

Державний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Онлайн-платформа Всеукраїнського учнівського конкурсу з
правознавства»**

Студента 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

Александрова Андрія
Юрійовича

підпис студента

Науковий керівник
кандидат педагогічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Котенко Наталія
Олексіївна

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення
та кібербезпеки

Рзаєва Світлана
Леонідівна

підпис гаранта

КИЇВ – 2022

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність «Інженерія програмного забезпечення»

Спеціалізація/ освітня програма «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

"09" листопада 2021 р.

Завдання

на випускню кваліфікаційну роботу студентіві

Александрову Андрію Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Онлайн-платформа Всеукраїнського
учнівського конкурсу з правознавства»

Затверджена наказом ректора від "29" листопада 2021 р. № 3928

2. Строк здачі студентом закінченої роботи 06.06.2022

3. Цільова установка та вихідні дані до роботи

Мета дослідження піддати науковому розгляду використання новітніх технологій у системах навчання, позанавчальної діяльності, та розробити програмний застосунок на основі отриманої інформації.

Об'єкт дослідження проектування та програмна реалізація front-end та back-end веб-сервісу учнівського конкурсу

Предмет дослідження розробка програмного забезпечення для Всеукраїнського учнівського конкурсу "Правознай", яке дозволяє проводити конкурси та брати участь у них.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи(перелік питань за кожним розділом)

ЗМІСТ

ВСТУП

РОЗДІЛ 1 ПОСТАНОВКА ТА ФОРМАЛІЗАЦІЯ ЗАВДАННЯ, СУТЬ І ПРАКТИЧНЕ ЗНАЧЕННЯ ПРЕДМЕТУ

Постановка та формалізація завдання

Формалізація вимог до розробки програмного забезпечення

Практичне значення розроблюваного програмного продукту та відповідної предметної області

Аналіз та вибір інструментів та засобів для розробки

Висновки до розділу 1

РОЗДІЛ 2 ПРОЄКТУВАННЯ КОМПОНЕНТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Проектування архітектури програмного забезпечення. Створення UML-діаграм

Структурні діаграми

Діаграми поведінки

Діаграми взаємодії

Проектування бази даних

Концептуальна модель

Логічна модель

Фізична модель

Проектування графічного інтерфейсу користувача

Висновки до розділу 2

РОЗДІЛ 3 ОПИС ТЕХНІЧНОГО РІШЕННЯ

Розробка програмного забезпечення

Розробка підсистеми авторизації

Розробка підсистеми тестування

Розробка підсистеми блогу

Безпекові рішення

Тестування та оптимізація

Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ



6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	
		За планом	Фактично
1.	Вибір теми випускної кваліфікаційної роботи	20.09.2021	20.09.2021
2.	Вступ та перелік літературних джерел	13.12.2021	13.12.2021
3.	Розділ 1. «ПОСТАНОВКА ТА ФОРМАЛІЗАЦІЯ ЗАВДАННЯ, СУТЬ І ПРАКТИЧНЕ ЗНАЧЕННЯ ПРЕДМЕТУ»	18.02.2022	18.02.2022
4.	Розділ 2. «ПРОЄКТУВАННЯ КОМПОНЕНТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	04.03.2022	04.03.2022
5.	Розділ 3. «ОПИС ТЕХНІЧНОГО РІШЕННЯ»	11.04.2022	11.04.2022
6.	Висновки	25.04.2022	25.04.2022
7.	Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)	25.05.2022	25.05.2022
8.	Підготовка автореферату та презентації доповіді	26.05.2022	26.05.2022
9.	Попередній захист випускної кваліфікаційної роботи	30.05.2022 - 03.06.2022	01.06.2022
10.	Зовнішнє рецензування випускної кваліфікаційної роботи	04.06.2022	04.06.2022
11.	Здача прошитою випускної кваліфікаційної роботи	06.06.2022	06.06.2022
12.	Публічний захист випускної кваліфікаційної роботи		

7. Дата видачі завдання «09» листопада 2021 р.

8. Науковий керівник випускної кваліфікаційної роботи

Котенко Н. О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Рзаєва С. Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Александров А. Ю.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

Робота присвячена дослідженню використання сучасних інформаційних технологій у сфері освіти, проектування та реалізації основних компонентів програмного забезпечення, його впровадження для оптимізації позанавчальної діяльності учнів у вигляді конкурсів із тестовими запитаннями.

Розробка здійснена з використанням мови програмування Python, із застосуванням фреймворку Django.

Випускна кваліфікаційна робота на тему «Онлайн-платформа Всеукраїнського учнівського конкурсу з правознавства» загальним обсягом 48 сторінок містить:

- 21 рисунок
- 2 таблиці

Ключові слова: тест, Python, конкурс, модель

ANNOTATION

The work is devoted to the research of the use of modern information technologies in the field of education, design and development of the main components of software, its implementation to optimize extracurricular activities of students in the form of contests with test questions.

The development was carried out using Python programming language and Django framework.

The final qualifying work on the topic “Online platform of the All-Ukrainian student contest in law” with a total volume of 48 pages contains:

- 21 figures
- 2 tables

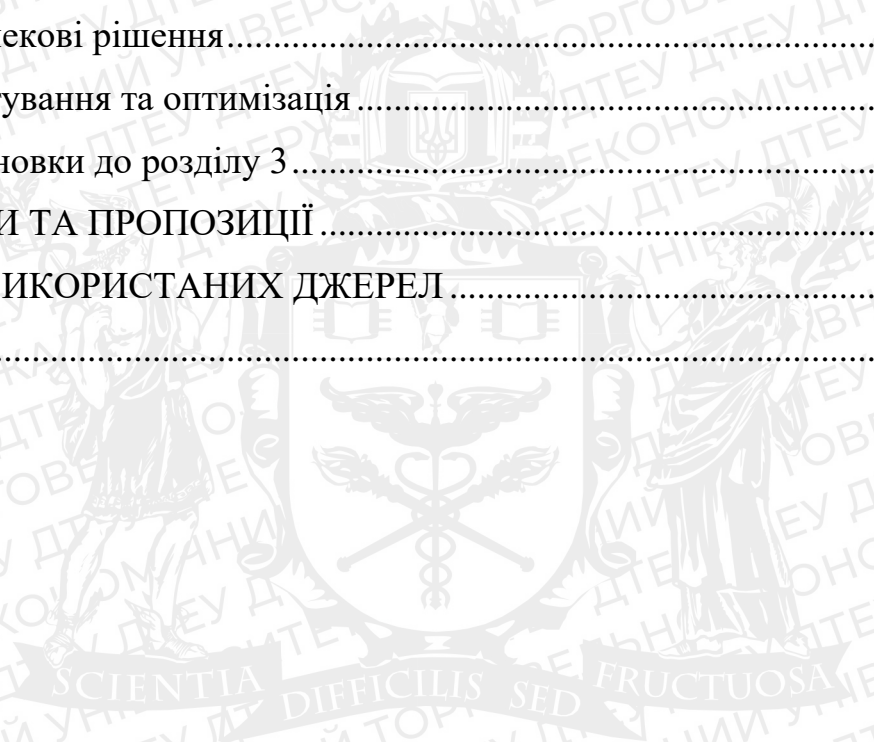
Keywords: test, Python, contest, model

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1	6
ПОСТАНОВКА ТА ФОРМАЛІЗАЦІЯ ЗАВДАННЯ, СУТЬ І ПРАКТИЧНЕ ЗНАЧЕННЯ ПРЕДМЕТУ	6
1.1 Постановка та формалізація завдання.....	6
1.2 Формалізація вимог до розробки програмного забезпечення.....	7
1.3 Практичне значення розроблюваного програмного продукту та відповідної предметної області	13
1.4 Аналіз та вибір інструментів та засобів для розробки.....	14
1.5. Технічне завдання.....	16
1.6 Висновки до розділу 1	16
РОЗДІЛ 2	18
ПРОЄКТУВАННЯ КОМПОНЕНТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	18
2.1 Проєктування архітектури програмного забезпечення. Створення UML- діаграм	18
2.1.1 Структурні діаграми.....	19
2.1.2 Діаграми поведінки	21
2.1.3 Діаграми взаємодії.....	23
2.2 Проєктування бази даних.....	24
2.2.1 Концептуальна модель.....	25
2.2.2 Логічна модель	26
2.2.3 Фізична модель	27
2.3 Проєктування графічного інтерфейсу користувача	28

					<i>ДТЕУ 121 06-01.БР</i>			
					<i>Онлайн-платформа Всеукраїнського учнівського конкурсу з правознавства</i>	<i>Зміст</i>	<i>2</i>	<i>49</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		25.04.22				
Керівник		Котенко Н. О.		25.04.22				
Гарант		Рзаєва С. Л.		25.04.22				
Розроб		Александров А. Ю.		25.04.22				
					<i>Зміст</i>	Факультет інформаційних технологій, 4 курс, 6 група		

2.4 Висновки до розділу 2	31
РОЗДІЛ 3	33
ОПИС ТЕХНІЧНОГО РІШЕННЯ	33
3.1 Розробка програмного забезпечення	33
3.1.1 Розробка підсистеми авторизації.....	37
3.1.2 Розробка підсистеми тестування	38
3.1.3 Розробка підсистеми блогу	40
3.2 Безпеківі рішення.....	40
3.3 Тестування та оптимізація.....	42
3.4 Висновки до розділу 3.....	44
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48
ДОДАТКИ.....	50



					<i>ДТЕУ 121 06-01.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		3

ВСТУП

Сьогодні, у 2022 році, можливості відвідування різних місць та навіть переміщення між ними є обмеженими і тому велика частина процесів почала виконуватися дистанційно – робота, розваги та будь-яка інша сфера людського життя. Це торкнулося і навчання – від шкіл та університетів до позанавчальної діяльності і курсів.

Розроблюваний програмний застосунок спрямований на забезпечення можливості проведення конкурсів онлайн, з мінімізацією будь-якої персональної взаємодії. Такий режим надає ряд переваг, а саме:

- зручність для учасників – немає необхідності здійснювати переміщення до різних місць, на різних етапах конкурсу; можна вирішувати завдання та переглядати результати у будь-який час у визначений проміжок;
- зручність для організаторів - автоматичний розрахунок набраних балів та висвітлення результатів по місцях за балами, початок та закінчення конкурсу;
- безпека - відсутність особистого контакту учасників та організаторів;
- паралельний розвиток учасниками навичок роботи з комп'ютерами та інформаційними технологіями загалом

Основним напрямком конкурсів, на які спрямовується програмне забезпечення, є правознавство. Правознавство як наука надає розуміння провідних засад, на яких побудований сучасний соціальний устрій, – законів. Наявність навичок з юриспруденції дозволяє більш раціонально здійснювати будь-яку діяльність пов'язану з соціумом, а уміння правильно інтерпретувати закони надає можливість уникати ситуацій, коли інші люди роблять спроби маніпулювати рамками цих законів з ціллю отримати вигоду.

					<i>ДТЕУ 121 06-01.БР</i>			
					<i>Онлайн-платформа Всеукраїнського учнівського конкурсу з правознавства</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		13.12.21		В	4	49
Керівник		Котенко Н. О.		13.12.21				
Гарант		Рзасва С. Л.		13.12.21				
Розроб		Александров А. Ю.		13.12.21	<i>Вступ</i>	Факультет інформаційних технологій, 4 курс, 6 група		

Конкурси є видом позанавчальної діяльності, що може використовуватися для стимулювання у потенційних учасників покращення знань, пов'язаних з тематикою конкурсу, який проводиться, у даному випадку – правознавства. Вони дозволяють вивчати новий матеріал, закріплювати його та перевіряти вже існуючі знання.

Для забезпечення найкращого результату потрібно створити найбільш зручні умови для учасників - це стосується як доведення інформації про конкурс, так і власне самого процесу тестування і отримання результатів. Найефективнішим способом досягнення таких умов є використання сучасних інформаційних технологій, включаючи Інтернет.

Актуальність: у 2022 році будь-яка діяльність, яка проводиться дистанційно, без особистої взаємодії є надзвичайно актуальною та відповідає усім вимогам ринку та потенційних користувачів.

Об'єкт: проектування та програмна реалізація front-end та back-end веб-сервісу учнівського конкурсу

Предмет: розробка програмного забезпечення для Всеукраїнського учнівського конкурсу “Правознай”, яке дозволяє проводити конкурси та брати участь у них.

Мета: піддати науковому розгляду використання новітніх технологій у системах навчання, позанавчальної діяльності, та розробити програмний застосунок на основі отриманої інформації.

Завдання:

- дослідження та аналіз предметної області, наявних конкурентних програмних засобів;
- проектування усіх компонентів майбутнього програмного забезпечення, включаючи проектування архітектури, бази даних, UI/UX інтерфейсу на основі проведених досліджень;
- власне розробка компонентів системи на основі створених проєктів.

									Аркуш
									5
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-01.БР				

РОЗДІЛ 1

ПОСТАНОВКА ТА ФОРМАЛІЗАЦІЯ ЗАВДАННЯ, СУТЬ І ПРАКТИЧНЕ ЗНАЧЕННЯ ПРЕДМЕТУ

1.1 Постановка та формалізація завдання

Результатом виконання повинен стати програмний продукт, який зможе забезпечити потребу в проведенні онлайн-конкурсів для учнів закладів середньої освіти, шкіл. Задля зацікавленості учасників кожен конкурс повинен мати можливість не бути схожим на інші, для цього потрібно надати можливість організаторам диверсифікувати створювані конкурси у максимально можливому об'ємі. Одним із компонентів, який повинен мати змогу варіюватися є запитання та їх типи. Можна виділити такі основні типи запитань:

- з однією правильною відповіддю;
- з декількома правильними відповідями;
- встановлення логічної відповідності;

Адміністратори або організатори повинні мати змогу в будь-який момент створювати конкурси - події з визначеними початковими і кінцевими датами, які характеризуються наборами питань і мають обмеження у часі на вирішення завдань. Для кожного питання можна обирати будь-який тип і визначати кількість балів.

Найважливішим компонент, на якому будуть базуватися конкурси є система тестування, яка відповідно надає можливість проводити тести.

Тест – це сукупність тестових завдань, за результатами виконання яких можна визначити рівень знань, вмінь і навичок у здобувачів вищої освіти з певної навчальної дисципліни.

					<i>ДТЕУ 121 06-01.БР</i>			
					<i>Онлайн-платформа Всеукраїнського учнівського конкурсу</i>	<i>Стадія РІ</i>	<i>Аркуш 6</i>	<i>Аркушів 49</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		18.02.22				
Керівник		Котенко Н. О.		18.02.22				
Гарант		Рзаєва С. Л.		18.02.22	<i>Постановка та формалізація завдання, суть і практичне значення предмету</i>	<i>Факультет інформаційних технологій, 4 курс, 6 група</i>		
Розроб		Александров А. Ю.		18.02.22				

Тестове завдання – це складова тесту, що відповідає рекомендаціям до формування тестових завдань, з якими здобувач вищої освіти (під час виконання тесту) виконує певну дію, а його результат реєструється у формі окремої відповіді, що оцінюється встановленою кількістю балів [1].

Тест має такі характеристики, як інструкція, що описує загальні правила виконання та час, відведений на проходження. Тестові запитання складаються зі таких невідчужуваних частин:

- інструкція (директива) - власне запитання, на яке потрібно правильно визначити відповідь;
- варіанти відповіді - опції, серед яких учасників повинні знайти відповідь на поставлене запитання (або, в залежності від типу запитання, - встановити відповідність між двома значеннями);
- правильні варіанти відповіді - опції, які є відповідями на запитання і надають учасникам, які обрали їх, бали;
- кількість балів - може бути однаковою на усі запитання із тесту або визначатися для кожного запитання окремо;

1.2 Формалізація вимог до розробки програмного забезпечення

Вимогами до програмного забезпечення називають описи різних аспектів системи, що розробляється. Якісно виявлені вимоги дозволяють визначити властивості майбутнього продукту, фіксують домовленості між замовником і виконавцем, а також є основою для планування, проектування інтерфейсів, архітектури, тестування і документування, тобто допомагають успішно завершити проект [2]. Збір, аналіз та опис вимог є важливим завданням, яким часто нехтують, але це може надати значну перевагу при проектування та власне розробці.

Існують різні методи визначення чи збирання вимог, серед них можна виділити такі:

- мозковий штурм;
- інтерв'ю;
- фокусна група;

								Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-01.БР			7

- спостереження;
- аналіз зацікавлених сторін (потенційних клієнтів, членів команди чи будь-кого, на кого впливає розроблюваний проєкт);

Основні вимоги до проєкту було визначено шляхом вивчення прикладів реалізації проєктів у таких самих або суміжних предметних областях, спостереження за їх роботою.

Розроблюване програмне забезпечення, яке призначене для проведення конкурсів у виглядів тестування, не є унікальним та має різноманітні аналоги. Багато з таких продуктів загальнодоступні та легкі у використанні, серед них можна виділити такі:

1. Kahoot – навчальна платформа, заснована на ігровому підході, де користувачі можуть створювати власні набори тестів. Використовується у школах та інших навчальних закладах. Серед безкоштовних функцій є можливість створення запитання з декількома відповідями та запитання з відповіддю Так/Ні. Всі інші види є платними. Використовується також для проведення онлайн-вікторин в реальному часі.
2. Quizizz – навчальна платформа, яка використовує схожий з Kahoot підхід, але додатково може використовуватися для призначення тестів, як позанавчальну діяльність, коли кожен з учнів має свій розпорядок, якого дотримується.
3. Quizlet – навчальний онлайн-застосунок, який дозволяє вивчати нове, та перевіряти знання, використовуючи флеш-картки та різні ігри. Quizlet має рекламу, а також платну підписку для додаткових функцій. Кожен з користувачів може створювати власні набори запитань і поширювати їх для інших.
4. Google Forms – частина комплекту інструментів Google Docs Editors, програмне забезпечення для управління опитуваннями.

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			8

Кожен з цих продуктів можна використовувати для створення, обробки та збереження результатів тестування. Але всі вони й мають свої недоліки у вигляді реклами або платних функцій, що погіршує загальний користувацький досвід.

Перевагами розроблюваного програмного забезпечення є унікальність дизайну та повна орієнтованість на єдину спеціалізацію – правознавство, відсутність реклами. Також розробка власної системи дозволяє повністю підлаштовувати її під власні потреби, змінювати як користувацький інтерфейс так і функціональність.

В ході аналізу конкурентних програмних засобів було визначено такий набір атрибутів якості (quality attributes), якими повинно бути наділене розроблюване програмне забезпечення:

- незалежність від платформи – програмне забезпечення має працювати з різними платформами та браузерами. Це передбачає адаптивність для доступності як на мобільних пристроях, так і на ПК;
- модульність – програмне забезпечення ділиться на окремі частини, які можуть працювати окремо та взаємодіють між собою за певним принципом;
- швидкодія - програмне забезпечення має відповідати на запити користувачів без видимих затримок;
- залежність від енергії - інформація зберігається в постійну пам'ять і при будь-яких технічних неполадках або вимкненнях електроенергії важлива інформація зберігає свій стан;
- доступність - графічний інтерфейс повинен відповідати сучасним трендам.

Важливим для правильного створення та подальшого функціонування будь-якого продукту є правильно створений портрет клієнта - уявна людина, якій притаманні всі характеристики цільової аудиторії. Визначення портрету клієнта дозволяє персоналізувати інформацію про надавані послуги, використовуючи правильну мову. Відштовхуючись від сутності програмного забезпечення можна визначити наступні основні характеристики:

- соціально-демографічні:

					ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

- вік: 15-19
- географія: Україна
- спосіб життя: надає перевагу отриманню послуг онлайн;
- інтереси і захоплення, хобі: юриспруденція;

Після проведених досліджень було визначено ряд вимог, які дозволяють кінцевому продукту займати конкурентне становище на ринку:

1. Розроблене програмне забезпечення відображає та отримує інформацію у вигляді вебсайту. Вебсайт повинен бути доступним з більшості сучасних операційних систем, включаючи мобільні; з основних браузерів.
2. Обрані основні інструменти розробки мають бути підтримуваними та сучасними задля можливості подальшого функціонування та розвитку програмного забезпечення без заміни інструментів.
3. Програмне забезпечення надає можливість автентифікації та авторизації користувачам та організаторам - для можливості збереження результатів та іншої важливої інформації. Автентифікація користувачів повинна бути надійною та зручною у використанні.
4. Чутлива інформація користувачів та така, що використовується у програмному коді, наприклад, паролі повинна зберігатися з дотриманням усіх необхідних заходів задля забезпечення максимального рівня безпеки.
5. Архітектура програмного забезпечення повинна бути побудована з максимальним рівнем декомпозиції її внутрішніх елементів на усіх рівнях, що дозволить легко переміщати ці елементи (наприклад, між різними серверами), а також масштабувати за потреби.

Розробка програмного забезпечення у вигляді вебсайту, на відміну від інших видів, серед яких мобільні застосунки, настільні програмні засоби, має ряд переваг та недоліків.

Серед переваг можна виділити доступність використання - усі користувачів, у яких є доступ до Інтернету, а також будь-який сучасний пристрій - смартфон, планшет, ноутбук чи інше можуть користуватися вебсайтом, при цьому створення

					<i>ДТЕУ 121 06-01.БР</i>	Аркуш
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		10

адаптивного для різних пристроїв та розмірів екрану інтерфейсу не є обов'язковим, хоча й дуже важливим, адже відповідно до статистичних даних, близько 70% всесвітнього трафіку до вебсайтів припадає саме на мобільні пристрої (Рис.1.1.).

Sum of Total Visits

PERFICIENT

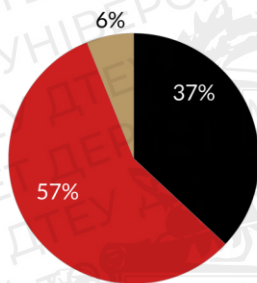
Desktop

Mobile

Tablet

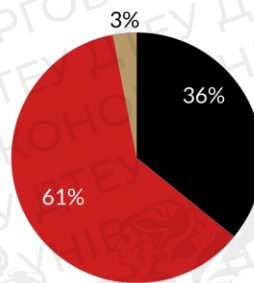
U.S. Only 2019

Based on 5.7 Trillion Visits



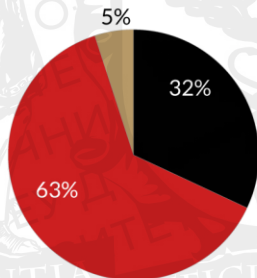
U.S. Only 2020

Based on 5.0 Trillion Visits



Global 2019

Based on 37.5 Trillion Visits



Global 2020

Based on 30.2 Trillion Visits

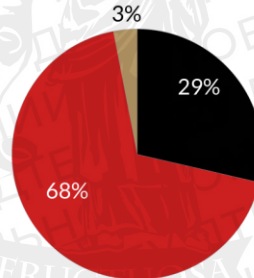


Рис. 1.1. Розподілення трафіку вебсайтів за пристроями

А серед недоліків вебсайту як форми програмного забезпечення є власне необхідність наявності в учасників доступу до мережі Інтернет та необхідність у додаткових налаштуваннях, які притаманні тільки вебсайтам, наприклад, домену. Попри усі недоліки вебсайти є найпопулярнішою, найбільш звичною та зручною для кінцевих користувачів формою відображення, збирання та обробки даних.

Необхідність у незалежності від енергії та одночасно у швидкодії забезпечується використанням бази даних. База даних (англ. database) – це поіменована, структурована сукупність взаємопов'язаних даних, що належать до певної предметної області і зберігається на комп'ютерних носіях, зазвичай разом з

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			11

прикладною програмою. Основне призначення баз даних – зберігання, накопичення, оновлення і пошук необхідної інформації [3].

Використання бази даних вимагає визначення системи керування базами даних - MySQL, sqlite, Microsoft SQL Server, PostgreSQL - приклади найпопулярніших.

Окрім усього наведеного вище, можна виділити вимоги, які хоча й не є обов'язковими, але дотримання яких допоможе підвищити ефективність розробки програмного забезпечення чи його подальшої роботи:

- логування усіх найважливіших подій, які відбуваються у ході роботи програмного забезпечення - помилок, попереджень та іншої інформації, яка може бути корисною для подальшого аналізу;
- використання асинхронності всюди або у необхідних для цього місцях, що дозволить значно підвищити ефективність роботи програмного забезпечення і відповідно покращити користувацький досвід;
- використання системи контролю версій, що значно покращить досвід розробки програмного забезпечення і прискорить її, надаючи можливість легко переглядати та переміщатися по історії версій програмного коду;
- використання об'єктно-реляційної проєкції для доступу до бази даних у формалізованому щодо SQL вигляді, що дозволить збільшити безпеку (наприклад, від SQL-ін'єкцій), а також зменшити кількість помилок, які могли б виникати при виконанні чистих SQL запитів;
- система сповіщень користувачів про різні події, наприклад, про результат тестування.

Увесь комплекс заходів для задоволення усіх вимог та забезпечення успішного виконання проєкту включає:

1. постановка задачі, визначення мети та актуальності проєкту, дослідження предметної області, аналіз конкурентів, схожих засобів та визначення їх переваг і недоліків;
2. проєктування архітектури програмного забезпечення;

					<i>ДТЕУ 121 06-01.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

3. проектування бази даних;
4. проектування UI/UX інтерфейсу;
5. розробка, впровадження усіх спроектованих частин та поєднання їх в одну цілісну систему;
6. тестування системи;
7. розгортання;

1.3 Практичне значення розроблюваного програмного продукту та відповідної предметної області

Освіта є надзвичайно важливою частиною суспільства і її розвиток повинен бути одним з перших його пріоритетів. Відповідно до опитування, близько 40% українців вважають, що проблема вдосконалення якості середньої освіти в Україні є важливою, а близько 34% вважають цю проблему першочерговою і з часом цей відсоток стрімко зростає (Рис.1.2.).

У більшості випадків розвиток освіти супроводжується, а у деяких випадках навіть є неможливим, без впровадження сучасних технологій. Розроблюваний проєкт може слугувати прикладом використання Інтернету та інших засобів, які надає сучасний світ, для задоволення потреб освіти і її удосконалення.

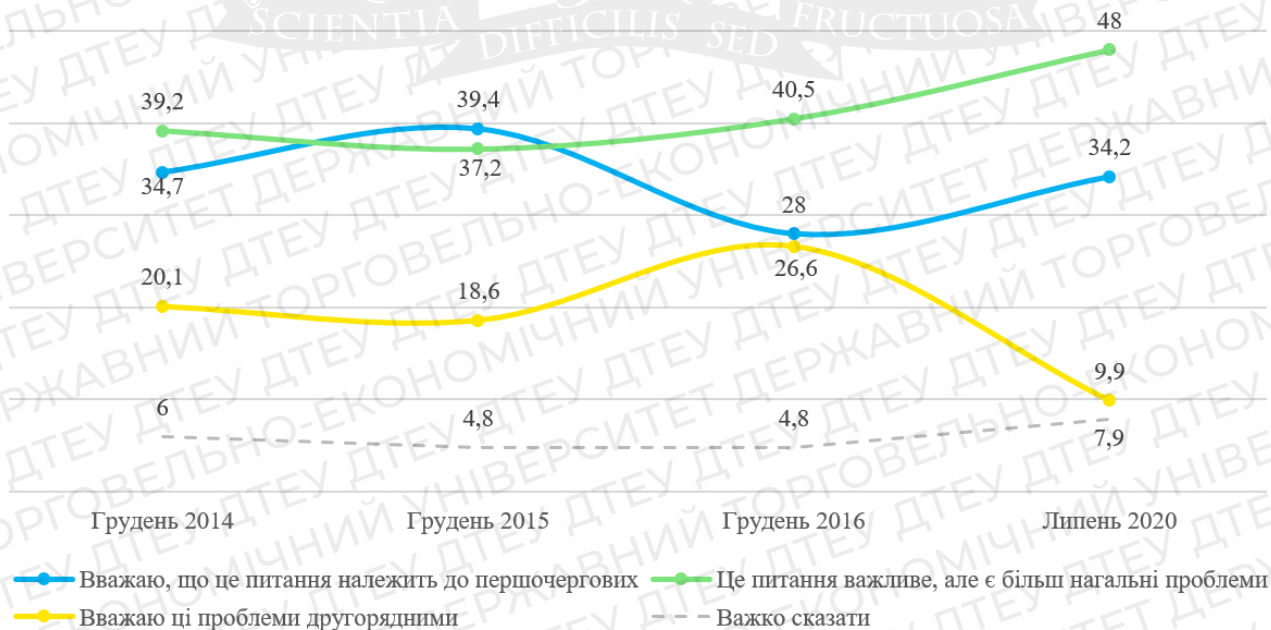


Рис. 1.2. Ставлення українців до проблеми вдосконалення середньої освіти в Україні (2020 рік)

Програмне забезпечення призначене для створення та проведення конкурсів з правознавства, обробку отриманих після завершення конкурсів даних, їх аналіз та визначення переможців.

Доцільність виконання проєкту, як інструменту дистанційної освіти, підтверджується наявними статистичними даними щодо використання новітніх технологій у навчанні.

Дистанційна освіта стала невід'ємною частиною освітньої системи в усьому світі. Офіційні статистичні дані з дистанційної освіти, підтверджують неухильне зростання числа учасників заочного навчання та курсів. Успіх дистанційного навчання можна пояснити на фоні мінливих умов у економіці, технології та суспільстві [4].

1.4 Аналіз та вибір інструментів та засобів для розробки

Розробка будь-якого програмного забезпечення вимагає певних інструментів. Кожен проєкт потребує різного набору таких інструментів і їх вибір значно впливає на ефективність подальшої розробки.

Основним інструментом розробки є мова програмування, відповідно до сучасних трендів та вимоги у форматі програмного забезпечення - вебсайті можна виділити такі:

Таблиця 1.1.

Порівняння мов програмування

Мова програмування для вебробки	Переваги	Недоліки
Golang	статично типізована, підтримка багатопотоковості, автоматичне управління пам'яттю	відносно нова - кількість наявних бібліотек замала, відсутність можливості управління пам'яттю вручну

Python	величезна кількість бібліотек - готових рішень, читабельність	використання пам'яті у великих об'ємах
C#	ефективність виконання програм, підтримка асинхронності	вимагає відносно великих зусиль для вивчення і порівняно низьку читабельність
PHP	популярна, легка для вивчення	складність подальшої підтримки програм, динамічна типізація
Ruby	ефективність розробки, активна спільнота	відносно низька ефективність виконання програм
Java	підтримка багатопотоковості, велика кількість бібліотек	потребує значної кількості ресурсів для запуску

Проаналізувавши різні мови програмування, Python було визначено найбільш зручною та такою, що зможе найкраще задовольнити потреби розробки проєкту онлайн-платформи Всеукраїнського учнівського конкурсу з правознавства. Python - динамічно типізована високорівнева мова програмування, яка підтримує об'єктно-орієнтованість, найбільшою перевагою є наявність великої кількості бібліотек, які, додаючи абстракції, полегшують розробку. Серед найпоширеніших сфер використання Python машинне навчання, аналіз та візуалізація даних, розробка ігор, а також веброботика.

Для ефективної розробки необхідно використовувати фреймворк для веброботики. Найбільш відповідним для проєкту є фреймворк Django, який надає можливості авторизації/автентифікації, розробки шаблонів для клієнтської частини вебсайту та величезну кількість інших додаткових функцій, є другим найбільш популярним вебфреймворком для Python і одним з найпопулярніших серед усіх інших мов програмування (Рис.1.3.).

Які фреймворки/бібліотеки ви використовуєте з Python?

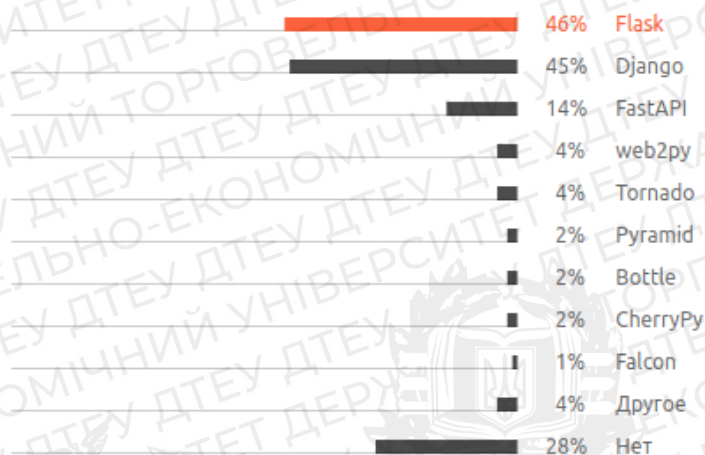


Рис. 1.3. Рейтинг фреймворків Python

1.5. Технічне завдання

Програмне забезпечення із користувацьким інтерфейсом у вигляді вебсайту дозволяє забезпечувати ефективно проведення конкурсів з правознавства, включаючи їх створення, проходження, а також перегляд результатів учасників. Проведення конкурсів супроводжується достатнім рівнем безпеки інформації, що зберігається в постійному сховищі – базі даних. Окрім цього, існує розподіл ролей користувачів із відповідними рівнями доступного функціоналу для учасників конкурсів, організаторів, модераторів відгуків.

1.6 Висновки до розділу 1

Отже, у будь-якій країні світу, як і в Україні, вдосконалення процесів освіти є важливою складовою розвитку всієї держави та має високі очікування у суспільства.

Розроблюване програмне забезпечення є прикладом удосконалення, впровадження у процеси середньої освіти сучасних технологій. Технологія Інтернет, як основа, та інші сучасні засоби допомагають досягненню максимально

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			16

зручного доступу та взаємодії, як учасників конкурсів, так і їх організаторів до розробленої системи тестування.

В ході дослідження було визначено основні вимоги, які ставляться до програмного забезпечення, способи та засоби, використання яких допоможе досягти виділених вимог та поставлених завдань. Крім цього, було визначено мінімальних список дій, які забезпечать успішне виконання проекту.



					<i>ДТЕУ 121 06-01.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

РОЗДІЛ 2

ПРОЄКТУВАННЯ КОМПОНЕНТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Проєктування архітектури програмного забезпечення. Створення UML-діаграм

Проєктування архітектури програмного забезпечення дозволяє описати структурні елементи та усі необхідні варіанти поведінки, взаємодії елементів між собою. Найпоширенішим способом опису архітектури ПЗ є UML (англ. Unified Modeling Language) - уніфікована мова моделювання, яка дозволяє описати проєкт при цьому не прив'язуючись до мови програмування, середі розробки чи інших чинників. Поділяється на три основних види - діаграми поведінки, взаємодії та структурні діаграми.

UML – це сімейство графічних нотацій, яке допомагає в описі та проєктуванні програмних систем, особливо систем, побудованих з використанням об'єктно-орієнтованої парадигми. Графічні мови моделювання вже тривалий час широко використовуються в індустрії програмування і проєктування інформаційних систем. UML є відносно відкритим стандартом, що знаходиться під управлінням групи OMG (Object Management Group), відкритого консорціуму компаній [5].

Цілі створення UML-діаграм:

- для проєктування великих проєктів, визначення дрібних деталей і на основі цього - створення коду;
- створення діаграм для вже існуючих проєктів на основі коду, для забезпечення документації;
- для використання як частини загальної документації проєкту.

					<i>ДТЕУ 121 06-01.БР</i>			
					<i>Онлайн-платформа Всеукраїнського учнівського конкурсу з правознавства</i>	<i>Р2</i>	<i>18</i>	<i>49</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		<i>04.03.22</i>				
<i>Керівник</i>		<i>Котенко Н. О.</i>		<i>04.03.22</i>				
<i>Гарант</i>		<i>Рзаєва С. Л.</i>		<i>04.03.22</i>				
<i>Розроб</i>		<i>Александров А. Ю.</i>		<i>04.03.22</i>				
					<i>Проєктування компонентів програмного забезпечення</i>		<i>Факультет інформаційних технологій, 4 курс, 6 група</i>	

2.1.1 Структурні діаграми

Структурні діаграми призначені для відображення структури статичних компонентів системи. До них відносяться такі:

- класів (class diagram);
- компонентів (component diagram);
- композитної структури (composite structure diagram);
- кооперації (collaboration - з'явилася у UML2.0);
- розгортання (deployment diagram);
- об'єктів (object diagram);
- пакетів (package diagram).

Діаграма класів призначена для моделювання класів об'єктно-орієнтованих додатків, визначення їх атрибутів та методів. Клас - конструкція, яка використовується для групування даних та функцій за певними ознаками, на основі одного класу, як на основі схеми, можна створювати безліч об'єктів, які вже містять в собі певні дані та можуть виконувати з цими даними різні операції.

Основні класи розроблюваного додатку використовуються для взаємодії з даними, які зберігаються в базі даних (Рис. 2.1.). Кожен з класів представляє окрему таблицю і кожен з атрибутів є колонкою бази даних. Такий підхід має назву ORM (англ. Object-relational mapping) - Об'єктно-реляційна проєкція). Це значно спрощує, формалізує і уніфікує взаємодію з даними, підвищує безпеку і знижує ймовірність помилок. За допомогою цих класів можна здійснювати отримання, запис, оновлення, фільтрування, видалення, сортування усіх даних, які зберігаються в постійній пам'яті.

Усі разом класи утворюють певну систему і є пов'язаними. Існують наступні види зв'язків:

- асоціація - вид зв'язку, який показує, що об'єкти класів пов'язані між собою;
- агрегація - зв'язок, за якого один з класів має вищий ранг і складається з декількох інших класів;

								ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата					19

Базовим елементом в процесі розгортання є база даних - основне постійне сховище даних, яке дозволяє як зберігати дані, так і записувати та зчитувати їх. Django-сервер - основний сервер, який здійснює обробку даних, їх збереження надає можливість функціонувати графічному інтерфейсу для взаємодії користувачів, взаємодіє з базою даних, функціонування підсистем авторизації, тестування та блогу реалізується саме цим елементом. Окрім цього, використовується Celery та Redis сервери, які забезпечують виконання асинхронних задач, та Nginx - вебсервер, через який користувачі отримують доступ до вебсайту, також забезпечує доступ до статичних і медіафайлів.

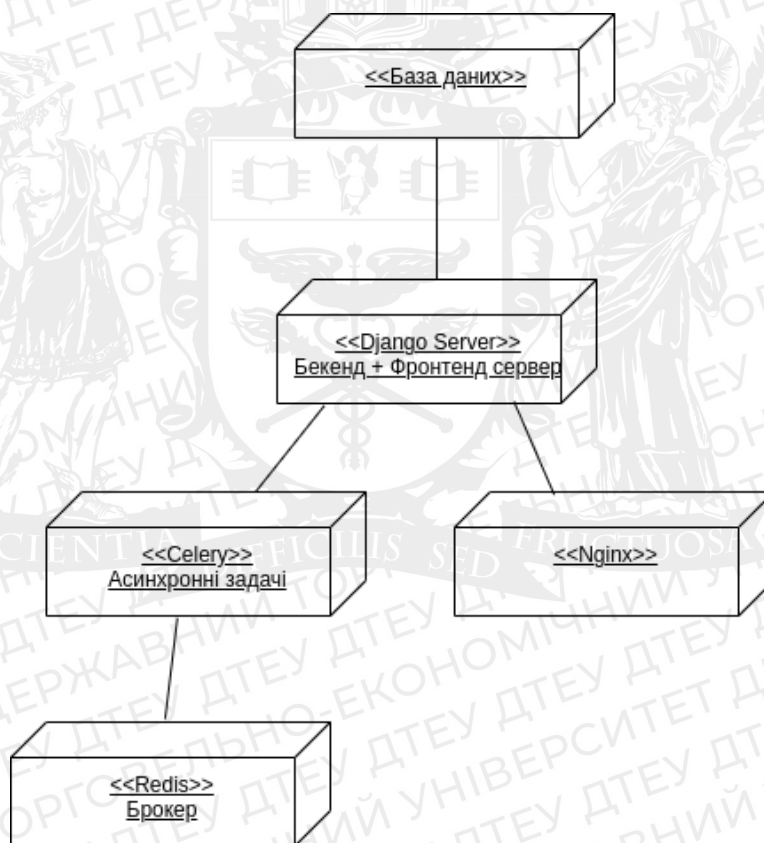


Рис. 2.2. UML-діаграма розгортання

2.1.2 Діаграми поведінки

Діаграми поведінки описують поведінку внутрішніх елементів системи

- діяльності (activity diagram);
- станів (state machine diagram);
- прецедентів (use case diagram).

					ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		21

Діаграма прецедентів або використання представляє найбільш узагальнене уявлення про поведінку системи, відображаючи об'єкти, які здійснюють взаємодію (актори) та дії, які дані об'єкти можуть здійснювати всередині системи (Рис. 2.3.).

На даній діаграмі можна побачити, що основними акторами є:

- учасник конкурсів, який здатний виконувати тести, здійснювати соціальну діяльність (писати коментарі), а також переглядати результати конкурсів;
- адміністратор (організатор конкурсів), основною можливістю якого є створення конкурсів - запитань та відповідей до них, а також модерація коментарів користувачів і створення записів у блозі;
- система тестування - здійснює підрахунок результатів, відправку пошти та підготовку такої інформації, як перегляд результатів конкурсів і відображення їх кінцевим користувачам.

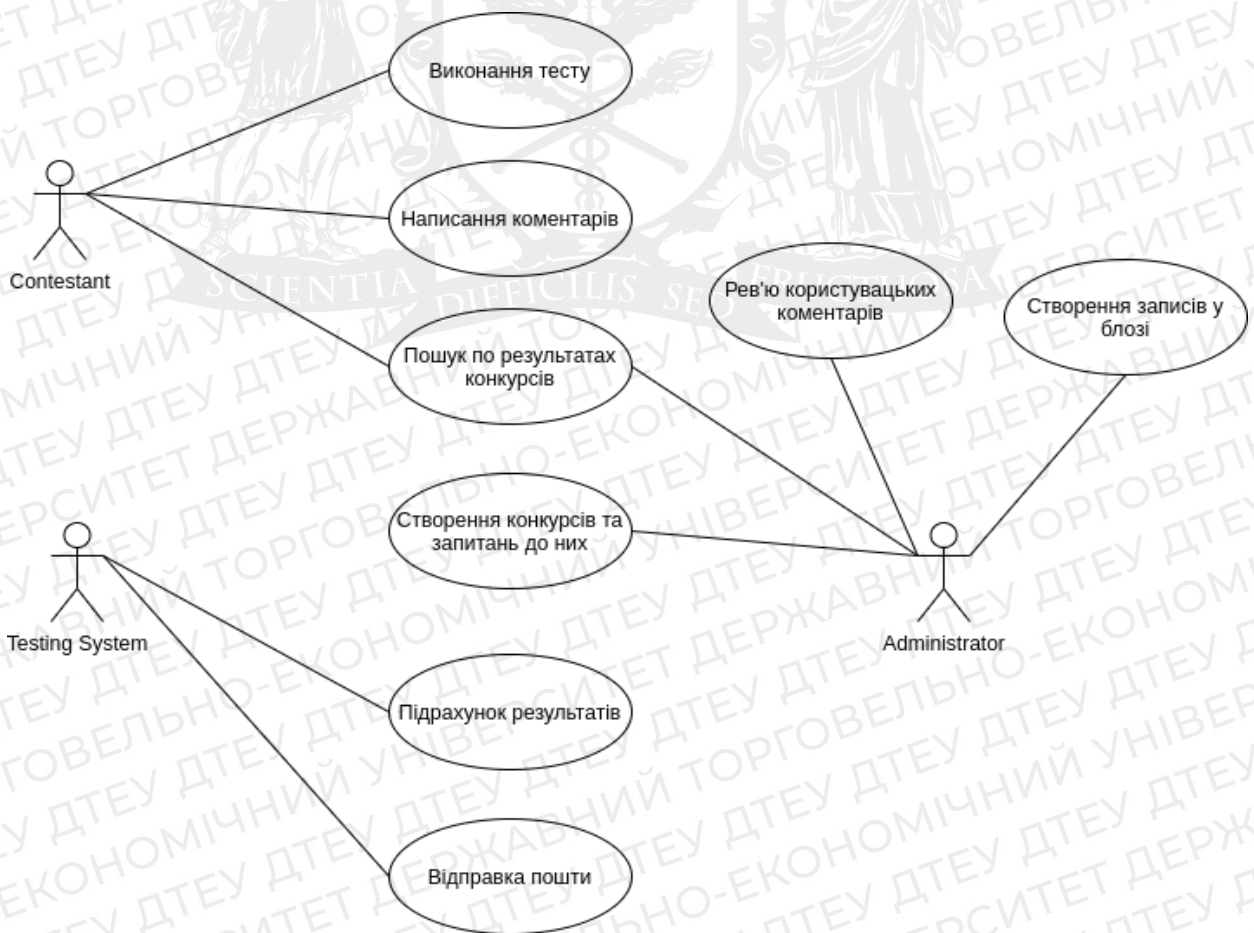


Рис. 2.3. UML-діаграма прецедентів

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			22

Діаграма діяльності - інша діаграма, яка відноситься до діаграм поведінки, демонструє кроки, які відбуваються при певному варіанті використання системи. Створена діаграма відображає процес проходження тесту користувачем, від авторизації чи створення облікового запису до виходу з нього (Рис. 2.4.).

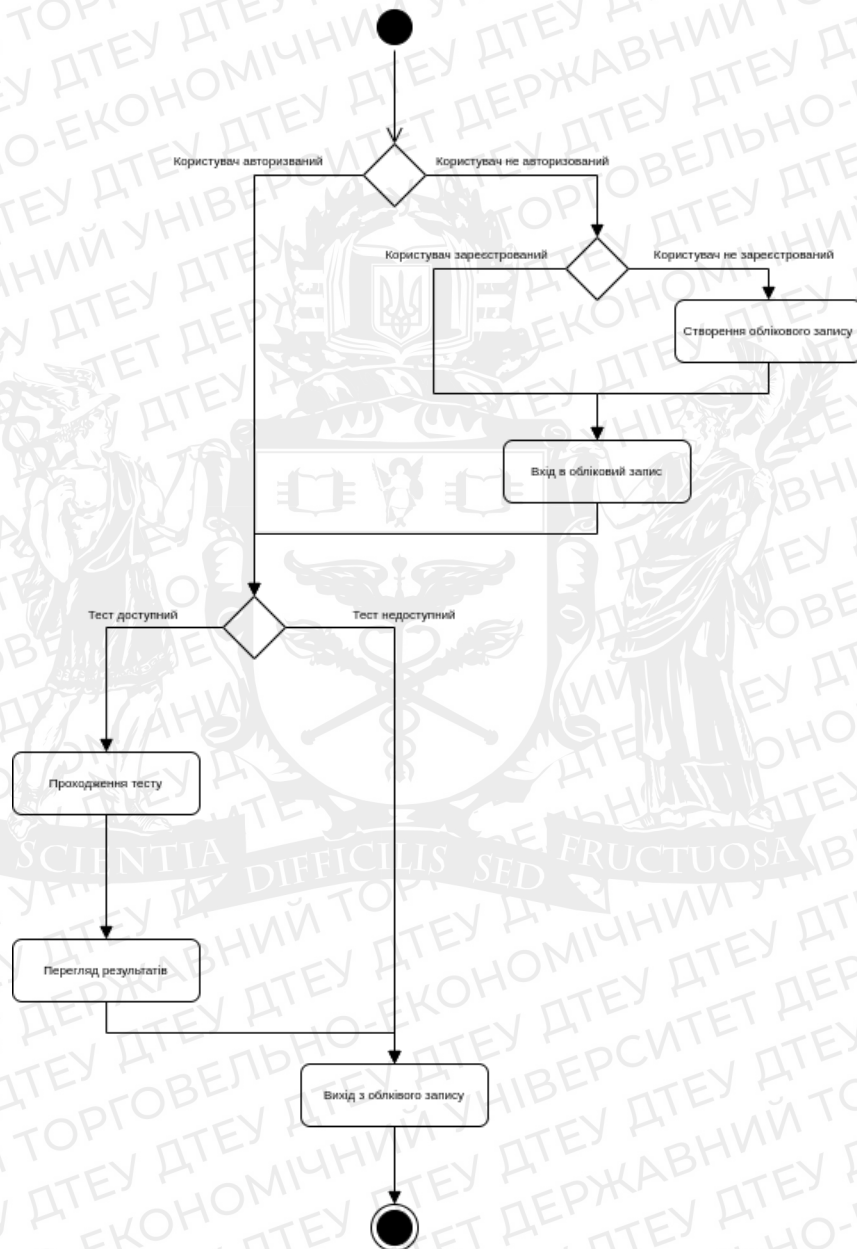


Рис. 2.4. UML-діаграма діяльності

2.1.3 Діаграми взаємодії

- комунікації (communication diagram - UML 2.0);
- огляду взаємодії (interaction overview diagram - UML 2.0);
- послідовності (sequence diagram);

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			23

- синхронізації (timing diagram - UML 2.0).

Діаграма послідовності - одна із UML-діаграм взаємодії, адже описує як і в якому порядку об'єкти взаємодіють між собою. Окрім цього, діаграма послідовності більш детально розкриває діаграму прецедентів, дозволяє змодельовувати і зрозуміти логіку функціоналу майбутніх можливих сценаріїв. На даній діаграмі можна побачити актора як початкову точку та подальші елементи програмного забезпечення між якими відбувається взаємодія, при цьому кожен з елементів має вертикальну лінію життя, на якій послідовно зображуються події (Рис.2.5.).

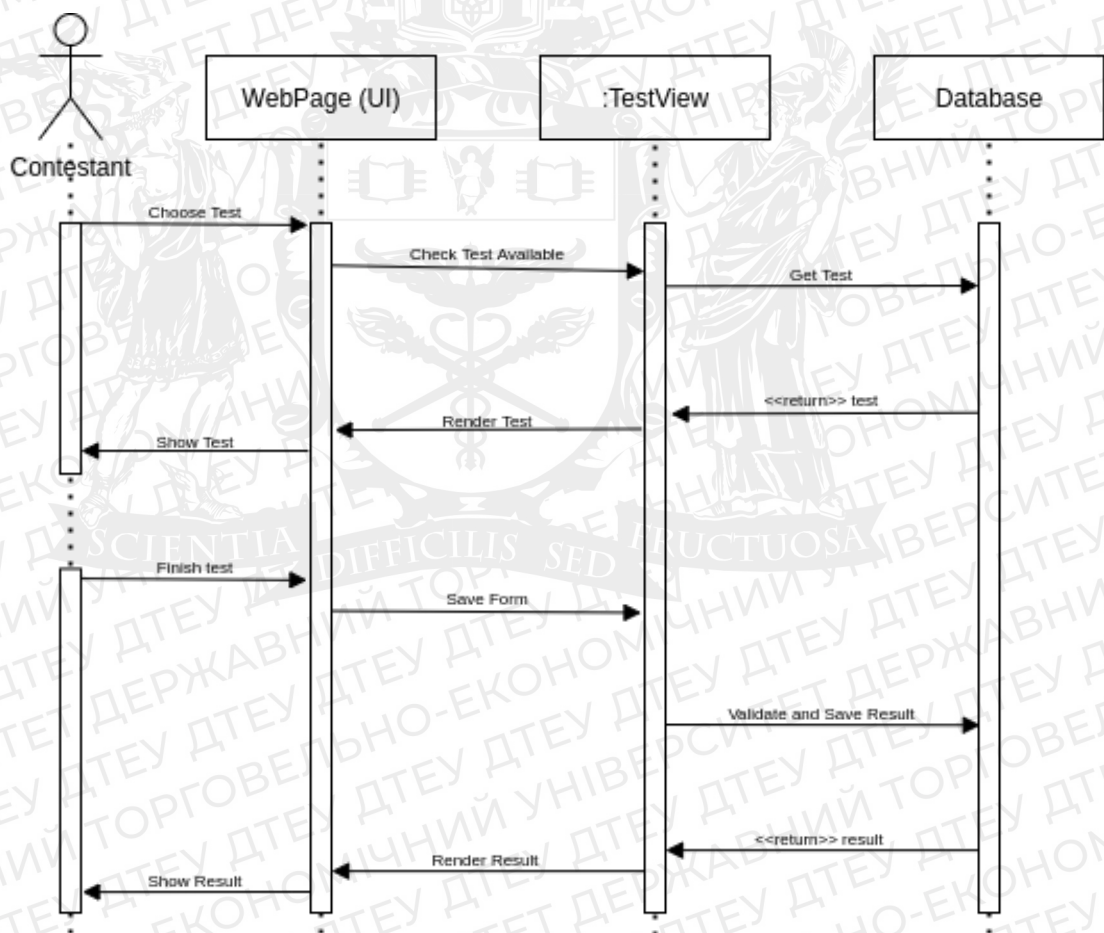


Рис. 2.5. UML-діаграма послідовності

2.2 Проектування бази даних

Найпоширенішим методом проектування баз даних є створення трьох моделей - концептуальної, логічної та фізичної, - де кожен наступний рівень стає

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		24

все менш абстрактним і надає все більше деталей про таблиці бази даних та зв'язки між ними. Цілями проектування бази даних є:

- визначення даних, які необхідно буде зберігати для використання у визначеному проєкті;
- зменшення надмірності, а також дублювання даних;
- забезпечення цілісності бази даних;
- створення умов, зв'язків між таблицями бази даних, за яких всі необхідні запити можуть бути ефективно виконаними.

2.2.1 Концептуальна модель

Найбільш примітивна високорівнева модель бази даних, показує таблиці та базові зв'язки між ними в абстрактному вигляді та не прив'язується до конкретної системи керування базами даних (СУБД) і не може бути використаною для побудови бази даних, так як містить недостатньо інформації. Основна мета для створення такої моделі - описати структури компонентів, які використовуються в бізнес-процесах, але не у власне фізичній базі даних (Рис. 2.6.).

На даній моделі можна побачити, що база даних застосунку складається з більш ніж десяти основних таблиць, які можна розділити на 3 основних групи по збереженим даним:

- для даних користувачів;
- для інформації про тести;
- для даних соціальної діяльності користувачів.

Ключовою таблицею усієї системи є таблиця user, яка зберігатиме інформацію про користувачів, їх обліковий запис, уся інша інформація так чи інакше стосується цієї таблиці.

Програмне забезпечення має також деякі інші таблиці, які є допоміжними та створюються автоматично:

- logentry - зберігає інформацію про створення, зміну чи видалення даних з будь-якої таблиці та користувачів, які це зробили;

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			25

- session - зберігає інформацію про сесії користувачів та є основою функціонування системи авторизації даного вебсайту;
- permission - налаштування дозволів для різних користувачів, разом з іншою таблицею Group надає можливість створювати такі групи користувачів, як модератори, з наданням можливості перегляду та перевірки коментарів користувачів;
- ряд таблиць (periodictask, crontabschedule, clockedschedule), які забезпечують роботу періодичних завдань Celery;

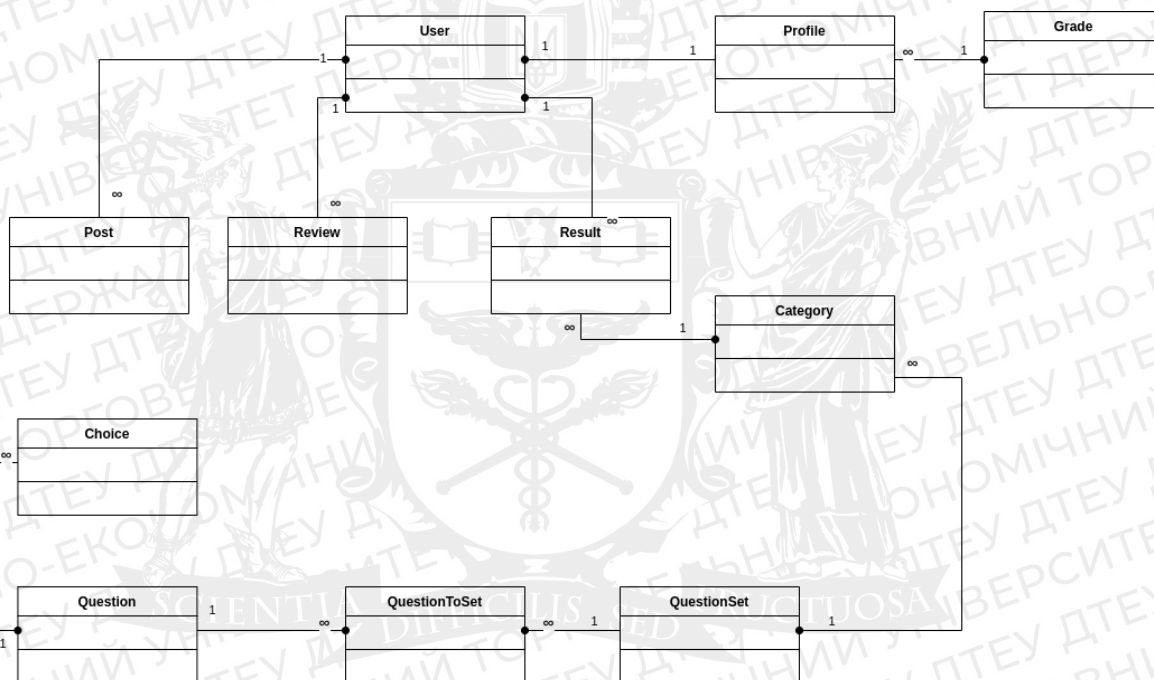


Рис. 2.6. Концептуальна діаграма даних

2.2.2 Логічна модель

Логічна модель значно розширює концептуальну, але ще є досить загальною і не використовує понять з конкретної системи керування базами даних.

На логічній моделі бази даних можна побачити, яку структуру має кожна з таблиць та їх характеристики, а саме поля, а також зв'язки між таблицями. Логічна модель даних створює основу для побудови фізичної моделі, допомагає уникнути можливої надмірності чи неузгодженості даних (Рис. 2.7.).

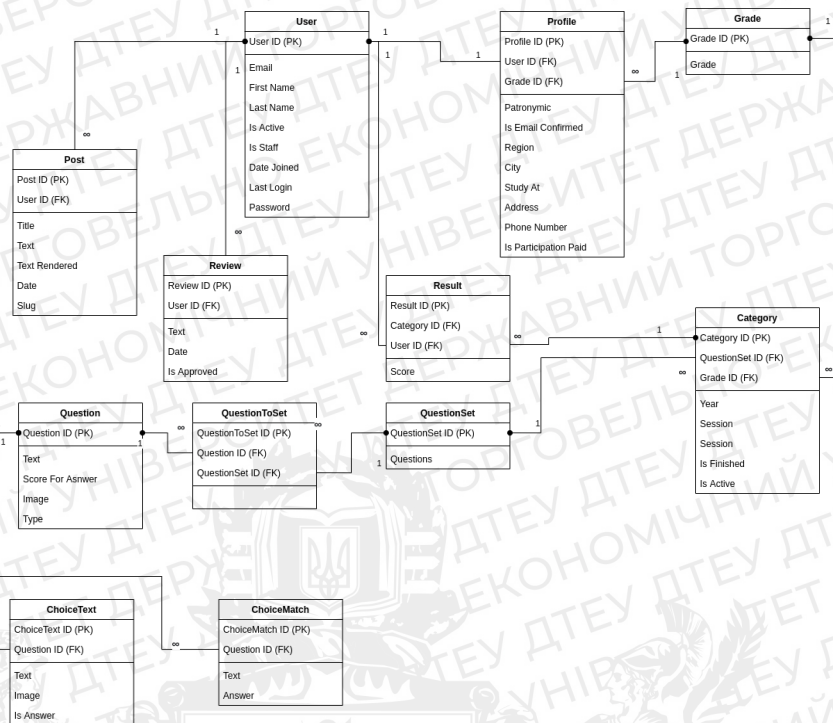


Рис. 2.7. Логічна модель даних

2.2.3 Фізична модель

Фізична модель бази даних - модель, найбільш приближена до реальної структури таблиць бази даних та їх зв'язків, з найменшим рівнем абстракції. На даній моделі можна побачити використовувані типи даних та назви колонок таблиць бази даних. Всі ці дані відповідають назвам, які використовуються у конкретній системі керування базою даних, у даному випадку - PostgreSQL.

На даній моделі можна також побачити такі характеристики, як:

- null-значення - можливість присутності пустих значень в колонці бази даних;
- ключ - визначає чи колонка бази даних є ключем - основний (primary key) або ключ для зв'язку з іншою таблицею (foreign key);
- індексоване - визначається для полів, по яким потрібен швидкий доступ, наприклад, для сортування чи пошуку, для цього створюється спеціальна структура, яка дозволяє замінити сканування всіх записів їх ефективним віднайднням. Індексованими полями позначені всі ключі;

- тип - тип колонки бази даних, у проєктованій базі даних використовуються різні типи даних, такі як VARCHAR, який дозволяє зберігати текстові дані із визначенням максимального розміру, TEXT - для збереження текстових даних необмежених розмірів, INTEGER - для збереження цілих чисел та інші.

На рисунку можна побачити фізичну модель частини бази даних, яка стосується користувачів, їх облікових записів, а також соціальної діяльності, яку вони можуть проводити (Рис. 2.8.).

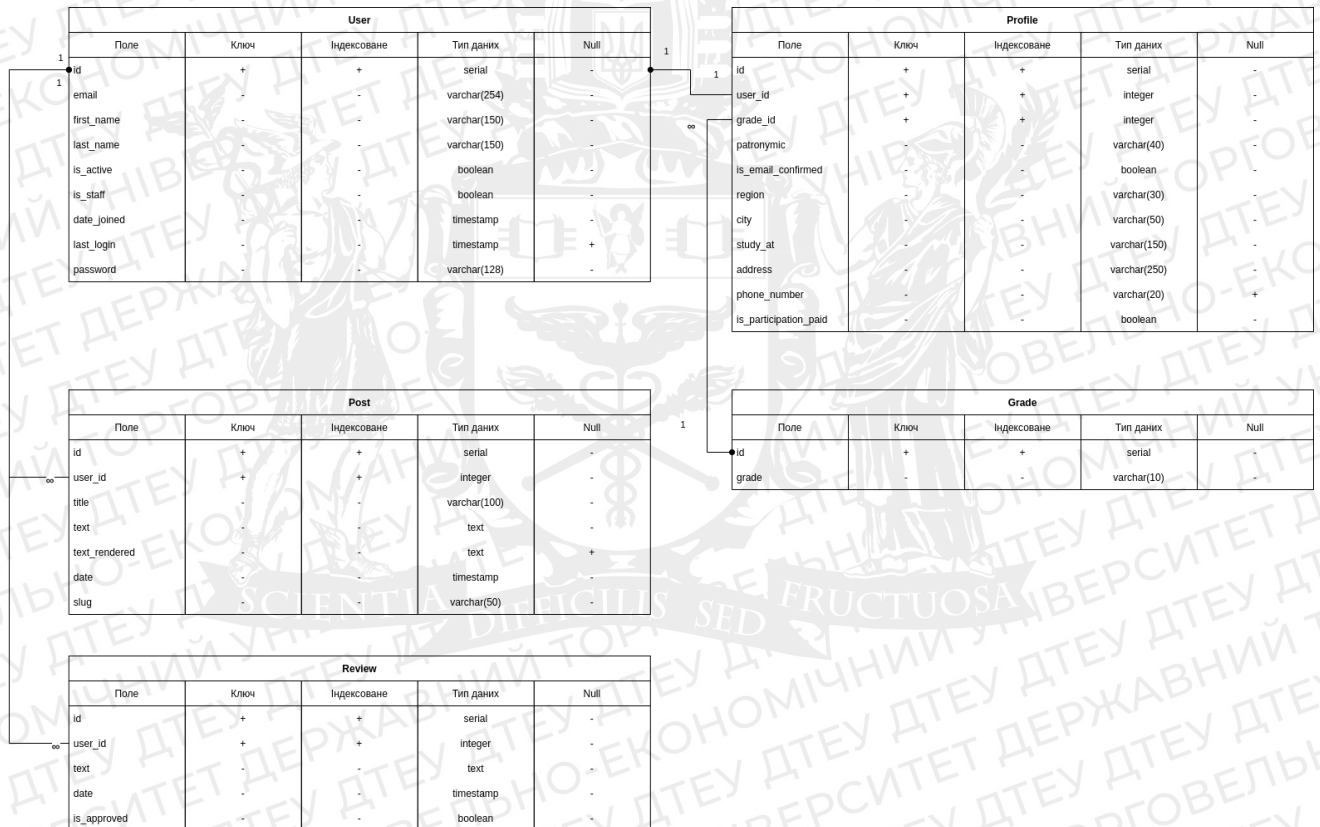


Рис. 2.8. Фізична модель даних користувача та таблиць для соціальної діяльності

2.3 Проєктування графічного інтерфейсу користувача

Графічний інтерфейс включає два основних елементи:

- користувацький інтерфейс (user interface - UI);
- користувацький досвід (user experience - UX);

Користувацький інтерфейс - поєднання типографіки, зображень чи інших візуальних елементів для створення інтерфейсу, з яким можна взаємодіяти для

досягнення певної цілі. Користувацький досвід - це те, що користувачі відчувають з моменту відкриття вебсайту до моменту його закриття - покликаний досягти найбільшої зручності та “очевидності” усіх компонентів графічного інтерфейсу.

Основним технічним рішенням, яке використовувалося при розробці графічного дизайну є Figma - багатоплатформний векторний сервіс для розробки та прототипування.

На першому етапі було розроблено каркас (wireframe) основних сторінок вебсайту, який відображає графічний інтерфейс, компоненти та їх розміщення в абстрактному вигляді (Рис. 2.9.).



Рис. 2.9. Каркас користувацького інтерфейсу

Наступний етап є власне розробка візуального дизайну із використанням заздалегідь визначених кольорів та шрифтів, розроблених таких графічних елементів, як логотипи. На цьому етапі можна побачити, яким чином остаточно буде виглядати вебсайт.

При цьому важливо враховувати правильність компоновання, щоб забезпечити найбільш оптимальний та доцільний варіант користувацького інтерфейсу, з найбільш високою ознакою його сприйняття кінцевими

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			29

користувачами. Для вибору кольору за теорією кольору існує чотири класичних типи комбінації колірних схем:

- доповнюючий (комплементарний);
- аналогічний;
- потрійний (тріадний);
- чотирьохколірний (тетраедний).

Окрім наведених існує монохроматична схема, яка використовує один колір і всі його відтінки – саме такий варіант компоновання кольорів було використано. (Рис. 2.10.).



Рис. 2.10. Візуальний дизайн користувацького інтерфейсу

Окрім візуального дизайну, його потрібно пристосувати до різних пристроїв та різних розмірів екранів, як візуально, так і функціонально - це дозволить значно збільшити коло користувачів, при цьому залишаючись зручним інтерфейсом для кожного з них. Розробка адаптивного дизайну (responsive design) включає створення аналогів усіх розроблених раніше сторінок, але для менших розмірів, із забезпеченням максимального рівня читабельності та доступності усіх елементів (Рис. 2.11.).

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			30

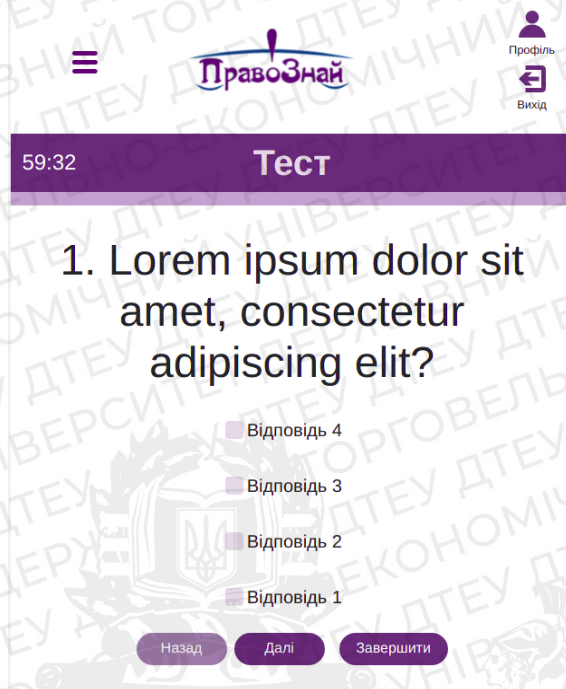


Рис.2.11.. Адаптивний дизайн на мобільному пристрої

Останнім етапом є розробка інтерактивного прототипу вебсайту - того, як елементи на сторінках та власне сторінки взаємодіють між собою.

2.4 Висновки до розділу 2

Отже, в ході роботи було спроектовано набір компонентів, які синтезуються в єдину систему і описують її із різних сторін. З допомогою уніфікованої мови моделювання (UML) спроектовано архітектуру, створено основні діаграми усіх наявних типів - структури, поведінки та взаємодії, які відображають основні елементи, які будуть наявні в програмному забезпеченні, та те, як вони між собою взаємодіятимуть.

Окрім архітектури, було спроектовано базу даних - основне місце постійного збереження даних, створено три базових моделі, які будуть використані для побудови таблиць бази даних та їх колонок. А також графічний інтерфейс користувача, включаючи каркас, візуальний дизайн і інтерактивний прототип, враховуючи користувацький досвід.

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			31

Усі спроектовані компоненти були створені із дотриманням різних рівнів абстракції, як концептуальна чи фізична модель бази даних, і можуть бути використані для різних цілей в подальшому.



									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 06-01.БР</i>				32

РОЗДІЛ 3

ОПИС ТЕХНІЧНОГО РІШЕННЯ

3.1 Розробка програмного забезпечення

Для розробки програмного забезпечення було обрано принцип мікросервісів, який реалізується контейнерами Docker (Рис. 3.1.). Робота контейнерів передбачає таку послідовність процесів:

1. створення Dockerfile (Додаток Б);
2. створення образу (Image) на основі Dockerfile;
3. запуск контейнера з образу.

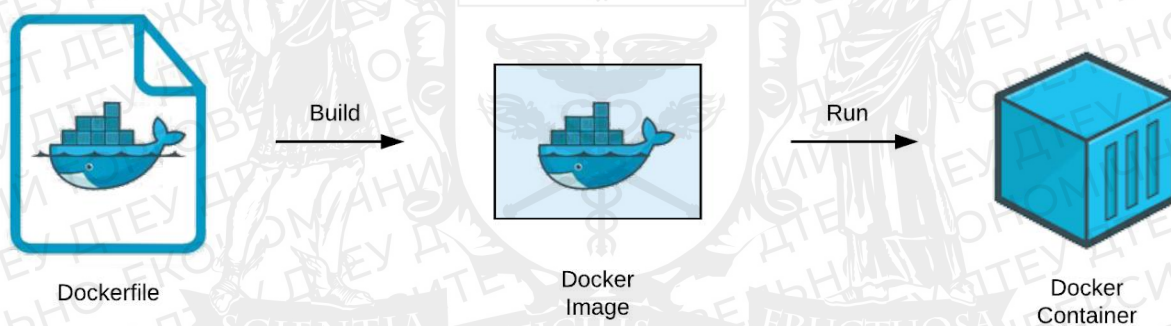


Рис. 3.1. Процес створення контейнеру Docker

Кожен з мікросервісів є сервером та має свій порт, через який може взаємодіяти з іншими. Основний мікросервіс – «web», займається обробкою запитів користувачів та відправкою запитів, забезпечує функціонування вебсайту.

«celery» та «redis» - мікросервіси для роботи з асинхронними задачами.

					<i>ДТЕУ 121 06-01.БР</i>					
					<i>Онлайн-платформа Всеукраїнського учнівського конкурсу</i>			<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				<i>РЗ</i>	<i>33</i>	<i>49</i>
Зав. каф.		Криворучко О.В.		11.04.22						
Керівник		Котенко Н. О.		11.04.22						
Гарант		Рзасва С. Л.		11.04.22						
Розроб		Александров А. Ю.		11.04.22	<i>Опис технічного рішення</i>			Факультет інформаційних технологій, 4 курс, 6 група		

«flower» - мікросервіс для візуалізації асинхронних задач.

«db» - мікросервіс для бази даних.

Для розробки використовуватиметься наступний стек технологій:

Таблиця 2.1.

Список використаних технологій

Призначення	Технологія
Мова програмування	Python
Вебфреймворк	Django
База даних середі розробки	sqlite
База даних основної середі та середі тестування	PostgreSQL
Шаблони	Django Templates
ORM	Django ORM
Реалізація клієнтської частини вебсайту	HTML, CSS, JavaScript, JQuery, Bootstrap
Асинхронна черга завдань	Celery
Брокер черги завдань	Redis
WSGI-сервер	Gunicorn
Вебсервер	Nginx
SMTP-сервер	Outlook
Віртуалізація середі розробки	Python Venv
Віртуалізація (контейнеризація) основної середі	Docker
Сервіс для збереження образів контейнерів	DockerHub
Система контролю версій	Git (GitHub)
Кешування	Cachalot

Профілювання (для аналізу вразливих місць в запитах, включаючи запити в базу даних)	Django Debug Toolbar
Логування	Sentry
Лінтер (для підтримання стилю і форматування коду)	Flake8

Основний мікросервіс також можна поділити на декілька частин за спільними ознаками:

- тестування - проведення конкурсів;
- акаунти - авторизація, автентифікація, реєстрація;
- блог - новини, відгуки.

Кожна з визначених частин пов'язана між собою, проте розробка може відбуватися послідовно. Для усіх частин потрібно забезпечити функціонування ряду стандартних компонентів Django:

- моделі (models) - використовуються для об'єктно-реляційного відображення, інкапсулює доступ до даних бази даних;
- представлення (views) - використовуються для обробки даних користувачів;
- шаблони (templates) - використовуються для відображення сторінок та даних у вигляді HTML-файлів;
- форми (forms) - використовуються для отримання даних користувачів;
- routes (urls) - направлення користувачів на потрібні сторінки з відповідними даними;
- тести (tests) - перевірка правильності функціонування системи;
- панель адміністратора (admin) - можливість створювати, змінювати, видаляти та переглядати дані, які використовуються на основному вебсайті;

Моделі, представлення та шаблони разом складають шаблон проектування програмного забезпечення MVT (Model, View, Template), який лежить в основні фреймворка Django (Рис. 3.2.).

									Аркуш	
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 06-01.БР</i>					35

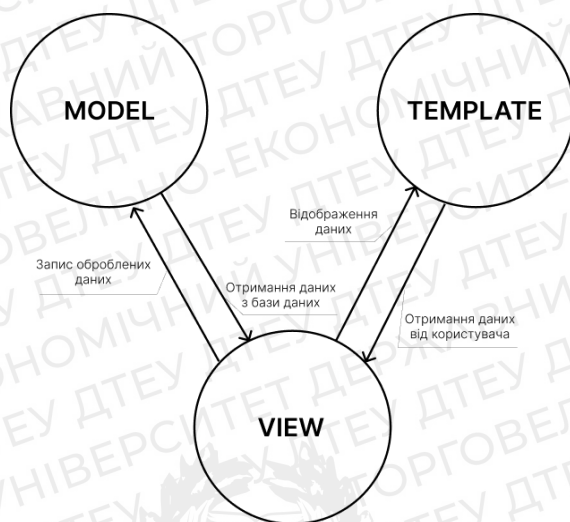


Рис. 3.2. Модель архітектури Django-проєкту

При роботі з мовою програмування Python для кожного проєкту, в якому використовуються сторонні бібліотекою, позитивною практикою є створення віртуального середовища. Віртуальне середовище – це інструмент, який дозволяє управляти залежностями та ізолює кожен проєкт разом з усіма наявними там додатковими бібліотеками, тобто будь-які зміни в одному проєкті не вплинуть на інші. Серед наявних інструментів можна виділити такі: virtualenv, ruenv, ruwin, anuenv і venv, який є частиною стандартної бібліотеки Python. Окрім різних версій модулів, деякі з них надають можливість вказувати потрібну версію Python. Використання віртуального середовища допомагає створювати самодостатні проєкти, підтримувати робоче середовище у чистоті, не нагромаджуючи його модулями, які не використовуються, та уникати великої кількості помилок.

Не менш важливим на даному етапі розробки є підключення до системи управління версіями. Система управління версіями (СУВ) - це система, що зберігає зміни в одному або декількох файлах так, щоб потім, за потреби, можна було відновити відповідні старі версії. Досить часто для управління версіями користувач копіює файли проєкту в інший каталог. На жаль, такий підхід найчастіше дає збої. За допомогою систем управління версіями можна зберігати минулі версії програм і завантажувати їх у разі потреби [6].

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			36

Для користувачів систем управління версіями відкриваються нові можливості:

- налаштування одночасного доступу різних користувачів до одного файлу, що значно полегшує командну розробку;
- розвиток різних версій одного проекту;
- документування кожної зміни, внесеної до будь-якого файлу.

Серед усіх аналогів СУВ, таких як Mercurial, SVN, для розробки проекту було обрано розподілену систему управління версіями git. Окрім цього, увесь програмний код може зберігатися на окремому сервері, що забезпечує розробників від втрати усього напрацьованого матеріалу. Для таких цілей було обрано сервіс GitHub.

Розробка кожної частини вебсайту для Всеукраїнського конкурсу з правознавства вимагає створення двох основних частин програмного коду:

- бекенду - та частина вебсайту, яка відповідає за роботу з базою даних, обробкою даних користувачів, їх виведення та запису, запуск періодичних завдань, відправку електронної пошти та інших завдань. Робота бекенду забезпечується такими технологіями як Django, Celery та іншими;
- фронтенду - частина вебсайту, яка виводить дані, отримані з бекенду, у розробленому користувацькому інтерфейсу, а також дозволяє приймати дані, які вводить користувач і відправляти їх на сервер для подальшої обробки. Забезпечується такими технологіями як Django Templates, JavaScript, HTML, CSS та іншими.

3.1.1 Розробка підсистеми авторизації

Авторизація є важливим компонентом, адже дозволяє користувачам вебсайтів зберігати дані та використовувати їх на різних пристроях у будь-який час. При розробці за основу була взята система, наявна в Django. Таким чином, основним суб'єктом, таблицею в базі даних та моделлю в об'єктно-реляційному відображенні є користувач, ним може бути як учасник конкурсу, так і адміністратор

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-01.БР				37

та/або організатор конкурсів. Така приналежність визначається окремими полями в базі даних.

Основним ідентифікатором користувачів є електронна пошта - в парі з паролем дозволяє авторизуватися на вебсайті. Для зручності та безпеки при реєстрації використовується відправка листів з посиланням для підтвердження власності email-адреси.

Отже, цілісний процес реєстрації облікового запису складається з наступних етапів:

1. Введення даних;
2. Валідація даних:
 - a. валідація формату електронної пошти користувача;
 - b. валідація достатнього рівня складності пароля;
 - c. валідація формату номера телефону;
 - d. валідація заповненості обов'язкових полів;
3. Створення хешу з пароля і збереження усіх даних в базу даних;
4. Відправка повідомлення на вказану електронну пошту з унікальним посиланням для підтвердження власності облікового запису;
5. Підтвердження облікового запису та, як результат, - його активація;

Окрім цього, в даній частині програмного забезпечення відбувається отримання історичних даних про результати користувачів, пошук та фільтрування по цих даних, а також розподілення місць при завершенні конкурсів.

3.1.2 Розробка підсистеми тестування

Підсистема тестування - основна частина програмного забезпечення, яка тісно пов'язана з підсистемою авторизації. Найважливішим компонентом цієї частини програми є форми, саме вони забезпечують функціонування конкурсів та їх тестових завдань. Не менш важливим є функціонал для підрахунку та збереження результатів користувача за конкретний тест.

Організатори конкурсів можуть створювати, оновлювати чи видаляти тести, контролювати їх проходження користувачами, при цьому валідація тестів (процес

					<i>ДТЕУ 121 06-01.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		38

перевірки відповідей користувачів і виведення інформації про це) проходить автоматично. Після завершення тесту користувачі можуть бачити такі варіанти результатів відповідей:

- без кольору - не відмічений користувачем варіант, який не є правильним;
- зелений колір - відмічений варіант є правильним;
- червоний колір - не відмічений варіант, який є правильним; або відмічений варіант не є правильним.

Відповідно до усіх цих даних та визначеної кількості балів за тестове запитання розраховується результат користувача. На основі результатів усіх користувачів визначаються місця учасників в рейтинговому списку.

Кожен конкурс складається з набору елементів, які роблять його унікальним та утворюють категорію конкурсу:

- сесія - тести проводяться один раз на сезон (осінній, зимовий, весняний чи літній);
- рік - рік проведення конкурсу;
- клас - клас учасника конкурсу у школі.

Для даного проєкту було налаштовано вбудований у Django автоматичний багатофункціональний інтерфейс адміністратора, який допомагає легко взаємодіяти з інформацією, наявною у базі даних. Форми для цього створюються на основі визначених моделей. Також додатково зберігається історія змін із вказаними датою, часом та іменем адміністратора. Є можливість змінювати вигляд інтерфейсу адміністратора, а також доповнення власним функціоналом у разі потреби.

Саме через панель адміністратора, до якої у звичайних користувачів немає доступу, відбувається створення конкурсів, тестових завдань та відповідей до них, запуск та завершення конкурсів (Рис. 3.3.).

					<i>ДТЕУ 121 06-01.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		39

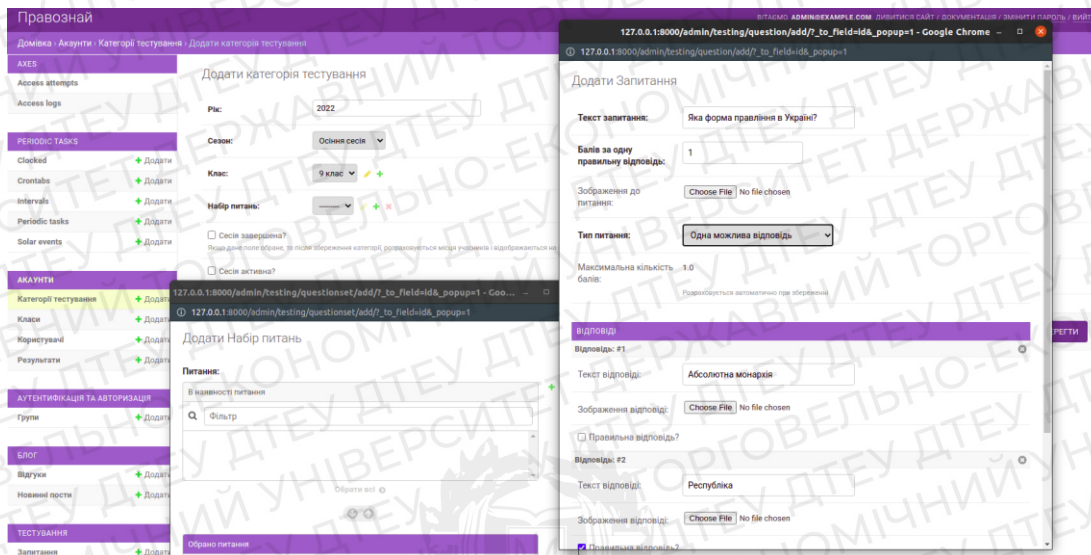


Рис. 3.3. Створення категорії конкурсу в панелі адміністратора

3.1.3 Розробка підсистеми блогу

Останнім елементом, який забезпечує соціальну взаємодію учасників та організаторів є підсистема блогу, він дозволяє адміністраторам вебсайту створювати дописи, новини, таким чином формуючи блог. Окрім цього, кожен учасник має змогу писати відгуки до конкурсів, які вони пройшли (Додаток В). Усі відгуки відображаються тільки після перевірки та затвердження модераторами вебсайту. Модерація відгуків відбувається на панелі адміністратора (Рис. 3.4.).

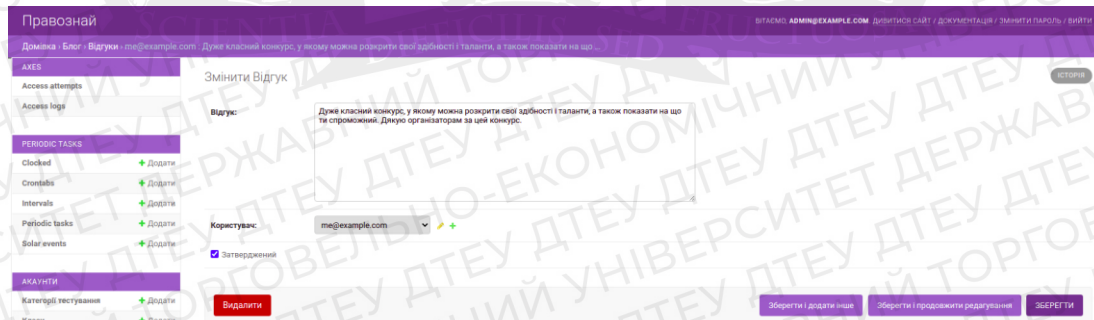


Рис. 3.4. Модерації відгука користувача

3.2 Безпекові рішення

В процесі розробки вебдодатків часто не враховуються питання, пов'язані з захистом цих систем від внутрішніх і зовнішніх небезпек, або не достатньо уваги приділяється даному процесу. Це своєю мірою зумовлює ситуацію, в якій проблеми інформаційної безпеки потрапляють у поле зору власника системи вже після

					Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	40

завершення проекту, а усунення вразливості у вже створеному вебдодатку є більш витратною часткою бюджету, ніж в процесі його розробки та впровадженні [7].

Оскільки розроблюваний додаток потребує збереження даних користувачів, які можуть бути конфіденційними, безпека повинна бути першим пріоритетом при впровадженні будь-яких рішень.

Будь-який сучасний додаток ніколи не зберігає паролів своїх користувачів у чистому вигляді, найбільш популярним підходом, який і був застосований, є використання алгоритмів хешування. Вони покладені змінити вхідний рядок даних, при цьому задовольняючи дві важливі умови:

1. однакові рядки завжди мають один результат;
2. зворотна конвертація неможлива або майже неможлива.

При реєстрації користувачів їх паролі обробляються таким алгоритмом і зберігаються в базі даних у вигляді хешу. При авторизації пароль знову конвертується в хеш і порівнюється з наявним у базі даних. Це унеможливить доступ зловмисників до паролів користувачів, наприклад, при безпосередньому доступі до бази даних.

Існує велика кількість алгоритмів для хешування, серед найпопулярніших - MD5, SHA256, SHA512, був використаний алгоритм Argon2 - переможець конкурсу Password Hashing Competition.

З іншого боку, при кожній реєстрації проводиться валідація паролів користувачів, що додає захист від можливих брутфорс чи інших видів атак, покликаних на отримання доступу до конкретного акаунта. Паролі перевіряються за такими принципами:

- перевірка схожості пароля з набором інших атрибутів, які має даний користувач, наприклад імені чи електронної пошти;
- мінімальна довжина - 9 символів;
- перевірка відсутності пароля у списку популярних паролів (20000 входжень);
- наявність не тільки цифр, а й букв чи символів.

									Аркуш
									41
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-01.БР				

Іншим важливим захисним комплексом є необхідність в підтвердженні власності електронної пошти користувачем, без цього вхід в обліковий запис буде обмежений.

3.3 Тестування та оптимізація

Окрім пакету автоматичних модульних тестів, було проведено обширне тестування відповідності вимогам до проекту, розробленим і визначеним раніше. Відповідно до вимог було створено набір тестових артефактів, які складаються тестову документацію, а саме:

- чек-ліст - список ідей, які необхідно перевірити, з позначками про результат проходження (Рис. 3.5.);
- тест-кейси - покроковий опис конкретної перевірки та очікуваний результат щодо неї (Додаток А);
- тест-план - опис процесу тестування, який визначає учасників команди, їх ролі та розподіл вимог до них, загальне завдання, обсяг роботи та доступні ресурси;
- баг-репорт - звіт про виконання тест-кейсів з детальним описом помилок, кроків для їх відтворення та інших характеристик, таких як важливість та терміновість.

	A	B	C	D	E
1	Ідентифікатор	Перевірка	Результат	Коментар	Ідентифікатор багу
2	CL-1	Перевірити можливість реєстрації	Passed		
3	CL-2	Перевірити можливість авторизації	Passed		
4	CL-3	Перевірити можливість оновлення інформації з профілю	Passed		
5	CL-4	Перевірити можливість зміни пароля облікового запису	Passed		
6	CL-5	Перевірити можливість перегляду інформації про тестування	Passed	Вимагає позначення користувача таким, що оплатив тест, вручну	
7	CL-6	Перевірити зміну курсору на pointer та зміну кольору	Passed		
8	CL-7	Перевірити зміну курсору при наведенні на текстове виділення	Passed		
9	CL-8	Перевірити зміну курсору на pointer при наведенні на клікабельні елементи	Passed		
10	CL-9	Перевірити адаптивність на коректну роботу	Passed		
11	CL-10	Перевірити швидкість роботи	Passed		
12	CL-11	Перевірити відповідність кнопок меню сторінкам	Passed		
13	CL-12	Перевірити можливість залишати відгуки	Passed		
14	CL-15	Перевірити коректність відображення карти	Passed		
15	CL-16	Перевірити відкриття головної сторінки	Passed		

Рис. 3.5. Чек-ліст

Результатом створених тест-кейсів, їх проходження та аналізу стало написання баг-репорту. Баг-репорт, який відображає наявні помилки програмного забезпечення, було використано для їх виправлення.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-01.БР	
					42	

Іншим важливим етапом в розробці програмного забезпечення є його оптимізація. Оптимізація зі свого боку вимагає детального профілювання. Для цього було використано Django Debug Toolbar, який дозволяє переглядати час завантаження сторінки, SQL-запити, які виконуються для завантаження сторінки, час їх виконання та наявні проблеми (наявність дублікатів - однакових запитів із однаковими параметрами, схожих запитів, які виконуються з різними параметрами).

За допомогою цього можна виявити слабкі сторони програмного коду та виправити їх, збільшивши продуктивність та зменшивши час завантаження сторінок (Рис. 3.6.). Так було виявлено та усунуто дублікати SQL-запитів, а також додано їх кешування.

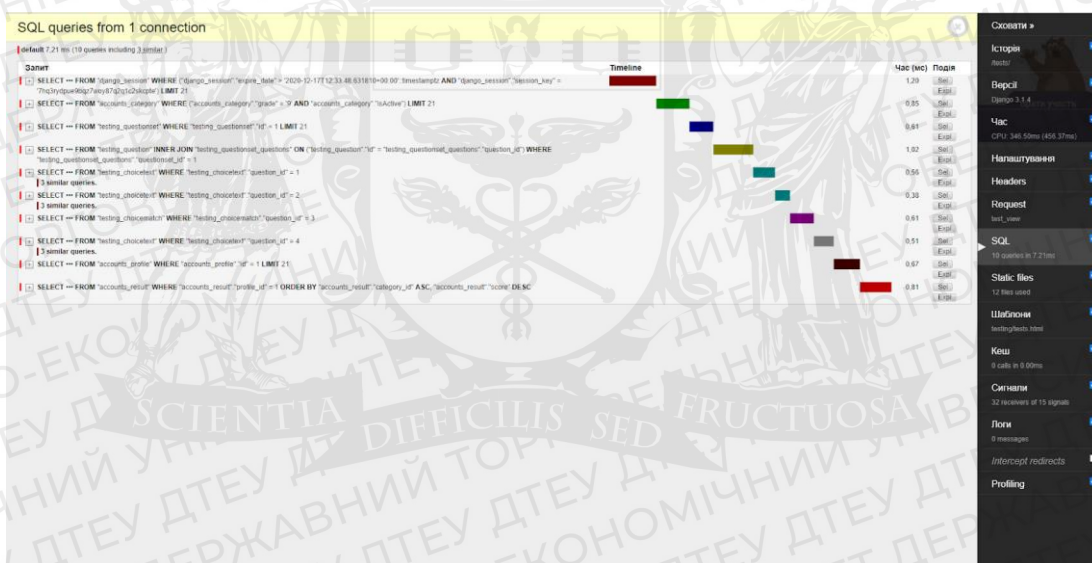


Рис. 3.6. Визначення неоптимізованих запитів з використанням Django Debug Toolbar

Іншим видом тестування є тестування продуктивності (performance testing), для таких цілей було використано програмне забезпечення Locust, яке дозволяє моделювати різні ситуації - визначати типи запитів та сторінки для запитів, кількість користувачів та запитів, які вони роблять за одиницю часу, а також переглядати статистику навантаженості ресурсу (Рис. 3.7.).

						ДТЕУ 121 06-01.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			43



Рис. 3.7. Тестування продуктивності з використанням Locust

Тестування продуктивності дозволяє перевірити наступні елементи:

- швидкість - чи достатньо швидко сервер відповідає на запити;
- розширюваність - максимальна кількість користувачів, яку програмне забезпечення може витримати;
- стабільність - чи достатньо стабільним є програмне забезпечення при різних рівнях навантаження.

В ході тестування було визначено слабкі місця та максимізовано ресурс кожного з визначених компонентів.

В ході розробки було налаштовано два основних середовища - розробки та виробниче. Серед основних відмінностей - у середовищі розробки використовується режим налагоджування Django проекту (debug), у такому режимі значно зменшується кількість різноманітних перевірок, а також відображається додаткова інформація для розробника, наприклад, детальна сторінка про помилку з мета-даними про оточення, також в режимі розробки Django запам'ятовує кожен SQL-запит. Так, у режимі розробки відображаються дані, які не повинні бути доступними кінцевому користувачу, а також використовується додаткова пам'ять, що не є ефективним для виробничого середовища.

3.4 Висновки до розділу 3

Отже, в ході розробки було технічно реалізовано поставлені завдання та спроектовані частини програмного забезпечення. Впроваджено основні частини кожної з визначених підсистем: авторизації, тестування та блогу, що забезпечує функціонування цілісної системи конкурсів, де користувачі можуть створювати

облікові записи, вирішувати тестові завдання, а також брати участь у соціальному житті вебсайту.

Окрім цього, було проведено глибоке тестування системи із різних боків: модульне тестування, тестування продуктивності, визначено місця, які працюють у розріз з вимогами та виправлено їх.



									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>ДТЕУ 121 06-01.БР</i>				45

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Виконуючи роботу, було створено цілісну систему - платформу для проведення Всеукраїнського конкурсу з правознавства. Для цього було пройдено через усі необхідні етапи для успішної розробки програмного забезпечення, а саме:

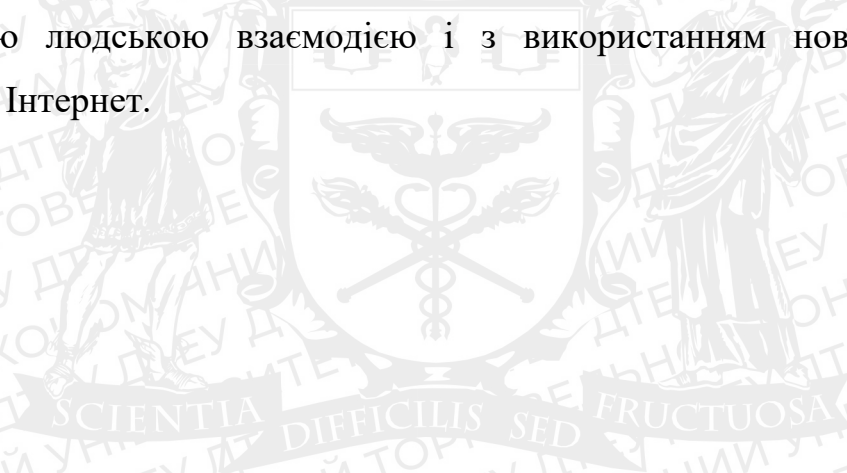
- визначено об'єкт, предмет та завдання, виконання якого потрібно було забезпечити;
- проаналізовано конкурентні програмні забезпечення, створено портрет потенційного користувача;
- визначено список вимог, яким повинна відповідати кінцева система;
- спроектовано архітектуру програмного забезпечення з використанням UML-діаграм, було створено ряд структурних діаграм, поведінки та взаємодії;
- спроектовано базу даних, створено концептуальну, логічну та фізичну моделі;
- спроектовано користувацький інтерфейс, розроблено каркас інтерфейсу, власне інтерфейс та його прототип, а також версію, адаптовану для мобільних пристроїв;
- впроваджено проекти кожної з частин технічно, реалізовано бекенд та фронтенд вебсайту, налаштовано базу даних;
- проведено тестування та оптимізацію.

Звичайно, розроблене програмне забезпечення підлягає подальшій модернізації, а саме:

					<i>ДТЕУ 121 06-01.БР</i>			
					<i>Онлайн-платформа Всеукраїнського учнівського конкурсу</i>	<i>Стадія</i> <i>ВП</i>	<i>Аркуш</i> <i>46</i>	<i>Аркушів</i> <i>49</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		25.04.22				
Керівник		Котенко Н. О.		25.04.22				
Гарант		Рзасва С. Л.		25.04.22	<i>Висновки та пропозиції</i>	Факультет інформаційних технологій, 4 курс, 6 група		
Розроб		Александров А. Ю.		25.04.22				

- впровадження автоматичної системи оплати, що могло б ще більше зменшити взаємодію організатор і користувачів, підвищити ефективність проведення тестування;
- створення можливості для авторизації з використанням різних сторонніх додатків, наприклад, соціальних мереж, це могло б значно прискорити процес створення облікового запису користувачами і відповідно покращити їх користувацьких досвід;

Таким чином, можна стверджувати що поставлена мета була досягнена, а завдання - виконані. Розроблений вебсайт дозволяє з максимальною зручністю створювати конкурси з тестовими завданнями організаторами з одного боку та проходити ці тести учасниками з іншого боку, все це відбувається із забезпеченою мінімальною людською взаємодією і з використанням новітніх технологій, включаючи Інтернет.



					<i>ДТЕУ 121 06-01.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		47

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кульчак Л. С. Методичні рекомендації щодо розробки та використання тестових завдань / Л. С. Кульчак, В. В. Шпіляревич. – Івано-Франківськ: Навчально-науковий юридичний інститут, 2020. – 17 с.
2. Вимоги до сучасного програмного забезпечення систем моніторингу навколишнього середовища об'єктів нафтогазової галузі / О. В. Храбатин, Р. І. Храбатин, Г. В. Григорчук, Л. В. Саманів. – Івано-Франківськ, 2011. – 103 с.
3. Адамик О.В. Бази і сховища даних – інформаційний фундамент бухгалтерського обліку та аналізу // Економічні, управлінські, правові та інформаційно-технічні проблеми діяльності підприємств: колективна монографія/ за заг. ред. Л. М. Савчук, М. Фіц.–Дніпро: Герда, 2016. – 528 с. ISBN 978-617-7097-58-6. – С. 330-341
4. Григоров Г. Тренд дистанційної освіти: проблеми та перспективи впровадження / Г. Григоров. – Дніпро: Дніпровська академія неперервної освіти, 2021. – 103 с. – (т. 3).
5. Гудзовата О. О. Використання уніфікованої мови візуального моделювання UML (Unitified Modeling Language) як інструменту підтримки проектування інформаційних систем / О. О. Гудзовата, В. І. Костирко, І. В. Артищук. – Львів, 2019. – 114 с.
6. Гриценко В. Г. Використання системи управління версіями Git для організації командної роботи над ІТ проектом / В. Г. Гриценко, О. М. Подолян. – Київ, 2014. – 263 с.

					<i>ДТЕУ 121 06-01.БР</i>					
					<i>Онлайн-платформа Всеукраїнського учнівського конкурсу</i>			<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				<i>СВД</i>	<i>48</i>	<i>49</i>
Зав. каф.		Криворучко О.В.		13.12.21						
Керівник		Котенко Н. О.		13.12.21						
Гарант		Рзасва С. Л.		13.12.21						
Розроб		Александров А. Ю.		13.12.21	<i>Список використаних джерел</i>			Факультет інформаційних технологій, 4 курс, 6 група		

7. Бондар В. О. Актуальні проблеми безпеки веб-додатків / В. О. Бондар,
В. В. Гринько. – Відень, 2021. – 140 с.



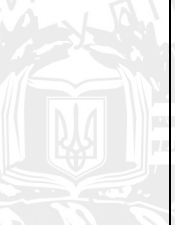
					<i>ДТЕУ 121 06-01.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		49

ДОДАТКИ

Додаток А

Фрагмент тест-кейсів

Ідентифікатор	Пріоритет	Номер вимоги	Модуль	Підмодуль	Попередні умови	Кроки
ТС-1	High	5	Автентифікація	Реєстрація		Невідповідність паролів 1. Відкрити сайт та перейти на сторінку з формою реєстрації 2. Ввести різні значення в поля "Пароль" та "Повторіть пароль". Заповнити всі інші поля 3. Натиснути кнопку підтвердження реєстрації
ТС-2	Low	5	Автентифікація	Реєстрація		Вибір Києва в полі "Область" 1. Відкрити сайт та перейти на сторінку з формою реєстрації 2. В полі "Область" обрати "м. Київ"
ТС-3	Highest	5	Автентифікація	Реєстрація		Некоректний формат email'у 1. Відкрити сайт та перейти на сторінку з формою реєстрації 2. В поле "Email" ввести "abcde". Заповнити всі інші поля 3. Натиснути кнопку підтвердження реєстрації
ТС-4	Highest	5	Автентифікація	Реєстрація		Підтвердження електронної

					<p>пошти</p> <ol style="list-style-type: none"> 1. Відкрити сайт та перейти на сторінку з формою реєстрації 2. Заповнити усі поля коректними даними 3. Натиснути кнопку підтвердження реєстрації 4. Перейти за посиланням з електронної пошти, вказаної при реєстрації
ТС-5	Highest	6	Автентифікація	Авторизація	<p>Авторизація</p> <ol style="list-style-type: none"> 1. Відкрити сайт та перейти на сторінку з формою реєстрації 2. Зареєструватися, активувати електронну пошту 3. Перейти на сторінку з авторизацією та ввести електронну пошту та пароль, використані при реєстрації
ТС-6	Highest	6	Автентифікація	Авторизація	<p>Авторизація з невірним паролем</p> <ol style="list-style-type: none"> 1. Відкрити сайт та перейти на сторінку з формою реєстрації 2. Зареєструватися, активувати електронну пошту 3. Перейти на сторінку з

						авторизацією та ввести електронну пошту, використану при реєстрації та інший пароль
--	--	--	--	--	--	---



Лістинг Dockerfile файлу для середовища розробки

```
FROM python:3
```

```
WORKDIR /usr/src/app
```

```
ENV PYTHONDONTWRITEBYTECODE 1
```

```
ENV PYTHONUNBUFFERED 1
```

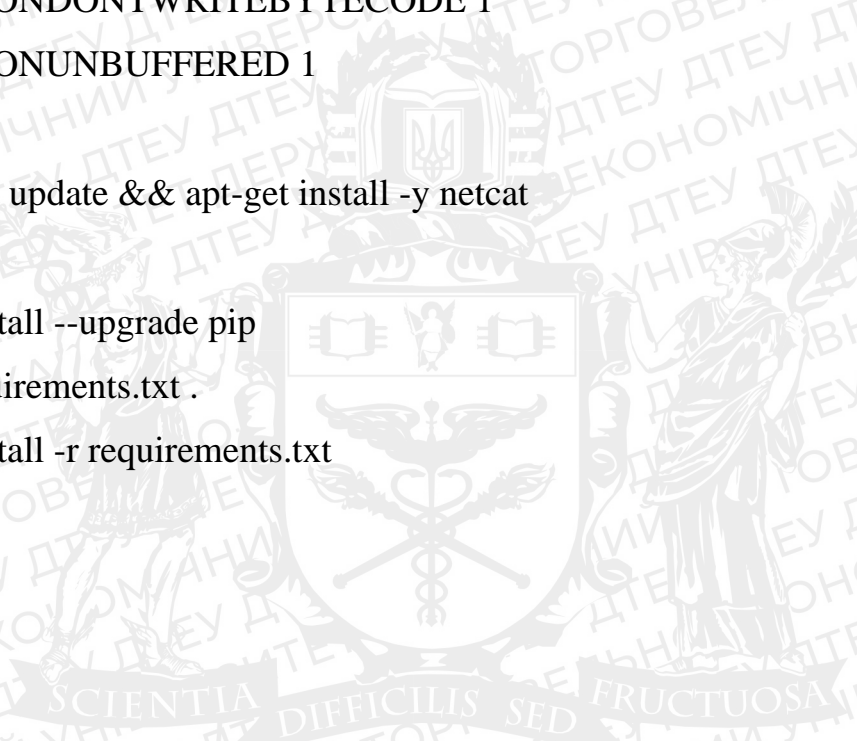
```
RUN apt-get update && apt-get install -y netcat
```

```
RUN pip install --upgrade pip
```



```
COPY ./requirements.txt .
```

```
RUN pip install -r requirements.txt
```

```
COPY ...
```



Користувачький інтерфейс сторінки для написання відгуків

Всеукраїнський конкурс з правознавства **ПравоЗнай**   **Брати участь**

Відгуки



- Участь у конкурсі
- Дати проведення
- Регистрація
- Нагородження
- Брати участь
- Результати
- Новини та оголошення
- Документи
- Запитання та відповіді
- Контакти
- Відгуки**
- Партнери

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi nec vehicula diam. Vestibulum a consectetur erat.


ПІБ

Ви є

Відгук

Надіслати



SCIENTIA DIFFICILIS SED FRUCTUOSA