

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

_____ грудня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційна система управління навчанням у освітньому закладі»
здобувача групи ІН.м-26 Мойсеєнка Дмитра Вікторовича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

_____ Дмитро МОЙСЕЄНКО
(підпис)

Керівник,
кандидат наук

Галина ОЛЕКСІЄНКО _____
(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук, освітньо-наукової програми «Інформатика»
здобувача групи ІН.м-26 Мойсеєвка Дмитра Вікторовича

1. Тема роботи: «Інформаційна система управління навчанням у освітньому закладі»
затверджую наказом по СумДУ від « » грудня 2023 р. № _____
2. Термін здачі здобувачем кваліфікаційної роботи до _____ грудня 2023 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Огляд технологій, що використовуються для навчання, оцінювання здобувачів освіти.
3) Вибір програмних засобів рішення задачі.
4) Проектування і реалізація додатку.
5) Аналіз результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____ Керівник _____
(підпис) (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		

2	<i>Огляд технологій, що використовуються для навчання, оцінювання здобувачів освіти</i>		
3	<i>Вибір програмних засобів рішення задачі</i>		
4	<i>Проектування і реалізація додатку</i>		
5	<i>Аналіз отриманих результатів</i>		
6	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти

(підпис)

Керівник

(підпис)

АНОТАЦІЯ

Записка: 57 стор., 15 рис., 1 додаток, 17 джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є надзвичайно актуальною в сучасному освітньому контексті. Інформаційні системи управління навчанням можуть допомогти ефективніше розподіляти ресурси, включаючи вчителів, час, можуть зробити навчальний контент доступним і віддалено, що особливо важливо в умовах дистанційного навчання, надаючи можливість навчання в будь-якому місці та часі.

Об'єкт дослідження — едьютейнмент-платформа для навчання. Платформа розглядається як складний інформаційно-технічний об'єкт, об'єднуючи в собі аспекти освіти та зрізу знань учнів.

Мета роботи — розробка web-додатку, з використанням фреймворку React, який відповідає потребам освітнього закладу та сприятиме вдосконаленню якості навчання та управління навчанням.

Методи дослідження — в процесі виконання програмної реалізації було застосовано комбінацію технологій такі, як: HTML, CSS, JavaScript, Bootstrap, React, Node.

Результати — проведений аналіз літератури, були проаналізовані альтернативні версії учбових веб-ресурсів на ринку, їх плюси та мінуси. Також був проведений аналіз по вибору методів реалізації даної роботи. Було розроблено алгоритм програми та її реалізація у вигляді web-додатку,, який структурує та оптимізує навчальний процес у освітньому закладі.

ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯМ НАВЧАННЯМ У
ОСВІТНЬОМУ ЗАКЛАДІ

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Дослідження актуальності проблеми.....	8
1.2 Постановка задачі.....	15
2 ВИБІР МЕТОДУ РІШЕННЯ.....	17
2.1 Вибір мови програмування.....	17
2.2 Вибір фреймворку.....	20
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ.....	24
3.1 Проектування користувацької частини.....	24
3.2 Опис програмної реалізації.....	32
3.3 Аналіз результатів.....	34
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	37
ДОДАТОК.....	39

ВСТУП

Актуальність. Сучасний етап розвитку освіти невіддільно пов'язаний із стрімкими змінами у сфері інформаційних технологій, що визначають необхідність впровадження інноваційних підходів до навчання та управління навчальним процесом у навчальних закладах. Інформаційні системи управління навчанням (ІСУН) являються ключовим елементом цього трансформаційного процесу, сприяючи створенню зручного та ефективного середовища для навчання та саморозвитку учнів. Тема дипломної роботи "Інформаційна система управління навчанням у освітньому закладі" актуальна та важлива в контексті сучасних викликів, які стоять перед системою освіти. Розробка веб-додатку для навчання учнів є практичним втіленням ідеї використання сучасних технологій для поліпшення якості навчання та підготовки фахівців в майбутньому.

Об'єкт дослідження. едьютейнмент-платформа для навчання.

Предмет дослідження. Сучасна освітня система потребує активного впровадження інформаційних технологій для оптимізації навчального процесу та підвищення його ефективності. Проте, багато освітніх закладів стикаються з проблемами управління навчанням, координації різноманітних даних та матеріалів, а також забезпеченням доступу до актуальної інформації. Таким чином, виникає необхідність в створенні інтегрованої інформаційної системи, яка забезпечить ефективне управління та навчання.

Гіпотеза. Розробка ІСУН для освітнього закладу має велике значення в контексті покращення якості освіти та підготовки висококваліфікованих фахівців. Нова інформаційна система сприятиме автоматизації процесів, пов'язаних із навчанням, взаємодією викладачів та учнів, а також моніторингом успішності та прогресу кожного учня.

Наукова новизна. Проведення даного дослідження є важливим етапом для впровадження сучасних підходів у навчальний процес. На відміну від існуючих аналогів ІСУН, описане у даній роботі програмне рішення дозволить

вирішити проблеми, пов'язані із розподілом ресурсів, контролем процесу навчання та забезпеченням доступу до актуальної освітньої інформації. В результаті дослідження буде розроблено веб-додаток, який відповідає потребам освітнього закладу та сприятиме вдосконаленню якості навчання та ефективності управління навчанням.

Структура. У подальших розділах дипломної роботи розглядатимуться технічні аспекти розробки веб-додатку, визначення вимог користувачів, вибір технологічного стеку та реалізація функціональності системи, яка враховує сучасні тенденції у сфері освіти та інформаційних технологій. Дане робота складається зі вступу, аналізу публікацій, постановки задачі дослідження, вибір методики та інструментів для рішення поставленої проблеми, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

Історія електронного навчання виникла на початку 20-го століття, коли в Сполучених Штатах Америки був розроблений аналоговий комп'ютер для симуляції польотів на літаках. Поява персональних комп'ютерів в середині 80-х років суттєво вплинула на розвиток програмного забезпечення, внаслідок чого виникли компанії, спеціалізовані на освітньому програмному забезпеченні. У світі було розроблено тисячі програм у цій області, зокрема в Україні, де початки електронного навчання в 90-х роках пов'язані зі створенням перших програм для взаємодії між студентами та викладачами віддалено.

В 1992 році в деяких великих університетах України були впроваджені перші прототипи систем дистанційної освіти після введення концепції дистанційної освіти. Однак у той час рівень комп'ютеризації в Україні був досить низький, і електронну освіту часто розглядали як форму заочного навчання. Зростання ринку бізнес-освіти сприяло розвитку електронного навчання, оскільки компанії, які надають бізнес-навчання, визнали його ефективність і перейшли на електронний формат.

На сьогоднішній день електронна освіта широко використовується у всьому світі. У США понад 90% вищих навчальних закладів, шкіл та компаній із значною кількістю працівників використовують цю форму навчання. У порівнянні з міжнародними стандартами, розвиток ринку електронного навчання в Україні, за експертними оцінками, відстає на 5-7 років.

Електронне навчання можна розділити на два основних сегменти:

Розробника програмного забезпечення:

Розробники, які спеціалізуються на створенні програмного забезпечення для електронного навчання. В цей сегмент входять компанії, які працюють у таких напрямках:

Створення систем управління навчанням (СУН) та систем управління вмістом навчання (СУНК).

Розробка інструментів для організації та проведення вебінарів або віртуальних класів.

Створення засобів для розробки електронних курсів та навчального контенту (системи розробки курсів).

Розробники електронних курсів і навчального контенту:

Компанії, що спеціалізуються на створенні електронних курсів та наповненні їх навчальним контентом. Цей сегмент займає меншу частку ринку і включає в себе розробників, які створюють архіви матеріалів, лекцій, тестів і іншого змісту для електронних курсів.

Поділ на сегменти є умовним, оскільки багато компаній працюють у всіх аспектах електронного навчання, тоді як інші можуть концентруватися лише на одному чи двох аспектах.

Система Управління Навчанням (СУН):

СУН виступає як платформа для реалізації електронного навчання. Ця платформа включає в себе:

- Інтерфейс для взаємодії між викладачем та учнями.

- Доступ до навчального порталу, що є центральною точкою для постачання навчального контенту.

- Інструменти для створення навчальних програм, контролю за їх проходженням та формування звітів.

- Управління комунікацією між студентами та викладачами.

- Керування учнями, включаючи реєстрацію та контроль доступу.

Система Управління Навчальним Контентом (СУНК):

СУНК фокусується на управлінні змістом навчального матеріалу, але не на самому процесі навчання. Це середовище орієнтоване на розробників, які створюють та управляють навчальним вмістом, зокрема через СУН. СУНК дозволяє розробникам створювати і зберігати навчальний контент, редагувати його та вести колективну роботу.

Ці два сегменти разом створюють ефективну систему електронного

навчання, об'єднуючи технологічний аспект (СУН), процеси навчання та управління (СДО) та розробку контенту (СУНК).

E-learning має ряд позитивних аспектів як для традиційного, так і для дистанційного навчання. По-перше, електронне навчання забезпечує легкий доступ до освітніх ресурсів. Якщо студент пропускає заняття з будь-якої причини, він може знайти матеріали у вигляді презентацій чи відеозаписів уроків для самостійного вивчення вдома.

Деякі вчителі використовують онлайн-трансляції для забезпечення можливості навчання учнів, незалежно від їхнього місця перебування. Це розширює можливості освітнього процесу та надає вільність вибору як для вчителя, так і для кожного учня.

Друга важлива перевага e-learning полягає в гнучкості організації навчального процесу. У випадку неможливості проведення уроків учитель та студенти можуть організувати заняття дистанційно. Викладач може проводити уроки в реальному часі, наприклад, за допомогою Zoom, або розміщувати записи уроків та текстові матеріали на спеціальних ресурсах, таких як LMS-платформа EDU від SendPulse, YouTube, Google Drive та інші.

Третій плюс електронного навчання - широкий охоплення. Зручний доступ до освітнього процесу дозволяє студентам обирати форму, місце і навіть час навчання. Технології e-learning дозволяють навчати та тестувати знання тисяч студентів одночасно.

Іншою важливою перевагою e-learning є можливість практикувати отримані знання та навички за допомогою симуляторів, додатків та тестів.

Однак, незважаючи на всі ці позитивні сторони, електронне навчання має свої недоліки. Зокрема, учні, що обрали дистанційну форму навчання, часто відчують відсутність особистого контакту. Відсутність прямого взаємодії з вчителем може призводити до зниження мотивації та успішності у навчанні.

Іншою проблемою є технічні вимоги та навички. Для того щоб бути

вчителем у форматі онлайн, потрібний доступ в Інтернет, технічне обладнання та вміння користуватися ним. Студентам також необхідне стабільне Інтернет-з'єднання та різні пристрої, такі як смартфони, планшети, комп'ютери чи ноутбуки для доступу та завантаження навчальних матеріалів.

E-learning може виступати як ефективним інструментом освіти, так і суттєвою перешкодою на шляху до здобуття знань.

Якість Освітніх Матеріалів:

Перший та найголовніший принцип ефективного e-learning. Інформація повинна бути подана послідовно, а всі матеріали (фото, відео, аудіозаписи, конспекти, презентації) мають бути інформативними, зрозумілими та високої якості.

Мультиформатність:

Допомагає підтримувати інтерес студентів незалежно від складності теми та дозволяє краще засвоювати інформацію. Використання різних форматів, таких як аудіозаписи, тексти, відеозаписи та презентації, гармонійно чергуються.

Простота та доступність:

Усі навчальні матеріали повинні бути доступні та легкі у використанні. Складність доступу може призвести до низької залученості та втрати інтересу студентів.

Результативність:

Кожен курс повинен мати конкретну мету, яку можна досягти тільки шляхом просування етап за етапом та вирішення проміжних завдань.

Персоналізація:

Якісне електронне навчання передбачає зорієнтованість на цільову аудиторію, врахування рівня підготовки та складності матеріалу для підтримки мотивації та успішності студентів.

Соціалізація:

Принцип взаємодії між студентами та викладачем. Це може включати групи чи чати для обговорень, де студенти обмінюються думками та підтримують один одного. Соціалізація підвищує мотивацію та забезпечує зворотний зв'язок для поліпшення якості курсу.

Дистанційна освіта та використання дистанційних освітніх технологій набирає все більше популярності. У "Концепції розвитку дистанційної освіти в Україні" дистанційне навчання визначається як система технологій, що забезпечує ефективну доставку вивченого матеріалу учням та дозволяє особам проводити самостійну роботу для освоєння навчального матеріалу [1]. Однак існують певні проблеми, з якими доводиться зіткнутися.

Перша проблема - це самоосвітні здібності учнів. Дистанційне навчання передбачає, що більшість навчального матеріалу учні опановують самостійно. Для вчителів виникає завдання як організувати та контролювати цю самостійну роботу.

Друга проблема - індивідуалізація навчання. Це важливий принцип дистанційного навчання, що надає можливість кожному обирати темп та ритм навчання. Виникає питання, як це забезпечити та контролювати.

Третя проблема - відсутність єдиної уніфікованої електронної платформи для навчання. Відсутність системності та структурованості у вивченні матеріалу призводить до складнощів, оскільки вчителі можуть використовувати різноманітні онлайн-сервіси, а учням доводиться пристосовуватися до різних платформ.

Наприкінці, тести та оцінювання в сучасному навчанні невід'ємно пов'язані. Для цього використовуються різні сервіси для побудови тестів, такі як Google Forms, Proprof. Також важливим напрямком є використання мобільних сервісів, наприклад, Kahoot!, який дозволяє не лише створювати тести, але і проводити онлайн вікторини для учнів.

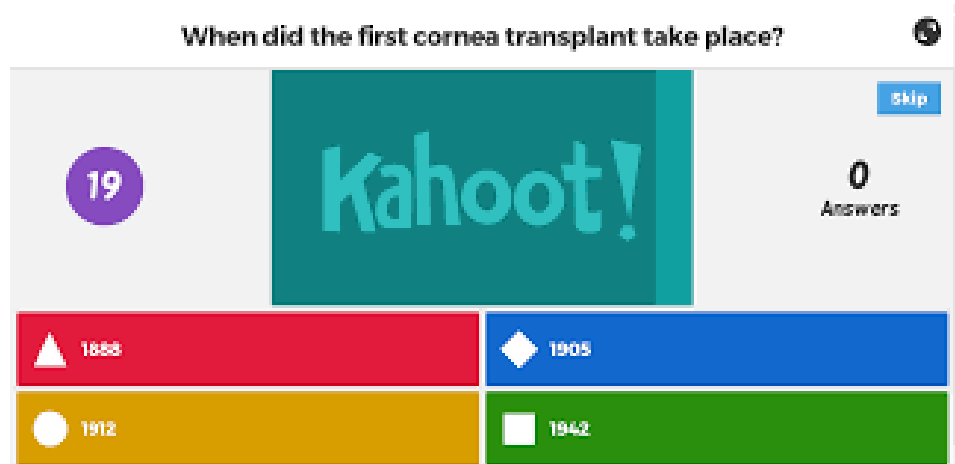


Рисунок 1.1 – Тестування за допомогою Kahoot!

2. Quizlet - це сервіс для створення тестів, в яких вибирається правильна відповідь із запропонованих варіантів.

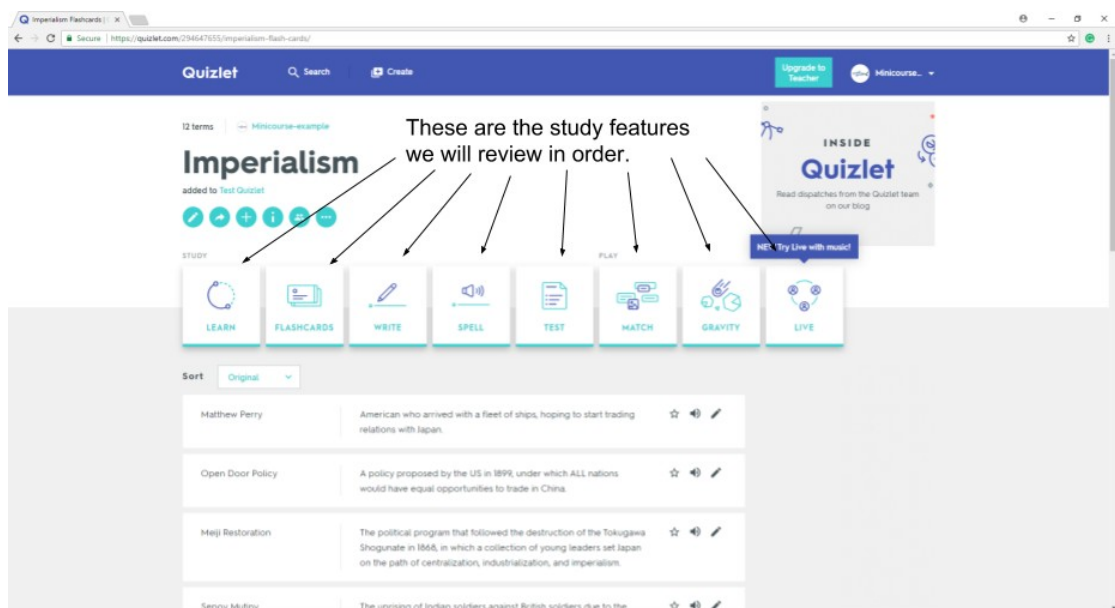


Рисунок 1.2 – Тестування за допомогою Quizlet

Важливо відзначити, що в сучасному світі дистанційне навчання реалізується за допомогою різноманітних технологій, які різняться за формою подання навчальних матеріалів, організацією контролю учбового процесу, та ступенем інтеграції звичайних методів ведення освітнього процесу тощо [3]. Для здійснення дистанційного навчання учнів можна використовувати різні засоби, такі як електронна пошта, Google Диск, соціальні месенджери, веб-

сайти тощо. Наприклад, використовуючи електронну пошту, вчитель висилає учням навчальні матеріали, посилання на онлайн-ресурси та завдання. Учні самостійно опрацьовують матеріали та виконують завдання, а потім надсилають виконану роботу на електронну адресу вчителя.

Можливі рішення: Розробка легкодоступних додатків для навчання, які можна використовувати на смартфонах або планшетах, дозволить учням отримати доступ до матеріалів і завдань, навіть якщо вони мають обмежений доступ до комп'ютерів, створення ресурсів, які структурують матеріал. Враховуючи ці аспекти, створення, оптимізація та вдосконалення веб-додатків для навчання, які суворо структуровані, в навчальному закладі може значно полегшити процес навчання та забезпечити більш ефективну освіту для учнів.

Висвітлюючи різноманітні проблеми та виклики, які стоять перед сучасними освітніми системами, цей розділ дипломної роботи формулює основні завдання, які будуть вирішуватися у процесі розробки інформаційної системи управління навчанням.

1.2 Постановка задачі

Розробити web-додаток для Міжшкільного навчального комбінату, який надасть учням 10-11 класів можливість систематичного вивчення лекційного матеріалу та проходження атестацій для здобуття навичок у обраній ними професії. Дане дослідження виконується в рамках в рамках кваліфікаційної роботи магістра за спеціальністю 122 «Комп'ютерні науки».

Розроблений програмний продукт має задовольняти наступні вимоги:

- Можливість завантаження, перегляду та взаємодії з лекційним матеріалом для кожного предмету.
- Реалізація системи тестування для кожної обраної професії
- Додавання розділу з інформацією про Міжшкільний навчальний комбінат, його викладачів та освітні програми.
- Додавання розділу для відображення матеріалів по кожній професії (галереї для кожної обраної професії та коротка інформація про неї)

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- 1) Вивчення потреб користувачів та визначення детальних вимог до функціоналу додатку.
- 2) Розробка дизайну та структури web-додатку з урахуванням зручності використання та естетичності.
- 3) Розробка модулів для завантаження та вивчення лекцій, проведення тестувань та атестацій.
- 4) Створення системи реєстрації та авторизації користувачів.
- 5) Впровадження функціоналу для кожної з обраних професій

Ці задачі спрямовані на створення високоефективного та користувацького дружнього веб-додатку, який надасть учням можливість

отримати якісну освіту та розвивати свої професійні навички в обраних професіях.

2 ВИБІР МЕТОДУ РІШЕННЯ

2.1 Вибір мови програмування

Кожна мова програмування має свої переваги та недоліки, і вибір повинен враховувати конкретні вимоги вашого проекту, команди розробників, а також тенденції ринку. Важливо також враховувати нові технології та тренди, які можуть впливати на вибір мови програмування у майбутньому.

Згідно з даними, що викладені у DOU [4] результати щорічного опитування щодо мов програмування. (рис.2.1).

Мова	Пишете для роботи зараз, %	Зміни	Пишете для роботи зараз	Використовуєте як додаткові	Використовуєте в своїх проектах	Індекс вподобання
JavaScript	19,1	+0,56 ▲	1695	3186	1012	51,6%
Java	14,0	0,00 ■	1241	851	491	72,4%
Python	13,4	+2,34 ▲	1190	1178	612	73,8%
C#	13,3	-1,21 ▼	1179	533	440	88,1%
TypeScript	13,3	+3,06 ▲	1179	2110	611	80,5%
PHP	7,1	-3,34 ▼	627	423	252	59,0%
Kotlin	3,2	-0,11 ▼	286	272	159	88,4%
C++	2,8	-0,41 ▼	251	321	256	67,3%

Рисунок 2.1 – Топ 5 мов програмування за даними DOU

Цього року найпопулярнішою мовою залишається JavaScript — 19% розробників пишуть нею комерційні проекти. Далі — Java (14%) і Python (13%). Остання вперше серед найпопулярніших. З мінімальним відривом за Python йдуть C# і TypeScript.

JavaScript - це високорівнева, інтерпретована мова програмування, яка використовується головним чином для розробки динамічних веб-сайтів.

JavaScript є стандартною мовою для веб-розробки та має велику кількість бібліотек та фреймворків, що робить його універсальним і дозволяє використовувати його як на клієнтській, так і на серверній стороні. Підтримка асинхронного програмування дозволяє виконувати операції без блокування інтерфейсу користувача, що особливо важливо для веб-розробки. Зручний синтаксис та велика кількість готових рішень дозволяють розробникам швидко

побудувати прототипи та вирішувати завдання. Інтерпретована, динамічна типізація та можливість змінювати об'єкти та функції в режимі роботи дозволяють забезпечити гнучкість коду. Всі розробники мають знати JavaScript так чи інакше. Неважливо, чи ти експерт JavaScript і ти його постійно використовуєш в роботі, чи ти просто вивчаєш її на елементарному рівні й зустрічаєшся з нею лише зрідка, — в будь-якому випадку ти маєш знати її основні концепції та розуміти принципи роботи.

Java - це високорівнева, об'єктно-орієнтована мова програмування, яка створювалася для забезпечення написання коду, який може працювати на будь-якій платформі, що підтримує віртуальну машину Java (Java Virtual Machine або JVM). Основна перевага Java - це здатність працювати на будь-якій платформі без необхідності перекомпіляції. Це досягається завдяки використанню віртуальної машини Java (JVM). Java побудована на основі об'єктно-орієнтованої парадигми, що сприяє створенню модульного та підтримуваного коду. Java підтримує многозадачність на рівні мови програмування та включає в себе вбудовані засоби для одночасного виконання багатьох завдань. Java була розроблена з особливим акцентом на безпеку. Відсутність вказівок на конкретну пам'ять та інші механізми дозволяють уникати багатьох типів програмних помилок.

Python - це високорівнева, інтерпретована, об'єктно-орієнтована мова програмування, яка визначається своєю простотою читання та чіткою синтаксичною структурою. Python має чистий та простий синтаксис, який допомагає розробникам вирішувати задачі меншим обсягом коду порівняно з іншими мовами. Використовується у різних областях, включаючи веб-розробку, наукові обчислення, штучний інтелект, машинне навчання, автоматизацію та інші. Має велику та активну спільноту розробників. Це означає, що ви зможете знайти велику кількість ресурсів, бібліотек та фреймворків для ваших проектів.

Підтримує модульність та можливість використовувати сторонні пакети. Це

дозволяє легко розширювати функціональність вашого коду. Python може працювати на різних платформах, забезпечуючи переносимість коду між операційними системами. Python почала користуватися популярністю протягом останніх років. В першу чергу, це сталося тому, що західні університети припинили викладати складні мови новачкам. Натомість навчальні заклади запропонували студентам вивчати Python, що спровокувало його широке використання у різних галузях, у тому числі у backend розробці.

C# (C-Sharp) - це об'єктно-орієнтована мова програмування, розроблена корпорацією Microsoft. Вона є частиною технологій .NET і широко використовується для розробки різноманітних додатків, включаючи веб-додатки, мобільні додатки, ігри та корпоративні рішення. C# є основною мовою програмування для платформи .NET, що надає велику бібліотеку класів і фреймворк для швидкої розробки. Мова побудована на основі об'єктно-орієнтованої парадигми, що дозволяє розробникам створювати модульний та легко розширюваний код. C# покладає великий акцент на безпеку програмного забезпечення, надаючи механізми для уникнення пам'ятевих витоків та інших типів помилок.

TypeScript - це сучасна мова програмування, розроблена на основі JavaScript, яка додає статичну типізацію та інші нові функції до мови для полегшення розробки складних програм. TypeScript додає можливість визначати типи змінних, що дозволяє виявляти помилки на етапі компіляції та полегшує роботу над великими проектами. Мова підтримує об'єктно-орієнтовану парадигму програмування, дозволяючи використовувати класи, модулі та інші концепції ООП. Базується на стандарті ECMAScript, тому весь валідний код JavaScript є валідним кодом TypeScript.

Критерії порівняння мов програмування для розробки web-додатку можуть включати наступні (Таблиця 2.1):

- Типізація;

- Призначення;
- Обробка помилок;
- Бібліотеки та Фреймворки;

Таблиця 2.1 Порівняння мов програмування за критеріями

Критерії	JavaScript	Java	C#	TypeScript
Типізація	Динамічна	Статична	Статична	Статична
Призначення	Веб-розробка (фронтенд, бекенд)	Мобільні додатки, веб-розробка	веб-розробка	веб-розробка
Обробка помилок	Динамічна	Статична	Статична	Статична
Бібліотеки та Фреймворки	jQuery, React, Angular, Vue	Spring, Hibernate, Apache Struts	Framework, Xamarin	React, Angular, Node.,

За наведеної інформації можна зробити висновок, що JS- це чудовий вибір для багатьох типів проектів, особливо для веб-розробки та побудови сучасних веб-додатків, завдяки можливості бути використаною як на фронтенді, так і на бекенді (за допомогою Node.js). Це дозволяє створювати повноцінні веб-додатки, використовуючи одну мову програмування.

2.2 Вибір фреймворку

Вибір JS-фреймворку є складним завданням, оскільки кожен з них має свої плюси та мінуси. Порівняємо деякі з найбільш популярних:

React. React визначається як бібліотека візуалізації, а не фреймворк, оскільки це лише частина концепції MVC (Model-ViewController), а саме - частина View.

Перший продукт, що використовує Virtual DOM для оптимізації оновлення DOM дерева, дозволяючи ефективніше визначати зміни та зменшувати навантаження на ресурси.

Vue.js. Vue.js представляє собою фреймворк, який забезпечує баланс між обмеженнями та гнучкістю. Його ядро зосереджено на завданнях представлення даних, і воно легко інтегрується в існуючі проекти. Призначений для створення повноцінних SPA застосунків з повним набором функціоналу, включаючи CLI, маршрутизацію, Virtual DOM та швидкий час розробки, аналогічний React.

Angular. Angular - це повноцінний MVC-фреймворк від Google з підтримкою Typescript та Web Workers. Має чітку і жорстку структуру, стабільний і надійний, але вимагає більше ресурсів для написання застосунку. Застосовується частіше в Enterprise-рішеннях через свою потужність та розширені можливості. Враховуючи різні особливості та потреби проекту, вибір між цими фреймворками може залежати від конкретних вимог та уподобань.

Згідно з даними, що викладені у DOU [5] популярність цих трьох фреймворків наведені в таблиці. (рис.2.2).

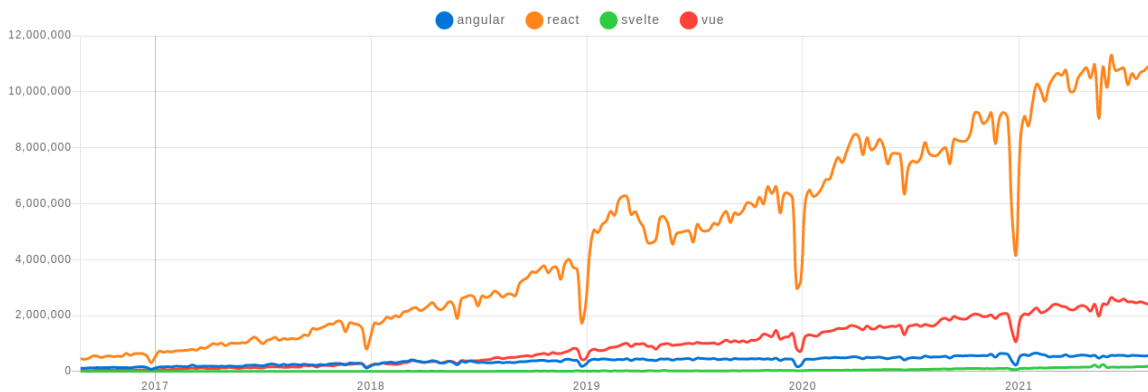


Рисунок 2.2 – Популярність фреймворків JS

Таблиця 2.2 Порівняння фреймворків JS за критеріями


Критерії	React	Angular	Vue.js
Розробник	Facebook		
Мова програмування	JavaScript, JSX	TypeScript, JavaScript	JavaScript
Спосіб роботи	Віртуальний DOM, Компоненти	Реальний DOM, Компоненти,	Віртуальний DOM, Компоненти
Декларативний синтаксис	Так	Так	Так
Продуктивність	Висока, завдяки віртуальному DOM	Висока, оптимізована для великих проектів	Висока, легкий та швидкий
Розширення	Широкий спектр бібліотек та модулів	Модульність, Розширення, Багатофункціональність	Широкий спектр бібліотек та модулів

Існує декілька причин, чому варто обрати React для розробки невеликого веб-додатку [6]:

- використовує віртуальний DOM, що дозволяє ефективно взаємодіяти з реальним DOM та оптимізує рендеринг компонентів. Це призводить до покращення продуктивності та відгуку додатка.

- використовує віртуальний DOM, що дозволяє ефективно взаємодіяти з реальним DOM та оптимізує рендеринг компонентів. Це призводить до покращення продуктивності та відгуку додатка.

- використовує декларативний підхід до опису вигляду інтерфейсу користувача, що полегшує розуміння та модифікацію коду.
- не є повноцінним фреймворком, але це лише бібліотека для створення інтерфейсів. Це означає, що ви вільні вибирати інші бібліотеки та інструменти для вирішення конкретних завдань.
- кросплатформенність.
- присутність інструментів для аналітики та аналізу.
- простота у освоєнні.





1		React A declarative, efficient, and flexible JavaScript library for building user inte...	+18.5k☆
2		Vue.js A progressive, incrementally-adoptable framework for building UI on the ...	+14.3k☆
3		Svelte Cybernetically enhanced web apps	+13.6k☆
4		Angular The modern web developer's platform	+9.3k☆
5		Solid A declarative, efficient, and flexible JavaScript library for building user inter...	+8.5k☆

Рисунок 2.3 – Список найпопулярніших JS фреймворків [7]

Отже, для розробки було обрано інструменти: JS фреймворк React для реалізації бек-енд частини web-додатку, мова програмування Node.js, та Mongo DB для реалізації бази даних.

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Проектування користувацької частини

Ключові аспекти проекту та його функціональність:

- Головна сторінка: Загальна інформація про навчальний заклад, його історія, місія та основні принципи.
- Інформація про вчителів та персонал.
- Сторінки професії: Окремі сторінки для кожної професії з описом, лекційним матеріалом, галереєю зображень та атестаційним тестом.
- Реєстрація користувача.
- Тестування та оцінювання: Тести з питань, що відповідають конкретній професії, автоматичне оцінювання результатів.

Використовуємо ліниве завантаження (Lazy Loading) для оптимізації завантаження сторінок та компонентів. Також використовуєте Redux для збереження стану автентифікації користувача.

```
import { Route, Routes, Navigate } from 'react-router-dom';
import { Layout } from './Layout/Layout';
import { useDispatch, useSelector } from 'react-redux';
import { useEffect, lazy } from 'react';
import { RestrictedRoute } from './RestrictedRoute';
import { Loader } from './Loader/Loader';
import { refreshUser } from '../redux/Authorization/operations';

const Register = lazy(() => import('../pages/Register/Register'));
const Home = lazy(() => import('../pages/Home/Home'));
const Login = lazy(() => import('../pages/Login/Login'));
const Contactlist = lazy(() => import('../pages/ContactList/ContactList'));
const ContactLec = lazy(() => import('../pages/ContactLec/ContactLec'));
const Quizz = lazy(() => import('../pages/Test/Quiz'));

export const App = () => {
  const dispatch = useDispatch();

  useEffect(() => {
    dispatch(refreshUser());
  }, [dispatch]);
}
```

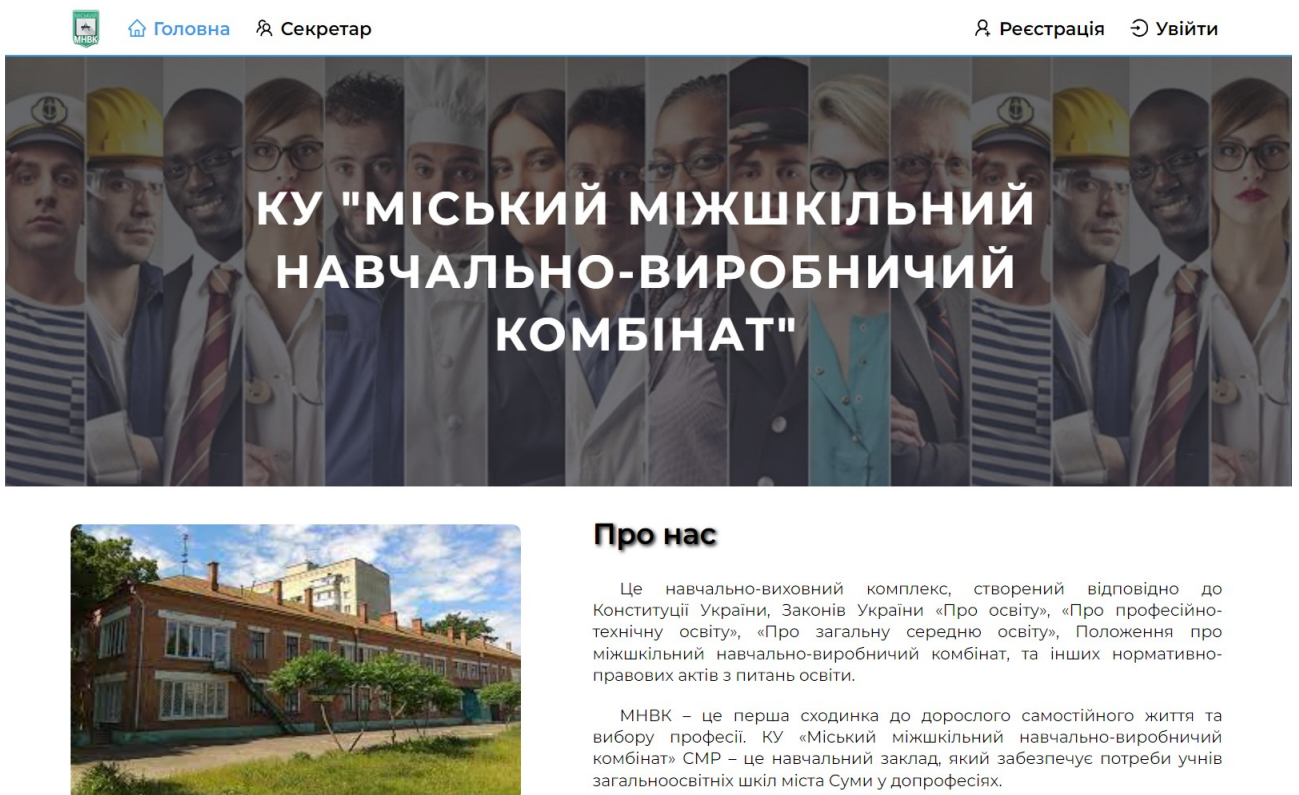
Рисунок 3.1 – Оптимізація завантаження сторінок та компонентів


```

16 export const App = () => {
17   const dispatch = useDispatch();
18
19   useEffect(() => {
20     dispatch(refreshUser());
21   }, [dispatch]);
22
23   const { isRefreshing } = useSelector(state => state.auth);
24
25   return !isRefreshing ? (
26     <>
27       <Routes>
28         <Route path="/" element={<Layout />} />
29         <Route index element={<Home />} />
30         <Route path="/secretary" element={<Contactlist />} />
31         <Route path="/secretary/lec" element={<ContactLec />} />
32         <Route path="/secretary/test" element={<Quizz />} />
33         <Route
34           path="/login"
35           element={
36             <RestrictedRoute redirectTo="/contacts" component={<Login />} />
37           }
38         />
39         <Route path="/register" element={<RestrictedRoute redirectTo="/contacts" component={<Register />} />} />
40         <Route path="*" element={<Navigate to="/" />} />
41       </Route>
42     </Routes>
43   </>
44 ) : (
45   <Loader />
46 );
47 };
48

```

Рисунок 3.2 – Програмна реалізація кореневого компонента App



[Головна](#)
[Секретар](#)
[Реєстрація](#)
[Увійти](#)

КУ "МІСЬКИЙ МІЖШКІЛЬНИЙ НАВЧАЛЬНО-ВИРОБНИЧИЙ КОМБІНАТ"

Про нас

Це навчально-виховний комплекс, створений відповідно до Конституції України, Законів України «Про освіту», «Про професійно-технічну освіту», «Про загальну середню освіту», Положення про міжшкільний навчально-виробничий комбінат, та інших нормативно-правових актів з питань освіти.

МНВК – це перша сходинка до дорослого самостійного життя та вибору професії. КУ «Міський міжшкільний навчально-виробничий комбінат» СМР – це навчальний заклад, який забезпечує потреби учнів загальноосвітніх шкіл міста Суми у допрофесіях.

Рисунок 3.3 – Головна сторінка додатку

Компонент **Home** - це центральний елемент головної сторінки веб-додатку.

В залежності від того, чи авторизований користувач відтворюються ті чи інші компоненти. Розглянути всі компоненти детальніше можливо в Додатку даної ПЗ.

Компонент **Professions** містить відомості про різні професії та їх опис. У цьому компоненті використовується бібліотека Framer Motion для анімації в залежності від прокрутки. Компонент містить заголовок "Професії та профілі", текстовий опис, а також список професій з зображеннями та текстовими описами. Описані ефективні методи використання **useEffect** для слідкування за подією прокрутки та анімації відповідно до видимості елементів.

Професії та профілі

Вибір професії – це одна з важливих складових умов людського щастя, усвідомлення своєї корисності для людей або відчуття «маленької людини» - гвинтика, виконавця чужої долі. Основна проблема для молоді – це пошук свого місця в житті, що тісно пов'язано з обранням майбутньої професії. Вибір професії – це той відповідальний рубіж, на якому оцінка своїх індивідуальних особливостей і можливостей піддається ретельному оцінюванню. Питання профорієнтації завжди було у центрі уваги держави і суспільства. Тому профорієнтаційна підготовка молоді розподіляється між усіма ланками суспільства, відповідальними за її виховання, - загальноосвітніми та професійно-технічними навчальними закладами.

Ми надаємо можливість апанувати сучасні технології, отримати актуальні знання та розвивати навички, які є важливими в сучасному ринку праці. Ви зможете не лише отримати необхідний досвід, але й розвинути свої таланти та знайти власний шлях до успіху у вибраній галузі.

-  Водій автотранспортних засобів (категорії «В», «С»)»
-  Слюсар механоскладальних робіт
-  Секретар керівника (організації підприємства, установи)
-  Основи web-дизайну та web-програмування
-  Основи психологічних знань - система особистісного росту
-  Кухар



Водій



Web-дизайнер



Слюсар

У нас ви навчитесь здійснювати збирання машин і механізмів з деталей, а також займатися ремонтом механоскладних пристроїв.



Психолог



Секретар



Кухар

Рисунок 3.4 – Професії, доступні для вивчення

Компонент **Content** містить інформацію про колектив закладу, а також включає компонент **Slider**. У цьому компоненті Slider використовується для відображення слайдів із зображеннями та текстовою інформацією. Слайди отримуються з масиву, а кнопки `prev` та `next` відповідають за переміщення між слайдами.

Колектив

КУ"ММНВК"СМР укомплектований кваліфікованими педагогічними кадрами. У закладі працює 13 вчителів та майстрів виробничого навчання, вищу освіту мають 10 вчителів та майстрів виробничого навчання, середню спеціальну три майстри виробничого навчання. Вчителів вищої категорії – 2, першої категорії – 5, спеціалістів – 6.



Рисунок 3.5 – Колектив закладу

Використовуємо локальний стейт (`useState`) для зберігання значень полів вводу, обробники подій `handleChange` та `handleSubmit`, які ефективно обробляють зміни у введеннях та сабміт форми відповідно. `Redux` для виклику операцій реєстрації.



Зареєструвати нового користувача

Name

Email

Password

[Зареєструватись](#)

Рисунок 3.6 – Сторінка реєстрації

Аналогічним чином оформлюємо сторінку авторизації.



Увійти

Email

Password

[Увійти](#)

Рисунок 3.7 – Сторінка авторизації



Секретар керівника (організації підприємства, установи)



[Зареєструйтеся](#) або [Увійдіть](#) , щоб користуватися додатком

[Про професію](#)[Галерея](#)[Лес](#)[Test](#)

Рисунок 3.8 – Сторінка професії «Секретар керівника»

Використання стану: використовуємо стан компонента (`showText` та `showGallery`), щоб керувати відображенням тексту про професію та галереї. Це дозволяє нам динамічно змінювати вміст сторінки відповідно до дій користувача.

Обробники подій - обробники подій (`handleLecturesButtonClick` та `handleGalleryButtonClick`) змінюють стан для управління відображенням відповідного контенту.

Використання React Router: використовуємо `<HomeLink>` для навігації на інші сторінки нашого додатку, а саме пропонуємо користувачеві зареєструватися або авторизуватися. Галерея професії: відображаємо галерею на сторінці, використовуючи `map` для перебору шляхів.

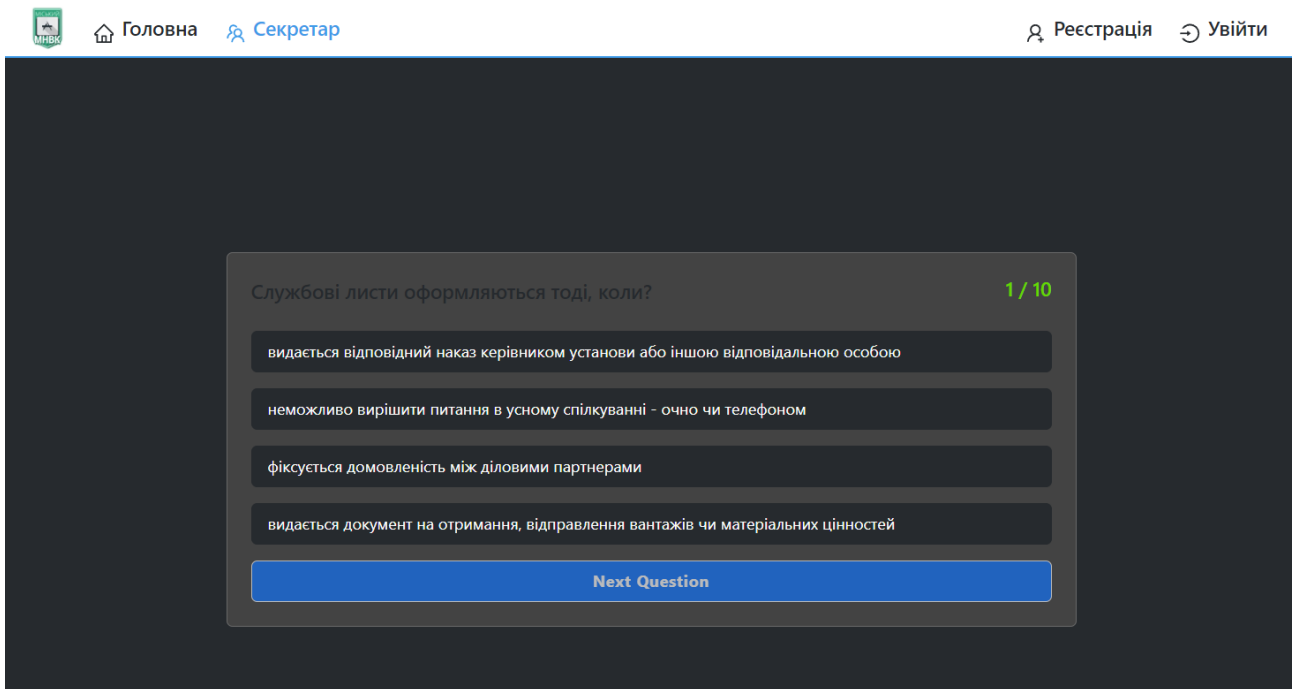


Рисунок 3.9 – Сторінка атестаційного тестування професії «Секретар керівника»

В даній частині додатку використовуємо контекст, який дозволяє передавати дані між компонентами безпосередньо, уникаючи необхідності передавати пропси через багато рівнів вкладеності. Кожен компонент виконує конкретний фрагмент функціоналу тестування, що робить код більш читабельним і підтримуваним.

Компонент Start: Використовує контекст `DataContext` для отримання даних та функцій. Має кнопку "Розпочати тест", яка викликає функцію `startQuiz`, яка розпочинає тестування.

Компонент Quiz: Використовує контекст для отримання даних та функцій. Викликає функцію `checkAnswer` при виборі відповіді, яка перевіряє правильність.

Забезпечує перехід до наступного питання за допомогою функції `nextQuestion`.

Компонент Result: Використовує контекст для отримання даних та функцій.

Відображає результат тестування та повідомлення. Відображає кількість правильних відповідей від загальної кількості питань.

`DataContext`-визначає контекст та надає значення та функції для керування

тестуванням. Включає стан, який зберігає поточне питання, відповіді, показ тесту, кількість правильних відповідей. Надає функції для початку тесту, перевірки відповідей, переходу до наступного питання та відображення результатів.

3.2 Опис програмної реалізації

Короткий огляд використаних інструментів та бібліотек у проєкті:

fontsource/montserrat: Бібліотека для підключення шрифту Montserrat у вашому проєкті. [15]

reduxjs/toolkit: Надмірна бібліотека для роботи з Redux. Надає кілька зручних інструментів для управління станом. . [11]

antd: Ant Design - бібліотека компонентів для React. Містить готові компоненти, які спрощують розробку інтерфейсу. [13]

axios: Бібліотека для виконання HTTP-запитів. Зручний інструмент для роботи з API. [9]

bootstrap: CSS-фреймворк для швидкого і простого створення адаптивних і красивих веб-сайтів. [14]

notiflix: Бібліотека для створення повідомлень та сповіщень.

framer-motion: Бібліотека для створення анімацій в React.

normalize.css: CSS-файл, який нормалізує стилі між різними браузерами.

react: Основна бібліотека React для створення інтерфейсів користувачів.

react-bootstrap: Інтеграція Bootstrap в React. Надає React-компоненти для використання в зручній для React-розробки.

react-dom: Допоміжна бібліотека React для роботи з DOM.

react-icons: Набір React-компонентів для використання популярних значків. [17]

react-loader-spinner: Компонент React для відображення анімованого завантажувача. [10]

react-redux: Офіційна бібліотека для інтеграції React та Redux. [12]

react-router-dom: Бібліотека для навігації між сторінками в React-додатках.

redux-persist: Бібліотека для збереження стану Redux через перезавантаження сторінки.

slick-carousel: Карусельна бібліотека для створення адаптивних слайдерів. [16]

web-vitals: Бібліотека для вимірювання важливих метрик продуктивності веб-сайтів.

styled-components: Бібліотека для написання CSS в JS, що дозволяє вбудовувати стилі безпосередньо в компоненти React. [8]

Ці інструменти і бібліотеки допомагають спростити розробку, надаючи готові рішення для часто зустрічаючихся задач та підтримуючи сучасні практики розробки.

```
13     },
14     "dependencies": {
15       "@fontsource/montserrat": "^5.0.15",
16       "@reduxjs/toolkit": "^1.9.1",
17       "antd": "^5.11.2",
18       "axios": "^1.6.1",
19       "bootstrap": "^5.1.3",
20       "notiflix": "^3.2.6",
21       "framer-motion": "^10.16.5",
22       "normalize.css": "^8.0.1",
23       "react": "^18.2.0",
24       "react-bootstrap": "^2.3.0",
25       "react-dom": "^18.2.0",
26       "react-icons": "^4.11.0",
27       "react-loader-spinner": "^5.4.5",
28       "react-redux": "^8.0.5",
29       "react-router-dom": "^6.19.0",
30       "redux-persist": "^6.0.0",
31       "slick-carousel": "^1.8.1",
32       "web-vitals": "^2.1.4",
33       "styled-components": "^6.1.1"
34     },
```

Рисунок 3.10 – Бібліотеки проекту

3.3 Аналіз результатів

Додаток, який був розроблений з використанням технологій, виявився вкрай функціональним та забезпечив високий рівень користувацької зручності. Використання інструментів, таких як React, Redux, а також інших сучасних пакетів на їх основі, дозволило створити додаток, демонструватиме ефективність, а також має зручний, простий у використанні і зрозумілий дизайн.

Один із аспектів аналізу є високий рівень функціональності. Додаток надає користувачеві доступ до інформації про обрану професію, надаючи зрозумілі та структуровані дані. Застосування анімацій, роботи з HTTP-запитами та інших бібліотек розширює можливості додатку, забезпечуючи сучасні та інноваційні рішення. Дизайн та інтерфейс додатку ретельно продумані, що забезпечує зручність та естетичний вигляд.

Аналіз результатів також відображає ефективність та досягнення поставлених цілей. Додаток успішно впроваджує інноваційні технології в галузі навчання, тестування та атестації знань. Високий рівень стабільності та відповідності сучасним стандартам робить його частиною інструментарію для освітніх потреб. Він створений для полегшення навчальних процесів, надаючи здобувачам освіти та викладачам зручний та ефективний інструмент для досягнення навчальних цілей.

ВИСНОВКИ

Розроблений додаток сприятиме автоматизації процесу, пов'язаних із навчанням, взаємодією викладачів та учнів, оцінюванням, а також моніторингом успішності та прогресу кожного учня. Дане рішення сприятиме покращенню якості навчання та ефективності управління навчанням, а здобувачам професії-полегшать вивчення і здачі предмету, так як все буде в одному додатку у вільному доступі. Ресурс відкриває нові перспективи для автоматизації навчальних процесів, полегшуючи взаємодію між викладачами та учнями. Зосереджений на оптимізації процесів, пов'язаних із навчанням, оцінюванням та моніторингом успішності, додаток спрямований на покращення якості навчання та управління навчанням в цілому.

В ході розробки враховано важливість комфортного та ефективного навчання для учнів, а також спрощення рутинних завдань для викладачів. Було обрано оптимальні інструменти для виконання поставленої мети. Також під час розробки проекту було враховано усі вимоги до програмного продукту. Застосування оптимальних технологій, таких як React, Redux, та інші, дозволило створити додаток, який об'єднує в собі функціональність та зручний інтерфейс.

У ході виконання кваліфікаційної роботи магістра було виконано наступні завдання:

1. Виконано аналіз проблемної області, визначено актуальність розробки системи.
2. Огляд технологій, що використовуються для навчання, оцінювання здобувачів освіти.
3. Обрані технології на розробку програмного продукту.
4. Проектування і реалізація додатку. Структура додатку, компонування елементів, збірка.

5. Проаналізовано отримані результати та зроблено висновки щодо якості роботи інформаційної технології.

Отримані результати підтверджують, що використання розробленого web-додатку значно полегшить навчання, зменшить витрати ресурсів та ефективно впровадить інновації у процес освіти.

За результатами роботи інформаційної системи, можна зробити висновок, що застосування розробленого web-додатку зменшить кількість необхідних ресурсів для навчання та скоротить час на підготовку і оцінку як для здобувачів освіти так і для викладачів. Можна з упевненістю стверджувати, що розроблений додаток (<https://dmytro1117.github.io/mnvk/>) є вагомим внеском у розвиток інноваційного навчання та автоматизації у галузі освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Концепція розвитку дистанційної освіти в Україні року: постанова Міністерства освіти і науки України від 20 грудня 2000 р. URL: <http://www.osvita.org.ua/distance/pravo/00.html>.
2. Організація освітнього процесу із застосуванням технологій дистанційного навчання у 2020/2021 навчальному році: методичні рекомендації / за заг. ред. В.І. Шуляра. Миколаїв: ОППО, 2020. 108 с
3. Толочко В.М. Проблемні аспекти дистанційної форми освіти та можливості її використання в Україні: URL: http://www.provisor.com.ua/archive/2009/N11/padfo_119.php.
4. Рейтинг мов програмування 2023. [Електронний ресурс] – Режим доступу до ресурсу: <https://dou.ua/lenta/articles/language-rating-2023/>
5. Огляд фреймворків JavaScript. [Електронний ресурс] – Режим доступу до ресурсу: <https://dou.ua/forums/topic/34739/>
6. Огляд фреймворків JavaScript. [Електронний ресурс] – Режим доступу до ресурсу: <https://dan-it.com.ua/uk/blog/5-najpopuljarnishih-frejmworkiv-dlja-vebrozrobki/>
7. Топ фреймворков [Електронний ресурс] – Режим доступу до ресурсу: <https://highload.today/top-frejmworkov-i-instrumentov-dlya-javascript-po-itogam-2021-goda-neozhidanno-pobedil-produkt-ot-google/>
8. CSS for the Component. [Електронний ресурс] – Режим доступу до ресурсу: <https://styled-components.com/>
9. Бібліотека Axios. [Електронний ресурс] – Режим доступу до ресурсу: <https://axios-http.com/ru/docs/intro>
10. Бібліотека React-spinner. [Електронний ресурс] – Режим доступу до ресурсу: <https://mhnpd.github.io/react-loader-spinner/docs/intro/>
11. Бібліотека React-toolkit. [Електронний ресурс] – Режим доступу до ресурсу: <https://redux-toolkit.js.org/>

12. Бібліотека React-redux. [Електронний ресурс] – Режим доступу до ресурсу: <https://react-redux.js.org/>
13. Бібліотека Ant design. [Електронний ресурс] – Режим доступу до ресурсу: <https://ant.design/>
14. Бібліотека Bootstrap. [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/docs/5.0/getting-started/download/>
15. Бібліотека fontsource/montserrat. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/package/@fontsource/montserrat>
16. Бібліотека Slick-carousel. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/package/slick-carousel>
17. Бібліотека Slick-carousel. [Електронний ресурс] – Режим доступу до ресурсу: <https://react-icons.github.io/react-icons/>

ДОДАТОК

Серверна частина роботи реалізована за допомогою застосунок Express.js, веб-фреймворк для Node.js. (частина коду, демонструє як реалізована автентифікація користувача: контролер, роут, модель, схема).

```
const express = require("express");
const logger = require("morgan");
const cors = require("cors");
const dotenv = require("dotenv");
const swaggerUi = require("swagger-ui-express");
const YAML = require("yamljs");
dotenv.config();
const authRouter = require("./routes/api/auth");
const testsRouter = require("./routes/api/contacts");

const swaggerDocument = YAML.load("./swagger.yaml");

const app = express();

const formatsLogger = app.get("env") === "development" ? "dev" : "short";

app.use(logger(formatsLogger));
app.use(cors());
app.use(express.json());
app.use(express.static("public"));

app.use("/api-docs", swaggerUi.serve, swaggerUi.setup(swaggerDocument));
app.use("/api/auth", authRouter);
app.use("/api/test", testsRouter);

app.use((req, res) => {
  res.status(404).json({ message: "Not found" });
});

app.use((err, req, res, next) => {
  const { status = 500 } = err;
  res.status(status).json({ message: err.message });
});

module.exports = app;
```

```

const express = require("express");
const validateJoi = require("../middlewares/validationJoi");
const {
  register,
  login,
  curent,
  logout,
  updateSubscription,
  updateAvatar,
  verifyEmail,
  resendVerifyEmail,
} = require("../controllers/authCtrloller");
const authenticate = require("../middlewares/authenticate");

const {
  registerJoiSchema,
  loginJoiSchema,
  subscriptionJoiSchema,
  verrifyEmailJoiSchema,
} = require("../models/userModel");
const multerUpload = require("../middlewares/multerUpload");

const router = express.Router();

router.post("/register", validateJoi(registerJoiSchema), register);
router.get("/verify/:verificationToken", verifyEmail);
router.post("/verify", validateJoi(verrifyEmailJoiSchema), resendVerifyEmail);
router.post("/login", validateJoi(loginJoiSchema), login);
router.get("/current", authenticate, curent);
router.post("/logout", authenticate, logout);
router.patch(
  "/subscription",
  authenticate,
  validateJoi(subscriptionJoiSchema),
  updateSubscription
);
router.patch(
  "/avatars",
  authenticate,
  multerUpload.single("avatar"),
  updateAvatar
);

module.exports = router;

```



```
const { Schema, model } = require("mongoose");
const Joi = require("joi");
const { handleMongooseError } = require("../helpers/handleMongooseError");

const emailRegexp = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*\\.\\w{2,3}+$/;
const typeSubscription = ["starter", "pro", "business"];

const userSchema = new Schema(
  {
    name: {
      type: String,
      required: [true, "Set name for contact"],
    },
    password: {
      type: String,
      required: [true, "Set password for user"],
      minlength: 10,
    },
    email: {
      type: String,
      match: emailRegexp,
      required: [true, "Email is required"],
      unique: true,
    },
    subscription: {
      type: String,
      enum: typeSubscription,
      default: "starter",
    },
    token: {
      type: String,
      default: "",
    },
    avatarURL: {
      type: String,
      required: true,
    },
    verify: {
      type: Boolean,
      default: false,
    },
    verificationToken: {
      type: String,
      required: [true, "Verify token is required"],
    },
  }
);
```

```

    },
  },
  { versionKey: false, timestamps: true }
);

userSchema.post("save", handleMongooseError);

const registerJoiSchema = Joi.object({
  name: Joi.string().required(),
  email: Joi.string().pattern(emailRegexp).required(),
  // email: Joi.string().required(),
  password: Joi.string().min(10).required(),
});

const verifyEmailJoiSchema = Joi.object({
  email: Joi.string().pattern(emailRegexp).required(),
  // email: Joi.string().required(),
});

const loginJoiSchema = Joi.object({
  email: Joi.string().pattern(emailRegexp).required(),
  // email: Joi.string().required(),
  password: Joi.string().min(10).required(),
});

const subscriptionJoiSchema = Joi.object({
  subscription: Joi.string()
    .valid(...typeSubscription)
    .required(),
});

const User = model("user", userSchema);

module.exports = {
  User,
  registerJoiSchema,
  verifyEmailJoiSchema,
  loginJoiSchema,
  subscriptionJoiSchema,
};

const { User } = require("../models/userModel");
const { ctrlWrapperRoutes } = require("../helpers/ctrlWrapperRoutes");
const Conflict = require("http-errors");

```

```

const bcryptjs = require("bcryptjs");
const jwt = require("jsonwebtoken");
const gravatar = require("gravatar");
const fs = require("fs");
const cloudinary = require("cloudinary").v2;
const { nanoid } = require("nanoid");
const { sendEmail } = require("../helpers/sendEmail");

const { SECRET_KEY, BASE_URL } = process.env;

const register = async (req, res) => {
  const { name, email, password } = req.body;
  const user = await User.findOne({ email });
  if (user) {
    throw new Conflict(409, `Sorry, user with ${email} in use`);
  }

  const hashBacrypt = bcryptjs.hashSync(password, bcryptjs.genSaltSync(10));
  const avatarURL = gravatar.url(email);
  const verificationToken = nanoid();

  const result = await User.create({
    name,
    email,
    password: hashBacrypt,
    avatarURL,
    verificationToken,
  });

  const verifyEmail = {
    to: email,
    subject: "Verify email",
    html: `Click verify email</a>`,
  };

  await sendEmail\(verifyEmail\);

  res.status\(201\).json\({
    status: "succes",
    code: 201,
    data: {
      user: {
        email,

```

```

    name,
  },
},
});
};

const verifyEmail = async (req, res) => {
  const { verificationToken } = req.params;
  const user = await User.findOne({ verificationToken });
  if (!user) {
    throw Conflict(401, "Email not found");
  }
  await User.findByIdAndUpdate(user._id, {
    verify: true,
    verificationToken: "",
  });

  res.json({
    status: "succes",
    code: 200,
    data: {
      message: "Email verify success",
    },
  });
};

const resendVerifyEmail = async (req, res) => {
  const { email } = req.body;
  const user = await User.findOne({ email });
  if (!user) {
    throw Conflict(401, "Email not found");
  }
  if (user.verify) {
    throw Conflict(401, "Email already verify");
  }

  const verifyEmail = {
    to: email,
    subject: "Re-verify email",
    html: `Click re-verify email</a>`,
  };

  await sendEmail\(verifyEmail\);
};

```

```

res.json({
  status: "succes",
  code: 200,
  data: {
    message: "Re-verify email send success",
  },
});
};

const login = async (req, res) => {
  const { email, password } = req.body;
  const user = await User.findOne({ email });
  if (!user) {
    throw new Conflict(401, `Sorry, user with ${email} or password is wrong`);
  }

  if (!user.verify) {
    throw new Conflict(401, "Email not verified");
  }

  const passwordCompare = bcryptjs.compare(password, user.password);

  if (!passwordCompare) {
    throw new Conflict(401, `Sorry, user with ${email} or password is wrong`);
  }

  const payload = {
    id: user._id,
  };

  const token = jwt.sign(payload, SECRET_KEY, { expiresIn: "5h" });
  await User.findByIdAndUpdate(user._id, { token });

  res.json({
    status: "succes",
    code: 200,
    data: {
      token,
    },
  });
};

const curent = async (req, res) => {

```

```

const { name, email } = req.user;
res.json({
  status: "succes",
  code: 200,
  data: {
    user: {
      name,
      email,
    },
  },
});
};

```

```

const logout = async (req, res) => {
  const { _id, name } = req.user;
  await User.findByIdAndUpdate(_id, { token: null });

```

```

res.json({
  status: "succes",
  code: 204,
  data: { message: `Logout ${name} success` },
});
};

```

```

const updateSubscription = async (req, res) => {
  const { _id } = req.user;

```

```

  const { subscription } = req.body;
  const update = await User.findByIdAndUpdate(
    _id,
    { subscription },
    {
      new: true,
    }
  );
  if (!update) {
    throw new Conflict(`Sorry, not found`);
  }

```

```

res.status(200).json({
  status: "success",
  code: 200,
  data: {
    message: `Subscription change success on ${subscription}`,
  }
});
};

```

```

    },
  });
};

```

```

const updateAvatar = async (req, res) => {
  const { _id } = req.user;
  const filePath = req.file.path;

  cloudinary.uploader.upload(
    filePath,
    {
      folder: "avatars",
      transformation: [{ width: 200, height: 200 }],
      allowedFormats: ["jpg", "jpeg", "png", "gif"],
    },
    async (error, result) => {
      if (error) {
        console.error("Помилка завантаження на Cloudinary:", error);
        res.status(500).json({
          status: "error",
          code: 500,
          message: "Помилка завантаження на Cloudinary",
        });
      } else {
        console.log("Завантажено на Cloudinary:", result.url);

        const avatarURL = result.url;
        await User.findByIdAndUpdate(_id, { avatarURL });

        fs.unlinkSync(filePath);

        res.json({
          status: "success",
          code: 200,
          data: { avatarURL },
        });
      }
    }
  );
};

```

```

module.exports = {
  register: ctrlWrapperRoutes(register),
  verifyEmail: ctrlWrapperRoutes(verifyEmail),
};

```

```

resendVerifyEmail: ctrlWrapperRoutes(resendVerifyEmail),
login: ctrlWrapperRoutes(login),
curent: ctrlWrapperRoutes(curent),
logout: ctrlWrapperRoutes(logout),
updateSubscription: ctrlWrapperRoutes(updateSubscription),
updateAvatar: ctrlWrapperRoutes(updateAvatar),
};

```

Фронтенд - частина роботи реалізована за допомогою застосунок

React,

```

const Home = () => {
  const { isLoaggedIn } = useSelector(state => state.auth);
  return (
    <Container>
      {!isLoaggedIn ? (
        <>
          <Hero>
            <Тех>КУ "Міський міжшкільний навчально-виробничий
комбінат"</Тех>
          </Hero>
          <AboutUs />
          <Professions />
          <Content />
          <Video />
        </>
      ) : (
        <>
          {/* <Img src={auth} alt="Mr.auth" /> */}
          <UnderTitle>
            Перейдіть на вкладку
            <HomeLink to="/contacts">Контакти</HomeLink>
            та керуйте телефонною книгою
          </UnderTitle>
        </>
      )}
    </Container>
  );
};

export default Home;

```

```

const Professions = () => {

```



```

const controls = useAnimation();

useEffect(() => {
  const handleScroll = () => {
    const element = document.getElementById('yourElementId'); // Replace with the
actual ID of your element
    if (element) {
      const rect = element.getBoundingClientRect();
      const windowHeight =
        window.innerHeight || document.documentElement.clientHeight;

      if (rect.top <= windowHeight) {
        controls.start({ opacity: 1, x: 0 });
      }
    }
  };

  // Attach the handleScroll function to the scroll event
  window.addEventListener('scroll', handleScroll);

  // Cleanup the event listener on component unmount
  return () => {
    window.removeEventListener('scroll', handleScroll);
  };
}, [controls]);

const variants = {
  hidden: { opacity: 0, x: -100 },
  hidden2: { opacity: 0, x: 100 },
  visible: { opacity: 1, x: 0 },
};

return (
  <Section>
    <Wrapper>
      <InfoContainer>
        <Title>Професії та профілі</Title>

        <Text>
          ...
        </Text>

        <ProfessionList>
          <ProfessionItem>

```

```

    <ProfessionImg src={target} alt="Водій" />
    <ProfessionText>
      Водій автотранспортних засобів (категорії «В», «С»)»
    </ProfessionText>
  </ProfessionItem>

```

```

export const Content = () => (
  <Section>
    <TeacherTitle>Колектив</TeacherTitle>
    <Paragraf>
      КУ"ММНВК"СМР укомплектований ...
    </Paragraf>
    <div className="content">
      <Slider data={sliderData} />
    </div>
  </Section>
);
export const Slider = ({ data }) => {
  const [activeId, setActiveId] = useState(0);
  const prev = () => {
    setActiveId(activeId => {
      if (activeId > 0) {
        return activeId - 1;
      }
      return activeId;
    });
  };
  const next = () => {
    setActiveId(activeId => {
      if (activeId < data.length - 1) {
        return activeId + 1;
      }
      return activeId;
    });
  };
  return (
    <div className="slider-wrap">
      <div className="slider">
        <div className="slider__actions left">
          <IconButton
            direction="left"
            onClick={prev}

```

```

        disable={activeId === 0}
      />
    </div>
    {data.map((slide, idx) => (
      <div
        key={idx}
        className={`slide${idx === activeId ? ' active' : ''}`}
      >
        <div className="slide__info">
          <div className="slide__label">{slide.label}</div>
          <div className="slide__title">{slide.title}</div>
        </div>
        <img src={slide.img} alt={`Slide ${idx + 1}`} />
      </div>
    ))}
    <div className="slider__actions right">
      <IconButton
        direction="right"
        onClick={next}
        disable={activeId === data.length - 1}
      />
    </div>
  </div>
</div>
);
};

```

```

import { register } from '../redux/Authorization/operations';
import { useState } from 'react';
import { useDispatch } from 'react-redux';
import { Input, Container, But, Title } from './Register.styled';

```

```

const Register = () => {
  const dispatch = useDispatch();

  const [name, setName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleChange = event => {
    const { name, value } = event.target;

```

```

switch (name) {
  case 'name':
    setName(value);
    break;
  case 'email':
    setEmail(value);
    break;
  case 'password':
    setPassword(value);
    break;
  default:
    break;
}
};

const handleSubmit = event => {
  event.preventDefault();
  dispatch(register({ name, email, password }));

  setName("");
  setEmail("");
  setPassword("");
};

return (
  <Container>
    <Title>Зареєструвати нового користувача</Title>
    <form onSubmit={handleSubmit}>
      <Input
        type="text"
        name="name"
        value={name}
        placeholder="Name"
        onChange={handleChange}
      />
      <Input
        type="email"
        name="email"
        value={email}
        placeholder="Email"
        onChange={handleChange}
      />
      <Input
        type="password"

```

```

    name="password"
    value={password}
    placeholder="Password"
    onChange={handleChange}
  />
  <But type="primary" htmlType="submit">
    Зареєструватись
  </But>
</form>
</Container>
);
};

```

```
export default Register;
```

```

const Login = () => {
  const dispatch = useDispatch();
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const handleChange = event => {
    const { name, value } = event.target;
    switch (name) {
      case 'email':
        setEmail(value);
        break;
      case 'password':
        setPassword(value);
        break;
      default:
        break;
    }
  };
  const handleSubmit = event => {
    event.preventDefault();
    dispatch(loginization({ email, password }));
    setEmail("");
    setPassword("");
  };
  return (
    <Container>
      <Title>Увійти</Title>
      <form onSubmit={handleSubmit}>
        <Input

```

```

        type="email"
        name="email"
        value={email}
        placeholder="Email"
        onChange={handleChange}
      />
      <Input
        type="password"
        name="password"
        value={password}
        placeholder="Password"
        onChange={handleChange}
      />
      <But type="primary" htmlType="submit">
        Увійти
      </But>
    </form>
  </Container>
);
};

const StartingPage = () => {
  const [showText, setShowText] = useState(false);
  const [showGallery, setShowGallery] = useState(false);
  const imagePath = [
    '/public/images/0.png',
    '/public/images/1.jpg',
    '/public/images/2.png',
  ];

  const handleLecturesButtonClick = () => {
    setShowText(!showText);
  };

  const handleGalleryButtonClick = () => {
    setShowGallery(!showGallery);
  };

  return (
    <Section>
      <Title>Секретар керівника (організації підприємства, установи)</Title>
      <Img src={hero} alt="Mr.Peabody" />
      <UnderTitle>

```

```

    <HomeLink to="/register">Зареєструйтеся </HomeLink>
    або
    <HomeLink to="/login">Увійдіть</HomeLink>, щоб користуватися
    ДОДАТКОМ
  </UnderTitle>
  <Button onClick={handleLecturesButtonClick}>Про професію</Button>
  <Button onClick={handleGalleryButtonClick}>Галерея</Button>
  <HomeLink to="/secretary/lec">Лес</HomeLink>
  <HomeLink to="/secretary/test">Test</HomeLink>
  {showText && (
    <Text> ... </Text>
  )}
  {showGallery && (
    <div>
      {imagePaths.map((path, index) => (
        <Img key={index} src={path} alt={`Image ${index}`} />
      ))}
    </div>
  )}
</Section>
);
};

const Start = () => {
  const { startQuiz, showStart } = useContext(DataContext);
  return (
    <section
      className="text-white text-center bg-dark"
      style={{ display: `${showStart ? 'block' : 'none'}` }}
    >
      <div className="container">
        <div className="row vh-100 align-items-center justify-content-center">
          <div className="col-lg-8">
            <h1 className="fw-bold mb-4">Атестаційний модуль</h1>
            <button
              onClick={startQuiz}
              className="btn px-4 py-2 bg-light text-dark fw-bold"
            >
              Розпочати тест
            </button>
          </div>
        </div>
      </div>
    </section>
  );
};

```

```
);
};
export default Start;
```

```
const Quiz = () => {
  const {
    showQuiz,
    question,
    quizzes,
    checkAnswer,
    correctAnswer,
    selectedAnswer,
    questionIndex,
    nextQuestion,
    showTheResult,
  } = useContext(DataContext);
```

```
  return (
    <section
      className="bg-dark text-white"
      style={{ display: `${showQuiz ? 'block' : 'none'}` }}
    >
      <div className="container">
        <div className="row vh-100 align-items-center justify-content-center">
          <div className="col-lg-8">
            <div
              className="card p-4"
              style={{ background: '#3d3d3d', borderColor: '#646464' }}
            >
              <div className="d-flex justify-content-between gap-md-3">
                <h5 className="mb-2 fs-normal lh-base">{question?.question}</h5>
                <h5
                  style={{
                    color: '#60d600',
                    width: '100px',
                    textAlign: 'right',
                  }}
                >
                  {quizzes.indexOf(question) + 1} / {quizzes?.length}
                </h5>
              </div>
              <div>
                {question?.options?.map((item, index) => (
```



```

    <button
      key={index}
      className={`option w-100 text-start btn text-white py-2 px-3 mt-3
rounded btn-dark ${
        correctAnswer === item && 'bg-success'
      }}
      onClick={event => checkAnswer(event, item)}
    >
      {item}
    </button>
  )))
</div>

{questionIndex + 1 !== quizzes.length ? (
  <button
    className="btn py-2 w-100 mt-3 bg-primary text-light fw-bold"
    onClick={nextQuestion}
    disabled={!selectedAnswer}
  >
    Next Question
  </button>
) : (
  <button
    className="btn py-2 w-100 mt-3 bg-primary text-light fw-bold"
    onClick={showTheResult}
    disabled={!selectedAnswer}
  >
    Show Result
  </button>
)}
</div>
</div>
</div>
</div>
</section>
);
};
export default Quiz;

```