення асинхронних Telegram-ботів, які забезпечують інтерактивну взаємодію учнів з системою автошколи. Вона дозволяє реалізувати перегляд результатів тестування та автоматичне надсилання нагадувань. Завдяки асинхронній архітектурі aiogram обробляє велику кількість запитів одночасно, що робить бота швидким і ефективним навіть при високих навантаженнях. Бібліотека також пропонує зручний інтерфейс для роботи з Telegram API, спрощуючи розробку і підтримку бота, який стане корисним інструментом для учнів автошколи.

PyCharm використовується як інтегроване середовище розробки для написання, редагування та тестування коду Telegram-бота. Це середовище забезпечує зручність роботи завдяки функціям автозаповнення, підсвічування синтаксису, інтеграції з системами контролю версій та вбудованому зневадженню. PyCharm значно підвищує ефективність роботи розробника, дозволяючи швидко створювати якісний код, перевіряти його на помилки та проводити налагодження. Використання цього інструмента спрощує процес розробки Telegram-ботів на основі бібліотеки aiogram.

4.2 Створення бази даних

База даних – інформаційної системи автошколи складається з 15 таблиць, які забезпечують збереження та організацію всіх необхідних даних. Під час інсталяції WordPress автоматично створюється 12 стандартних таблиць, кожна з яких виконує свою специфічну функцію:

- wp_commentmeta – зберігає метадані коментарів, які можуть містити додаткову інформацію, пов’язану з коментарями на сайті;
- wp_comments – таблиця для збереження інформації про коментарі, залишені користувачами;
- wp_links – використовується для управління списком посилань (наприклад, блогролів);
- wp_options – зберігає налаштування сайту, зокрема параметри тем, плагінів та конфігурації WordPress;
- wp_postmeta – містить додаткові метадані, пов’язані з публікаціями, наприклад, інформацію про додані медіафайли чи специфічні атрибути записів;
- wp_posts – основна таблиця для зберігання всіх типів контенту, таких як записи, сторінки та вкладення;

- wp_termmeta – містить метадані, пов’язані з термінами таксономій, наприклад, додаткові характеристики категорій;
- wp_terms – зберігає терміни (наприклад, назви категорій чи міток), які організовують контент;
- wp_term_relationships – визначає зв’язок між контентом (наприклад, записами) та термінами таксономій;
- wp_term_taxonomy – описує таксономії для термінів, наприклад, чи є термін категорією або міткою;
- wp_usermeta – містить метадані користувачів, такі як додаткова інформація про профілі;
- wp_users – таблиця, що зберігає основну інформацію про зареєстрованих користувачів (логін, електронну пошту, пароль тощо).

Окрім стандартних таблиць, які створюються автоматично при встановленні WordPress, до бази даних було додано ще 3 спеціалізовані таблиці. Вони забезпечують реалізацію функцій, необхідних для управління навчальним процесом автошколи:

- exercise_confirmed_day – ця таблиця містить інформацію про підтверджені дати занять. Вона дозволяє системі зберігати дані про доступні дні для проведення практичних занять, враховуючи графіки викладачів. Основними атрибутами є: ідентифікатор запису (ID), дата заняття та ідентифікатор викладача, закріпленого за цим днем;
- schedule_exercise – таблиця, яка відповідає за управління розкладом практичних занять. У ній зберігаються дані про учасників занять (учнів і викладачів), дату, час проведення заняття та унікальні ідентифікатори записів. Це дозволяє автоматизувати формування розкладу, забезпечуючи унікальність кожного запису та синхронізацію з іншими таблицями;
- student – таблиця, яка зберігає інформацію про учнів автошколи. Вона включає атрибути, такі як повне ім’я, група навчання, дата початку та завершення навчання, а також кількість витрачених коштів. Ця таблиця забезпечує зручний доступ до персональних даних учнів, їхнього прогресу та статусу навчання.

Для кожної таблиці були визначені типи даних та додаткові властивості для кожного з її атрибутів. Оскільки детальний опис атрибутів сутностей «exercise_confirmed_day», «schedule_exercise» та «student» наведено в пункті 3.3, у даному розділі надано лише їхні структури, що відображені на рисунках 4.1 – 4.3. Це дозволяє зосередитись на основних характеристиках, зокрема на взаємозв'язках між атрибутами та їхньому функціональному призначенню в системі.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID 	bigint(20)			No	None		AUTO_INCREMENT
2	Data 	date			No	None		
3	Teacher_id 	bigint(20)		UNSIGNED	No	None		

Рисунок 4.1 – Структура таблиці «exercise_confirmed_day»

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID  	bigint(20)		UNSIGNED	No	None		
2	group	text	latin1_swedish_ci		No	None		
3	data_start	date			No	None		
4	data_end	date			No	None		
5	spend_money	int(11)			No	None		
6	name	text	utf8mb4_unicode_520_ci		No	None		

Рисунок 4.2 – Структура таблиці «student»

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID 	bigint(20)			No	None		AUTO_INCREMENT
2	Student_id 	bigint(20)		UNSIGNED	No	None		
3	Teacher_id 	bigint(20)		UNSIGNED	No	None		
4	Date_id 	bigint(20)			No	None		
5	time	int(11)			No	None		

Рисунок 4.3 – Структура таблиці «schedule_exercise»

Опис атрибутів для сутності «wp-user» також детально розглянутий у пункті 3.3, а її структура представлена на рисунку 4.4. Це дозволяє зрозуміти, як саме організовані дані цієї сутності та їх взаємозв'язки з іншими частинами системи.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	user_login	varchar(60)	utf8mb4_unicode_520_ci		No			
3	user_pass	varchar(255)	utf8mb4_unicode_520_ci		No			
4	user_nicename	varchar(50)	utf8mb4_unicode_520_ci		No			
5	user_email	varchar(100)	utf8mb4_unicode_520_ci		No			
6	user_url	varchar(100)	utf8mb4_unicode_520_ci		No			
7	user_registered	datetime			No	0000-00-00 00:00:00		
8	user_activation_key	varchar(255)	utf8mb4_unicode_520_ci		No			
9	user_status	int(11)			No	0		
10	display_name	varchar(250)	utf8mb4_unicode_520_ci		No			

Рисунок 4.4 – Структура таблиці «wp-user»

Опис атрибутів таблиці «wp_posts», яка є однією з основних базових сутностей у системі WordPress, представлений на рисунках 4.5. Ця таблиця містить наступні атрибути:

- ID – унікальний ідентифікатор публікації;
- post_author – ідентифікатор користувача, який створив публікацію;
- post_date – час та дата створення публікації;
- post_date_gmt – час та дата створення в форматі GMT;
- post_content – містить повний вміст публікації, включаючи HTML, шорткоди та інші елементи;
- post_title – назва публікації;
- post_excerpt – короткий опис або версія вмісту публікації;
- post_status – статус публікації (наприклад, опублікована, чернетка);
- post_password – необов'язковий пароль для доступу до публікації;
- post_name – URL-адреса публікації (slug);
- to_ping – список URL-адрес, яким WordPress має надіслати pingbacks при оновленні публікації;
- pinged – список URL-адрес, яким вже було надіслано pingback;

- `post_modified` – дата та час останньої модифікації публікації;
- `post_modified_gmt` – дата та час останньої модифікації в форматі GMT;
- `post_content_filtered` – використовуються для кешування версії вмісту публікації, однак цей атрибут не використовується в ядрі WordPress;
- `post_parent` – використовується для створення ієрархії публікацій, визначаючи батьківську публікацію;
- `guid` – постійний URL до публікації;
- `menu_order` – визначає порядок відображення публікації у меню сторінок;
- `post_type` – ідентифікатор типу вмісту (наприклад, публікація, сторінка, медіафайл);
- `post_mime_type` – тип MIME для завантаженого файлу (застосовується для медіафайлів);
- `comment_count` – кількість коментарів, pingbacks та trackbacks, що відносяться до публікації.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	ID 🗄️	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	post_author 🗄️	bigint(20)		UNSIGNED	No	0		
3	post_date 🗄️	datetime			No	0000-00-00 00:00:00		
4	post_date_gmt	datetime			No	0000-00-00 00:00:00		
5	post_content	longtext	utf8mb4_unicode_520_ci		No	None		
6	post_title	text	utf8mb4_unicode_520_ci		No	None		
7	post_excerpt	text	utf8mb4_unicode_520_ci		No	None		
8	post_status 🗄️	varchar(20)	utf8mb4_unicode_520_ci		No	publish		
9	comment_status	varchar(20)	utf8mb4_unicode_520_ci		No	open		
10	ping_status	varchar(20)	utf8mb4_unicode_520_ci		No	open		
11	post_password	varchar(255)	utf8mb4_unicode_520_ci		No			
12	post_name 🗄️	varchar(200)	utf8mb4_unicode_520_ci		No			
13	to_ping	text	utf8mb4_unicode_520_ci		No	None		
14	pinged	text	utf8mb4_unicode_520_ci		No	None		
15	post_modified	datetime			No	0000-00-00 00:00:00		
16	post_modified_gmt	datetime			No	0000-00-00 00:00:00		
17	post_content_filtered	longtext	utf8mb4_unicode_520_ci		No	None		
18	post_parent 🗄️	bigint(20)		UNSIGNED	No	0		
19	guid	varchar(255)	utf8mb4_unicode_520_ci		No			
20	menu_order	int(11)			No	0		
21	post_type 🗄️	varchar(20)	utf8mb4_unicode_520_ci		No	post		
22	post_mime_type	varchar(100)	utf8mb4_unicode_520_ci		No			
23	comment_count	bigint(20)			No	0		

Рисунок 4.5 – Структура таблиці «wp_posts»

Таблиця «wp_options», представлена на рисунках 4.6, містить такі атрибути:

- option_id – унікальний ідентифікатор для кожної опції в таблиці;
- option_name – ідентифікаційний ключ для конкретного фрагмента даних, що зберігається в таблиці;
- option_value – фактичний фрагмент даних, що відповідає зазначеній опції;
- autoload – вказує, чи повинна опція автоматично завантажуватися при виклику функції wp_load_alloptions(), якщо значення цього поля встановлене в "yes", опція завантажується автоматично при старті WordPress.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	option_id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	option_name 	varchar(191)	utf8mb4_unicode_520_ci		No			
3	option_value	longtext	utf8mb4_unicode_520_ci		No	None		
4	autoload	varchar(20)	utf8mb4_unicode_520_ci		No	yes		

Рисунок 4.6 – Структура таблиці «wp_options»

Таблиця «wp_terms», яка використовується для зберігання термінів, що використовуються в таксономіях представлена на рисунках 4.7, має такі атрибути:

- term_id – унікальний ідентифікатор, призначений кожному терміну;
- name – назва терміна;
- slug – ім'я терміна, яке використовується в URL для зручності доступу;
- term_group – поле, яке дозволяє темам або плагінам групувати терміни разом для певних цілей (наприклад, для створення груп термінів або таксономій).

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	term_id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	name 	varchar(200)	utf8mb4_unicode_520_ci		No			
3	slug 	varchar(200)	utf8mb4_unicode_520_ci		No			
4	term_group	bigint(10)			No	0		

Рисунок 4.7 – Структура таблиці «wp_terms»

4.3 Програмна реалізація вебсервісу

Вебресурс має модульну архітектуру, що забезпечує ефективний розвиток і масштабованість системи. Всі інформаційні матеріали, зокрема текстові дані, зображення та файли сценаріїв, організовані в окремі папки для спрощення доступу та керування ними. Це дозволяє знижувати складність підтримки і розширення вебсайту.

Адміністративна частина сайту чітко відокремлена від користувацького інтерфейсу, що забезпечує безпеку і зручність у керуванні ресурсом. Така структура також дозволяє більш гнучко управляти правами доступу для різних типів користувачів.

Вебресурс побудований з урахуванням принципів мінімізації числа переходів між сторінками, що покращує користувацький досвід і робить навігацію інтуїтивно зрозумілою. Для досягнення високих результатів в SEO-оптимізації, структура URL, мета-дані та контент налаштовані відповідно до найкращих практик, що сприяє кращій видимості вебсайту в пошукових системах.

4.3.1 Розробка загальної частини системи

У цьому розділі описано програмну реалізацію частини системи, доступної для перегляду користувачами, які не авторизовані. Ця частина включає основний інтерфейс сайту, який взаємодіє з відвідувачами.

Для забезпечення зручного та швидкого редагування загальної інформації, яка доступна для перегляду, використовується механізм користувацьких полів. Це дозволяє адміністраторам сайту створювати та оновлювати контент без потреби в знаннях мови розмітки. Усі зміни здійснюються через адміністративну частину сайту, до якої має доступ лише адміністратор.

На кожній сторінці вебресурсу, окрім адміністративної частини, застосовується однакова структура header та footer:

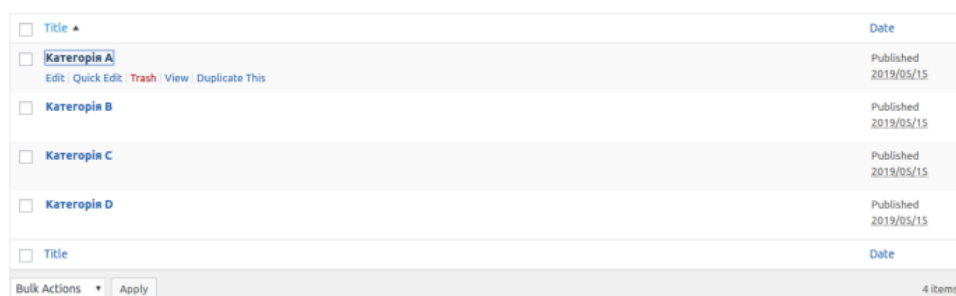
– header – це заголовок сторінки, що містить важливі елементи навігації: логотип сайту, меню для переходу між розділами, форму пошуку та корзину для інтернет-магазину;

– footer – це нижня частина сторінки, яка забезпечує доступ до важливої інформації, такої як копірайт, контактні дані, а також посилання на важливі сторінки (наприклад, політику конфіденційності, умови використання тощо). Ці елементи забезпечують зручність користувачам при перегляді сайту, оскільки потрібна інформація завжди знаходиться під рукою незалежно від того, на якій сторінці вони перебувають.

У системі використовується заголовок сайту, лістинг якого подано в Додатку А, лістинг А.1. Заголовок містить логотип сайту та навігаційне меню. Логотип та кольорове забарвлення заголовка можна легко редагувати за допомогою користувацьких форм, що дозволяє адміністраторам оперативно змінювати візуальні елементи без необхідності втручання в код.

Футер сайту, лістинг якого представлений в Додатку А, лістинг А.2, містить контактну та важливу інформацію. Ці елементи також редагуються за допомогою користувацьких форм, що забезпечує зручність у підтримці актуальності інформації на сайті.

На головній сторінці сайту відображаються категорії, за якими учні автошколи можуть навчатися. Ці категорії наведені на рисунку 4.8. Для створення кожної категорії використовується шаблон – тип спеціальних повідомлень, що має ідентичний стиль відображення. Структура категорії наведена у лістингу 4.1, але з різним змістом, який вказує адміністратор через користувацькі форми.



☐ Title	Date
☐ Kateropia A Edit Quick Edit Trash View Duplicate This	Published 2019/05/15
☐ Kateropia B	Published 2019/05/15
☐ Kateropia C	Published 2019/05/15
☐ Kateropia D	Published 2019/05/15
☐ Title	Date

Bulk Actions ▼ Apply 4 items

Рисунок 4.8 – Категорії навчання

Лістинг 4.1 – Відображення спеціальних категорій навчання

```

<?php
$attr = array(
    'posts_per_page' => 9,
    'post_type' => 'projekt-projects');
$all_bedrift = get_posts($attr);
$all_project = get_field( 'projekt_content');
if($all_bedrift)
{ echo '<section id="projectbox">
<div class="container">
<div class="row">';
    echo      '<div      class="title"><span>Категорії</span>
навчання</div>';
    foreach($all_bedrift as $row)
    { echo '<div class="project_simple_box">';
    echo '<div class="project_simple_box_context text">';
    if ( has_post_thumbnail( $row->ID ) ) {
    $thumb = wp_get_attachment_image_src(
    get_post_thumbnail_id($row->ID), 'large' );}
    $description = get_field( 'projekt_content', $row->ID
);
    $publish = get_field( 'publish_inner_page', $row->ID );
    if ( !empty( $thumb[0] ) ) {
    echo      'post_title.'">';}
    echo '<h5>'. $row->post_title. '</h5>';
    echo '<h6>' . $description['sub_heading_project_field']
.
    '</h6>';
    echo      '<p>' . $description['text_project_field'] .
'</p>';
    if ($publish) {
    echo '<a href="'. get_permalink( $row->ID ) .'"> Se
projekt
    </a>'; }
    echo '</div></div>';}
    echo '</div></div></div></section>';}

```

Відображення підтверджених відгуків на сторінці відгуків приведені у лістингу 4.2.

Лістинг 4.2 – Відображення підтверджених відгуків

```

<div id="comment-<?php comment_ID();?>" class="my_commentlist">
<div class="comment-meta">
    <div class="comment-author vcard">
        <?php
        $avatar_size = 68;
        if ( '0' != $comment->comment_parent )
        $avatar_size = 48;
        echo get_avatar( $comment, $avatar_size );

```

```

        printf( __( '%1$s %2$s' ),
        sprintf( ' <span                                class="fn">%s</span>',
get_comment_author_link() ),
        sprintf( '<a href="%1$s"><time pubdate datetime="%2$s"> %3$s
</time></a>', esc_url( get_comment_link(
$comment->comment_ID ) ), get_comment_time(
'c' ),
        sprintf( __( '%1$s %2$s' ), get_comment_date(),
get_comment_time() ) ); ?>
</div>
</div>
<div class="comment-content">
    <?php comment_text(); ?>
</div>
    <div class="reply" style="padding-bottom: 10px">
        <?php comment_reply_link( array_merge( $args, array(
'reply_text' => __( 'Відповісти', 'my_press' ), 'depth' =>
$depth,
'max_depth' => $args['max_depth'] ) ) ); ?>
    </div>
    <?php edit_comment_link( __( 'Редагувати' ), ' <span
class="editlink">', '</span>' ); ?> </div>

```

4.3.2 Розробка особистого кабінету учня

У цьому розділі описується розробка частини системи, доступної для перегляду авторизованими користувачами, які є учнями автошколи.

Для переходу до частини сайту, призначеної для авторизованих користувачів, в навігаційному меню необхідно вибрати пункт "Учень". Це відкриє сторінку з формою авторизації, де користувач має ввести свій логін та пароль, надіслані системою після успішної реєстрації учня адміністратором.

Адміністратор реєструє учня, вводючи його особисті дані, які будуть використовуватися в системі. При цьому адміністратор також надає роль користувача – абонент системи. Роль абонента обмежує можливості користувача, надаючи йому право лише змінювати налаштування свого облікового запису, не дозволяючи змінювати інші налаштування системи. Це забезпечує безпеку та стабільність роботи вебресурсу.

Після успішної авторизації учень не потрапляє до адміністративної частини сайту, а переходить на закриту сторінку, де доступні функції та інструменти, спеціально розроблені для учнів автошколи. Ці функції дозволяють учням

переглядати матеріали навчання, слідкувати за прогресом і виконувати інші завдання, визначені в рамках навчальної програми.

Лістинг 4.3 – Перехід до особистого кабінету учня

```
if ( is_user_logged_in() && is_admin() ) {
    global $current_user;
    get_currentuserinfo();
    $user_info = get_userdata($current_user->ID);
    if ( $user_info->wp_user_level == 0 )
    {
        header( 'Location: ' . get_home_url() . '/учень/' );
    }
}
```

На сторінці відображення розкладу, зверху розміщується форма для запису на практичне заняття, лістинг якої наведено в лістингу 4.4. Адміністратор сайту має можливість вибирати доступні для запису дні, коли учні можуть записатися на заняття з водіння. У формі необхідно вказати інструктора та дату. Після заповнення цих полів, з'являється додаткове поле, в якому відображаються вільні години для запису, що залежать від вибраного інструктора та дати, як показано в лістингу 4.5.

Запит на отримання вільного часу для запису здійснюється без перезавантаження сторінки завдяки використанню аях-запиту, лістинг якого наведено в лістингу 4.6. Після вибору конкретного часу та натискання кнопки підтвердження, інформація про заняття записується в базу даних і автоматично відображається у розкладі, як показано в лістингу 4.7.

Лістинг 4.4 – Форма для вибору дати, викладача та часу

```
<form action="" method="post">
<?php $blogusers = get_users('role=Author');
echo "<p>Інструктор: <select size='1' name='prep'
onchange=\"selectTime();\">";
echo "<option disabled>Виберіть інструктора</option>";
foreach ($blogusers as $user) {
echo "<option value='".$user->ID.">".$user-
>display_name."</option>"; }
```

```

echo "</select></p>"; ?>
<?php $blogdate = $wpdb->get_results("SELECT * FROM
exercise_confirmed_day");
echo "<p>Дата: <select size='1' name='date'
onchange=\"selectTime();\">";
echo "<option disabled>Виберіть дату практичного
заняття</option>";
foreach ($blogdate as $date_ex) {
if ( $date_ex->Data >= date("Y-m-d")) {
echo "<option value='".$date_ex->ID."'>".$date_ex-
>Data."</option>"; } }
echo "</select></p>";
echo "<p>Час: <select size='1' name='time'>";
echo "</select></p>"; ?>
<p><input type="submit" value="Запис"/></p>
</form>

```

Лістинг 4.5 – Показ доступного часу для вибраної дати та інструктора

```

<?php
switch ($_POST['action']){
case "showTimeForInsert":
$link =
mysqli_connect("localhost","dymilchuam11","Masddasia");
mysqli_select_db($link, "avtoschool-diploma");
$Teacher_id = $_POST['id_prep'];
$Date_id = $_POST['id_data'];
echo "<option disabled>Виберіть час практичного
заняття</option>";
for ($i = 8; $i < 17; $i++) {
$strCHEK = "SELECT * FROM schedule_exercise WHERE Teacher_id =
".$Teacher_id." AND Date_id = ".$Date_id." AND time = ".$i."";
$chek = mysqli_query($link, $strCHEK);
$rowChek = mysqli_fetch_array($chek);
if ( empty($rowChek) ) {
echo "<option value='".$i."'>".$i.":00</option>";}
}
break;
};

```

Лістинг 4.6 – Показ доступного часу для запису без оновлення сторінки

```

function selectTime() {
var id_prep = jQuery('select[name="prep"]').val();
var id_data = jQuery('select[name="date"]').val();
if(!id_prep || !id_data){
jQuery('select[name="time"]').html('');
}else{
jQuery.ajax({

```

```

    }else{
jQuery.ajax({
    type: "POST",
    url: "http://tmp.risingbear.no/avtoschool-diploma/wpcontent/
themes/avtoscholl/page-templates/time_ajax.php",
    data: { action: 'showTimeForInsert', id_prep: id_prep ,
id_data: id_data },
    cache: false,
    success: function(responce){
jQuery('select[name="time"]').html(responce); }
});
};
};
};

```

Лістинг 4.7 – Запис інформації про заняття до бази даних

```

<?php global $wpdb;
if (isset($_POST['date']) && isset($_POST['time']) &&
isset($_POST['prep'])) {
$empty_date = $wpdb->get_results("SELECT * FROM
schedule_exercise WHERE Date_id = '".$_POST['date']."' AND
Teacher_id =
".$_POST['prep']."' AND time = '".$_POST['time']."' ");
if (empty($empty_date)) {
$wpdb->query(" INSERT INTO `schedule_exercise`(`Student_id`,
`Teacher_id`, `Date_id`, `time`) VALUES (
'".$_current_user_id."'
,'".$_POST['prep']."','".$_POST['date']."','".$_POST['time']."'
)
");
$result = 'запис на практичне заняття успішний';}
else { $result = 'запис на практичне заняття невдалий';}
}
else {$result = 'практичне заняття на визначений час вже є';}
}?>

```

Для сповіщення користувача про заняття в системі використовується механізм cron, який забезпечує планування завдань на визначені періоди, фіксовані часи чи дати. У середовищі WordPress цей функціонал реалізується за допомогою класу WP-Cron, який імітує роботу стандартного системного планувальника завдань.

WP-Cron дозволяє виконувати регулярні завдання, такі як надсилання нагадувань про заняття, автоматичне оновлення даних чи перевірка статусу користувачів. Наприклад, для нагадування учню про майбутнє заняття WP-Cron

може бути налаштований так, щоб відправляти повідомлення за 24 години та за годину до його початку.

Лістинг 4.8 – Нагадування про запис на практичне заняття

```
add_filter( 'cron_schedules', 'cron_add_five_min' );
function cron_add_five_min( $schedules ) {
    $schedules['five_min'] = array(
        'interval' => 60 * 60,
    );
    return $schedules;
}
add_action( 'wp', 'my_activation' );
function my_activation() {
    if ( ! wp_next_scheduled( 'my_five_min_event' ) ) {
        wp_schedule_event( time(), 'five_min', 'my_five_min_event' );
    }
}
add_action( 'my_five_min_event', 'do_every_five_min' );
function do_every_five_min() {
    $link = mysqli_connect("localhost","dymilchuaml1","Masddasia");
    mysqli_select_db($link, "avtoschool-diploma");
    $strCHEK = "SELECT * FROM schedule_exercise";
    $chek = mysqli_query($link, $strCHEK);
    while ($row = mysqli_fetch_array($chek))
    {
        $Student_id = get_user_by('id', $row['Student_id']);
        $strCHEKdata = "SELECT * FROM exercise_confirmed_day WHERE ID
        = ".$row['Date_id']."";
        $date_name_chek = mysqli_query($link, $strCHEKdata);
        $date_name = mysqli_fetch_array($date_name_chek);
        $message = 'Нагадуем практичне заняття відбудеться сьогодні в
        '.$row['time'].':00';
        if ( $date_name['Data'] == date("Y-m-d")) {
            if ($row['time'] < (date("G")+6) && $row['time'] >
            (date("G")+5)) {
                wp_mail($Student_id->user_email, 'Заняття з
                практики', $message );
            }
        }
    }
}
```

На сторінці відображення розкладу в нижній частині розміщена інтерактивна форма, яка дозволяє учням переглядати графік практичних занять. Учень може вибрати дату та викладача, після чого система відобразить таблицю розкладу із зазначенням імен та часу занять.

Якщо обраний час і дата відповідають дійсному (наприклад, заняття вже відбулося або заплановане на поточний день), відповідний рядок у таблиці автоматично змінює колір для візуального виділення. Дану функцію продемонстровано у лістингу 4.9.

Лістинг 4.9 – Відображення графіку практичних занять

```
<div style="padding-top: 30px;">
<?php $blogdate = $wpdb->get_results("SELECT * FROM
exercise_confirmed_day");
echo "<form action='' method='post'><p>Дата: <select size='1'
name='date_to_table'>";
echo "<option disabled>Виберіть дату практичного
заняття</option>";
foreach ($blogdate as $date_ex) {
if ( $date_ex->Data >= date("Y-m-d")) {
echo "<option value='". $date_ex->ID. "'>". $date_ex-
>Data."</option>"; }
} echo "</select>";
$blogusers = get_users('role=Author');
echo "<p>Інструктор: <select size='1' name='prep_table'>";
echo "<option disabled>Виберіть інструктора</option>";
foreach ($blogusers as $user) {
echo "<option value='". $user->ID. "'>". $user-
>display_name."</option>"; } echo "</select></p>";
echo "<input type='submit' value='Перегляд' /></p></form>";
?>

<?php if (isset($_POST['date_to_table']) &&
isset($_POST['prep_table'])) {
$prep_name = get_user_by('id', $_POST['prep_table']);
$date_name = $wpdb->get_row("SELECT * FROM
exercise_confirmed_day WHERE ID = ".$_POST
['date_to_table'].");?>
<h4>Інструктор: <?php echo $prep_name->display_name;?></h4>
<h4>Дата: <?php echo $date_name->Data;?></h4>
<table class="schedule_table">
<tr><th>Учень</th><th>Час</th></tr>
<?php for ($i = 8; $i < 17; $i++) {
$blogtime = $wpdb->get_row("SELECT * FROM schedule_exercise
WHERE Date_id = ".$_POST['date_to_table']. " AND Teacher_id =
".$_POST['prep_table']. " AND time = ".$i."");
if ( $blogtime ) {
$user_name = $wpdb->get_row("SELECT * FROM student WHERE ID =
".$blogtime->Student_id."");
if ($blogtime->time == date("G") && $date_name->Data ==
date("Y-m-d")) {echo "<tr><td style='background:
darkgreen;'>". $user_name->name."</td><td style='background:
darkgreen;'>". $blogtime->time.":00</td></tr>"; }
else { echo "<tr><td>". $user_name->name."</td><td>". $blogtime->
time.":00</td></tr>"; }
}
}
```

```

else { echo "<tr><td> </td><td> </td></tr>"; }
} ?>
</table><?php } ?></div>

```

Сторінка з відео заняттями містить відеоплеєр з онлайн трансляцією та коментарями, наведено в лістингу 4.10.

Лістинг 4.10 – Сторінка з відео заняттям та коментарями

```

<?php the_content();
$video_url = get_field( 'rtmp_url' ); ?>
<div style="display: inline-block; text-align: center
padding-left:15px;">
<iframe width="560" height="315"
src="https://www.youtube.com/embed/<?php echo $video_url; ?>"
frameborder="0" allow="accelerometer; autoplay; encrypted-
media;
gyroscope; picture-in-picture" allowfullscreen></iframe>
</div>
<div style="display: inline-block; text-align: center;">
<iframe height="315px"
src="https://www.youtube.com/live_chat?v=<?php echo $video_url;
?>&embed_domain=tmp.risingbear.no"></iframe>
</div>

```

При натисканні на кнопку тестування користувач буде перенаправлений на сторінку, де відкриється сервіс «Офіційні тести ПДР» для проходження онлайн-тестів [18]. На цій сторінці, загальний вигляд якої представлено на рисунку 4.10, буде доступний список доступних тестів, з можливістю вибору конкретного тесту для проходження. Користувач зможе ознайомитися з описом тесту, розпочати його проходження та отримати результати по завершенню.

Дана функція описана у лістингу 4.11.

Лістинг 4.11 – Кнопка перенаправлення на вебресурс з тестуванням

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Кнопка в меню</title>
<style>

```

```

.sidebar {
width: 200px;
background-color: #f4f4f4;
padding: 20px;
height: 100vh;
box-shadow: 2px 0 5px rgba(0, 0, 0, 0.1);}
.menu-button {
display: block;
width: 100%;
padding: 10px;
text-align: center;
background-color: #007bff;
color: white;
text-decoration: none;
border-radius: 5px;
margin-top: 10px;
font-size: 16px;
}
.menu-button:hover {
background-color: #0056b3;
}
</style>
</head>
<body>
<div class="sidebar">
<a href="https://pdr.infotech.gov.ua/tests" target="_blank"
class="menu-button">
Тестування
</a>
</div>
</body>
</html>

```

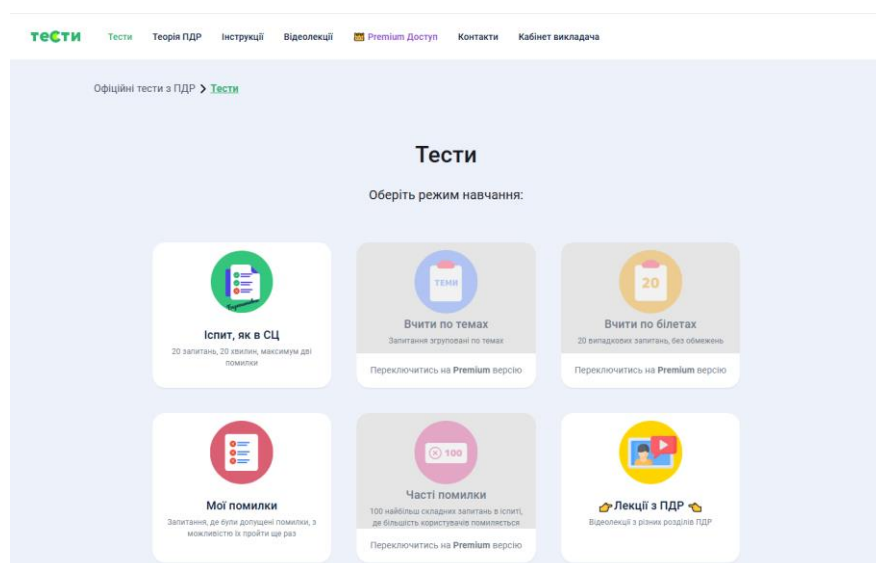


Рисунок 4.10 – Загальний вид сторінки тестування вебсервісу «Офіційні тести ПДР»

4.3.3 Розробка адміністративної частини системи

Адміністратор здійснює реєстрацію нового учня, вводячи його дані, які використовуватимуться системою. Процес показано на рисунку 4.10. Під час створення облікового запису адміністратор призначає учневі логін, після чого на його електронну пошту надходить підтвердження про реєстрацію з посиланням для встановлення пароля. У нижній частині сторінки відображається список усіх активних учнів автошколи, що дозволяє адміністратору переглядати їхні облікові записи та за необхідності видаляти їх із системи.

Лістинг 4.12 – Реєстрація нового користувача

```
<?php
add_action('admin_menu', function() {
add_menu_page( 'Реєстрація користувача', 'user registration',
'manage_options', 'registration-options', 'register_user', '',
4 );
} );
function register_user() {
global $wpdb;
if (isset($_POST['login']) && isset($_POST['mail'])) {
$error = register_new_user($_POST['login'], $_POST['mail']);
if (!is_wp_error($error)) {
$wpdb->query("
INSERT INTO `student`(`ID`, `group`, `data_start`, `data_end`,
`spend_money`, `name`) VALUES ( ' ".$error."
', ' ".$_POST['group']."', ' ".$_POST['start_study']."', ' ".$_POST['e
nd_stud
y']."', ' ".$_POST['money']."', ' ".$_POST['name']."'");
$result = 'користувач зареєстрований'; }
else {$result = 'користувач не зареєстрований'; }
}
if(isset($_POST['submitted']))
{$wpdb->query("DELETE FROM `student` WHERE ID =
' ".$_POST['submitted']."'");
wp_delete_user( $_POST['submitted'] ); }
?>
<div class="wrap">
<form action="" method="post">
<p>Login: <input type="text" name="login" /></p>
<p>Ім'я: <input type="text" name="name" /></p>
<p>Email: <input type="email" name="mail" /></p>
<p>Група: <input type="text" name="group" /></p>
<p>Дата реєстрації: <input type="date" name="start_study"/></p>
<p>Кінець навчання: <input type="date" name="end_study" /></p>
<p>Внесено грошей при реєстрації: <input type="number"
name="money" /></p>
```

```

<p><input type="submit"/></p>
<h1> Result: <?php echo $result ?></h1>
<div> <?php
$blogusers = get_users('role=subscriber');
foreach ($blogusers as $user) {
echo '<ul class="admin-user-list">';
$userid = $user->ID;
$userlist = $wpdb->get_row("SELECT * FROM student WHERE ID =
".$userid.");
echo '<li>' . $user->user_login . '</li>';
echo '<li>' . $userlist->name . '</li>';
echo '<li>' . $user->user_email . '</li>';
echo '<li>' . $userlist->group . '</li>';
echo '<li>' . $userlist->data_start . '</li>';
echo '<li>' . $userlist->data_end . '</li>';
echo '<li>' . $userlist->spend_money . '</li>';
echo '<li><button type="submit" name="submitted"
value="" . $userid . ">Видалити</button></li>';
echo '</ul>';} ?>
</div> </div><?php>

```

The screenshot shows a web interface for user registration. At the top, there is a form with the following fields: Login (with a password icon), Im'я (Name), Email, Група (Group), Дата реєстрації (Registration date) with a date picker showing dd/mm/yyyy, Кінець навчання (End of study) with a date picker showing dd/mm/yyyy, and Внесено грошей (Money paid) with a text input. A Submit button is located below the form. Below the form, the result message "Result: користувач не зареєстрований" (User not registered) is displayed. At the bottom, there is a table listing registered users with columns for name, surname, email, group, registration date, end date, and amount paid. Each row has a "Видалити" (Delete) button next to it.

Ім'я	Прізвище	Електронна пошта	Група	Дата реєстрації	Дата закінчення навчання	Сума грошей	Дії
chumak	Ілля Чумак	chumachok01@gmail.com	kt_2024	2024-10-01	2024-11-01	2500	Видалити
sherbakov	Максим Щербаків	snapok634@gmail.com	kt_2024	2024-10-01	2024-11-01	1223	Видалити

Рисунок 4.11 – Сторінка реєстрації

Адміністратор системи визначає доступні дні та призначає інструкторів для проведення практичних занять, що стають доступними для запису учням.

Лістинг 4.13 – Реєстрація наступного дня для практичних занять

```

<?php
add_action('admin_menu', function(){
add_menu_page( 'Реєстрація днів для практики', 'day
registration', 'manage_options', 'registration-day-options',

```

```

'register_day', '', 4 );
} );
function register_day() {
global $wpdb;
if (isset($_POST['date']) && isset($_POST['prep_table'])) {
$user_exists = $wpdb->get_var(
$wpdb->prepare("SELECT Data FROM exercise_confirmed_day WHERE
Data = '". $_POST['date']."' AND Teacher_id =
".$_POST['prep_table']."'"));
if ($user_exists) {
$result = 'Дата для практичного завдання вже зареєстрована';}
else {$wpdb->query(" INSERT INTO
`exercise_confirmed_day`(`Data`, `Teacher_id`) VALUES
('".$_POST['date']."','".$_POST['prep_table']."'"));
$result = 'Дата для практичного завдання підтверджена';} }
if(isset($_POST['submitted'])) {
$wpdb->query(" DELETE FROM `exercise_confirmed_day` WHERE ID
= '".$_POST['submitted']."'"); } ?>
<div class="wrap">
<form action="" method="post">
<p>Дата реєстрації: <input type="date" name="date" /></p>
<?php $blogusers = get_users('role=Author');
echo "<p>Інструктор: <select size='1' name='prep table'>";
echo "<option disabled>Виберіть інструктора</option>";
foreach ($blogusers as $user) {
echo "<option value='". $user->ID."'>". $user->
display_name."</option>"; } echo "</select></p>"; ?>
<p><input type="submit"/></p>
<h1> Result: <?php echo $result ?></h1>
<div>
<?php $userlist = $wpdb->get_results("SELECT * FROM
exercise_confirmed_day");
foreach ($userlist as $user) {
if ( $user->Data >= date("Y-m-d")) {
echo '<ul class="admin-user-list">';
echo '<li>' . $user->Data . '</li>';
$prep_name = get_user_by('id', $user->Teacher_id);
echo '<li>' . $prep_name->display_name . '</li>';
echo '<li><button type="submit" name="submitted"
value="" . $user->ID . "'>Видалити</button></li>';
echo '</ul>';
} } ?> </div> </div> <?php }

```

4.3.4 Розробка телеграм-бота для перевірки тестування

Наступним кроком у роботі є розробка Телеграм-бота для перевірки результатів тестування. Цей інструмент дозволить автоматизувати процес оцінювання, забезпечуючи зручність для користувачів і ефективність для розробників [19]. Далі ми розглянемо основні етапи створення такого бота, включаючи його функціонал.

Для початку необхідно зареєструвати бота. Відкриваємо додаток Telegram та знаходимо @BotFather, вводимо команду /start. Етап реєстрації нашого бота зображено на рисунку 4.12.

Далі необхідно написати боту @BotFather команду /newbot, задати ім'я і тег. Як результат, ми отримуємо токен, використовуючи який, будемо далі працювати з Телеграм-ботом. Даний процес можна побачити на рисунку 4.13.

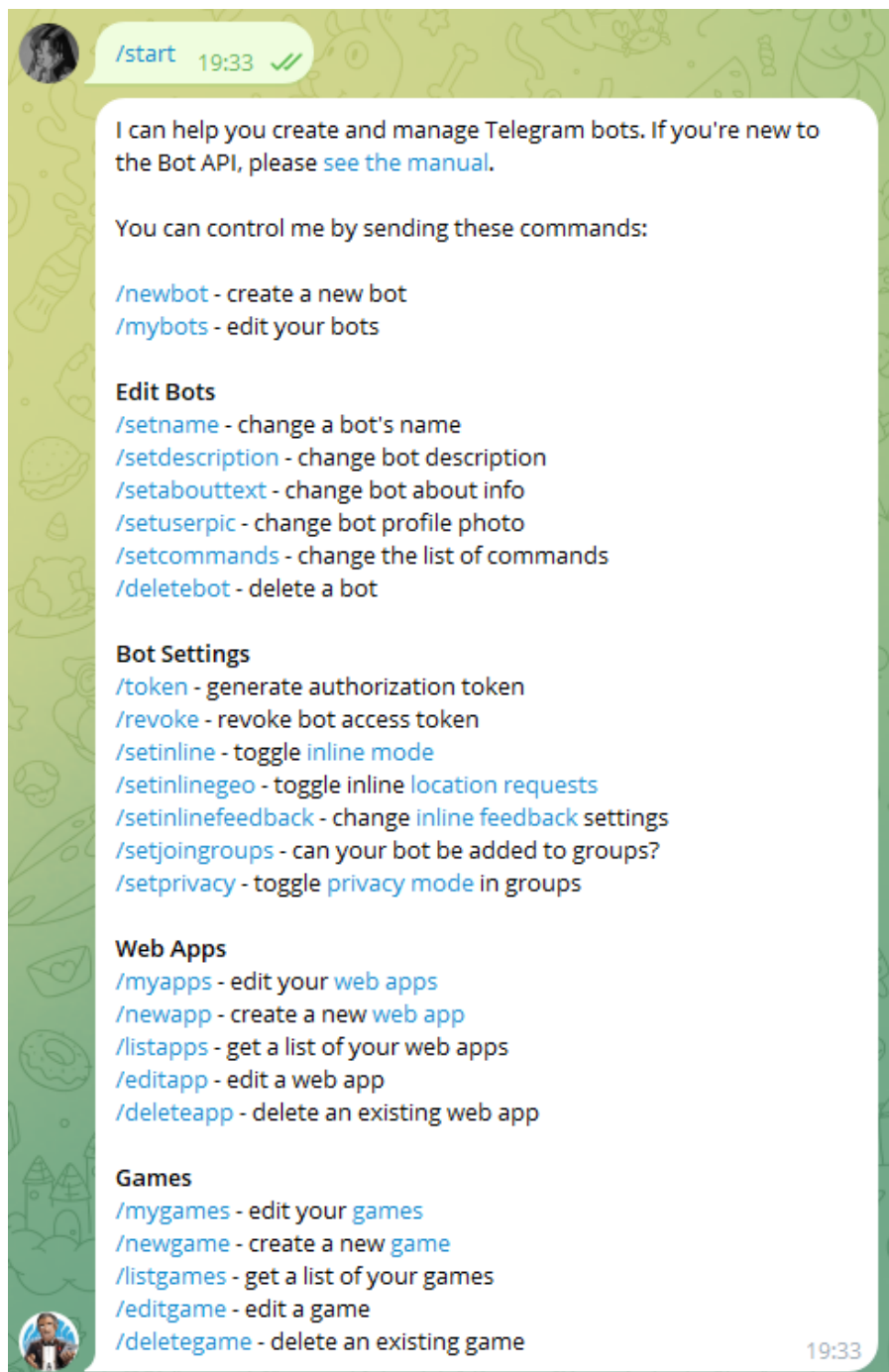


Рисунок 4.12 – Реєстрація бота

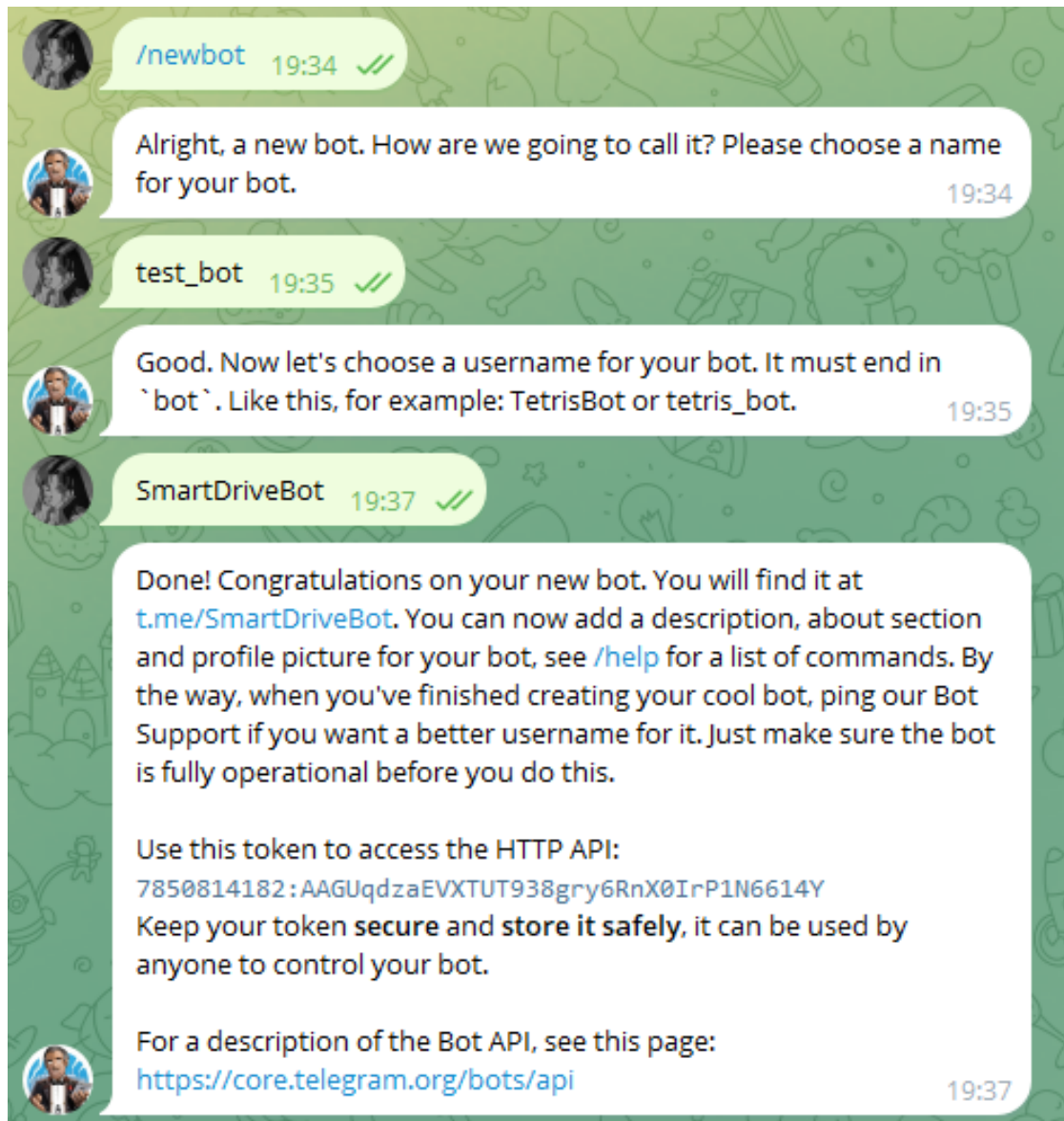


Рисунок 4.13 – Отримання токена для Телеграм-бота

Telegram-бот призначений для спрощення взаємодії учнів з системою тестування автошколи. Він забезпечує швидке отримання результатів тестів та аналіз помилок. Основна функція бота – автоматичне надання посилань на питання, в яких були допущені помилки, для подальшого перегляду і навчання.

Запускаємо бота при натисканні на плаваючий елемент на сторінці вебресурсу автошколи, пишемо команду `/start`. Після запуску бот автоматично надсилає повідомлення з вітанням, повідомляє про отримання результатів і пропонує переглянути зроблені помилки у тесті (рис. 4.14).



Рисунок 4.14 – Загальний інтерфейс телеграм-бота SmartDriveBot

Наступний етап – вибір дії користувачем. Бот надає два варіанти відповіді через інтерактивні кнопки:

- якщо натиснути «Так», то користувач бажає переглянути свої помилки й бот запропонує перейти за посиланням на ті теми, у яких користувач допустив помилку (рис. 4.15);

- якщо натиснути «Ні», то бот нейтрально реагує на відмову і завершує поточний процес (рис. 4.16).

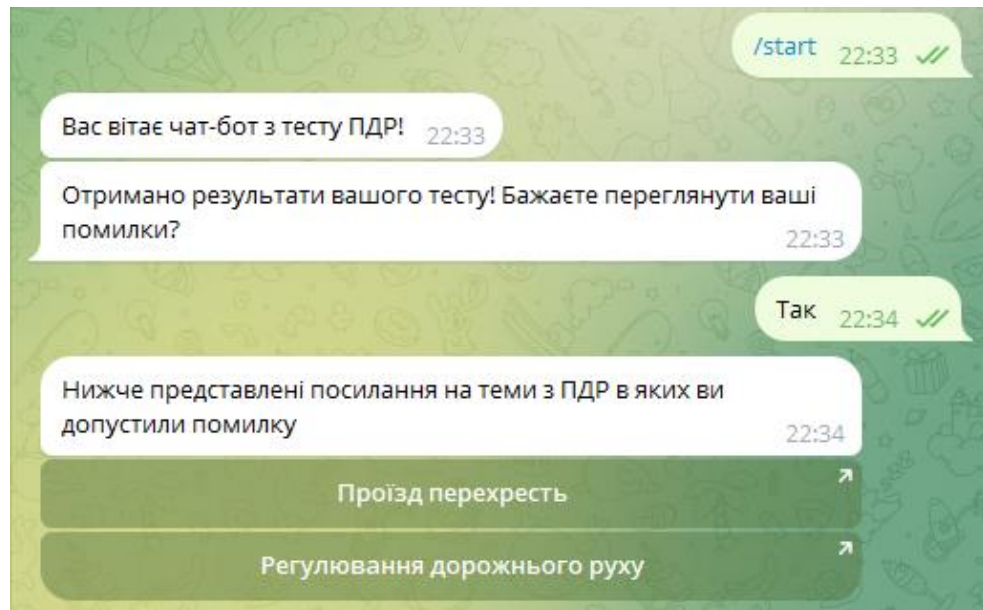


Рисунок 4.15 – Процес при натисканні «Так» у телеграм-боті

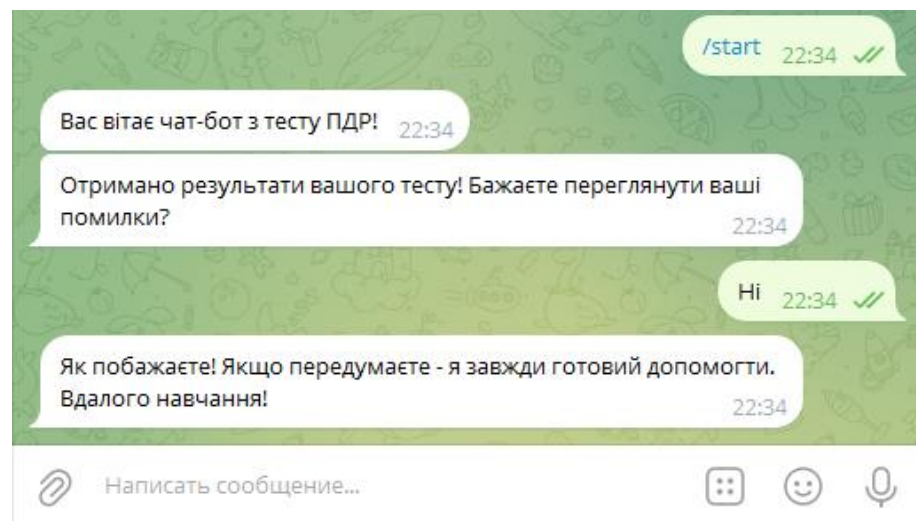


Рисунок 4.16 – Процес при натисканні «Ні» у телеграм-боті

Повний код розробки телеграм-бота «SmartDriveBot» наведено у додатку А, лістинг А3.

4.4 Інтерфейс користувача

На основі макетів сайту, представлених у пункті 3.4, було розроблено сторінки вебресурсу, такі як: головна сторінка, сторінка з відгуками, сторінка особистого кабінету учня з розкладом занять та відеоматеріалами для навчання.

Головна сторінка виконує роль лендінгу, основною метою якого є привернення уваги користувачів і стимулювання їх до виконання певних дій. Її структура, зображена на рисунку 4.17, складається з декількох функціональних блоків:

- слайдер містить заголовок і кнопки для швидкого переходу до форми зворотного зв'язку;
- секція з категоріями навчання демонструє основні напрями підготовки, які пропонує автошкола;
- інформаційні секції дві окремі частини з корисною інформацією для користувачів, що описують особливості автошколи та її послуги.

Сторінка з налаштуваннями включає секції для редагування слайдера головної сторінки, хедера та футера, забезпечуючи зручність у внесенні змін до вмісту, що відображається на сайті.

Налаштування хедера сайту, містять форми для редагування таких елементів:

- логотипу та favicon сайту;
- кольорової гами хедера та кольору шрифтів.

Налаштування футера сайту, включають форми для зміни:

- додаткової текстової інформації;
- контактних даних;
- посилань на соціальні мережі.

Сторінка особистого кабінету учня, стає доступною лише після реєстрації користувача адміністратором у системі та успішного проходження авторизації (форма авторизації показана на рисунку 4.18).

У кабінеті розміщується важлива інформація, вказана адміністратором, а також меню навігації, яке дає доступ до сторінок із розкладом занять, відеоматеріалами для навчання та тестування.



[Контакти](#)
[Ціни](#)
[Відгуки](#)
[Про нас](#)
[Блог](#)



SmartDrive

Автошкола у місті Запоріжжя

ЗАПИСАТИСЬ

Доступні категорії

A

B

B

C

CE

BE

CIE

Етапи отримання прав

1 **Запис до автошколи**
Подача документів та запис до автошколи

2 **Теоретичний курс**
Очі заняття проводяться в ранковий або вечірній час. Також є можливість проводити навчання [ONLINE](#)

3 **Тестові запитання**
Теоретичний іспит в сервісному центрі МВС

Які документи потрібні для навчання в автошколі

Паспорт

ІПН

Фото 2шт. 3x4

Медицина

4 **Практичний курс**
Опановуємо маневри, рух у потоці машин та застосуємо ПДР на вулицях міста

5 **Фінішна праця**
Практичний іспит в сервісному центрі МВС

6 **Отримання посвідчення водія**
Три версії керування 2

НАШІ ЦІЛІ І ПЕРЕВАГИ

Автошкола - це зручний графік теоретичних та практичних занять, сучасний автопарк, право вибору інструктора та автомобіля. Наша мета - підготувати нове покоління водіїв, які будуть досконало знати ПДР та застосовувати ці знання на практиці.

Особливість SmartDrive – розширений спектр послуг. Ми пропонуємо програми не тільки для водіїв-початківців, але і для водіїв з посвідченнями: для відновлення навичок водіння, підвищення кваліфікації.

ПРАКТИЧНІ ЗАНЯТТЯ

«SmartDrive» - автошкола, водіння в якій практикують уже за кілька тижнів після початку навчання. Перші візди здійснюються на спеціально обладнаному автодромі з твердим покриттям та розміткою. Інструктор розповість про основні водіння, дасть відчуття, як працює автомобіль, розповість про особливості керування у різних дорожніх ситуаціях.

Водіння автодромом та автошколами відбувається виключно на обладнаних автомобілях з дублюючими подалеми, що забезпечить безпеку процесу. Учню надається вибір марки автомобіля. Автопарк «SmartDrive» — це ланос, шевроле, опель, шкода, кіа та ауді. Маршрути узгоджуються заздалегідь: вони зображені на схемі, розташованій у класі на стіні, із зазначеними світлофорами, знаками, небезпечними ділянками доріг. Підготовкою водіїв займаються атестовані спеціалісти із відповідним досвідом водіння.

Не знайшли відповідь?

Це не проблема, ми завжди допоможемо!

ЗАПИСАТИСЬ



Ми надаємо авторитетні та професійні послуги. Наша мета – дати вам впевненість у тому, що ви зможете успішно пройти тест та безпечно їздити дорогами.

- вулиця Гмирі 21-а
- 380 63 992 5632
- admin@smartdrive.com

Рисунок 4.17 – Головна сторінка сайту автошколи SmartDrive

Username or Email Address

Password

☐ Remember Me

Log In

Рисунок 4.18 – Форма авторизації користувача

Сторінка з відгуками користувачів, яку розміщено на рисунку 4.19, включає спеціальну секцію, де розміщуються коментарі, попередньо перевірені та підтверджені адміністратором. Також на ній є інтерактивна форма, яка дозволяє відвідувачам залишати нові відгуки. Окрім цього, до кожного відгуку можна додати відповідь — як від адміністратора сайту, так і від інших користувачів. Коментарі від адміністратора публікуються автоматично, а відповіді від звичайних користувачів можуть вимагати перевірки.

SmartDrive

Контакти Ціни Відгуки Про нас Блог

Відгуки

НАВІГАЦІЯ

[Учень](#)

[Розклад занять](#)

[Відео заняття](#)

[Тестування](#)

Дуже радий що обрав саме цю автошколу, професійний та індивідуальний підхід, чинний та ввічливий персонал, також приємний бонус у вигляді знижки задають дуже позитивний темп з самого початку навчання. Справді тут не тільки підготують до екзамену на здану на права, тут навчать їздити, це я вважаю головне. Окрему подяку інструктору Руслану. Компетентний і висококваліфікований інструктор. Зрозуміло пояснив на практиці всі правила дорожнього руху, знає все про "страхи за кермом" і допоможе їх побороти! РЕКОМЕНДУЮ!!!

★★★★★

Павел
14.09.2024

Рекомендую автошколу "Legend" як професійну школу підготовки з нуля. Програма теорії онлайн – це знахідка! Займалась у той час, коли комфортно саме мені, тестові питання в програмі ідеально запитані на іспиті, тому відчувала себе підготовленою та здала іспит з теорії без проблем з 1 помилкою. Особлива подяка моєму інструктору Станіславу за його професіоналізм, терплячість, підтримку та індивідуальний підхід. Відчувалось, що він любить свою справу, терплячий, ввічливий у спілкуванні, вчасно міг підтримати та змотивувати, коли щось одразу не виходило. Якісно підготував до іспиту в сервісному центрі, за що велика подяка. Дуже вдячна цій школі за професійний колектив і високий рівень підготовки. Завдяки школі "Legend" та професійній роботі інструктора, як тільки отримала водійське посвідчення, з першого дня одразу сіла за кермо автомобіля і вільно відчуваю себе на дорозі. Обравши цю школу і отримавши права, впевнилась, що "Legend" найкраща!

★★★★★

Ірина
31.07.2024

Велика подяка автошколі "Legend" Теорія онлайн це дуже зручно, можна навчатися в будь яку годину, будь де! Повна підтримка автошколи, від початку навчання до останнього екзамену! Велика подяка моєму інструктору Станіславу! Рекомендую його, найкращий інструктор! Сдав теорію і практику з першого разу. Дуже вам вдячний!

★★★★★

Роман
30.07.2024

Рисунок 4.19 – Сторінка відгуків до вебресурсу

Сторінка відео занять, показана на рисунку 4.20, має схожу структуру зі сторінкою особистого кабінету учня. Її основний контент – це онлайн-відеотрансляція та реальний час чат для спілкування з викладачем через коментарі.

Для організації онлайн-трансляції використовується сервіс YouTube, що забезпечує широку доступність і простоту використання. Викладач створює або планує трансляцію, після чого генерується URL-посилання, яке включає унікальний ідентифікатор відеотрансляції. Цей ідентифікатор вказується у спеціальному полі налаштувань трансляції на сторінці відео занять, як показано на рисунку 4.21.

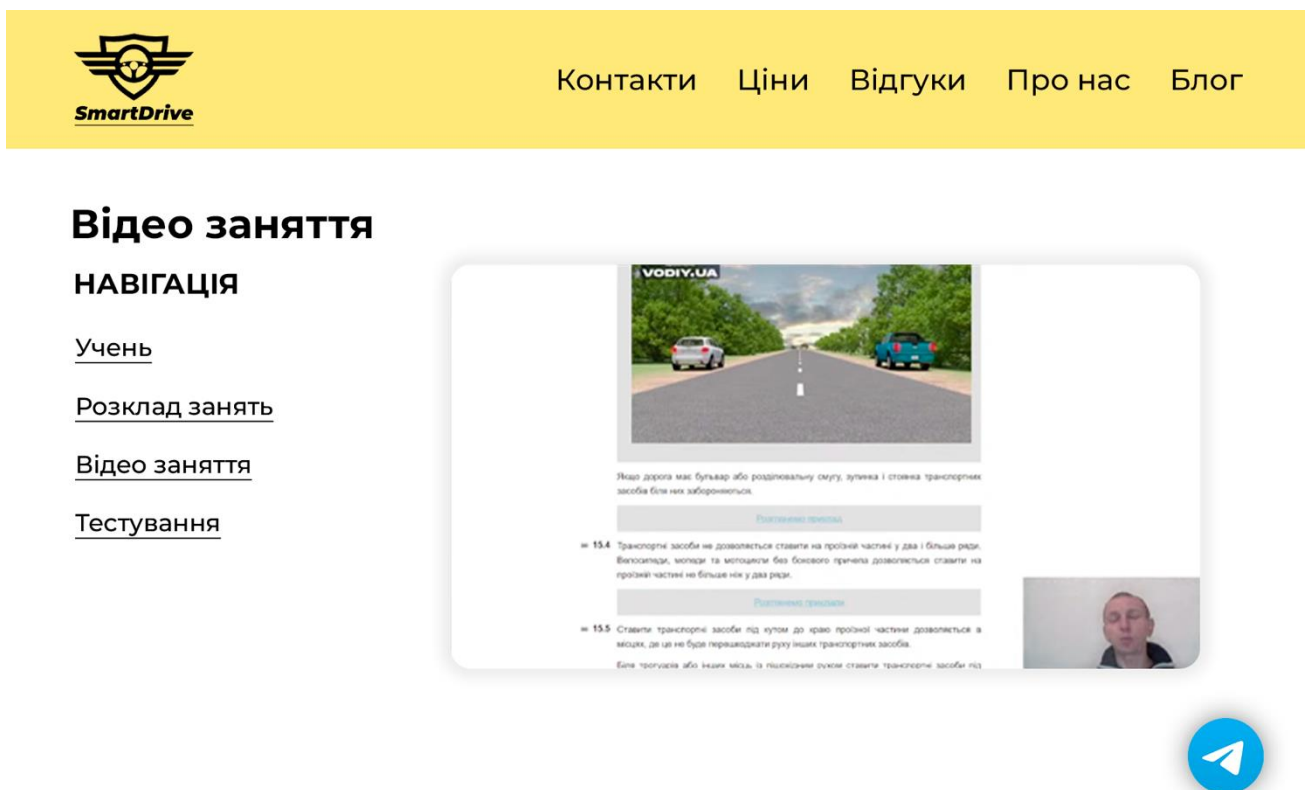


Рисунок 4.20 – Сторінка відео заняття

rtmp
rtmp url
KLUXbCsqbmA

Рисунок 4.21 – Поле з налаштуванням відео трансляції

Сторінка розкладу занять, представлена на рисунку 4.20, має аналогічну структуру зі сторінкою особистого кабінету учня. Основний контент цієї сторінки включає:

- форму для запису на практичне заняття;
- розклад занять, відфільтрований за вибраною датою та викладачем.
-



[Контакти](#)
[Ціни](#)
[Відгуки](#)
[Про нас](#)
[Блог](#)

Розклад занять

НАВІГАЦІЯ

[Учень](#)
[Розклад занять](#)
[Відео заняття](#)
[Тестування](#)

Інструктор

Коваль Володимир Володимирович ▼

Дата

10.10.2024 ▼

Час

12:00 ▼

ЗАПИС

Інструктор

Коваль Володимир Володимирович ▼

Дата

10.10.2024 ▼

ПЕРЕГЛЯД

УЧЕнь	ЧАС
Бицюта Кирило	12:00



Рисунок 4.20 – Сторінка запису на заняття

Розділ тестування, при натисканні надає доступ до онлайн-тестів через сервіс «Офіційні тести ПДР». Для забезпечення ефективного тестування

використовується спеціальна платформа, яка дозволяє користувачам проходити тести в онлайн-режимі та отримувати результат у реальному часі.

Після переходу на сторінку тестування користувач отримує доступ до списку доступних тестів, кожен з яких має свою назву та опис. Для кожного тесту генерується унікальний ідентифікатор, який дозволяє ідентифікувати тест та його результати. Користувачі можуть вибирати тест, що їм підходить, та починати його проходження. Результати тестування відображаються на екрані після завершення тесту, що дозволяє миттєво оцінити рівень знань.

ВИСНОВКИ

У процесі виконання роботи було розроблено інформаційну систему для автошколи, яка спрямована на автоматизацію ключових процесів, покращення якості навчання та підвищення зручності для учнів і адміністрації. Система забезпечує реєстрацію користувачів, запис на заняття, проведення автоматизованого тестування та аналіз результатів.

У роботі проведено детальний аналіз предметної області, визначено основні вимоги до програмного забезпечення та обрано клієнт-серверну архітектуру. У ході роботи створено моделі системи, включаючи діаграми прецедентів і послідовності, які допомогли оптимізувати розробку. Проектування бази даних забезпечило зручне зберігання та доступ до інформації про учнів, заняття та інші елементи навчального процесу.

Особливу увагу приділено розробці інтуїтивно зрозумілого інтерфейсу та інтеграції чат-бота, що сприяє персоналізації навчання й оперативному отриманню зворотного зв'язку. Реалізований вебресурс відповідає сучасним вимогам безпеки, стабільності та продуктивності.

Розроблена система дозволяє знизити адміністративне навантаження, покращити організацію навчання та забезпечити учнів зручним інструментом для доступу до матеріалів і тестування. Подальший розвиток проєкту може включати розширення функціоналу, інтеграцію нових технологій і адаптацію до змін у навчальному процесі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1) Чат-бот. URL: <https://sendpulse.ua/support/glossary/chatbot> (Дата звернення: 01.10.2024)
- 2) Як проходить навчання в автошколі. URL: <https://forsage21.com/ua/blog/kak-prohodit-obuchenie> (Дата звернення: 01.10.2024)
- 3) Автошкола – СТАРТ АВТО. URL: <https://www.startauto.zp.ua> (Дата звернення: 01.10.2024)
- 4) Автошкола – GREEN-WAY. URL: <https://green-way.com.ua/uk> (Дата звернення: 01.10.2024)
- 5) Теорія та методика навчання математики, фізики, інформатики: Збірник наукових праць. Випуск VI: В 3-х томах. – Кривий Ріг: Видавничий відділ НМетАУ, 2006. – Т. 3: Теорія та методика навчання інформатики. – 615 с. (Дата звернення: 01.10.2024)
- 6) Малахов Є.В. Основи проєктування баз даних: Навчальний посібник для студентів вищих навчальних закладів / Є.В. Малахов. – О.: Наука і техніка, 2006. – 156 с. (Дата звернення: 01.10.2024)
- 7) Веб-сервер. URL: <https://sites.znu.edu.ua/webprog/lect/1170.ukr.html> (Дата звернення: 01.10.2024)
- 8) Мова програмування PHP. URL: <https://dou.ua/lenta/articles/how-to-learn-php/> (Дата звернення: 01.10.2024)
- 9) СКБД MySQL. URL: <http://www.znannya.org/?view=mysql-intro> (Дата звернення: 01.10.2024)
- 10) СКБД PostgreSQL. URL: <https://foxminded.ua/postgresql-shcho-tse/> (Дата звернення: 01.10.2024)
- 11) Телеграм-бот та чат-бот. URL: <https://sendpulse.ua/knowledge-base/chatbot/telegram/create-telegram-chatbot> (Дата звернення: 01.10.2024)
- 12) PyCharm. URL: <https://foxminded.ua/pycharm-tse/> (Дата звернення: 01.10.2024)

- 13) Visual Studio Code (VS Code). URL: <https://romul.name/blog/znayomstvo-z-visual-studio-code/> (Дата звернення: 01.10.2024)
- 14) Python-telegram-bot. <https://spacelab.ua/articles/yak-stvoriti-telegram-bota-na-python/> (Дата звернення: 01.10.2024)
- 15) Aiogram. URL: <https://docs.aiogram.dev/uk-ua/latest/dispatcher/middlewares.html> (Дата звернення: 01.10.2024)
- 16) PyTelegramBotApi. URL: <https://devzone.org.ua/post/stvoriuyemo-telegram-bota-na-python-chastyna-1> (Дата звернення: 01.10.2024)
- 17) Wordpress. URL: <https://interkassa.com/blog/oglyad-cms-wordpress-plyusiminusi-prikladi> (Дата звернення: 01.10.2024)
- 18) Офіційні тести з ПДР в Україні. URL: <https://pdr.infotech.gov.ua/tests> (Дата звернення: 01.11.2024)
- 19) Тягунова М.Ю., Бицюта К.В. Використання чат-бота в інформаційній системі автошколи. XII Міжнародна науково-практична конференція "Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій", 10-12 груд. 2024: тези доп. – Запоріжжя. (Дата звернення 01.12.2024)

ДОДАТОК А

Лістинг А.1 – Відображення заголовку сайту

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>SmartDrive - Заголовок</title>
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:
wght@300;400;500;700&display=swap" rel="stylesheet">
  <style>
    body {
      font-family: 'Poppins', sans-serif;
      margin: 0;
    }
    header {
      background-color: #ffcc00;
      padding: 20px 0;
      text-align: center;
    }
    header .container {
      max-width: 1200px;
      margin: 0 auto;
      display: flex;
      justify-content: space-between;
      align-items: center;
    }
    header img {
      max-width: 120px;
    }
    header nav {
      display: flex;
      justify-content: center;
      gap: 20px;
    }
    header nav a {
      text-decoration: none;
      font-weight: 500;
      color: #000;
    }
    header a.signup {
      text-decoration: none;
      background-color: #000;
      color: #fff;
      padding: 10px 20px;
      border-radius: 5px;
    }

```

```

        font-size: 16px;
    }
</style>
</head>
<body>
    <header>
        <div class="container">
            <!-- Логотип -->
            <div>
                
            </div>
            <nav>
                <a href="#categories">Категорії</a>
                <a href="#steps">Етапи навчання</a>
                <a href="#advantages">Переваги</a>
                <a href="#contacts">Контакти</a>
            </nav>
            <div>
                <a href="#signup" class="signup">Записатися</a>
            </div>
        </div>
    </header>
</body>
</html>

```

Лістинг А.2 – Відображення футера сайту

```

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>SmartDrive - Футер</title>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;
500;700&display=swap" rel="stylesheet">
    <style>
        body {
            font-family: 'Poppins', sans-serif;
            margin: 0;
        }
        footer {
            background-color: #ffcc00;
            padding: 20px 0;
            text-align: center;
            color: #000;
        }
        footer .container {
            max-width: 1200px;
            margin: 0 auto;
            display: flex;
            flex-wrap: wrap;
            justify-content: space-between;

```

```

        align-items: center;
    }
    footer img {
        max-width: 120px;
    }
    footer p {
        margin: 5px 0;
    }
    footer a {
        color: #000;
        text-decoration: none;
    }
    footer a:hover {
        text-decoration: underline;
    }
</style>
</head>
<body>
    <footer>
        <div class="container">
            <div>
                
            </div>
            <div>
                <p>вулиця Гмири 21-а</p>
                <p>Телефон: <a href="tel:+380639925632">
+380 63 992 5632</a></p>
                <p>Email: <a href="mailto:admin@smartdrive.com">
admin@smartdrive.com</a></p>
            </div>
        </div>
    </footer>
</body>
</html>

```

Лістинг А.3 – Телеграм-бот з тестування

```

from aiogram.types import ReplyKeyboardMarkup,
KeyboardButton, InlineKeyboardMarkup, InlineKeyboardButton
import requests
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton,
InlineKeyboardMarkup, InlineKeyboardButton
from bs4 import BeautifulSoup
question = ReplyKeyboardMarkup(keyboard=[
    [KeyboardButton(text='Так')],
    [KeyboardButton(text='Hi')]],
    resize_keyboard=True,
    one_time_keyboard=True,
    input_field_placeholder='Оберіть пункт меню...')
topics = InlineKeyboardMarkup(inline_keyboard=[])
def get_mistakes():
    headers = {
        "user-agent": "Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212
Safari/537.36"

```

```

    }
    url = "https://pdr.infotech.gov.ua/tests/my-mistakes"
    r = requests.get(url=url, headers=headers)
    soup = BeautifulSoup(r.text, "lxml")
    topic = soup.find_all("div", class_="exam_question")
    for t in topic:
        tname = t.find("div", class_="exam_question_topic").text.strip()
        turl = t.find("a").get("href")
        topics.add(InlineKeyboardButton(text=tname,
url=turl,callback_data=turl))
    from aiogram import F, Router
    from aiogram.types import Message, CallbackQuery
    from aiogram.filters import CommandStart, Command
    from getmk import get_mistakes
    import keyboards as kb
    router = Router()
    @router.message(CommandStart())
    async def cmd_start(message: Message):
        await message.answer('Вас вітає чат-бот з тесту ПДР!')
        await message.answer('Отримано результати вашого тесту!
Бажаєте переглянути ваші помилки?', reply_markup=kb.question)
    @router.message(Command('help'))
    async def cmd_help(message: Message):
        await message.answer('Ви натиснули кнопку допомоги')
    @router.message(F.text == 'Так')
    async def mistakes(message: Message):
        get_mistakes()
        #await message.answer('Нижче представлені посилання на теми
з ПДР в яких ви допустили помилку')
        await message.answer('Нижче представлені посилання на теми
з ПДР в яких ви допустили помилку',reply_markup=kb.topics)
    @router.message(F.text == 'Hi')
    async def mistakes(callback: Message):
        #await message.answer('Нижче представлені посилання на теми
з ПДР в яких ви допустили помилку')
        await callback.answer('Як побажаєте! Якщо передумаєте - я
завжди готовий допомогти. Вдалого навчання!', )
    import asyncio
    from aiogram import Bot, Dispatcher
    from handlers import router
    async def main():
        bot = Bot(token='7831618709:AAHILuiKR-
New7SOC2Lu2ILWG73nWMTzqc4')
        dp = Dispatcher()
        dp.include_router(router)
        await dp.start_polling(bot)
    if name == 'main':
        asyncio.run(main())

```