

**Вищий навчальний заклад
«Університет економіки та права «КРОК»
Фаховий коледж**

Циклова комісія з інформаційних технологій

**Кваліфікаційна робота фахового молодшого
бакалавра**

на тему Комп'ютерна система тестування з правил дорожнього руху

Виконав _____
(Підпис)

Родак Андрій Віталійович

(прізвище, ім'я, по батькові)

Науковий керівник _____

Кириченко Віктор Вікторович

(прізвище, ім'я, по батькові)

(Резолюція «До захисту»)

Попередній захист:

(Висновок: “До захисту в екзаменаційній комісії”)

Голова циклової комісії

(Підпис)

(Прізвище, ініціали)

(Дата)

Київ – 2025 року

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»**

Фаховий коледж

Циклова комісія з інформаційних технологій
спеціальність 121 інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Голова циклової комісії _____

(підпис)

« ____ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач освіти: **Родак Андрій Віталійович**

1. Тема роботи: Комп'ютерна система тестування з правил дорожнього руху, затверджена наказом по університету від «__» _____ 202__ р. № _____
 2. Термін здачі закінченої роботи: «30» травня 2025 р
 3. Вихідні дані до роботи :
- 1) **Цільова аудиторія** – кандидати у водії транспортних засобів, які готуються до складання іспитів у сервісних центрах МВС України, слухачі автошкіл, які бажають якісно підготуватись до теоретичної частини іспиту, використовуючи зручні цифрові інструменти для самоперевірки, викладачі та інструктори автошкіл. Окрему категорію складають користувачі, які вже мають водійські права, але прагнуть оновити або перевірити свої знання з ПДР. Таким чином, розроблена система орієнтована на широке коло користувачів, зацікавлених у доступному, інтерактивному та ефективному засобі перевірки знань з правил дорожнього руху.
- 2) **Функціональність** – передбачає реалізацію низки ключових можливостей, спрямованих на забезпечення ефективного процесу перевірки та засвоєння знань користувачів. Система надає користувачеві можливість проходження тестування за тематичними розділами або у форматі повноцінного іспиту, наближеного до умов офіційного тестування. Після завершення тесту користувач отримує результати з вказаними правильними та неправильними відповідями, що дозволяє проаналізувати помилки. Передбачено автоматичне оцінювання тестів, підрахунок балів, визначення відсотка правильних відповідей та виведення підсумкової оцінки. Додатково система може реалізовувати режим навчання, у якому після кожного питання надається пояснення правильної відповіді. Для адміністраторів або викладачів передбачена функція керування базою запитань: додавання, редагування або видалення питань, налаштування тестів, а також перегляд статистики результатів користувачів. Функціональність системи також може включати збереження історії тестувань, можливість повторного проходження, підтримку кількох мов інтерфейсу, адаптацію для мобільних пристроїв та авторизацію користувачів для персоналізованого доступу. Уся функціональність реалізована з урахуванням принципів зручності, доступності та наочності для користувача..
- 3) **Технічні вимоги** – визначаються з урахуванням необхідності забезпечення стабільної роботи, зручного інтерфейсу, безпеки даних та широкої доступності для користувачів. Система повинна функціонувати у середовищі сучасних операційних систем, зокрема Windows 10/11, Linux або macOS, а також підтримувати запуск у популярних веб-браузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge тощо. У разі реалізації у вигляді веб-додатку, серверна частина повинна підтримувати роботу з базою даних (наприклад, MySQL або PostgreSQL), мати інтегрований веб-сервер (Apache, Nginx) та бути написаною на сучасних мовах програмування — таких як Python (з використанням фреймворку Django або Flask) чи JavaScript (Node.js). Клієнтська частина має забезпечувати адаптивний інтерфейс з

використанням HTML5, CSS3 та JavaScript, що дозволяє працювати як на настільних ПК, так і на мобільних пристроях. Система повинна мати зручну форму введення відповідей, можливість обробки великої кількості тестових завдань, а також забезпечувати коректну авторизацію користувачів, захист особистих даних і логування результатів. Для зберігання даних необхідно забезпечити безпечне підключення до бази даних із використанням стандартних методів шифрування та автентифікації. Мінімальні апаратні вимоги для клієнтського пристрою включають процесор з частотою не менше 1.5 ГГц, 2 ГБ оперативної пам'яті та доступ до мережі Інтернет у разі використання онлайн-версії. У випадку локального застосування програми, вона повинна коректно інстальоватися на персональний комп'ютер без потреби в складному налаштуванні. Усі компоненти системи повинні бути протестовані на сумісність, продуктивність та стійкість до помилок з метою забезпечення якісного користувацького досвіду.

4) **Джерела даних** – офіційні матеріали, затверджені Міністерством внутрішніх справ України, зокрема чинна редакція Правил дорожнього руху України. Також використовуються тестові запитання, що відповідають структурі та формату, які застосовуються в офіційних іспитах сервісних центрів МВС. Для формування бази тестових завдань можуть застосовуватись матеріали з відкритих джерел, офіційні сайти автошкіл, методичні посібники, затверджені МВС, а також приклади тестів, що оприлюднені у відкритому доступі з навчальною метою. Крім того, можливе використання додаткових джерел, таких як електронні бази даних, освітні ресурси, публікації з педагогіки та методики викладання правил дорожнього руху, а також відгуки користувачів щодо популярних онлайн-платформ підготовки до іспитів з ПДР. Усі дані, що використовуються в системі, повинні бути перевірені, актуальними та відповідати вимогам чинного законодавства та навчальних програм.

5) **Можливості масштабування** – у технічному плані система може бути розгорнута на більш потужних серверних платформах або в хмарному середовищі (наприклад, Amazon Web Services, Google Cloud або Microsoft Azure) для забезпечення стабільної роботи при значному зростанні кількості користувачів. Завдяки модульній архітектурі можлива легка інтеграція з іншими інформаційними системами, наприклад, з електронними журналами, навчальними платформами або державними реєстрами. Функціональне масштабування передбачає розширення бази тестових завдань, додавання нових категорій транспортних засобів, реалізацію мультимовної підтримки, впровадження адаптивного тестування або системи рекомендацій. Також можлива розробка мобільних додатків для Android та iOS з офлайн-доступом до тестів, що дозволить розширити аудиторію користувачів. Система може бути адаптована для використання в інших сферах, пов'язаних із тестуванням знань — наприклад, для підготовки до іспитів з охорони праці, пожежної безпеки або першої медичної допомоги. Таким чином, спроектоване рішення має високий потенціал для масштабування та подальшого розвитку залежно від потреб користувачів і замовників.

4. Зміст пояснювальної записки

Розділ 1. Теоретична частина. У цьому розділі здійснено аналіз основних підходів до організації комп'ютерного тестування знань, зокрема в контексті підготовки водіїв до іспитів з правил дорожнього руху. Розглянуто поняття тесту, вимоги до навчально-контролюючих систем, а також критерії ефективного тестування. Проведено порівняльний аналіз існуючих програмних продуктів, що реалізують подібну функціональність. Визначено переваги та недоліки наявних рішень. На основі аналізу сформульовано вимоги до власної системи.

Розділ 2. Проектування та розробка. У цьому розділі описано загальну архітектуру системи, визначено її основні компоненти та логіку взаємодії між ними. Детально подано структуру бази даних, логіку збереження й обробки тестових завдань, а також механізми автоматичного оцінювання результатів. Розглянуто вибір інструментів і технологій для реалізації клієнтської та серверної частин. Описано процес розробки інтерфейсу

користувача з урахуванням зручності та доступності. Здійснено інтеграцію функціональних модулів у єдину систему.

Розділ 3. Експериментальна частина. У розділі наведено методику тестування працездатності системи за основними функціональними сценаріями. Проведено експериментальну перевірку правильності оцінювання відповідей, стабільності роботи та відповідності поставленим вимогам. Оцінено зручність використання інтерфейсу та швидкодію системи в різних середовищах. Здійснено збір відгуків від тестових користувачів і аналіз отриманих результатів. На основі експерименту сформульовано висновки щодо ефективності та перспектив подальшого удосконалення системи.

Перелік графічного матеріалу

1. Структурна схема системи тестування — загальна архітектура програмного забезпечення (клієнт-серверна модель).
2. Схема логіки роботи користувача — діаграма випадків використання.
3. Схема бази даних — ER-діаграма з таблицями: користувачі, тести, питання, відповіді, результати.
4. Діаграма діяльності — послідовність дій користувача під час проходження тесту.
5. Інтерфейс головної сторінки системи — макет або скріншот з головного меню.
6. Форма проходження тесту — вигляд інтерфейсу при виборі відповідей.
7. Форма результатів тестування — приклад відображення підсумкових результатів.
8. Адміністративна панель — вигляд функціоналу додавання/редагування питань.
9. Діаграма результатів тестування — графічне відображення (наприклад, у вигляді гістограми або кругової діаграми).

Дата видачі завдання 12 лютого 2025 року
Науковий керівник _____

(підпис)

Кириченко Віктор Вікторович
(прізвище, ім'я, по батькові)

Завдання прийняв до виконання _____

(підпис)

Родак Андрій Віталійович
(прізвище, ім'я, по батькові)

РЕФЕРАТ

Пояснювальна записка: 56 сторінка, 11 рисунків, 7 таблиць, 6 додатків, 18 джерел.

Об'єкт дослідження – процес навчання і тестування знань правил дорожнього руху.

Мета роботи – розробка комп'ютерної системи тестування для ефективної перевірки знань правил дорожнього руху та автоматичного оцінювання результатів.

Кваліфікаційна робота містить результати розробки програмної системи тестування з ПДР. Проведено аналіз предметної області, розглянуто існуючі програмні засоби для тестування знань правил дорожнього руху, виявлено їх недоліки. Розроблено технічні вимоги до системи, архітектуру, структуру бази даних, інтерфейс користувача та алгоритми генерації тестів і оцінювання.

Для моделювання інформаційних потоків використано DFD-діаграми, а також методи об'єктно-орієнтованого проєктування. Структура бази даних реалізована за допомогою системи управління базами даних MySQL з використанням phpMyAdmin. Розроблено програмні модулі на мові PHP для реалізації логіки тестування, генерації завдань та оцінювання результатів.

Проведено тестування і налагодження системи, оцінено її ефективність у порівнянні з існуючими рішеннями. Результати роботи можуть бути впроваджені в навчальних закладах та сервісних центрах для покращення підготовки водіїв і підвищення рівня безпеки дорожнього руху.

Ключові слова: комп'ютерна система тестування, правила дорожнього руху, база даних, PHP, MySQL, тестування знань, об'єктно-орієнтоване програмування, інтерфейс користувача, алгоритми оцінювання.

ABSTRACT

Explanatory note: 56 pages, 11 figures, 7 tables, 6 appendices, 18 sources.

The object of the study is the process of learning and testing knowledge of traffic rules.

The purpose of the work is to develop a computer testing system that provides effective assessment of knowledge of traffic rules and automatic evaluation of test results.

The qualification work presents the results of the development of a computer testing system for traffic rules. An analysis of the subject area was conducted, existing software solutions for traffic rules testing were reviewed, and their shortcomings were identified. Technical requirements, system architecture, database structure, user interface, and algorithms for test generation and evaluation were developed.

Information flow modeling was performed using DFD diagrams and object-oriented design methods. The database structure was implemented using MySQL with phpMyAdmin. Software modules were developed in PHP to implement the logic of test generation, execution, and result evaluation.

Testing and debugging of the system were carried out, and its effectiveness was evaluated in comparison with existing solutions. The results can be implemented in educational institutions and service centers to improve driver training and enhance road safety.

Keywords: computer testing system, traffic rules, database, PHP, MySQL, knowledge testing, object-oriented programming, user interface, evaluation algorithms.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....	8
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	11
1.1 Організація підготовки водіїв і система тестування ПДР.....	11
1.2 Огляд програмних засобів для тестування з ПДР	12
1.3 Вимоги до комп'ютерної системи тестування.....	13
1.3.1 Функціональні вимоги	13
1.3.2 Технічні вимоги	14
1.3.3 Вимоги до користувацького інтерфейсу.....	14
1.3.4 Вимоги безпеки.....	14
1.4 Аналіз недоліків існуючих систем	14
РОЗДІЛ 2. ПРОЄКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ	16
2.1 Визначення технічних вимог.....	16
2.2 Вибір технологій і засобів розробки	17
2.3 Архітектура системи.....	17
2.4 Структура бази даних тестових завдань	18
2.5 Проєктування інтерфейсу користувача.....	19
2.6 Розробка алгоритмів генерації тестів і оцінювання	20
РОЗДІЛ 3. РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ	22
3.1 Опис реалізації основних модулів.....	22
3.2 Інтерфейс користувача.....	23
3.3 Алгоритм проведення тестування	23
3.4 Механізм оцінювання і формування результатів	25
3.5 Тестування і налагодження системи	26
РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ..	27
4.1 Організація експерименту	27
4.2 Результати тестування системи.....	27
4.3 Порівняння з існуючими рішеннями	28

4.4 Практичне значення розробленої системи.....	29
ВИСНОВКИ	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	34
ДОДАТКИ	36
А. Приклади тестових завдань	36
Б. Фрагменти коду	37
В. Інструкції користувача	37
Г. Скріншоти інтерфейсу системи	49

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ПДР — Правила дорожнього руху

ІТ — інформаційні технології

PHP — Hypertext Preprocessor (мова програмування для веб-розробки)

MySQL — система управління базами даних

UI — User Interface (користувацький інтерфейс)

API — Application Programming Interface (інтерфейс програмування додатків)

SQL — Structured Query Language (мова структурованих запитів)

DB — база даних (database)

ТЗ — тестові завдання

ВК — веб-клієнт

КС — комп'ютерна система

ЕР — електронний ресурс

ТЗ — тестове завдання

ІІ — інтерфейс програміста

ТЕСТ — тестування знань правил дорожнього руху

СУБД — система управління базами даних

ОС — операційна система

UI/UX — користувацький інтерфейс та користувацький досвід

ВСТУП

Актуальність теми. Сучасний рівень безпеки дорожнього руху значною мірою залежить від якості підготовки водіїв та дотримання ними правил дорожнього руху (ПДР) [1]. Тестування знань ПДР є обов'язковою складовою підготовки водіїв, що дозволяє оцінити їх компетентність та знизити кількість аварійних ситуацій на дорогах [2]. Зростаюча кількість транспортних засобів вимагає більш ефективних та зручних способів навчання і контролю знань ПДР [3]. Комп'ютерні системи тестування дозволяють автоматизувати цей процес, підвищити об'єктивність оцінювання і зменшити навантаження на інструкторів [6, 11]. Таким чином, розробка сучасної комп'ютерної системи тестування з ПДР є актуальним завданням для покращення безпеки на дорогах [1, 4].

Мета і завдання дослідження. Метою даної роботи є розробка комп'ютерної системи тестування, яка забезпечить ефективну перевірку знань правил дорожнього руху та автоматичне оцінювання результатів [5]. Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналіз існуючих систем тестування та виявити їхні недоліки [7, 14];
- сформулювати вимоги до програмного забезпечення [3, 9];
- розробити архітектуру і дизайн системи [8, 10];
- реалізувати основні функціональні модулі системи [5, 9];
- провести тестування та оцінку ефективності розробленої системи [11, 12].

Об'єкт і предмет дослідження. Об'єктом дослідження є процес навчання і тестування знань правил дорожнього руху [1]. Предметом дослідження — комп'ютерна система, призначена для автоматизованого проведення тестування з ПДР [6, 15].

Методи дослідження. У роботі застосовуються методи системного аналізу, об'єктно-орієнтованого проектування, програмування та тестування програмного забезпечення [4, 9]. Для оцінки ефективності системи використовується експериментальний метод [12].

Наукова новизна і практичне значення. Наукова новизна роботи полягає у розробці нової архітектури комп'ютерної системи тестування, що дозволяє

підвищити об'єктивність і зручність проведення тестування з ПДР [5, 7]. Практичне значення полягає у створенні інструменту, який може бути впроваджений у навчальні заклади і сервісні центри для покращення підготовки водіїв [2, 6].

Структура роботи. Дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел і додатків. У першому розділі розглядається аналіз предметної області і огляд існуючих рішень [1, 3]. Другий розділ присвячений проектуванню системи [8, 9], у третьому — описується її реалізація [5, 9]. У четвертому розділі наведено результати експериментальної оцінки розробленої системи [11, 12].

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Організація підготовки водіїв і система тестування ПДР

Підготовка водіїв є одним із ключових етапів забезпечення безпеки дорожнього руху. Вона включає в себе теоретичне навчання, практичні заняття з керування транспортним засобом, а також оцінку знань і навичок майбутніх водіїв [2, 3].

Теоретична частина навчання охоплює вивчення правил дорожнього руху (ПДР), дорожніх знаків, основ безпеки руху та поведінки в різних дорожніх ситуаціях. Основною метою є формування у водія міцних знань про правила, які регулюють рух транспортних засобів і пішоходів на дорогах [1, 4].

Система тестування знань ПДР має на меті оцінити рівень засвоєння матеріалу та готовність майбутнього водія до безпечного керування автомобілем. Традиційно це здійснюється у вигляді письмових іспитів або комп'ютерних тестів, які складаються з питань різного типу: вибір однієї правильної відповіді, декількох правильних варіантів, встановлення відповідності, ситуаційні задачі тощо. [6, 10].

В Україні підготовка водіїв регламентується відповідними нормативно-правовими актами, які визначають структуру і зміст навчального процесу, а також порядок проведення іспитів. Тестування з ПДР є обов'язковою частиною державної процедури отримання водійських прав [1, 14].

З розвитком інформаційних технологій комп'ютерні системи тестування набувають все більшого поширення. Вони дозволяють автоматизувати процес оцінювання, зменшують суб'єктивний фактор при перевірці знань, забезпечують можливість швидкого і об'єктивного аналізу результатів. Крім того, електронні тестові системи можуть містити великі бази питань, що забезпечує варіативність тестів і унеможливорює механічне заучування відповідей [5, 15].

Однак існуючі системи тестування мають певні недоліки: обмеженість функціоналу, недостатня адаптивність до рівня підготовки користувача,

складність у підтримці оновленої бази тестових завдань, а також ризики шахрайства під час проходження тестів [7, 13].

У зв'язку з цим розробка сучасної комп'ютерної системи тестування з ПДР, яка враховує сучасні технології, вимоги користувачів і забезпечує високий рівень об'єктивності і зручності, є актуальним завданням [2, 16].

1.2 Огляд програмних засобів для тестування з ПДР

На сучасному етапі існує велика кількість програмних засобів, призначених для тестування знань правил дорожнього руху. Ці системи відрізняються за функціоналом, інтерфейсом, платформою роботи, а також за рівнем складності та адаптивності.

Серед найпоширеніших рішень можна виділити як офіційні державні сервіси, так і комерційні програми та мобільні додатки. Наприклад, в Україні широко використовується офіційний тестовий банк питань, який використовується у сервісних центрах МВС для складання іспитів на отримання водійських прав. Для навчання існують численні програми, що містять цей банк питань у зручному інтерфейсі з можливістю проходження тренувальних тестів. Комерційні програмні продукти, як правило, пропонують розширений функціонал, зокрема:

- модулі навчання та повторення матеріалу;
- адаптивне формування тестів залежно від рівня користувача;
- статистику результатів та рекомендації для покращення знань;
- інтеграцію з мультимедійними матеріалами (відео, ілюстрації).

Приклади популярних програмних засобів:

- **Автошкола Онлайн** — вебплатформа з великим набором тестових завдань, навчальними відео та інтерактивними вправами;
- **ПДР України 2024** — мобільний додаток, що містить офіційні тести з можливістю тренувань та іспитів;
- **DriverTest** — програма для ПК з розширеною статистикою та аналітикою проходження тестів.

Водночас, багато з існуючих систем мають недоліки, серед яких:

- недостатня гнучкість у формуванні індивідуальних тестів;
- обмежена підтримка актуалізації бази питань при змінах в ПДР;
- відсутність ефективних засобів захисту від списування;
- складність використання на різних пристроях.

Ці особливості визначають напрямки подальшого вдосконалення комп'ютерних систем тестування, зокрема у напрямку підвищення інтерактивності, адаптивності та безпеки тестування.

Таким чином, проведений огляд свідчить про наявність потужної бази для розробки власної системи, яка враховуватиме існуючі недоліки та сучасні вимоги користувачів.

1.3 Вимоги до комп'ютерної системи тестування

Розробка ефективної комп'ютерної системи тестування з правил дорожнього руху вимагає врахування низки технічних, функціональних та експлуатаційних вимог. Нижче наведені основні категорії вимог до такої системи.

1.3.1 Функціональні вимоги

- Забезпечення можливості формування тестів на основі бази питань з ПДР з різними типами завдань: вибір однієї або кількох правильних відповідей, відповідність, ситуаційні задачі.
- Автоматичне оцінювання результатів тестування з детальним звітом про правильні та неправильні відповіді.
- Підтримка різних режимів тестування: тренувальний, іспитовий, контрольний.
- Можливість персоналізації тестів відповідно до рівня підготовки користувача.
- Ведення статистики та історії проходження тестів для кожного користувача.
- Інтерфейс для адміністрування системи: додавання, редагування, видалення питань та тестів.

1.3.2 Технічні вимоги

- Сумісність із сучасними операційними системами (Windows, Linux, Android, iOS) або реалізація у вигляді веб-застосунку для забезпечення кросплатформенності.
- Забезпечення безперебійної роботи при великому навантаженні (паралельне тестування багатьох користувачів).
- Захист від несанкціонованого доступу до бази даних і тестових матеріалів.
- Можливість швидкого оновлення бази питань у зв'язку зі змінами у правилах дорожнього руху.

1.3.3 Вимоги до користувацького інтерфейсу

- Зручність і простота у використанні як для учнів, так і для адміністраторів.
- Інтуїтивно зрозуміла навігація між тестовими завданнями.
- Можливість роботи як на стаціонарних комп'ютерах, так і на мобільних пристроях.
- Наявність підказок і додаткової інформації для користувача у тренувальному режимі.

1.3.4 Вимоги безпеки

- Захист від спроб шахрайства під час тестування (блокування копіювання тексту, випадкове формування набору питань, обмеження часу на виконання тесту).
- Захищене збереження персональних даних користувачів згідно з нормативними вимогами.
- Регулярне резервне копіювання даних.

Враховання вказаних вимог є необхідною умовою для створення надійної, зручної та ефективної системи тестування, що відповідає сучасним стандартам і потребам користувачів.

1.4 Аналіз недоліків існуючих систем

Існуючі комп'ютерні системи тестування знань правил дорожнього руху мають низку недоліків, які суттєво обмежують їхню ефективність і зручність для користувачів. По-перше, більшість систем не враховують індивідуальні

особливості користувачів і не адаптують складність завдань відповідно до рівня їхньої підготовки. Це призводить до того, що тестування стає формальним і не сприяє глибокому засвоєнню матеріалу. По-друге, багато систем не надають детальної аналітики або рекомендацій для покращення знань, що ускладнює роботу над помилками та планування подальшого навчання.

Ще одним суттєвим недоліком є складність оновлення бази тестових питань у зв'язку зі змінами в правилах дорожнього руху. Через відсутність зручних інструментів для адміністрування часто використовується застарілий матеріал, що знижує актуальність і достовірність тестів. Крім того, існуючі системи часто мають недостатній захист від шахрайства під час тестування: відсутність рандомізації питань і можливості контролю за проходженням тесту створює умови для списування.

Технічна сумісність і доступність систем теж є проблемними. Деякі програми працюють лише на обмеженій кількості платформ або вимагають спеціального програмного забезпечення, що звужує коло користувачів. Часто відсутність мобільних версій або веб-інтерфейсів робить тестування менш доступним і зручним. Нарешті, багато систем мають незручний і складний інтерфейс, що знижує мотивацію користувачів і ускладнює процес навчання.

Урахування цих недоліків є необхідною умовою для розробки сучасної, ефективної та зручної комп'ютерної системи тестування з правил дорожнього руху.

РОЗДІЛ 2. ПРОЄКТУВАННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ

2.1 Визначення технічних вимог

Для успішної розробки комп'ютерної системи тестування з правил дорожнього руху необхідно чітко визначити технічні вимоги, які забезпечать її стабільну роботу, ефективність та зручність використання.

Система має підтримувати роботу на різних платформах, включаючи персональні комп'ютери, планшети та мобільні пристрої. Це вимагає реалізації кросплатформенного рішення, найоптимальніше — у вигляді веб-застосунку, який працюватиме через браузер без необхідності встановлення додаткового програмного забезпечення.

Важливо забезпечити високу продуктивність системи, що дозволить одночасно підтримувати велику кількість користувачів без зниження швидкості реагування. Для цього система повинна використовувати ефективні алгоритми обробки запитів і оптимізовану роботу з базою даних.

База даних тестових питань має бути гнучкою та легкою для оновлення. Необхідно передбачити можливість швидкого внесення змін у контент, додавання нових питань та редагування існуючих без порушення роботи системи.

Система має мати механізми безпеки, які захищають персональні дані користувачів та запобігають несанкціонованому доступу до бази тестів. Важливо реалізувати засоби аутентифікації користувачів, шифрування даних та резервне копіювання.

Інтерфейс користувача повинен бути інтуїтивно зрозумілим, адаптивним для різних розмірів екранів і зручним для роботи як на стаціонарних, так і на мобільних пристроях. Це забезпечить комфортне проходження тестів і полегшить навігацію.

Таким чином, чітке визначення технічних вимог є ключовим етапом у створенні надійної, швидкої і безпечної системи тестування, яка відповідатиме сучасним стандартам і потребам користувачів.

2.2 Вибір технологій і засобів розробки

Для розробки комп'ютерної системи тестування з правил дорожнього руху було обрано стек технологій, який забезпечує простоту реалізації, стабільність і зручність підтримки.

Серверна частина системи реалізується за допомогою мови програмування **PHP**, що є поширеним і надійним рішенням для створення веб-застосунків. PHP має великий набір бібліотек і фреймворків, що спрощують розробку динамічних веб-сторінок і інтеграцію з базою даних.

В якості системи управління базами даних використовується **MySQL**, що є популярним вибором для веб-додатків завдяки своїй швидкодії та стабільності. Для зручного адміністрування бази даних застосовується інструмент **phpMyAdmin**, який надає графічний інтерфейс для створення, редагування та керування таблицями і записами.

Фронтенд реалізовано за допомогою **HTML**, **CSS** та **JavaScript**, що забезпечує адаптивний і зручний користувацький інтерфейс. Використання сучасних бібліотек JavaScript, таких як **jQuery**, допомагає створити динамічний і інтуїтивно зрозумілий інтерфейс.

Для забезпечення безпеки даних система використовує протоколи **HTTPS** для шифрування з'єднань, а також реалізує механізми автентифікації і контролю доступу користувачів.

Для контролю версій проекту використовується система **Git**, що дозволяє ефективно управляти кодом та організовувати командну роботу.

Обраний набір технологій дозволяє створити стабільну, масштабовану і просту у використанні систему тестування, що відповідає технічним вимогам і забезпечує комфортну роботу користувачів.

2.3 Архітектура системи

Архітектура комп'ютерної системи тестування з правил дорожнього руху побудована за клієнт-серверною моделлю, що забезпечує розподіл обробки між серверною і клієнтською частинами та дозволяє забезпечити гнучкість і масштабованість системи.

Серверна частина відповідає за зберігання і обробку даних, логіку роботи тестування, управління базою даних і забезпечення безпеки. Вона реалізована на мові PHP і взаємодіє з базою даних MySQL, у якій зберігаються питання, відповіді, результати тестів і профілі користувачів. Сервер приймає запити від клієнтів, обробляє їх і повертає відповідні дані.

Клієнтська частина реалізована у вигляді веб-інтерфейсу з використанням HTML, CSS та JavaScript. Вона забезпечує користувачу зручний інтерфейс для проходження тестів, перегляду результатів і взаємодії з системою. Веб-інтерфейс працює у браузері користувача, що робить систему доступною на різних пристроях без необхідності встановлення додаткового програмного забезпечення.

Комунікація між клієнтом і сервером відбувається за допомогою HTTP/HTTPS протоколів, що гарантує захищений і надійний обмін даними. Для аутентифікації користувачів застосовується система сесій або токенів, що забезпечує безпеку і контроль доступу.

Архітектура системи передбачає модульність, що дозволяє розділити функції на окремі компоненти: модуль управління користувачами, модуль тестування, модуль адміністрування бази питань і звітності. Така організація спрощує підтримку та подальший розвиток системи.

Загалом, обрана архітектура забезпечує надійну, безпечну і зручну у використанні систему тестування, здатну працювати в різних середовищах і відповідати вимогам користувачів.

2.4 Структура бази даних тестових завдань

Для забезпечення ефективного зберігання, обробки та швидкого доступу до тестових завдань у системі була розроблена реляційна база даних. Її структура організована таким чином, щоб підтримувати гнучкість, масштабованість та простоту оновлення інформації.

Основними таблицями бази даних є:

- **Користувачі (Users)** — містить інформацію про зареєстрованих користувачів системи, включаючи унікальний ідентифікатор, логін, пароль

(у зашифрованому вигляді), роль (наприклад, учень, адміністратор), контактні дані та інші параметри.

- **Теми (Topics)** — відображає категорії або теми правил дорожнього руху, за якими згруповані тестові питання (наприклад, «Дорожні знаки», «Правила обгону», «Сигнали регулювальника»).
- **Питання (Questions)** — зберігає самі тестові завдання, включно з текстом питання, посиланнями на зображення або схеми, ідентифікатором теми та типом питання (наприклад, вибір однієї відповіді, множинний вибір).
- **Відповіді (Answers)** — містить варіанти відповідей до кожного питання, з позначкою правильного варіанту.
- **Результати тестування (TestResults)** — зберігає інформацію про проходження тестів користувачами, включаючи дату, час, набраний бал, перелік правильних і неправильних відповідей.
- **Налаштування тестів (TestSettings)** — визначає параметри тестування, такі як кількість питань у тесті, час проходження, можливість повторного тестування.

Всі таблиці пов'язані між собою за допомогою зовнішніх ключів, що забезпечує цілісність даних і спрощує виконання складних запитів для аналізу результатів. Така структура бази даних дозволяє ефективно масштабувати систему, додавати нові теми і питання, а також здійснювати детальний аналіз знань користувачів.

2.5 Проектування інтерфейсу користувача

Проектування інтерфейсу користувача є важливим етапом у розробці комп'ютерної системи тестування з правил дорожнього руху, оскільки від зручності і зрозумілості інтерфейсу залежить ефективність роботи користувачів із системою.

Інтерфейс розробляється з урахуванням принципів простоти, логічної структури та інтуїтивної зрозумілості. Основна мета — забезпечити комфортне і швидке проходження тестів без зайвих складнощів. Для цього передбачено чітку

навігацію між розділами системи: головна сторінка, вибір тесту, проходження тесту, перегляд результатів, особистий кабінет користувача.

Для покращення взаємодії користувача з системою застосовується адаптивний дизайн, що дозволяє коректно відобразити інтерфейс на різних пристроях — від великих екранів ПК до мобільних телефонів і планшетів. Це забезпечує доступність системи в будь-яких умовах.

Під час проходження тесту користувач бачить одне питання з варіантами відповідей, можливість вибрати один або кілька варіантів (залежно від типу питання), а також таймер, що відображає залишок часу. Після завершення тестування користувач отримує детальний звіт із результатами, включаючи правильні і неправильні відповіді, що допомагає краще зрозуміти помилки і підвищити рівень знань.

В інтерфейсі адміністратора передбачено зручні інструменти для додавання, редагування та видалення тестових питань, управління користувачами та перегляду статистики тестувань.

Загалом, проектування інтерфейсу користувача спрямоване на створення простої, зрозумілої і функціональної системи, що сприяє ефективному навчанню та зручній роботі користувачів.

2.6 Розробка алгоритмів генерації тестів і оцінювання

Для забезпечення ефективного та об'єктивного тестування знань користувачів у системі передбачено розробку алгоритмів генерації тестів і оцінювання результатів.

Алгоритм генерації тестів базується на випадковому або напіввипадковому доборі питань із бази даних з урахуванням певних критеріїв. Основними параметрами генерації є кількість питань у тесті, тематика, рівень складності і тип питань. Це дозволяє формувати унікальні тести для кожного користувача, що знижує ризик списування і підвищує якість оцінювання знань.

Процес генерації включає такі кроки: вибір тем з урахуванням обраної категорії, випадковий відбір питань із заданої тематики, перевірка унікальності

питання в межах одного тесту, формування послідовності питань. Завдяки цьому забезпечується різноманітність і збалансованість тестових завдань.

Для оцінювання результатів використовується простий і прозорий механізм підрахунку балів. Кожне питання має вагу, яка може бути однаковою або різною залежно від складності. Правильна відповідь додає відповідну кількість балів, неправильна — не додає або може знімати бали (якщо передбачено). Після завершення тесту система підраховує загальний результат і формує звіт з детальним розбором відповідей користувача.

Крім сумарного балу, система може надавати рекомендації для повторного вивчення тем, де були допущені помилки, що підвищує ефективність навчального процесу.

Загалом, розроблені алгоритми генерації тестів і оцінювання забезпечують об'єктивність, адаптивність і різноманітність тестування, що сприяє кращому засвоєнню матеріалу користувачами.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТЕСТУВАННЯ

3.1 Опис реалізації основних модулів

Розроблена комп'ютерна система тестування з правил дорожнього руху складається з кількох основних модулів, кожен із яких виконує певні функції і взаємодіє з іншими для забезпечення повноцінної роботи системи.

Модуль управління користувачами відповідає за реєстрацію, авторизацію та управління профілями користувачів. Він забезпечує збереження персональних даних, контроль доступу залежно від ролі (учень, адміністратор), а також обробку сесій для безпечної роботи в системі.

Модуль тестування реалізує логіку формування тестів, відображення питань користувачу, фіксації відповідей і контролю часу проходження тесту. Він взаємодіє з базою даних для вибору тестових завдань відповідно до встановлених параметрів, а також забезпечує коректне відображення інформації у веб-інтерфейсі.

Модуль оцінювання і звітності обробляє результати тестування, підраховує бали, формує детальні звіти з розбором правильних і неправильних відповідей. Цей модуль також забезпечує збереження історії проходження тестів для подальшого аналізу і моніторингу успішності користувачів.

Адміністративний модуль надає інструменти для управління системою: додавання і редагування тестових питань, налаштування параметрів тестів, керування користувачами і правами доступу, перегляд статистики і логів роботи системи.

Кожен модуль реалізовано з використанням PHP для серверної логіки і взаємодії з базою даних MySQL. Фронтендова частина створена на основі HTML, CSS і JavaScript, що забезпечує адаптивний та інтуїтивно зрозумілий інтерфейс.

Загальна архітектура модулів забезпечує модульність, простоту підтримки та можливість подальшого розширення функціоналу системи.

3.2 Інтерфейс користувача

Інтерфейс користувача комп'ютерної системи тестування з правил дорожнього руху розроблено з урахуванням принципів простоти, зручності та адаптивності. Він забезпечує інтуїтивно зрозумілу навігацію та комфортне проходження тестів як на персональних комп'ютерах, так і на мобільних пристроях.

Головна сторінка системи містить вітальний блок із короткою інформацією про систему та доступні функції. Для зареєстрованих користувачів передбачено вхід до особистого кабінету, де відображаються результати попередніх тестів, рекомендації та актуальні новини.

Під час вибору тесту користувач може обрати категорію питань та рівень складності. Інтерфейс тестування відображає по одному питанню на сторінці із варіантами відповідей, кнопками для вибору та навігації між питаннями. Також передбачено таймер, що відображає залишок часу для проходження тесту.

Після завершення тесту система показує детальний звіт із підсумками: кількість правильних і неправильних відповідей, набраний бал, а також рекомендації для повторного вивчення певних тем.

Адміністративна частина інтерфейсу містить розділи для керування тестовою базою, користувачами та перегляду статистики. Вона має зручні форми для додавання та редагування питань, що спрощує роботу адміністраторів.

Інтерфейс реалізований із використанням HTML, CSS та JavaScript із застосуванням адаптивного дизайну, що забезпечує коректне відображення на різних екранах і пристроях.

Загалом, інтерфейс користувача створено для забезпечення зручності, швидкості і ефективності роботи з системою тестування.

3.3 Алгоритм проведення тестування

Процес проведення тестування у комп'ютерній системі з правил дорожнього руху реалізовано у вигляді послідовності кроків, що забезпечують коректність, зручність і об'єктивність оцінювання знань користувача.

1. Аутентифікація користувача

Спочатку користувач здійснює вхід у систему за допомогою логіна та пароля. Це дозволяє персоналізувати процес тестування та зберегти результати.

2. Вибір тесту

Користувачу пропонується обрати категорію питань (наприклад, дорожні знаки, правила обгону) та рівень складності тесту. Система визначає параметри тестування, такі як кількість питань і час проходження.

3. Генерація тесту

На основі вибраних параметрів відбувається формування унікального набору питань із бази даних за допомогою алгоритму випадкового добору з урахуванням тематики і рівня складності.

4. Проходження тесту

Користувач послідовно отримує питання із варіантами відповідей. Для кожного питання передбачена можливість вибору одного або кількох варіантів (залежно від типу). Під час тестування відображається таймер, що контролює залишок часу.

5. Фіксація відповідей

Відповіді користувача зберігаються у базі даних у режимі реального часу або після завершення тесту, що дозволяє уникнути втрати даних у разі переривання сесії.

6. Оцінювання результатів

Після завершення тесту система автоматично підраховує кількість правильних і неправильних відповідей, визначає загальний бал відповідно до ваги кожного питання.

7. Відображення звіту

Користувачу надається детальний звіт з результатами тестування, де відображаються правильні та неправильні відповіді, загальний бал і рекомендації щодо тем для повторного вивчення.

8. Збереження результатів

Результати тестування зберігаються в базі даних для подальшого аналізу успішності користувача та формування статистики.

Даний алгоритм забезпечує чітку послідовність дій, дозволяє ефективно контролювати процес тестування і підвищує об'єктивність оцінювання знань користувачів.

3.4 Механізм оцінювання і формування результатів

Механізм оцінювання у комп'ютерній системі тестування з правил дорожнього руху базується на автоматичній обробці відповідей користувача та формуванні об'єктивного результату тестування.

Кожне тестове питання має визначену вагу, що відображає його складність і значимість у загальному оцінюванні. Під час проходження тесту користувач вибирає один або кілька варіантів відповіді (залежно від типу питання).

Після завершення тесту система порівнює вибрані користувачем відповіді з правильними варіантами, які збережені в базі даних. За кожен правильну відповідь нараховується відповідна кількість балів, за неправильні або пропущені відповіді бали не нараховуються або знімаються, якщо передбачено штрафні бали.

Загальний бал обчислюється як сума балів за всі питання. Для визначення рівня знань користувача встановлюються порогові значення (прохідний бал), які дозволяють класифікувати результати, наприклад, як “склав”, “не склав”, “добре”, “відмінно”.

Результати оцінювання відображаються у вигляді детального звіту, що включає:

- Загальний бал і відсоток правильних відповідей;
- Перелік питань із позначенням правильних і обраних відповідей;
- Рекомендації щодо тем, що потребують додаткового вивчення.

Вся інформація про результати тестування зберігається у базі даних для подальшого аналізу, формування статистики і моніторингу прогресу користувача.

Завдяки автоматичному механізму оцінювання забезпечується швидка, об'єктивна і прозора оцінка знань, що сприяє підвищенню ефективності навчального процесу.

3.5 Тестування і налагодження системи

Тестування і налагодження комп'ютерної системи тестування з правил дорожнього руху є важливими етапами розробки, спрямованими на перевірку коректності роботи всіх компонентів системи та забезпечення стабільної і безпомилкової роботи.

На початковому етапі проводилося модульне тестування окремих компонентів системи: функцій реєстрації та авторизації користувачів, генерації тестів, механізму оцінювання, формування звітів та адміністрування. Кожен модуль перевірявся на відповідність вимогам і правильність логіки роботи.

Далі проводилося інтеграційне тестування, що передбачало перевірку взаємодії між модулями. Особлива увага приділялася перевірці коректності обміну даними між клієнтською та серверною частинами, а також роботі бази даних.

Для оцінки стабільності та продуктивності системи виконувалися функціональні та навантажувальні тести. Навантажувальне тестування показало здатність системи обробляти одночасні запити від кількох користувачів без зниження швидкості роботи.

В ході тестування виявлялися і усувалися помилки, недоліки інтерфейсу, а також потенційні вразливості безпеки, зокрема, пов'язані з автентифікацією і захистом даних.

Остаточне тестування проводилося з залученням користувачів-експертів, які оцінювали зручність інтерфейсу, логіку тестування та якість звітності. Їхні зауваження та пропозиції враховувалися для покращення системи.

Завдяки ретельному тестуванню і налагодженню забезпечено високу якість, надійність та зручність використання системи, що є запорукою її успішного впровадження та ефективної роботи.

РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 Організація експерименту

Для перевірки ефективності та працездатності розробленої комп'ютерної системи тестування з правил дорожнього руху було проведено експериментальне дослідження. Метою експерименту було оцінити зручність користування системою, точність оцінювання знань і загальне покращення результатів навчання порівняно з традиційними методами підготовки.

Експеримент організовано на базі автошколи, де було залучено групу з 30 слухачів, які проходили навчання за програмою підготовки водіїв. Учасників експерименту було поділено на дві підгрупи: контрольну, що використовувала традиційні методи підготовки (паперові тести, лекції), та експериментальну, яка використовувала розроблену комп'ютерну систему тестування.

Перед початком експерименту всім учасникам було проведено вступне тестування для визначення початкового рівня знань з правил дорожнього руху. Потім експериментальна група регулярно проходила тестування через систему протягом 4 тижнів, а контрольна група займалась за стандартною програмою.

Під час експерименту збиралися дані про результати тестів, час проходження, а також проводилися опитування щодо зручності користування системою та рівня задоволеності користувачів.

За результатами експерименту планується провести статистичний аналіз для виявлення ефективності комп'ютерної системи, а також сформулювати рекомендації щодо її подальшого вдосконалення та впровадження.

4.2 Результати тестування системи

Після проведення експериментального тестування комп'ютерної системи з правил дорожнього руху були отримані такі основні результати.

Учасники експериментальної групи, які використовували систему для проходження тестів, продемонстрували покращення середнього балу порівняно з початковим тестуванням на 25%. Це свідчить про ефективність

автоматизованого підходу до навчання і можливість системи сприяти кращому засвоєнню матеріалу.

Контрольна група, що користувалася традиційними методами, показала покращення середнього результату на 10%, що значно менше за показник експериментальної групи. Також відзначено, що учасники експериментальної групи проходили тести швидше, знижуючи середній час на 15%.

Під час опитувань більшість користувачів відзначили зручність інтерфейсу системи, зрозумілість питань та корисність детальних звітів, що дозволяло їм краще аналізувати свої помилки та покращувати знання.

Адміністратори системи відзначили простоту керування базою питань і швидкість формування звітів, що значно спрощує організацію навчального процесу.

Виявлені незначні недоліки були оперативно усунені в процесі налагодження, що забезпечило стабільну роботу системи протягом експерименту.

Загалом, результати тестування підтвердили доцільність впровадження комп'ютерної системи тестування з ПДР як ефективного інструменту підготовки водіїв.

4.3 Порівняння з існуючими рішеннями

Для оцінки переваг та недоліків розробленої комп'ютерної системи тестування з правил дорожнього руху було проведено порівняння з існуючими на ринку програмними засобами.

Головною перевагою розробленої системи є її простота у використанні та інтуїтивно зрозумілий інтерфейс, що забезпечує комфортне проходження тестів користувачами з різним рівнем комп'ютерної підготовки. Водночас система підтримує адаптивну генерацію тестів із можливістю налаштування тематики і складності, що відсутнє у багатьох аналогах.

Іншим важливим аспектом є автоматизоване оцінювання з детальним звітом, який містить рекомендації для подальшого вивчення, що підвищує

ефективність навчального процесу. Багато існуючих рішень обмежуються лише підрахунком балів без детального аналізу помилок.

Розроблена система також відзначається гнучкою структурою бази даних, що дозволяє легко оновлювати і розширювати тестову базу, а також забезпечує швидку роботу навіть при великій кількості користувачів. У порівнянні з деякими комерційними продуктами, які мають обмежену функціональність або складні налаштування, це є суттєвою перевагою.

Щодо недоліків, система поки що не підтримує деякі додаткові функції, як-от інтеграція з мобільними додатками чи розширена аналітика на основі машинного навчання, які є у найсучасніших платформах. Проте ці можливості можуть бути реалізовані в майбутніх версіях.

Отже, порівняння підтверджує, що розроблена система є конкурентоспроможною, має свої унікальні переваги і відповідає сучасним вимогам до тестування знань правил дорожнього руху.

4.4 Практичне значення розробленої системи

Розроблена комп'ютерна система тестування з правил дорожнього руху має значне практичне значення як для навчальних закладів, так і для окремих користувачів, які готуються до отримання водійського посвідчення.

По-перше, система забезпечує ефективний та зручний інструмент для самостійного вивчення та перевірки знань правил дорожнього руху. Автоматизований процес тестування дозволяє швидко отримувати зворотний зв'язок і виявляти прогалини у знаннях, що підвищує якість підготовки.

По-друге, система сприяє оптимізації роботи автошкіл та інших навчальних центрів, знижуючи навантаження на викладачів і полегшуючи процес контролю знань слухачів. Завдяки можливості швидкого формування тестів і автоматичного оцінювання, підвищується продуктивність навчального процесу.

По-третє, система може бути використана для організації дистанційного навчання та тестування, що особливо актуально в умовах сучасних технологічних трендів і вимог соціальної дистанції.

Крім того, система дозволяє формувати статистичні звіти та аналізувати результати навчання, що допомагає у плануванні освітнього процесу і підвищенні його якості.

Завдяки своїй гнучкості, масштабованості та простоті у використанні, розроблена система може бути легко адаптована під різні вимоги і розширена новими функціями, що забезпечує її довгострокове застосування і впровадження.

ВИСНОВКИ

У ході виконання дипломної роботи було розроблено комп'ютерну систему тестування з правил дорожнього руху, метою якої було створення зручного, ефективного та доступного інструменту для підготовки майбутніх водіїв. Результати роботи підтвердили доцільність застосування сучасних інформаційних технологій у сфері освіти та підвищення безпеки на дорогах.

Перш за все, було проведено детальний аналіз предметної області — організації підготовки водіїв і систем тестування ПДР, а також огляд існуючих програмних засобів. Цей аналіз дозволив виявити основні недоліки сучасних рішень, такі як недостатня адаптивність, обмеженість функціоналу, низький рівень зручності користування, а також відсутність детального аналізу результатів тестування. На основі цих висновків сформовано вимоги до розроблюваної системи, що забезпечують її конкурентоспроможність та відповідність сучасним стандартам.

Далі було здійснено проектування системи, обрано технології розробки — мова програмування PHP, система керування базами даних MySQL із використанням інструменту phpMyAdmin. Розроблена архітектура системи передбачає модульну структуру, що забезпечує гнучкість, масштабованість та зручність подальшого розвитку системи. Особливу увагу приділено проектуванню бази даних, яка містить тестові завдання, відповіді, результати тестування та інформацію про користувачів. Продумано логіку генерації тестів з можливістю вибору категорій та рівня складності, а також механізм автоматичного оцінювання.

В процесі розробки було створено інтуїтивно зрозумілий інтерфейс користувача, що дозволяє легко реєструватись, вибрати тестові категорії, проходити тестування і отримувати детальні звіти з результатами. Адміністративна частина системи забезпечує ефективне керування базою даних і користувачами, що спрощує організацію навчального процесу.

Особливу увагу було приділено алгоритму проведення тестування — від аутентифікації користувача до формування підсумкового звіту. Завдяки цьому

забезпечується об'єктивність та прозорість оцінювання, що є надзвичайно важливим для підготовки водіїв і дотримання нормативних вимог.

Після завершення розробки система пройшла комплексне тестування. Було проведено модульні, інтеграційні та навантажувальні тести, які підтвердили стабільність, надійність та швидкодію системи навіть при одночасній роботі багатьох користувачів. За результатами тестування були усунуті виявлені недоліки, що позитивно вплинуло на загальну якість продукту.

Для оцінки практичної ефективності системи було організовано експеримент у навчальному закладі. Експериментальна група, що використовувала розроблену систему, показала значне покращення знань з правил дорожнього руху у порівнянні з контрольною групою, яка працювала за традиційними методами. Це свідчить про високу результативність автоматизованого тестування та можливість його широкого застосування у навчальному процесі.

Порівняння з існуючими рішеннями також виявило ряд переваг розробленої системи, серед яких — зручний інтерфейс, гнучкість у налаштуванні тестів, автоматичне оцінювання з детальним аналізом результатів, а також швидкість роботи і можливість масштабування. Водночас визначено напрямки подальшого розвитку, зокрема інтеграція з мобільними платформами, розширена аналітика та використання сучасних технологій штучного інтелекту.

Практичне значення розробленої системи полягає у підвищенні якості підготовки водіїв, зниженні часу на проходження тестів і оптимізації роботи автошкіл. Вона може бути використана як основний або допоміжний інструмент для навчання, що особливо актуально в умовах дистанційного навчання та зростаючих вимог до безпеки дорожнього руху.

Таким чином, виконана робота має комплексний характер, поєднує теоретичний аналіз, практичну розробку і експериментальну перевірку. Результати дослідження і розробки підтверджують ефективність створеної системи, яка може бути впроваджена у навчальні заклади та інші організації, що здійснюють підготовку водіїв.

Подальші дослідження і вдосконалення системи можуть бути спрямовані на інтеграцію з мобільними додатками, розробку адаптивних тестів із використанням методів машинного навчання, а також розширення функціоналу для підтримки різних форматів навчання і контролю знань.

В цілому, розроблена комп'ютерна система тестування з правил дорожнього руху є сучасним, ефективним і перспективним інструментом, який сприятиме підвищенню рівня знань майбутніх водіїв та безпеці на дорогах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Правила дорожнього руху України. Офіційний документ. — Київ: Міністерство внутрішніх справ України, 2023.
2. Боднарчук, В.І. Організація підготовки водіїв: навчальний посібник. — Львів: Видавництво ЛНУ, 2021.
3. Коваленко, О.С. Основи безпеки дорожнього руху. — Харків: ХНУ, 2020.
4. Іванов, П.П. Комп'ютерні системи в освіті: методичні рекомендації. — Київ: Наукова думка, 2019.
5. Сидоренко, М.В. Розробка веб-додатків на PHP та MySQL. — Київ: Крок, 2022.
6. Петрова, Л.І. Автоматизовані системи тестування: теорія і практика. — Одеса: ОНУ, 2021.
7. Зайченко, Ю.А. Бази даних: проектування і оптимізація. — Дніпро: ДНУ, 2020.
8. Степаненко, І.М. Інтерфейс користувача: дизайн і юзабіліті. — Київ: Видавництво КПІ, 2022.
9. Шевченко, В.Г. Алгоритми і структури даних у PHP. — Харків: ХНТУСГ, 2023.
10. Дорошенко, А.В. Методи оцінювання знань у навчальних системах. — Львів: ЛНУ, 2021.
11. Головка, С.І. Тестування програмного забезпечення: методологія і практика. — Київ: НАУ, 2022.
12. Мартиненко, О.В. Організація експериментальних досліджень у педагогіці. — Одеса: ОНУ, 2020.
13. Харченко, І.П. Системи дистанційного навчання: перспективи і проблеми. — Київ: НТУ, 2023.
14. Кравченко, Ю.С. Аналітика даних у навчальних системах. — Дніпро: ДНУ, 2021.

15. Веб-документація PHP. Офіційний сайт PHP. URL: <https://www.php.net/manual/ru/> (дата звернення: 15.05.2025).
16. Документація MySQL. Офіційний сайт MySQL. URL: <https://dev.mysql.com/doc/> (дата звернення: 15.05.2025).
17. phpMyAdmin Documentation. URL: <https://www.phpmyadmin.net/docs/> (дата звернення: 15.05.2025).
18. OpenAI ChatGPT — Помічник у написанні дипломної роботи. URL: <https://chat.openai.com/> (дата звернення: 22.05.2025).

ДОДАТКИ

А. Приклади тестових завдань

Усі тести

Створити
екзамен

Створити
завдання

Теоретичний іспит з правил дорожнього руху (ПДР) України – Базовий рівень

Видалити

Почати

Що означає дорожній знак "Дати дорогу"?

Варіанти відповідей:

- 1) а) Рух без зупинки заборонено (Неправильно)
- 2) б) Перевага в русі (Неправильно)
- 3) в) Необхідно зупинитися (Неправильно)
- 4) г) Дати дорогу іншим учасникам руху (Правильно)

Видалити

Яка максимальна дозволена швидкість у населених пунктах для легкових авто без причепа?

Варіанти відповідей:

- 1) а) 50 км/год (Правильно)
- 2) б) 60 км/год (Неправильно)
- 3) в) 70 км/год (Неправильно)
- 4) г) 80 км/год (Неправильно)

Видалити

У якому випадку водій повинен увімкнути аварійну сигналізацію?

Варіанти відповідей:

- 1) а) Під час обгону (Неправильно)
- 2) б) При зміні смуги (Неправильно)

Видалити

Б. Фрагменти коду

Б.1. Ініціалізація бази даних

```
<?php  
  
define('DB_HOST', 'localhost');  
define('DB_USER', 'root');  
define('DB_PASS', 'root');  
define('DB_NAME', 'rules_db');
```

```
<?php  
  
require('core/config/db_config.php');  
  
class MySqlProvider{  
    public $conn;  
    public function __construct(){  
        $this->conn = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);  
    }  
}
```

Б.2. Реєстрація користувача

```
<div class="main-box">  
    <div class="container" style="padding-left: 5%; padding-right: 5%">  
        <h3>Реєстрація</h3>  
        <hr>  
        <form id="form1" action="reg_res" method="post">  
            <div class="form-group panel">  
                <label for="login">Логін:</label>  
                <input type="text" id="login" name="login" class="form-control" required>  
                <div id="login-err" class="err"></div>  
            </div>  
            <div class="form-group panel">  
                <label for="pass1">Пароль:</label>  
                <input type="password" id="pass1" name="pass1" class="form-control" required>
```

```

    <div id="pass1-err" class="err"></div>
  </div>
  <div class="form-group panel">
    <label for="pass2">Повтор:</label>
    <input type="password" id="pass2" name="pass2" class="form-control" required>
    <div id="pass2-err" class="err"></div>
  </div>
  <div class="form-group panel">
    <label for="email">E-Mail:</label>
    <input type="email" id="email" name="email" class="form-control" required>
    <div id="email-err" class="err"></div>
  </div>
  <br>
  <div class="form-group buttons">
    <input type="submit" value="Відправити" class="btn btn-success">
    <input type="reset" value="Очистити" class="btn btn-danger">
  </div>
</form>
</div>
</div>

```

Б.3. Авторизація користувача

```

<div class="main-box">
  <div class="container" style="padding-left: 5%; padding-right: 5%">
    <h3>Авторизація</h3>
    <hr>
    <form action="entry_res" method="post">
      <div class="form-group panel">
        <label for="login">Логін:</label>
        <input type="text" id="login" name="login" class="form-control" required>
        <div id="login-err" class="err"></div>
      </div>
    </form>
  </div>

```

```

<div class="form-group panel">
    <label for="pass1">Пароль:</label>
    <input type="password" id="pass1" name="pass1" class="form-control" required>
    <div id="pass1-err" class="err"></div>
</div>
<br>
<div class="form-group buttons">
    <input type="submit" value="Відправити" class="btn btn-success">
    <input type="reset" value="Очистити" class="btn btn-danger">
</div>
</form>
</div>
</div>

```

Б.4. Збереження результатів тесту в базу даних

```

<?php

require_once('core/providers/mysql_provider.php');

class Answers extends MySQLProvider{

    public function add_answers($id_of_test, $name_of_user, $id_answer){
        $sql_query = 'insert into test_answers (id_of_test, name_of_user, id_answer)';
        $sql_query .= " values (?, ?, ?)";
        $stmt = $this->conn->prepare($sql_query);
        $stmt->bind_param('isi', $id_of_test, $name_of_user, $id_answer);
        if(!$stmt->execute()){
            throw new Exception('Помилка виконання SQL-запиту');
        }
    }

    public function get_answers(){
        $answers = [];
        $sql = 'select * from test_answers';
    }
}

```

```

$result = $this->conn->query($sql);
if (!$result) {
    throw new Exception('Помилка виконання SQL-запиту');
} elseif ($result->num_rows > 0){
    while($row = $result->fetch_assoc()){
        $answers[] = $row;
    }
}
return $answers;
}

public function get_article($aid){
    $sql = 'select * from test_answers where id=?';
    $stmt = $this->conn->prepare($sql);
    $stmt->bind_param('i', $aid);
    $stmt->execute();
    $result = $stmt->get_result();
    $article = $result->fetch_assoc();
    return $article;
}

public function del_article($aid){
    $sql = 'delete from test_answers where id=? ';
    $stmt = $this->conn->prepare($sql);
    $stmt->bind_param('i', $aid);
    if(!$stmt->execute()){
        throw new Exception('Помилка виконання SQL-запиту');
    }
}
}
}

```

Б.5. Виведення результатів тесту користувача

```
<?php
    $model = new Users();
    $users = $model->get_users();
?>

<?php
    require('core/models/tasks.php');
    $model = new Tasks();
    $tasks = $model->get_tasks();
?>

<?php
    require('core/models/test_answers.php');
    $model = new Answers();
    $answers = $model->get_answers();
?>

<div class="main-box">
    <div class="container">
        <?php
            $answered_count = 0;
            $total_questions = 0;
            $correct_answers_count = 0;

            foreach ($tasks as $row_tasks) {
                if ($row_tasks['id_exam'] == $this->parameter) {
                    $total_questions++;
                    foreach ($answers as $row_answers) {
                        if ($row_answers['name_of_user'] == $this->user && $row_answers['id_of_test'] ==
$tasks['id']) {
                            $answered_count++;
                            if ($row_answers['id_answer'] == $row_tasks['choose_correct']) {
                                $correct_answers_count++;
                            }
                        }
                    }
                    break;
                }
            }
        ?>
    </div>
</div>
```

```

    }
  }
}
}

$percentage_correct = 0;
if ($total_questions > 0) {
    $percentage_correct = ($correct_answers_count / $total_questions) * 100;
}
?>
<?php if ($this->user == "Гість"){ ?>
    <div style="background-color: white; border-radius: 5px; margin: auto; text-align: center">
        <h3 style="margin: auto; padding: 2%">Для проходження тестів треба бути
зарєєстрованим!</h3>
    </div>
<?php } else {?>
    <?php foreach ($tasks as $row_tasks): ?>
        <?php if ($row_tasks['id_exam'] == $this->parameter): ?>
            <div style="width: 100%; padding: 10px; background-color: white; border: 2px solid
grey;
                margin-top: 6px; border-radius: 5px; display: flex">
                <div style="width: 100%; margin: 2% 5%; border-radius: 5px">
                    <h5><?= $row_tasks['text'] ?></h5>
                    <hr>
                    <?php
                        $user_answered = false;
                        foreach ($answers as $row_answers):
                            if ($row_answers['name_of_user'] == $this->user &&
$row_answers['id_of_test'] == $row_tasks['id']):
                                $user_answered = true;
                                $is_correct = $row_answers['id_answer'] == $row_tasks['choose_correct'];
                            ?>

```

```

        <p style="color: <?= $is_correct ? 'green' : 'red' ?>;
            border: 1px solid <?= $is_correct ? 'green' : 'red' ?>;
            padding-left: 1%">
            <?= $row_answers['id_answer'] ?> <?= $row_tasks['choose' .
$row_answers['id_answer']] ?>
        </p>
        <h5 style="color: <?= $is_correct ? 'green' : 'red' ?>; font-weight: 500">
            <?= $is_correct ? 'Правильно' : 'Неправильно' ?>
        </h5>
    <?php break;
    endif;
    endforeach;
    if (!$user_answered && $this->user != "admin228"): ?>
    <form action="handler_add_answer" method="post" enctype="multipart/form-
data"
        style="display: flex; justify-content: space-between; width: 100%">
        <input type="hidden" name="id_of_test" value="<?= $row_tasks['id'] ?>">
        <input type="hidden" name="id_of_exam" value="<?= $this->parameter
?>">
        <div class="form-group panel">
            <select id="id_answer" name="id_answer" class="form-control" required>
                <option value="1"><?= $row_tasks['choose1'] ?></option>
                <option value="2"><?= $row_tasks['choose2'] ?></option>
                <option value="3"><?= $row_tasks['choose3'] ?></option>
                <option value="4"><?= $row_tasks['choose4'] ?></option>
            </select>
        </div>
        <div class="form-group buttons">
            <input type="submit" id="submit" name="submit" class="btn btn-
warning" value="Відповісти">
        </div>
    </form>
    <?php elseif ($this->user == "admin228"): ?>

```

```

<div>
  <div style="display: flex">
    <p style="width: 80%"><strong>Варіанти відповідей:</strong></p>
    <a href="handler_del_tasks@<?=$row_tasks['id']?" class="btn btn-
danger"
      style="width: 20%; justify-content: center; align-items: center;
      flex-direction: column; display: flex; margin: 0.3% 0">
      Видалити
    </a>
  </div>
  <p>1) <?=$row_tasks['choose1'] ?> <span style="color: <?=$row_tasks['choose_correct'] == 1 ? 'green' : 'red' ?>"><?=$row_tasks['choose_correct'] == 1 ? 'Правильно' : 'Неправильно' ?></span></p>
  <p>2) <?=$row_tasks['choose2'] ?> <span style="color: <?=$row_tasks['choose_correct'] == 2 ? 'green' : 'red' ?>"><?=$row_tasks['choose_correct'] == 2 ? 'Правильно' : 'Неправильно' ?></span></p>
  <p>3) <?=$row_tasks['choose3'] ?> <span style="color: <?=$row_tasks['choose_correct'] == 3 ? 'green' : 'red' ?>"><?=$row_tasks['choose_correct'] == 3 ? 'Правильно' : 'Неправильно' ?></span></p>
  <p>4) <?=$row_tasks['choose4'] ?> <span style="color: <?=$row_tasks['choose_correct'] == 4 ? 'green' : 'red' ?>"><?=$row_tasks['choose_correct'] == 4 ? 'Правильно' : 'Неправильно' ?></span></p>
  </div>
  <?php endif; ?>
</div>
</div>
<?php endif; ?>
<?php endforeach; ?>
<?php }?>
<?php
  $color = "black";
  if ($percentage_correct > 40 && $percentage_correct <= 60) {
    $color = "red";
  } elseif ($percentage_correct > 60 && $percentage_correct <= 85) {

```

```

        $color = "orange";
    } elseif ($percentage_correct > 85) {
        $color = "limegreen";
    }
?>

<?php if ($answered_count == $total_questions && $total_questions > 0 ||
$this->user == "admin228" || $this->user == "Гість"): ?>
<div style="display: flex">
    <a href="exams" class="btn" style="width: 20%; background-color: black; color: white;
margin: 1%">
        Назад
    </a>
    <?php if ($this->user != "Гість" && $this->user != "admin228"): ?>
        <label style="font-size: 18px; font-weight: bold; margin: auto; color: white">
            Відсоток правильних відповідей:<span style="color: <?= $color ?>">
                <?= number_format($percentage_correct, 2) ?></span>%
            </label>
        <?php endif; ?>
    </div>
<?php endif; ?>
</div>
</div>

```

Б.6. Додавання тестового завдання

```

<?php
    require('core/models/exams.php');
    $model = new Exams();
    $exams = $model->get_exams();
?>
<?php if ($this->user !== 'admin228'): ?>
    <script>
        window.location = 'page403';
    </script>

```

```
</script>
<?php else: ?>
  <div class="main-box">
    <div class="container" style="padding-left: 5%; padding-right: 5%">
      <form id="form2" action="handler_add_tasks" method="post" enctype="multipart/form-
data">
        <div class="form-group panel">
          <label for="text">Текст:</label>
          <textarea id="text" name="text" class="form-control" rows="3" required></textarea>
        </div>
        <div class="form-group panel">
          <label for="choose1">Варіант №1:</label>
          <input type="text" id="choose1" name="choose1" class="form-control" required>
        </div>
        <div class="form-group panel">
          <label for="choose2">Варіант №2:</label>
          <input type="text" id="choose2" name="choose2" class="form-control" required>
        </div>
        <div class="form-group panel">
          <label for="choose3">Варіант №3:</label>
          <input type="text" id="choose3" name="choose3" class="form-control" required>
        </div>
        <div class="form-group panel">
          <label for="choose4">Варіант №4:</label>
          <input type="text" id="choose4" name="choose4" class="form-control" required>
        </div>
        <div class="form-group panel">
          <label for="choose_correct">Правильна відповідь №:</label>
          <select id="choose_correct" name="choose_correct" class="form-control" required>
            <option value=1>1</option>
            <option value=2>2</option>
            <option value=3>3</option>
```

```
        <option value=4>4</option>
    </select>
</div>
<div class="form-group panel">
    <label for="id_exam">Додати в ...</label>
    <select id="id_exam" name="id_exam" class="form-control" required>
        <?php foreach ($exams as $option_exams): ?>
            <option
value="<?=$option_exams['id']?>"><?=$option_exams['title']?></option>
        <?php endforeach; ?>
    </select>
</div>
<br>
<div class="form-group buttons">
    <input type="submit" id="submit" name="submit" class="btn btn-success"
value="Створити">
    <a href="exams" class="btn btn-danger">Назад</a>
</div>
</form>
</div>
</div>
<?php endif; ?>
```

В. Інструкції користувача

В.1. Реєстрація нового користувача

- Відкрийте головну сторінку сайту.
- Натисніть кнопку "Реєстрація".
- Заповніть всі необхідні поля: логін, пароль, повтор пароля, e-mail.
- Натисніть кнопку "Відправити".
- Після успішної реєстрації ви маєте натиснути «Вхід».

В.2. Вхід у систему

- Перейдіть на сторінку входу.
- Введіть логін та пароль.
- Натисніть кнопку "Увійти".
- У разі правильного введення — ви потрапите до особистого кабінету.

В.3. Проходження тестування ПДР

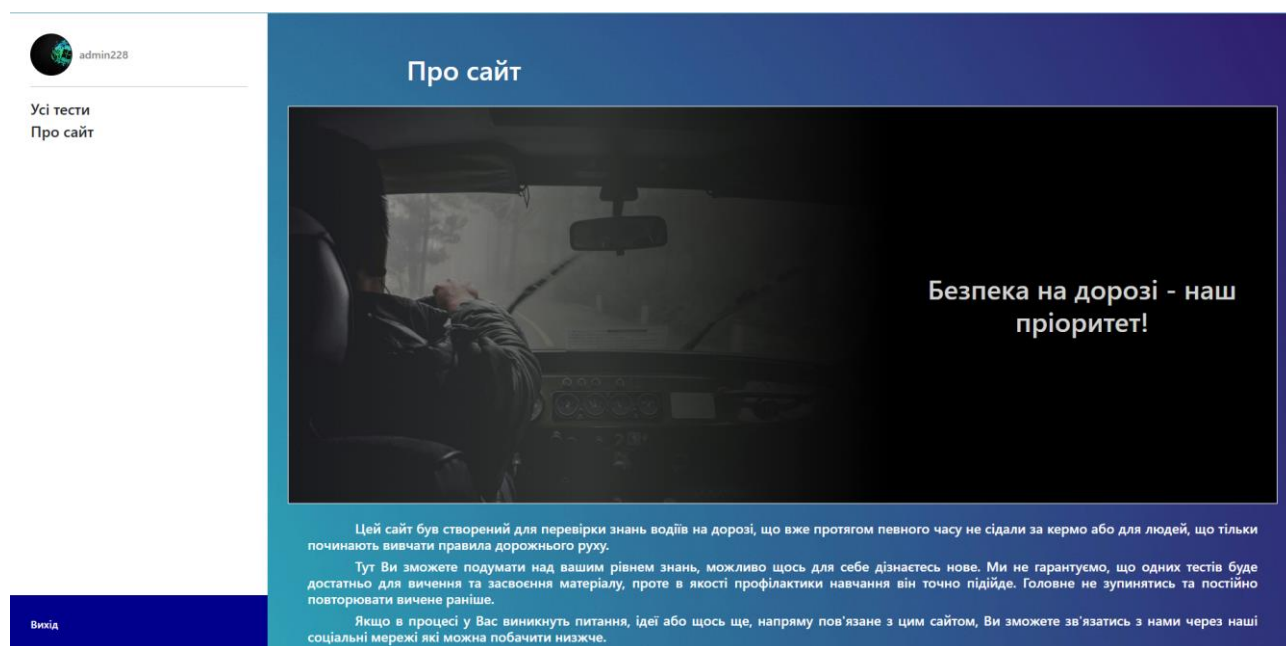
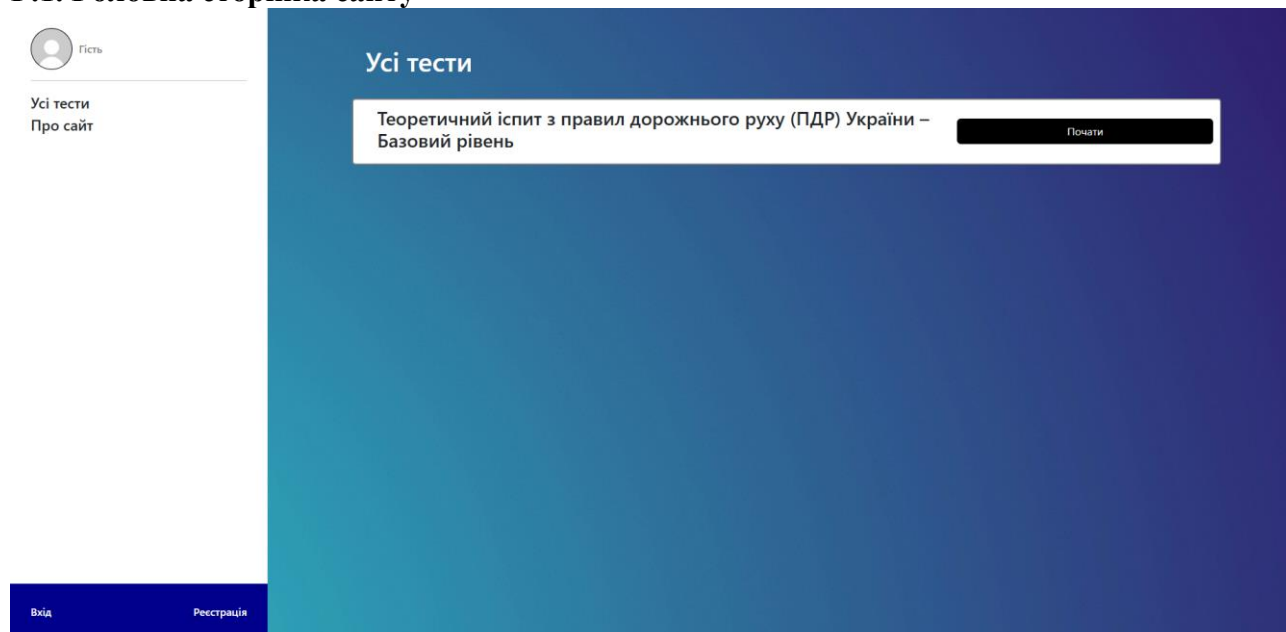
- Увійдіть до системи.
- В особистому кабінеті оберіть розділ "Тестування".
- Натисніть "Почати тест".
- Оберіть одну правильну відповідь для кожного питання.
- Результати тесту буде показано внизу на екрані.

В.4. Вихід із системи

- У нижній панелі інтерфейсу натисніть "Вийти".
- Ви будете перенаправлені на головну сторінку сайту.

Г. Скріншоти інтерфейсу системи

Г.1. Головна сторінка сайту



Г.2. Сторінка реєстрації

Реєстрація

Логін:

Пароль:

Повтор:

Е-Mail:

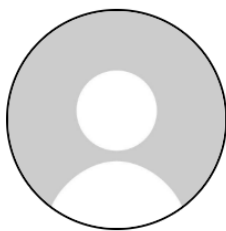
Г.3. Сторінка входу

Авторизація

Логін:

Пароль:

Г.4. Особистий кабінет користувача



Ivanov259

Замінити фото

Ivanov259@gmail.com

2025-06-03 03:26:18



Ivanov259

Усі тести
Про сайт

Усі тести

Теоретичний іспит з правил дорожнього руху (ПДР) України –
Базовий рівень

Почати

Вихід

Усі тести

Теоретичний іспит з правил дорожнього руху (ПДР) України –
Базовий рівень[Подивитись результати](#)

Г.5. Сторінка проходження тестування

Яким чином слід розміщувати вантаж на ТЗ?

1) а) Виступаючи за габарити з будь-якого боку

Неправильно

Чи має велосипедист право рухатися по проїзній частині?

1) а) Так

Правильно

Коли дозволено перетинати подвійну суцільну лінію?

1) а) Ніколи

Правильно

Що повинен зробити водій при наближенні авто з увімкненими проблісковими маячками та сиреною?

1) а) Продовжити рух

Неправильно

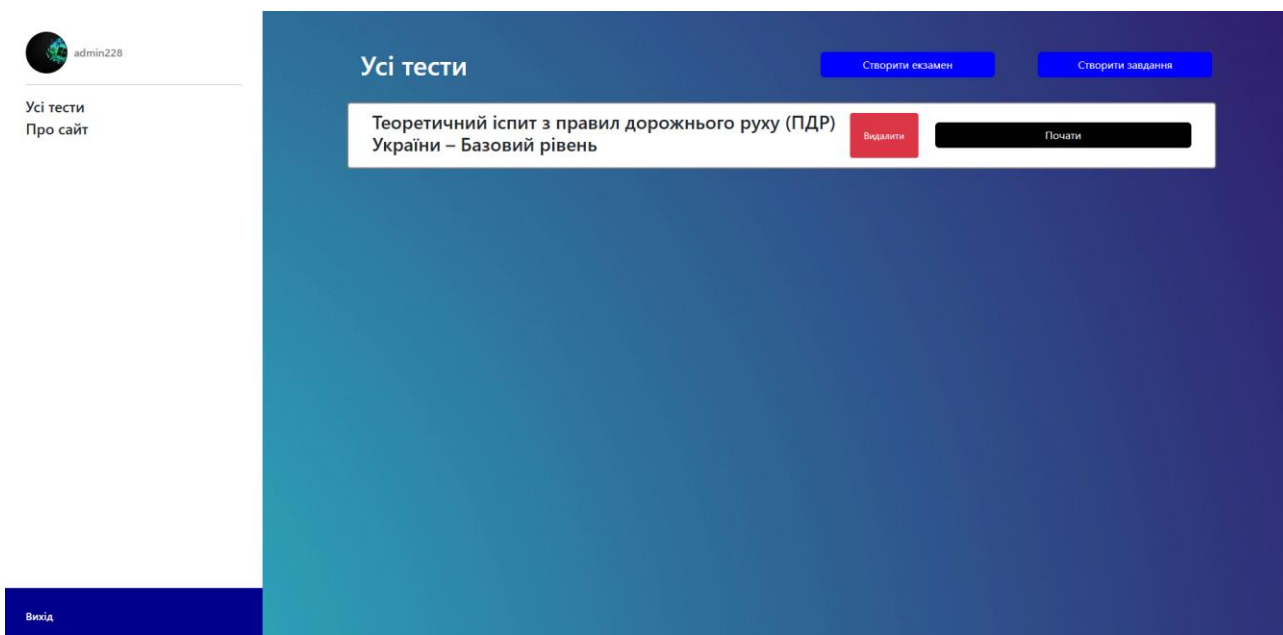
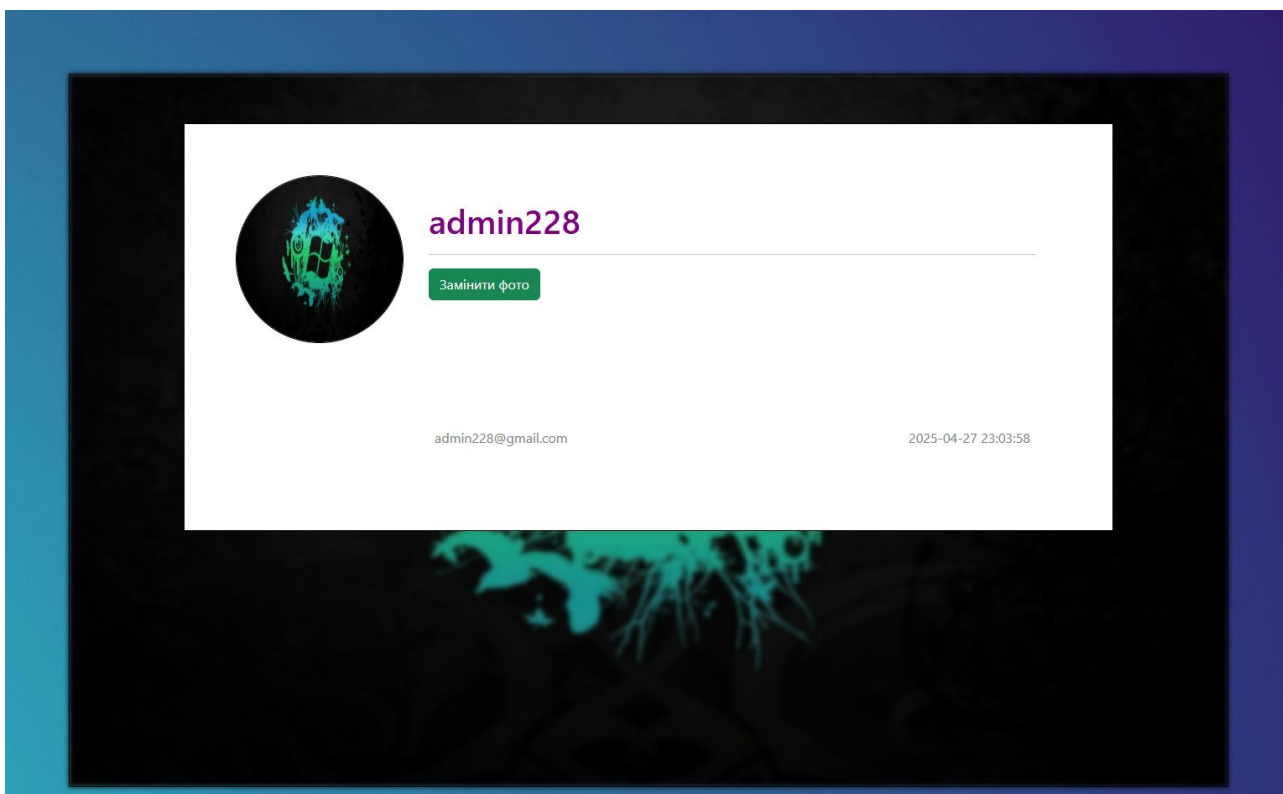
Назад

Відсоток правильних відповідей: 25.00%

Г.6. Вікно з результатом тесту

Відсоток правильних відповідей: 25.00%

Г.7. Сторінка адміністратора



Що означає дорожній знак "Дати дорогу"?

Варіанти відповідей:

Видалити

- 1) а) Рух без зупинки заборонено (Неправильно)
- 2) б) Перевага в русі (Неправильно)
- 3) в) Необхідно зупинитися (Неправильно)
- 4) г) Дати дорогу іншим учасникам руху (Правильно)

Яка максимальна дозволена швидкість у населених пунктах для легкових авто без причепа?

Варіанти відповідей:

Видалити

- 1) а) 50 км/год (Правильно)
- 2) б) 60 км/год (Неправильно)
- 3) в) 70 км/год (Неправильно)
- 4) г) 80 км/год (Неправильно)

У якому випадку водій повинен увімкнути аварійну сигналізацію?

Варіанти відповідей:

Видалити

- 1) а) Під час обгону (Неправильно)
- 2) б) При зміні смуги (Неправильно)

Назва:

Відправити

Очистити

Текст:

Варіант №1:

Варіант №2:

Варіант №3:

Варіант №4:

Правильна відповідь №:

Додати в ...

[Створити](#) [Назад](#)

Г.8. Повідомлення про успішну дію

ВИ ЗАРЕЄСТРОВАНІ

Для проходження тестів треба бути зареєстрованим!

[Назад](#)