

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних
технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Заєць Сергій Вікторович
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.15:378.14

КВАЛІФІКАЦІЙНА РОБОТА

Проектування веб-орієнтованої системи для занять з репетитором

(тема роботи)

Спеціальності 122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

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

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи
Тимонін Юрій Олександрович
(прізвище, ім'я, по батькові)

кандидат технічних наук, доцент
(науковий ступінь, вчене звання)

Житомир – 2024

Висновок кафедри _____
за результатами попереднього захисту: _____

Протокол засідання кафедри _____
№ _____ від « _____ » _____ 20 _____ р.

Завідувач кафедри _____

_____ (науковий ступінь, вчене звання)

_____ (підпис)

_____ (прізвище, ім'я, по батькові)

« _____ » _____ 20 _____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

_____ (науковий ступінь, вчене звання)

_____ (підпис)

_____ (прізвище, ім'я, по батькові)

АНОТАЦІЯ

Заєць С. В. Проектування веб-орієнтованої системи для занять з репетитором. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня магістра за спеціальністю 122 – комп'ютерні науки. – Поліський національний університет, Житомир, 2024.

Кваліфікаційна робота присвячена дослідженню та розробці веб-орієнтованої системи для занять з репетиторами. Робота починається з аналізу існуючих платформ навчання, щоб визначити їхні переваги та недоліки. На основі цього аналізу формулюються вимоги до нової системи. Далі виконується розробка програмного забезпечення, включаючи реєстрацію користувачів, пошук та вибір репетитора, проведення уроків та спілкування між користувачами.

Ключовою частиною роботи є реалізація бази даних та функціональності системи. На завершальних етапах розробки проводиться тестування та вдосконалення системи для забезпечення її ефективності та надійності.

Результатом роботи є створення зручної та ефективної платформи для навчання з репетиторами, що може полегшити процес навчання для учнів та репетиторів у веб-середовищі.

Ключові слова: веб-орієнтована система, інформаційна система, навчання, програмне забезпечення, репетитор, сучасні технології в освіті технології веб-розробки.

ABSTRACT

Zaiets S.V. Designing a web-based system for classes with a tutor. - Qualification work on manuscript rights.

Qualification work for obtaining a master's degree in specialty 122 - computer science. – Polis National University, Zhytomyr, 2024.

The qualification work is dedicated to the research and development of a web-oriented system for classes with tutors. The work begins with an analysis of existing learning platforms to determine their strengths and weaknesses. Based on this analysis, the requirements for the new system are formulated. Subsequently, software development is carried out, including user registration, tutor search and selection, lesson delivery, and user-to-user communication.

The key part of the work involves implementing the database and system functionality. In the final stages of development, testing and improvement of the system is carried out to ensure its efficiency and reliability.

The result of the work is the creation of a convenient and effective platform for learning with tutors, which can facilitate the learning process for both students and tutors in the web environment.

Keywords: web-oriented system, information system, training, software, tutoring, current technologies in web development technology education.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ	8
1.1 Аналіз інформаційних потреб та визначення предметної області дослідження	8
1.2 Аналіз вимог до інформаційної системи.....	12
Висновки до першого розділу.....	14
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ЗАНЯТЬ З РЕПЕТИТОРОМ.....	16
2.1 Вибір засобів розробки програмного модуля	16
2.2 Засоби проєктування бази даних	18
2.3 Проєктування структури системи	19
Висновки до другого розділу	22
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	24
3.1 Програмна реалізація	24
3.2 Інструкція користувача	26
Висновки до третього розділу.....	31
ВИСНОВКИ.....	33
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	34

ВСТУП

Оскільки освіта в сучасному світі є ключовим фактором особистого та професійного розвитку, заняття з репетитором стають все більш популярним та ефективним методом набуття знань. Використання веб-орієнтованих систем спрощує процес навчання та спілкування з репетитором. Дослідження цих аспектів сприяє розвитку платформ для занять з репетиторами та інформаційних технологій в освіті.

Метою кваліфікаційної роботи є розробка веб-орієнтованої платформи для планування та проведення занять з репетитором. Для досягнення цієї мети необхідно вирішити наступні **завдання**: провести аналіз існуючих систем для онлайн-навчання; створити структуру бази даних; розробити веб-орієнтовану платформу для занять з репетитором; розробити інструкцію користувача;

Предметом дослідження даної роботи є процес розробки веб-системи для організації занять з репетиторами.

Об'єктом дослідження даної роботи є методи та засоби розробки веб-платформи для занять з репетиторами. Крім того, досліджується і предметна область, у якій буде застосовуватися веб-платформа для занять з репетиторами, зокрема, освітня сфера.

У процесі дослідження використовувався аналіз даних з різних джерел. Спочатку було проведено збір необхідної інформації з відповідних джерел. Потім ці дані були систематизовані та проаналізовані з метою виявлення закономірностей, тенденцій та ключових факторів, що впливають на досліджувану проблему.

Порівняльний аналіз був важливим компонентом дослідження. Це включало в порівнянні різні аспекти досліджуваної теми або об'єктів. Метод порівняння дозволив виявити подібності та відмінності між різними аспектами, що допомогло зробити об'єктивні висновки.

У дослідженні також використовувались засоби автоматизації обробки даних. Це включало в себе використання спеціалізованих програмних засобів для обробки та аналізу великих обсягів даних. Автоматизація дозволила прискорити процес аналізу та зменшити ймовірність помилок.

Дослідження також включало в себе тестування гіпотез та моделей, щоб перевірити їхню відповідність даним. Після тестування проводилася детальна аналітика результатів тестування для формулювання висновків та рекомендацій.

Апробація результатів дослідження здійснювалася під час Міжнародної науково-практичної конференції «Актуальні проблеми сучасної науки та освіти» (м. Львів, 2024 р.) та Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених «Інформаційні технології та моделювання систем» (м. Житомир, 2024 р.), за результатами яких опубліковано тези у збірниках конференцій.

Структура та обсяг роботи. Кваліфікаційна робота складається з вступу, трьох розділів, висновку та списку використаних джерел. Загальний обсяг роботи становить 36 сторінок.

РОЗДІЛ 1 ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Аналіз інформаційних потреб та визначення предметної області дослідження

Аналіз літературних джерел та практичного досвіду використання інформаційних систем у визначеній предметній області показав, що існують інші системи, які вирішують аналогічні або схожі завдання. Серед таких систем, для порівняння, було виділено наступні: Preply, Verbling та Italki.

Системи Preply, Verbling та Italki вибрані для порівняння через їх популярність, функціонал і відомість на ринку онлайн-навчання.

- Популярність і репутація: Preply, Verbling та Italki вже відомі в онлайн-навчальній галузі та користувачам з усього світу. Їхній успіх і популярність свідчать про те, що вони успішно вирішують завдання, схожі на ті, що стоять перед власним проектом.

- Функціонал: ці системи надають широкий спектр функціональності для організації занять з репетиторами онлайн, включаючи можливість пошуку репетиторів за різними критеріями, вибір підходящого часу для занять, спілкування з репетиторами через онлайн-чат або відеодзвінки, оплату занять.

- Міжнародний охоплення: Preply, Verbling та Italki працюють з репетиторами та учнями з усього світу, надаючи можливість навчання та викладання мов будь-де, де є доступ до Інтернету. Це робить їх хорошими об'єктами порівняння для проекту, що також має амбіції працювати на міжнародному рівні.

Також було встановлено декілька критеріїв для аналізу, а саме: функціональні можливості, інтерфейс та юзабіліті, цінова політика, кількість користувачів та їх відгуки.

– Під функціональними можливостями розуміється аналіз доступних функцій і можливостей платформи, таких як можливість пошуку репетиторів, організація онлайн-уроків, система рейтингів та відгуків, можливість спілкування між репетиторами та учнями тощо.

– Аналіз інтерфейсу та юзабіліті передбачає, оцінку наскільки простим та зручним є інтерфейс користувача платформи, чи легко користуватися системою, чи зручно знаходити потрібну інформацію.

– Під час аналізу цінової політики досліджується цінова стратегія платформи, включаючи вартість послуг репетиторів, наявність безкоштовних пробних уроків та інші аспекти, що впливають на фінансову доступність користувачів.

– Оцінка кількості користувачів та їх відгуків передбачає оцінку кількості зареєстрованих користувачів та наявність репетиторів у системі, а також аналізуються відгуки користувачів щодо якості наданих послуг та функціоналу платформи.

За наведеними критеріями створено порівняльну таблицю (табл. 1.1).

Табл. 1.1

Порівняльний аналіз існуючих веб-орієнтованих систем для занять з репетитором

Критерії	Preply	Verbling	Italki
Функціональні можливості	Пропонує пошук репетиторів за різними параметрами. Можливість організації онлайн-уроків через вбудований веб-інтерфейс. Є система рейтингів і відгуків. Можливість спілкування між репетиторами та учнями через чат.	Пошук репетиторів, організація онлайн-уроків, система рейтингів і відгуків, спілкування через вбудований чат.	Пошук репетиторів, організація онлайн-уроків, система відгуків, спілкування через повідомлення та відеодзвінки.

Інтерфейс та юзабіліті	Інтерфейс досить простий та зрозумілий. Легко знаходити потрібну інформацію та користуватися системою.	Простий та зрозумілий інтерфейс, легко знаходити інформацію та користуватися системою.	Інтерфейс зрозумілий, але може бути трохи складнішим для новачків. Інформація легко доступна.
Цінова політика	Ціни варіюються від репетитора до репетитора, але зазвичай на середньому рівні. Безкоштовні пробні уроки можливі з деякими репетиторами.	Ціни можуть бути дещо вищими порівняно з іншими платформами. Пропонуються безкоштовні пробні уроки.	Різноманітні ціни, відносно доступні. Безкоштовні пробні уроки доступні з деякими репетиторами
Кількість користувачів	Має велику базу користувачів та репетиторів з різних країн.	Має значну кількість зареєстрованих користувачів та репетиторів.	Має велику базу користувачів та репетиторів з усього світу.
Відгуки користувачів	Переважно позитивні відгуки, але деякі користувачі вказують на проблеми з якістю деяких репетиторів та нерівномірне обслуговування.	Загалом позитивні відгуки, з високою оцінкою якості навчання та обслуговування.	Переважно позитивні відгуки, але деякі користувачі вказують на проблеми з роботою деяких репетиторів та підтримкою користувачів.

За результатами порівняння існуючих веб-орієнтованих систем для занять з репетитором можна виділити наступні переваги та недоліки, що відображають як їхню ефективність, так і обмеження у певних аспектах:

Табл 1.2

Переваги та недоліки існуючих веб-орієнтованих систем для занять з репетитором

	Переваги	Недоліки
Preply	<ul style="list-style-type: none"> – Широкий вибір репетиторів з різних країн та предметних областей. – Система рейтингів і відгуків допомагає вибрати відповідного репетитора. – Можливість організувати уроки в зручний для вас час. 	<ul style="list-style-type: none"> – Ціни можуть бути високими порівняно з іншими платформами. – Якість репетиторів може варіюватися, іноді зустрічаються неякісні викладачі.
Verbling	<ul style="list-style-type: none"> – Висока якість викладання завдяки відбору репетиторів. – Дуже зручний інтерфейс та швидке знаходження репетиторів. – Широкий вибір мов і предметів для навчання. 	<ul style="list-style-type: none"> – Вищі ціни порівняно з іншими платформами. – Можуть виникати проблеми з доступністю репетиторів, особливо для менш популярних мов чи предметів.

Italki	<ul style="list-style-type: none"> – Різноманітність цін на уроки, включаючи недорогі варіанти. – Широкий вибір репетиторів та предметів для навчання. – Велика активна спільнота користувачів, що допомагає знайти потрібного репетитора. 	<ul style="list-style-type: none"> – Інтерфейс може здаватися менш зручним порівняно з іншими платформами. – Якість викладання може варіюватися, адже репетитори самостійно встановлюють свої ціни та розклад.
--------	---	--

За результатами порівняння функціональних можливостей, інтерфейсу та юзабіліті, цінової політики, кількості користувачів та їх відгуків платформ для веб-орієнтованих платформ для занять з репетиторами (Preply, Verbling, Italki), можна зробити наступні висновки:

- Кожна з платформ має свої переваги та недоліки у всіх аспектах порівняння, і жодна з них не є ідеальною для користувачів.

- Розробка аналогічної системи може бути виправдана для створення інтегрованої платформи, яка поєднує кращі риси кожної з існуючих платформ. Наприклад, забезпечення широкого вибору репетиторів, зручного інтерфейсу та конкурентоспроможних цін.

- Така система може бути кращою за існуючі аналоги завдяки комбінації найкращих практик і функціональностей з різних платформ, а також врахуванню потреб користувачів у зручності, доступності та якості послуг.

- Розробка нової системи також може стати кращим варіантом для користувачів, які не задоволені або не знайшли потрібні функції на існуючих платформах. Нова система може врахувати їхні потреби та вимоги.

Отже, розробка унікальної системи може бути виправдана для створення інноваційного та конкурентоспроможного продукту, який задовольняє потреби користувачів у всіх аспектах навчання з репетиторами.

Предметна область даного дослідження визначається необхідністю удосконалення системи навчання з репетиторами, орієнтованої на веб-середовище, з метою ефективної співпраці між репетиторами та їхніми учнями за допомогою сучасних технологій. Інформаційне забезпечення цієї області

охоплює різноманітні аспекти, такі як потоки даних, користувачі та джерела інформації.

У цій системі інформаційні потоки включають обмін даними між репетиторами та учнями, розклади занять, інформацію про результати навчання та інші навчальні аспекти. Користувачі даних складаються з репетиторів та учнів, які взаємодіють у віртуальному середовищі для досягнення навчальних цілей. Джерела інформації включають різноманітні бази даних, навчальні матеріали, а також інші ресурси, необхідні для навчання та взаємодії.

У системі існують різні канали обміну інформацією між учнями та репетиторами, кожен з яких відіграє важливу роль у забезпеченні ефективної комунікації та організації навчального процесу:

Учасниками системи є:

Студенти: надають системі свої особисті дані та визначають свої навчальні цілі, а також записуються на заняття з репетиторами.

Викладачі: надають інформацію про свої послуги, цінову політику, графік проведення занять, а також можуть надавати учням додаткові навчальні матеріали та завдання.

1.2 Аналіз вимог до інформаційної системи

Аналіз предметної області дослідження дозволив визначити конкретні функціональні вимоги до системи, відповідно до яких система має:

1. надавати користувачам можливість здійснювати пошук репетиторів за різними критеріями, такими як предмет, вартість послуг тощо.

2. забезпечувати можливість організації та проведення онлайн-уроків між репетиторами та учнями з використанням відповідних інструментів для взаємодії та навчання.

3. підтримувати систему рейтингів та відгуків, яка дозволить користувачам оцінювати та залишати відгуки про репетиторів та їхню роботу.

4. забезпечувати зручний механізм спілкування між репетиторами та учнями, включаючи обмін повідомленнями, планування зустрічей тощо.

5. дозволяти користувачам управляти своїм розкладом, планувати та регулювати час проведення занять з репетиторами.

6. гарантувати безпеку та конфіденційність особистих даних користувачів, включаючи дані про оплату та особисту інформацію.

Ці функціональні вимоги є ключовими для створення ефективної інформаційної системи для занять з репетитором, яка задовольнить потреби як учнів, так і репетиторів. При цьому не менш важливо врахувати і нефункціональні вимоги до інформаційної системи для занять з репетитором. Серед них можна виділити наступні:

1. Система повинна працювати стабільно, тобто мати стійкість до відмов, забезпечуючи безперервний доступ користувачів до послуг.

2. Інформаційна система має бути швидкодіюююю, забезпечуючи оперативну відповідь на запити користувачів та швидкий доступ до необхідної інформації.

3. Система повинна бути легко масштабованою, здатною адаптуватися до зростання кількості користувачів та обсягу навчальних матеріалів без втрати продуктивності.

4. Забезпечення захищеності особистих даних користувачів є важливою нефункціональною вимогою. Система повинна використовувати сучасні методи шифрування та забезпечувати конфіденційність та цілісність інформації.

5. Інформаційна система має бути сумісною з різними пристроями та платформами, щоб забезпечити зручний доступ до неї з будь-якого пристрою.

6. Інтерфейс користувача повинен бути інтуїтивно зрозумілим та зручним для використання, щоб навіть недосвідчені користувачі могли легко користуватися системою.

У веб-орієнтованій системі для занять з репетитором будуть використовуватися різноманітні дані. Це можуть бути особисті дані користувачів, такі як ім'я, адреса електронної пошти, контактні дані та інші особисті відомості, які вони надають при реєстрації. Крім того, це також можуть бути дані про їхні навчальні потреби, попередні досягнення, вимоги до репетиторів та інші відомості, які допомагають належному підбору репетитора для кожного учня. Крім того в системі будуть зберігатись всі навчальні матеріали, які завантажують репетитори та учні.

Для забезпечення безпеки та достовірності інформації, дані користувачів будуть зберігатися в захищеній базі даних з використанням сучасних методів шифрування. Доступ до цих даних буде обмежений лише авторизованим користувачам за допомогою системи аутентифікації.

Висновки до першого розділу

У першому розділі було вивчено предметну область системи навчання з репетиторами, зорієнтованої на веб-середовище, з метою забезпечення ефективної співпраці між репетиторами та їхніми учнями за допомогою сучасних технологій. Аналіз існуючих інформаційних систем, таких як Preply, Verbling та Italki, дозволив виявити їхні переваги та недоліки у всіх аспектах порівняння, і жодна з них не є ідеальною для користувачів, тому розробка аналогічної системи може бути виправдана.

Відповідно до аналізу предметної області дослідження були сформульовані конкретні функціональні та нефункціональні вимоги до веб-орієнтованої системи для занять з репетитором. Основними функціональними вимогами є можливість пошуку репетиторів, організація онлайн-уроків, система рейтингів та відгуків, можливість спілкування між учасниками та управління розкладом та заняттями. Поряд з цим, не менш важливими є нефункціональні вимоги, такі як надійність,

швидкодія, масштабованість, безпека даних, сумісність та зручність використання інтерфейсу. Впровадження відповідної інформаційної системи має потенціал полегшити процес навчання та підвищити якість освіти.

РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ЗАНЯТЬ З РЕПЕТИТОРОМ

2.1 Вибір засобів розробки програмного модуля

Для розробки програмного модуля, що відповідає за пошук репетиторів та проведення онлайн занять, і повністю задовольняє вимоги поставлених завдань, для досягнення максимальної ефективності та успішної реалізації завдання буде задіяно такий набір інструментів:

- мова програмування TypeScript;
- для створення front-end частини веб-системи буде використано ReactJS;
- для створення інтерфейсу користувача веб-додатка буде використовуватися UI бібліотека PrimeReact;
- для централізованого управління станом веб-додатка буде використано Redux;
- для оплати онлайн уроків буде використано платіжну систему Fondy;
- для створення back-end частини веб-системи буде використовуватися фреймворк NestJS;
- СКБД PostgreSQL для зберігати та ефективного управління даними.

Вибір вказаних інструментів обґрунтований наступними причинами:

- React: забезпечує швидку та ефективну роботу з користувацьким інтерфейсом завдяки віртуальному DOM; дозволяє побудувати додаток з використанням компонентів, що полегшує розробку та підтримку коду; має велику активну спільноту та багато ресурсів для навчання та розвитку;
- Redux: дозволяє ефективно керувати станом додатку та забезпечує простий та однозначний спосіб керування даними; легко інтегрується з React, що робить його ідеальним вибором для управління станом додатку;

- **PrimeReact**: надає багато готових компонентів, які можна легко і швидко використовувати для побудови інтерфейсу користувача; має сучасний та естетичний дизайн, що допоможе підвищити користувацький досвід;

- **Fondu**: Fondu надає зручний інтерфейс для проведення оплати, що дозволить учням легко та швидко здійснювати платежі за уроки без зайвих складнощів; забезпечує високий рівень безпеки платежів; Fondu має гнучкий API, що дозволяє легко інтегрувати платіжну систему з іншими сервісами та платформами, такими як система управління курсами або веб-орієнтована система для занять з репетиторами;

- **Nest.js**: дозволяє будувати додатки з використанням модульної архітектури, що полегшує організацію та підтримку коду; Nest.js підтримує TypeScript з коробки, що дозволяє писати безпечний та підтримуваний код; легко інтегрується з багатьма іншими бібліотеками та фреймворками, що робить його ідеальним вибором для розробки веб-орієнтованих додатків;

- **PostgreSQL**: має високу надійність завдяки реплікації, резервному копіюванню та відновленню; підтримує широкий спектр типів даних, функцій та інструментів, що робить його універсальним рішенням для багатьох завдань; має високий рівень безпеки завдяки аутентифікації, шифруванню та аудиту.

Обрані інструменти відповідають потребам системи для занять з репетиторами з урахуванням їхніх особливостей та можливостей. React.js забезпечить швидко та ефективно розробку користувацького інтерфейсу, тоді як Nest.js забезпечить масштабованість, безпеку та реальний час взаємодії. Це дозволить створити зручну та ефективно систему для організації та проведення занять з репетиторами.

2.2 Засоби проєктування бази даних

Для використання СКБД PostgreSQL було розгорнуто хмарну базу даних ElephantSQL, яка спеціалізується на PostgreSQL. Вона надає можливість створення та управління базами даних PostgreSQL без необхідності встановлення та налаштування бази даних на власному сервері. Завдяки цьому, ми можемо швидко розпочати розробку проєкту, не зважаючи на складнощі налаштування бази даних.

Принцип роботи ElephantSQL досить простий:

1. Після створення облікового запису в ElephantSQL можна створити нову базу даних. Для цього зазвичай використовується веб-інтерфейс.
2. Після створення бази даних отримуємо URL-адресу, ім'я користувача та пароль для підключення. Ці дані використовуються для налаштування підключення до бази даних з додатка або сервера.
3. Після успішного підключення можна використовувати базу даних для зберігання та отримання даних у додатку.

Для управління базою даних було використано TablePlus, що дозволило ефективно працювати з даними та спростити аналіз інформації.

TablePlus - це могутній клієнт для керування базами даних, який надає зручний та інтуїтивно зрозумілий інтерфейс для роботи з різними типами баз даних, включаючи PostgreSQL.

Також було сконфігуровано Swagger UI для взаємодії з базою даних у реальному часі, використовуючи візуальний інтерфейс Swagger UI. Використання Swagger є обґрунтоване з різних поглядів. По-перше, Swagger дозволяє автоматизувати процес створення документації для API, що є важливим етапом при розробці будь-якого програмного інтерфейсу. Завдяки цьому, розробники та інші зацікавлені сторони можуть легко ознайомитися з функціональністю API, розуміти його можливості та обмеження.

Другим важливим аспектом є можливість тестування API безпосередньо з інтерфейсу Swagger UI. Це дозволяє швидко та ефективно перевірити правильність роботи кожного ендпоінту, зменшуючи час, потрібний для тестування та відлагодження.

2.3 Проєктування структури системи

На основі встановлених функціональних вимог формується алгоритм, який забезпечує процес від пошуку репетитора до проведення заняття (Додаток В).

Контекстна діаграма представляє процес пошук та вибору репетитора, а також проведення заняття. Процес складається з восьми кроків:

1. Введення критеріїв пошуку: користувач вводить критерії пошуку репетитора, включаючи предмет, ціну та інші дані.
2. Пошук репетиторів: система шукає репетиторів, які відповідають критеріям пошуку.
3. Сортування результатів пошуку: система сортує результати пошуку за заданим критерієм.
4. Відображення результатів пошуку: система відображає результати пошуку користувачу.
5. Вибір репетитора: користувач вибирає репетитора.
6. Оформлення замовлення: користувач та репетитор узгоджують ціну та час заняття.
7. Оплата заняття: користувач оплачує заняття за узгодженою ціною.
8. Проведення заняття: користувач та репетитор проводять заняття

Діаграма взаємодії з чотирма ресурсами:

– Критерії пошуку: це інформація, яку користувач вводить для пошуку репетитора.

- БД репетиторів: це база даних, яка містить інформацію про репетиторів.
- Платіжна система: це система, що дозволяє учням оплатити заняття.
- Інтерфейс веб орієнтованої системи: це сукупність засобів, за допомогою яких користувач взаємодіє з вебплатформою.

Алгоритм, який забезпечує процес узгодження та проведення занять відображено на додатку Г. Діаграма представляє процес узгодження та проведення заняття у веб-орієнтованій системі для занять з репетитором. Процес складається з семи кроків:

1. Узгодження заняття: репетитор і учень узгоджують час та дату заняття;
2. Генерація посилання для онлайн-конференції: до онлайн уроку прикріплюється посилання на онлайн-конференцію;
3. Відправка посилання на онлайн-конференцію;
4. Вхід до онлайн-конференції;
5. Проведення заняття: репетитор і учень проводять заняття за допомогою онлайн-конференції;
6. Оцінка заняття: репетитор і учень оцінюють заняття;
7. Система зберігає інформацію про заняття в базі даних.

Для відображення послідовності дій, які відбуваються під час онлайн-заняття з репетитором створено діаграму послідовності (Додаток Б).

Діаграма складається з п'яти основних дій:

1. Репетитор та користувач реєструються на платформі онлайн-занять. Реєстрація є обов'язковою для всіх учасників онлайн-заняття. Під час реєстрації користувачі повинні надати свої особисті дані, такі як ім'я, прізвище, email та номер телефону. Репетитори також повинні надати свої кваліфікаційні дані, такі як освіта, професійний досвід та сертифікати.

2. Репетитор створює портфоліо, яке демонструє його кваліфікацію та досвід. Створення портфоліо є необов'язковим для репетиторів. Однак воно може бути корисним для залучення клієнтів.

3. Користувач шукає репетитора, який відповідає його потребам.

4. Користувач та репетитор узгоджують час і дату заняття.

5. Користувач та репетитор в узгоджений час підключаються до онлайн конференції та проводять заняття. Під час онлайн заняття репетитор передає знання та навички користувачеві за допомогою програмного модуля для відеозв'язку. Користувач може задавати запитання та отримувати відповіді від репетитора в режимі реального часу.

Далі доцільно розробити діаграму прецедентів, яка призначена для відображення взаємозв'язків між акторами та прецедентами, встановлення меж системи та опису функціональних вимог.

В додатку А представлено діаграму прецедентів для веб-орієнтованої системи. Система складається з двох основних компонентів: програмного модуля і користувачів (репетитор та учень).

На діаграмі прецедентів зображено користувачів системи та доступний їм функціонал для реєстрації, створення онлайн-конференцій, завантаження матеріалів, спілкування та інших операцій.

Для вирішення поставлених задач розроблено структуру бази даних. Структура бази даних інформаційної системи представлена на додатку Д.

Основні таблиці бази даних:

- *Уроки*: таблиця зберігає такі параметри як назва, ціна, тривалість, дата створення, автор, тема уроку та ін;
- *Зарезервовані уроки*: таблиця зберігає такі параметри як назва, тема уроку, статус (може бути «проведено» або «заплановано»), тривалість, дата початку, посилання на онлайн зустріч та ід всіх учасників уроку;
- *Файли*: таблиця зберігає інформацію про файли, пов'язані з уроками, такі як ім'я файлу, тип файлу, розмір, дата створення та урок, з яким він пов'язаний;

- *Відгуки*: містить інформацію про відгуки, залишені учнями про репетитора, включаючи повідомлення, рейтинг, відправника та одержувача;
- *Діалоги*: містить інформацію про діалоги між учнями та вчителями, включаючи учасників, дату створення та повідомлення;
- *Повідомлення*: містить інформацію про повідомлення в діалогах, включаючи текст, час надсилання, відправника, одержувача та роль користувача;
- *Репетитори*: містить інформацію про репетиторів, включаючи ім'я, електронну пошту, пароль, URL-адресу аватара, опис, рейтинг, роль користувача, дату створення та кількість учнів (для вчителів);
- *Учні*: містить інформацію про користувачів учнів, включаючи ім'я, електронну пошту, пароль, URL-адресу аватара, роль користувача та дату створення.

Розроблена структура системи відповідає поставленим задачам і забезпечує ефективний процес пошуку репетитора та проведення онлайн-занять, враховуючи індивідуальні потреби учнів та вчителів, забезпечуючи гнучкість у виборі розкладу та методів навчання, а також забезпечуючи високий рівень безпеки та конфіденційності даних користувачів.

Крім того, система забезпечує зручність у взаємодії завдяки інтуїтивно зрозумілому інтерфейсу та доступності для користувачів різних вікових груп та рівнів освіти. В цілому, розроблена система є не лише зручним інструментом для навчання та саморозвитку, але й потужним інструментом у підвищенні якості освіти та доступності навчальних ресурсів.

Висновки до другого розділу

Для розробки програмного модуля було задіяно ряд інструментів, таких як TypeScript, ReactJS, PrimeReact, Redux, Fondy, NestJS та PostgreSQL. Кожен з них має свої переваги, що робить їх ідеальними для цієї задачі. Обрані інструменти

відповідають потребам системи для занять з репетиторами, роблячи її зручною та ефективною.

Для реалізації інформаційної системи було обрано хмарну базу даних ElephantSQL, спеціалізовану на PostgreSQL. Це дозволило ефективно використовувати базу даних без необхідності встановлення та налаштування на власному сервері, що спростило процес розробки. Використання TablePlus для керування базою даних стало ефективним рішенням, що дозволило швидко та зручно працювати з даними.

У процесі проектування структури системи було створено контекстну діаграму A1, яка відображає процес пошуку репетиторів та узгодження занять, діаграму послідовності, що відображає реєстрацію, пошук, узгодження та проведення занять у веб-орієнтованій системі.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Програмна реалізація

Система розроблена відповідно до принципів клієнт-серверної архітектури, де фронтенд та бекенд розділені для забезпечення більшої масштабованості та швидкодії. Основні архітектурні складові системи:

- Клієнтська частина (frontend): розроблена з використанням ReactJS та PrimeReact для створення інтерактивного та зручного інтерфейсу для користувачів. Redux використовується для управління станом додатку, забезпечуючи консистентність даних та швидкий доступ до них.
- Серверна частина (backend): реалізована з використанням NestJS, що забезпечує ефективну обробку запитів користувачів, масштабованість та безпеку додатку. PostgreSQL використовується для зберігання та управління даними, а Fondy інтегровано для обробки платежів.

Реалізація клієнтської частини системи для занять з репетитором включала в себе кілька ключових етапів:

1. Дизайн і прототипування;
2. Реалізація компонентів;
3. Створення взаємодій з сервером;
4. Управління станом;
5. Тестування та налагодження.

Першим кроком було створення дизайну користувацького інтерфейсу та його прототипування. Для цього використовувалася спеціалізована програма Figma, що дозволяє створювати макети інтерфейсу.

Після створення дизайну розпочалася реалізація компонентів інтерфейсу. Використовувалися бібліотеки ReactJS та PrimeReact для швидкої і зручної розробки компонентів інтерфейсу, таких як кнопки, форми, таблиці тощо.

Для реалізації взаємодії між клієнтською та серверною частиною системи використовувалися асинхронні запити HTTP, за допомогою бібліотеки Axios.

Для зручного управління станом додатку та передачі даних між компонентами використовувалася бібліотека управління станом Redux. Redux дозволяє зберігати стан додатку в одному централізованому місці і легко керувати ним за допомогою дій та редукторів.

Після реалізації фронтенду проводилися тестування та налагодження для виявлення та виправлення можливих помилок і недоліків у роботі інтерфейсу.

Лістинг програмного коду клієнтської частини проекту наведено у додатку X. Структура проекту організована, як показано в додатку III.

Реалізація серверної частини системи для занять з репетитором включала в себе розробку API, взаємодію з базою даних, забезпечення безпеки та автентифікації, а також обробку запитів користувачів.

1. Розробка API

– Початково, створювалася структура проекту за допомогою фреймворка NestJS. NestJS дозволяє швидко створити потужний та ефективний сервер, використовуючи концепції модульності та ін'єкції залежностей.

– Для визначення маршрутів та обробки запитів використовувався декоратор @Controller та декоратори методів, такі як @Get, @Post тощо.

– Запити оброблялися у відповідних контролерах, де виконувалася логіка опрацювання запитів, виклик методів сервісів та повернення відповідей клієнту.

– Для валідації вхідних даних використовувалася бібліотека class-validator, яка дозволяє встановлювати правила валідації та автоматично перевіряти дані.

2. Забезпечення безпеки та автентифікації

Для забезпечення безпеки використовувався механізм автентифікації та авторизації. Користувачі аутентифікувалися за допомогою токенів, які генерувалися після успішної аутентифікації.

3. Взаємодія з базою даних

– Налаштовувалась взаємодія з базою даних PostgreSQL за допомогою модуля TypeORM. TypeORM дозволяє створювати моделі даних та виконувати запити до бази даних з використанням об'єктно-реляційного відображення.

– Створювалася схема бази даних, яка відображала структуру даних, необхідних для системи, та відносини між ними.

Лістинг програмного коду серверної частини проекту наведено у додатку Щ. Структура проекту організована, як показано на додатку Ж.

Сервер встановлює кілька груп маршрутів для роботи з усіма елементами системи. Ці маршрути отримують інформацію про репетиторів, учнів, уроки або ж інші структурні елементи системи від клієнтської сторони. Сервер надсилає відповідний HTTP-запит до відповідної кінцевої точки API та повертає дані у відповідь.

3.2 Інструкція користувача

У **розробленій** веб-орієнтованій системі для занять з репетитором доступ до функціоналу можливий лише після проходження процесу авторизації (додаток И) Якщо користувач вже має обліковий запис, йому необхідно ввести свою електронну пошту, пароль та свою роль (“репетитор” або “студент”) для входу.

В іншому випадку, якщо користувач новий, він має пройти процес реєстрації (додаток К).

Процес реєстрації включає в себе наступні кроки:

1. Користувач повинен ввести своє повне ім'я.
2. Вказання дійсної електронної адреси, яка буде використовуватися для входу в систему.
3. Встановлення пароля, який буде використовуватися для захисту облікового запису.
4. Користувач має обрати свою роль у системі, репетитор або студент.

Після успішної реєстрації користувач може використовувати введені дані для авторизації в системі. Під час авторизації, крім електронної пошти та пароля, користувач також вказує свою роль, щоб система змогла забезпечити відповідний функціонал залежно від потреб користувача.

Після успішного входу в систему користувача автоматично перенаправить на головну сторінку веб-додатку (додаток Л), де можна користуватися всіма його можливостями, доступними авторизованим користувачам.

Головна сторінка відображатиме список репетиторів, які зареєстровані у системі, з можливістю фільтрації за різними критеріями, а також навігаційні елементи для швидкого доступу до різних функціональних модулів та розділів веб-додатку.

Управління веб-додатком здійснюється за допомогою різних елементів навігації, таких як меню, кнопки та посилання. Ці елементи дозволяють користувачам швидко переміщатися між різними розділами, сторінками та функціями веб-додатку.

У верхній частині головної сторінки розміщено головне меню з основними розділами веб-застосунку, такими як «Репетитори», «Календар», «Повідомлення», «Уроки», «Налаштування» та «Мій профіль» для репетиторів.

Бронювання онлайн уроку:

Однією з основних функцій вебзастосунку є можливість бронювати онлайн уроки з репетиторами. Онлайн уроки є основною одиницею контенту вебзастосунку. Нижче наведені кроки для бронювання онлайн уроку:

1. Користувач виконує фільтрацію репетиторів за дисципліною та ціною за урок, використовуючи доступні фільтри або пошуковий рядок.
2. Після виконання пошуку користувач переглядає список репетиторів, які відповідають його критеріям.
3. Користувач обирає репетитора, який відповідає його потребам, переглядаючи інформацію про нього, таку як профіль та відгуки.
4. Після обрання репетитора користувач натискає кнопку "Забронювати урок" на сторінці профілю репетитора або в списку репетиторів на головній сторінці.
5. Далі з'являється модальне вікно з календарем репетитора, де користувач може вибрати зручний для нього час проведення уроку.
6. Після вибору дати і часу користувач підтверджує обраний час для уроку.
7. Після підтвердження часу користувач переходить до оплати за урок, вибираючи доступний метод оплати.
8. Після успішної оплати користувач отримує підтвердження бронювання.
9. На дату та час уроку користувач та репетитор зустрічаються в віртуальному класі для проведення уроку, використовуючи засоби відеоконференційного зв'язку та додаткові навчальні матеріали, які можуть бути завчасно завантажені на платформі.

На платформі для занять з репетитором також є особистий календар з розкладом занять (додаток М), який доступний як для учнів, так і для репетиторів. У цьому календарі користувачі можуть переглядати свій розклад занять та отримувати коротку інформацію про кожен урок. Вони можуть бачити дату та час проведення кожного заняття, а також назву предмету або тему уроку. Крім того, у цьому календарі користувачі можуть перейти на сторінку онлайн урока для та переглянути детальну інформацію про урок, таку як матеріали, які необхідно підготувати до уроку, посилання на відеоконференцію тощо. Такий особистий календар дозволяє користувачам ефективно організовувати свій час і планувати свої навчальні заняття.

Також є особистий чат (додаток П), в якому репетитор та учень можуть спілкуватись один з одним. Цей чат дозволяє їм обговорювати деталі уроків, обмінюватись інформацією, вирішувати питання, що виникають під час навчання, а також будь-які інші питання або коментарі, пов'язані з процесом навчання. Особистий чат є зручним інструментом для побудови ефективного комунікаційного процесу між репетитором та учнем, що сприяє зручності та ефективності навчання.

Репетитори та студенти можуть переглядати список своїх уроків (запланованих або проведених) на сторінці «Уроки» (додаток Р). Ця сторінка дозволяє користувачам керувати своїми навчальними заняттями. Користувачі можуть переглядати усі заплановані та вже проведені уроки, а також фільтрувати їх за дисципліною та статусом уроку. Крім того, вони можуть сортувати список уроків за датою або ціною для зручності.

Для репетитора доступна кнопка редагування уроку, натиснувши на яку відкриється модальне вікно (додаток С). В цьому вікні репетитор має можливість відредагувати параметри уроку. Він може змінити тему уроку, додати посилання на онлайн конференцію для зручного підключення до уроку, встановити ціну за урок та вказати тривалість уроку, якщо це було узгоджено зі студентом.

З цієї сторінки користувачі можуть перейти на сторінку онлайн уроку (додаток Т), де детально інформуються про урок, матеріали, які необхідно підготувати до уроку, посилання на відеоконференцію тощо. Такий функціонал дозволяє користувачам зручно керувати своїми навчальними заняттями та швидко отримувати доступ до необхідної інформації.

Для завантаження навчальних матеріалів (репетитором) або домашніх завдань (учнем), користувачу потрібно здійснити одну з двох опцій:

1. Натиснути на кнопку "Вибрати" та обрати відповідний файл зі свого комп'ютера.

2. Перетягнути відповідний файл зі свого комп'ютера на відповідну область для завантаження.

Після виконання одного з цих кроків, завантажений файл буде відображений для попереднього перегляду (додаток У).

Якщо все коректно, користувач повинен натиснути кнопку "Завантажити". Після цього файл буде успішно завантажений на сервер та стане видимим в списку файлів на сторінці онлайн уроку (додаток Ф).

На сторінці «Налаштування» (додаток Х) представлені наступні параметри для редагування:

1. Фото профіля: Користувач може змінити своє зображення-профілю, вибравши нове фото зі свого комп'ютера або завантажуючи її з іншого джерела.

2. Ім'я: Можливість зміни імені, яке відобразатиметься у профілі користувача.

3. Адреса електронної пошти: Користувач може змінити свою електронну адресу, що використовується для входу в систему.

4. Пароль: Користувач може змінити пароль, що використовується для входу в систему.

5. Опис: Можливість додавання або редагування опису профілю (для репетитора), де користувач може представити себе або надати будь-яку іншу інформацію.

Ці параметри дозволяють користувачам персоналізувати свій профіль на платформі та змінювати основні дані для кращого використання системи.

Для репетиторів доступна сторінка "Мій профіль" (додаток Ц), яка містить повну інформацію про репетитора. Ця сторінка є особистим простором, де репетитор може представити себе та свою робочу діяльність. Вона включає в себе такі розділи:

На початку сторінки відображається ім'я репетитора та опис профілю репетитора. Тут репетитор може представити себе, описати свої навички та

досвід роботи, а також надати іншу важливу інформацію, яка може бути корисною для студентів.

Далі слідує розділ з 3 вкладками: «Розклад», «Відгуки» та «Уроки».

На вкладці «Розклад» представлений розклад роботи репетитора. Це дозволяє студентам знати, коли репетитор доступний для проведення уроків.

Далі на вкладці «Відгуки», відображаються всі відгуки, які залишають студенти про роботу репетитора. Це допомагає іншим користувачам отримати інформацію про якість його роботи.

На вкладці «Уроки» репетитор може відредагувати, оновити або видалити дисципліни зі свого переліку, щоб відповідати своїм поточним можливостям та інтересам.

Завершує сторінку рейтинг репетитора на платформі. Який відображає загальну оцінку та популярність репетитора серед користувачів. Чим вищий рейтинг, тим більше довіри він має серед студентів.

Висновки до третього розділу

Проект включає реалізацію як клієнтської, так і серверної частини для веб-орієнтованої системи занять з репетитором. Для створення клієнтської частини було створено проект React. На серверній частині, для реалізації операцій CRUD, використовувався фреймворк Nest. Для взаємодії з API використовувалась бібліотека Axios.

Така архітектура дозволяє забезпечити ефективну роботу системи, включаючи навігацію між сторінками, обробку даних користувачів та взаємодію з сервером для отримання та збереження необхідної інформації.

Однією з ключових функцій додатку є можливість бронювання онлайн уроків з репетиторами, що передбачає пошук, вибір репетитора та узгодження часу уроку.

Учні мають можливість вибору дати та часу для уроку, після чого підтверджують обрані параметри та переходять до оплати (додаток Н). На запланований час уроку користувач та репетитор зустрічаються в віртуальному класі для проведення уроку з використанням засобів відеоконференційного зв'язку та додаткових навчальних матеріалів.

Додатково, в системі реалізовано особистий календар з розкладом занять, що дозволяє ефективно організувати час і планувати навчальні заняття. Також надано можливість комунікації між репетитором та учнем за допомогою особистого чату, що сприяє зручності та ефективності навчання.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проведено аналіз існуючих систем для онлайн-навчання, що дозволило з'ясувати основні переваги та недоліки таких платформ. На основі цього аналізу була розроблена концептуальна модель веб-платформи для занять з репетиторами, яка враховує найбільш ефективні підходи та потреби користувачів.

Далі, розпочався етап розробки програмного забезпечення. Було спроектовано базовий функціонал системи, включаючи реєстрацію, авторизацію, налаштування профілю, пошук та вибір репетитора, проведення онлайн-уроків, чат для спілкування між користувачами та інші основні функції.

Наступним кроком було проектування бази даних. Структура бази даних була створена з урахуванням потреб системи та оптимальної організації даних. Це дозволило забезпечити ефективне зберігання та обробку інформації про репетиторів, учнів та занять.

Реалізація веб-додатку включала створення як клієнтської, так і серверної частин. На клієнтській стороні була використана бібліотека React для створення користувацького інтерфейсу, зручного для використання та навігації. На серверній частині було використано фреймворк Nest для реалізації операцій створення, читання, оновлення та видалення даних. Для забезпечення взаємодії між сервісами була використана бібліотека Axios.

В результаті була розроблена ефективна та зручна система для навчання з репетиторами в онлайн-середовищі. Цей процес вимагав багато досліджень, розробки та тестування, але на виході була створена продуктивна та корисна платформа, яка може значно полегшити процес навчання для учнів та репетиторів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Шило С. Г. Щербак Г.В. Огурцова К.В. Інформаційні системи та технології: навч. посіб. Вид. ХНЕУ, 2021. – 220 с.
2. Особливості веб-орієнтованих інформаційних систем веб-сайт. URL: <https://studfile.net/preview/5535715/page:3> (дата звернення: 11.12.2023);
3. Коннолли Т. Бегг К. Бази даних. Проектування, реалізація та супроводження. 3 видання: Пер. С англ.: Видавничий дім «Вільямс», 2019. 844с.
4. Дідух О.В., Дяченко О.В. Розробка веб-додатків: навч. посіб. Київ: «Вища школа», 2022 308с.
5. Веб-технології. веб-сайт. URL: <https://plarson.ru/technologies.html> (дата звернення: 01.12.2023);
6. Основи розробки баз даних. веб-сайт. URL: <https://support.microsoft.com/uk-ua/office/основи-розробки-баз-даних-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5> (дата звернення: 11.11.2023);
7. Бази даних веб-сайт. URL: <https://shpora.me/Pedro/database-520> (дата звернення: 13.11.2023);
8. Шевчук О.І., Білецький В.В. Веб-технології: навч. посіб. Київ: "Вища школа", 2021 172с.
9. Петрик М. Р., Петрик О. Ю. Моделювання програмного забезпечення : науково-методичний посібник. Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. 200 с.
10. Пількевич І. А., Молодецька К. В., Сугоняк І. І., Лобанчикова Н. М. Основи побудови автоматизованих систем управління : навч. посібник. Житомир: ЖДУ ім. І. Франка, 2014. 226 с.
11. Морзе Н.В. Інформаційні системи. Навч. посіб. Івано-Франківськ, «ЛілеяНВ», – 2019. – 384 с.

12. Difference Between Static and Dynamic Web Pages. веб-сайт. URL: <https://techdifferences.com/difference-between-static-and-dynamic-web-pages.html> (дата звернення: 15.11.2023);
13. Анісімов А. В. Інформаційні системи та бази даних: навч. посіб. для студентів факультету комп'ютерних наук та кібернетики. Київ. 2017. – 110 с.
14. Web Development веб-сайт. URL: <https://www.w3schools.com/whatis/> (дата звернення: 18.11.2023);
15. Марченко А.В. Проектування інформаційних систем: навч. посіб. Хмельницький. 2015. 375с.
16. Основні етапи створення сайту. веб-сайт. URL: <https://pbb.lviv.ua/statti-inovyny/statti-shchodo-stvorennia-saitu/osnovni-etapy-stvorennia-saitu> (дата звернення: 18.11.2023);
17. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навч. посіб. Черкаси Черкаський національний університет ім. Б. Хмельницького, 2017. 434 с.
18. Павлиш В. А., Гліненко Л. К. Основи інформаційних технологій і систем: Навч. посіб. Львів: Видавництво Львівської політехніки, 2021. – 500 с.
19. Мальцев И. П. Проектування сайтів : монографія /И.П. Мальцев – Нижній Новгород : Видавництво SelfPub, 2018. – 12 с.
20. Codesido I. What is front-end development? веб-сайт. URL: <https://www.theguardian.com/help/insideguardian/2009/sep/28/blogpost>. (дата звернення: 09.03.2024);
21. Томашевський О.М. Інформаційні технології та моделювання бізнеспроцесів: навч. посіб. для студ. ВНЗ. О.М. Томашевський, Г.Г. Цегелик, М.Б. Вітер, В.І. Дубук. — К.: Центр уч. лри, 2012. — 295 с.
22. User Experience (UX): Process and Methodology. URL: <https://uiuxtrend.com/user-experience-ux-process/>

23. Соколовська З.М., Клепікова О.А. Комп'ютерне моделювання складних економічних систем : монографія. Одеса : Астропринт, 2021. 512 с. 488-489
24. Томашевський О.М. Інформаційні технології та моделювання бізнес-процесів: навч. посіб. для студ. ВНЗ / О.М. Томашевський, Г.Г. Цегелик, М.Б. Вітер, В.І. Дубук. — К.: Центр уч. лри, 2022. — 295 с.
25. Stack Overflow - Where Developers Learn, Share, & Build Careers. URL: <https://stackoverflow.com/> (дата звернення: 01.02.2024);
26. TypeScript: JavaScript With Syntax For Types. веб-сайт. URL: <https://www.typescriptlang.org/> (дата звернення: 09.03.2024);
27. PostgreSQL: The World's Most Advanced Open Source Relational Database веб-сайт. URL: <https://www.elephantsql.com/> (дата звернення: 01.04.2024);
28. ElephantSQL веб-сайт. URL: <https://www.elephantsql.com/> (дата звернення: 01.04.2024);
29. Swagger веб-сайт. URL: <https://swagger.io/> (дата звернення: 05.04.2024);
30. React.dev веб-сайт. URL: <https://react.dev/> (дата звернення: 21.03.2024);
31. PrimeReact | React UI Component Library веб-сайт. URL: <https://react.dev/> (дата звернення: 22.03.2024);
32. React Router веб-сайт. URL: <https://reactrouter.com/en/main> (дата звернення: 21.03.2024);
33. Axios Docs URL: <https://axios-http.com/ru/docs/intro> (дата звернення: 21.03.2024);
34. Документація FONDY веб-сайт. URL: <https://docs.fondy.eu/uk/docs/page/1/> (дата звернення: 21.03.2024);
35. Nestjs веб-сайт. URL: <https://nestjs.com/> (дата звернення: 29.03.2024);
36. GitHub веб-сайт. URL: <https://github.com/> (дата звернення: 05.04.2024);