ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ «ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ» Навчально-науковий інститут денної освіти Форма навчання денна Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту Завідувач кафедри _____Олена ОЛЬХОВСЬКА

«___» ____ 2023 p.

КВАЛІФІКАЦІЙНА РОБОТА

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ: ВЕБ-ІНТЕРФЕЙС

зі спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра

Виконавець роботи Комар Ілля Ігорович

_____ «____»____ 2023 p.

Науковий керівник к.пед.н., доцент, Кошова Оксана Петрівна _____ «____»____ 2023 p.

Рецензент

ПОЛТАВА 2023 р.

	3	АТВЕРДЖУ	Ю
Завідувач кафедри	Олена (ЭЛЬХОВСЬ	КА
	(підпис)		
	«»	202_	_ p.

ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему «<u>Розробка програмного забезпечення системи дистанційного навчання:</u> веб-інтерфейс»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові Комар Ілля Ігорович

Затверджена наказом ректора № <u>144</u>-Н від «<u>01</u>» <u>вересня</u> 2022 р.

Термін подання студентом роботи «____»____ 202__ р.

Вихідні дані до кваліфікаційної роботи: <u>публікації з теми, системи</u> дистанційного навчання.

Зміст пояснювальної записки (перелік питань, які потрібно розробити) ВСТУП

1. РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

2. РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

- 2.1. Платформа дистанційного навчання Moodle.
- 2.2. Платформа дистанційного навчання Blackboard.
- 2.3. Платформа дистанційного навчання Edmodo.
- 2.4. Платформа дистанційного навчання Canvas.
- 3. ТЕОРЕТИЧНА ЧАСТИНА
 - 3.1. Опис проектних рішень, інструментів та підходів до розробки.
 - 3.2. Опис методології створення програмного забезпечення.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис та побудова діаграми прецедентів роботи з системою.

4.2. Інструкція з використання системи.

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Перелік графічного матеріалу: <u>1 аркуш блок-схем, 37 ілюстрацій.</u>

Розділ	Прізвище,	Підпі	ис, дата
	ініціали, посада	завдання	завдання
	консультанта	видав	прийняв
Постановка задачі	Кошова О.П.		
Інформаційний огляд	Кошова О.П.		
Теоретична частина	Кошова О.П.		
Практична частина	Кошова О.П.		

Консультанти розділів кваліфікаційної роботи

Календарний графік виконання кваліфікаційної роботи

	Зміст роботи	Термін	Фактичне
		виконання	виконання
1.	Вступ		
2.	Вивчення методичних рекомендацій та		
ста	андартів та звіт керівнику		
3.	Постановка задачі		
4.	Інформаційний огляд джерел бібліотек та		
інт	сернету		
5.	Теоретична частина		
6.	Практична частина		
7.	Закінчення оформлення		
8.	Доповідь студента на кафедрі		
9.	Доробка (за необхідністю), рецензування		

Дата видачі завдання «	>>>	_ 202_	p.
Здобувач вищої освіти			<u>Комар Ілля Ігорович</u>
Науковий керівник	(підпис)		<u>к.пед.н., доц. Кошова О.П</u> . (науковий ступінь, вчене звання, ініціали та прізвище)
Результати захисту кв Кваліфікаційна робота	аліфікаційн опінена на	ої роб	боти
	(ő	алів, оцін	ка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «____» _____ 2023 р. Секретар ЕК______(ініціали та прізвище)

Затверджую Зав. кафедрою _____ к.ф.-м.н. Олена ОЛЬХОВСЬКА «____» ____ 202__ р.

Погоджено Науковий керівник _____ к.пед.н., Оксана КОШОВА «____» _____ 202___ р.

План

кваліфікаційної роботи на тему «<u>Розробка програмного забезпечення системи дистанційного навчання: вебінтерфейс</u>» зі спеціальності 122 Комп'ютерні науки освітня програма 122 «Комп'ютерні науки» ступеня бакалавр Прізвище, ім'я, по батькові <u>Комар Ілля Ігорович</u>

ВСТУП

1. РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

- 2. РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ
 - 2.1. Платформа дистанційного навчання Moodle.
 - 2.2. Платформа дистанційного навчання Blackboard.
 - 2.3. Платформа дистанційного навчання Edmodo.
 - 2.4. Платформа дистанційного навчання Canvas.
- 3. РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Опис проектних рішень, інструментів та підходів до розробки.

- 3.2. Опис методології створення програмного забезпечення.
- 4. РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис та побудова діаграми прецедентів роботи з системою.

4.2. Інструкція з використання системи.

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А

Здобува	ач вищої освіти	Ілля КОМАР
«»	202 p.	

РЕФЕРАТ

Записка: 57 сторінок, 37 рисунків, 1 додаток, 16 літературних джерел.

СИСТЕМА ДИСТАНЦІЙНОГО НАВЧАННЯ, WEB-ЗАСТОСУНОК, REACT, BEБ-IHTEPФЕЙС, ПРЕЦЕДЕНТ.

Мета роботи – розробка програмного забезпечення системи дистанційного навчання: веб-інтерфейс.

Об'єкт розробки – система дистанційного навчання.

Методи дослідження – використання векторного онлайн-сервісу розробки інтерфейсів та прототипування «Figma», редактор коду «Visual Studio Code» мову програмування Туре Script, Css-препроцесор Sass, систему контролю версіями Git та бібліотеки React, Redux, Redux-Saga, Zod, Ant Design.

Зроблено огляд систем дистанційного навчання, виділено їх позитивні та негативні сторони. Виконано опис проектних рішень, інструментів та підходів до розробки системи дистанційного навчання. Обрано методологію створення програмного забезпечення. Побудовано діаграму прецедентів. Розроблено вебінтерфейс системи дистанційного навчання. Розроблено інструкцію з користування системою дистанційного навчання.

ВСТУП	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ	9
РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ	13
2.1. Платформа дистанційного навчання Moodle	13
2.2. Платформа дистанційного навчання Blackboard	15
2.3. Платформа дистанційного навчання Edmodo	17
2.4. Платформа дистанційного навчання Canvas	19
РОЗДІЛ З. ТЕОРЕТИЧНА ЧАСТИНА	21
3.1. Опис проектних рішень, інструментів та підходів до розробки	
3.2. Опис методології створення програмного забезпечення	32
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА	34
4.1. Опис та побудова діаграми прецедентів роботи з системою	34
4.2. Інструкція з використання системи	37
ВИСНОВКИ	47
СПИСОК ЛІТЕРАТУРИ	48
ДОДАТОК А. ВИХІДНІ КОДИ	50

3MICT

ВСТУП

відбувається Актуальність. Останнім часом розвиток технологій З розвитком Інтернет-зв'язку швидкими темпами. відкриваються нові можливості для освітнього процесу. Інтернет надає необмежені можливості студенту для отримання необхідної інформації. І вже зараз перед людством постала проблема вдосконалення системи освіти, оскільки необхідно готувати спеціалістів гідних свого часу. Одним з варіантів вирішення цієї проблеми є використання системи дистанційного навчання. Дистанційна форма навчання найбільш відповідає сучасному рівню розвитку суспільства. На відміну від класичних форм отримання освіти, дистанційне навчання здійснюється з використанням всіх новітньої технічних досягнень в сфері телекомунікаційних технологій і мережі Інтернет. Системи віддаленого навчання дозволяють навчати як школярів, студентів, так і працюючих людей без відриву від виробництва. Дистанційні освітні технології з використанням Інтернету застосовуються як для освоєння окремих курсів підвищення кваліфікації користувачів, так і для здобуття вищої освіти.

Основною перевагою дистанційної форми навчання над очною формою є можливість отримання освіти з будь-якого місця, де є доступ до Інтернету. Це дає студентам можливість отримувати знання зручним для них способом, що дозволяє їм підлаштовувати навчання до свого графіку, стилю життя і особистих потреб. Крім того, дистанційна форма навчання дозволяє студентам самостійно визначати темп і глибину свого навчання, що забезпечує більш гнучкий підхід до навчання. Також дистанційна форма навчання може бути менш витратною, оскільки не потрібно сплачувати витрати на проживання та харчування в університеті чи коледжі.

Таким чином, дистанційне навчання має ряд переваг над традиційною освітою, а саме:

- дозволяє студентам здобувати освіту, не покидаючи свій дім;

- надає доступ до електронних матеріалів і відеозаписів;

використання онлайн-інструментів, за допомогою яких студенти можуть
 зв'язуватися з іншими студентами та викладачами;

 – дистанційне навчання може бути доступним для студентів з інвалідністю або тим, хто має обмеження в русі;

 – дистанційне навчання може бути менш коштовним для студентів, оскільки вони не повинні платити за житло, проїзд та інші витрати, пов'язані з перебуванням в університеті;

- використання нових технології та методів навчання;

– можливість працювати та навчатися.

Мета роботи є розробка програмного забезпечення системи дистанційного навчання: веб-інтерфейс.

Об'єктом розробки є система дистанційного навчання.

Предметом розробки є програмне забезпечення системи дистанційного навчання з використанням JavaScript бібліотеки «React» для створення інтерфейсів користувача.

Перелік використаних методів полягає у проектування системи дистанційного навчання.

При реалізації проекту використано: векторний онлайн-сервіс розробки інтерфейсів та прототипування «Figma», редактор коду «Visual Studio Code» мову програмування Туре Script, Css-препроцесор Sass, систему контролю версіями Git та бібліотеки React, Redux, Redux-Saga, Zod.

Пояснювальна записка складається з чотирьох розділів, а саме: постановка задачі, огляд систем аналогічного призначення, теоретичної та практичної частини. Структура побудована так, що дає змогу логічного представлення матеріалу та розкриття теми роботи.

Обсяг пояснювальної записки: 57 сторінок, 37 рисунків, 1 додаток, 16 літературних джерел.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Розробку веб-інтерфейсу системи дистанційного навчання можна поділити на наступні етапи:

- розробка технічного завдання;
- розробка макету дизайну сайту;
- верстка та програмування;
- наповнення контентом;
- тестування.

Технічне завдання представляє собою фундамент для подальшої роботи. В ньому прописуються всі деталі: структура або мапа сайту (кількість сторінок, розділів, категорій, блоків), вимоги стосовно дизайну, функціонального, візуального та текстового наповнення, а також технічні можливості.

Макет дизайну сайту – це візуальне відображення того, як веб-сайт буде виглядати перед його розробкою та реалізацією. Він може містити такі елементи, як макети сторінок, типографію, колірну гаму, фотографії, іконки та інші елементи дизайну.

Верстка та програмування – це процес створення HTML-коду та CSSстилів для веб-сайту на основі макету дизайну. Під час верстки веб-сайту розробник створює HTML-структуру сторінок та встановлює стилі, які визначають вигляд та розміри елементів на сторінках.

Наповнення контентом веб-сайту – це процес додавання текстового, візуального та іншого типу контенту на веб-сайт. Це можуть бути статті, новини, фотографії, відео, аудіо, інфографіка, електронні книги, інформаційні графіки та інше. Додавання якісного та релевантного контенту на веб-сайт допомагає залучати аудиторію та збільшувати її зацікавленість в вашому продукті чи послузі. Контент допомагає підтримувати відвідувачів на веб-сайті, допомагає вивчати їх потреби та інтереси, вирішувати їх проблеми та забезпечує зв'язок з аудиторією. Тестування – це процес перевірки функціональності, продуктивності, безпеки та інших аспектів веб-додатку перед його випуском або оновленням. Під час тестування веб-додатку перевіряються різні аспекти, такі як коректність роботи функцій, швидкість відгуку веб-сторінок, відповідність дизайну, безпека, сумісність з різними браузерами та пристроями [1].

Розроблений веб-додаток системи дистанційного навчання повинен мати такі сторінки:

головна сторінка;

мої курси;

– деталі курсу;

 налаштування курсу (матеріали, студенти, журнал оцінок, редагувати курс);

– журнал оцінок;

- зустрічі;

налаштування профілю.

Головна сторінка. На головній сторінці повинно бути реалізовано можливість перегляду всіх власних курсів, розкладу на сьогодні та проведення зустрічей. Для перегляду інформації про курс повинна бути можливість натискання на потрібний курс, в результаті чого користувач потрапляє на сторінку з деталями курсу. Щоб розпочати зустріч повинна бути можливість вибору потрібного курсу.

Мої курси. Повинна бути реалізована сторінка для перегляду всіх курсів у вигляді таблиці з відповідними заголовками: назва курсу, група, спеціальність, дата створення курсу, дата оновлення курсу.

Для додавання нового курсу повинна бути відповідна кнопка, після натискання якої, буде відкриття модального вікна з відповідними полями: назва курсу, група та опис.

Деталі курсу. Після того, як користувач вибрав курс повинна відкриватися нова сторінка з відповідним курсом. На цій сторінці повинна бути

можливість перегляду шапки курсу (назва курсу, група, опис та картинка курсу), тем та матеріалів (лекцій та практичних). Для перегляду матеріалів повинна бути можливість натискання на потрібну тему, після чого повинна з'являтися лекція або практична. Необхідно реалізувати можливість завантажування матеріалу.

Щоб здати практичну роботу повинна бути відповідна кнопка, після натискання якої, буде відкриватися модальне вікно з відповідною формою для прикріплення файлів.

Для перегляду зданих робіт повинна бути можливість натискання на потрібну практичну роботу, після чого повинна з'явитися таблиця з відповідними заголовками: П.І.Б. користувача, дата здачі, оцінка, стан перевірки роботи. Для того щоб перевірити роботу, повинна бути можливість натискання на відповідну роботу, після чого повинно з'являтися модальне вікно, з відповідними полем для виставлення оцінки.

Налаштування курсу. Повинна бути реалізована сторінка для перегляду налаштування курсу з відповідними вкладками: матеріали, студенти, журнал оцінок, розклад та налаштування.

Для перегляду матеріалів курсу повинна бути можливість вибору вкладки матеріали, після чого повинні з'явитися дві таблиці. Перша таблиця повинна мати заголовки: назва теми, дата створення теми, дата оновлення та дії (можливість редагувати та видаляти тему). Друга таблиця повинна мати заголовки: назва матеріалу, тип матеріалу, дата створення, дата оновлення та дії (можливість редагувати та видаляти матеріалу).

Для перегляду студенів курсу повинна бути можливість вибору вкладки студенти, після чого повинно з'явитися таблиця з відповідними заголовками, а саме: П.І.Б студента, пошта та можливість видалити студента.

Для перегляду журналу оцінок курсу повинна бути можливість вибору вкладки журнал оцінок, після чого повинно з'явитися форма вибору студента та таблиця з результатом вибору з відповідними заголовками, а саме: назва практичної роботи та оцінка.

Для перегляду налаштування курсу повинна бути можливість вибору вкладки налаштування, після чого повинна з'явитися форма, з відповідними полями: назва курсу, опис курсу та оформлення.

Журнал оцінок. Повинна бути реалізована сторінка для перегляду журналу оцінок з формами вибору курсу та студента, після чого повинна з'явитися таблиця з результатами вибору.

Зустрічі. Повинна бути реалізована сторінка для проведення зустрічей, де користувач повинен мати можливість розпочати або зупинити зустріч. Для того, щоб розпочати зустріч користувачу потрібно ввести ім'я та скопіювати посилання на зустріч.

Налаштування профілю. Повинна бути реалізована сторінка для редагування профілю, де користувач повинен мати можливість завантажити або замінити зображення профілю.

РОЗДІЛ 2. ОГЛЯД СИСТЕМ АНАЛОГІЧНОГО ПРИЗНАЧЕННЯ

На даний момент у світі існує значне число систем дистанційного навчання, й далі ми розглянемо деякі з них.

2.1. Платформа дистанційного навчання Moodle

Moodle – це відкрите програмне забезпечення для створення та управління електронними курсами та навчальними матеріалами в онлайнсередовищі. За допомогою Moodle можна створювати віртуальні класи, де викладачі можуть розміщувати матеріали для навчання, виставляти завдання та тестування для студентів, проводити дистанційні заняття та спілкуватися з ними (Рисунок 2.1).





Мооdle має багатий функціонал, який дозволяє налаштувати курси під потреби конкретного навчального закладу та забезпечити ефективне навчання. Він є популярним інструментом для онлайн-навчання та дистанційного навчання, який дозволяє забезпечити доступ до якісної освіти у будь-якому місці та в будь-який час. Він також має розгалужену глобальну спільноту розробників та користувачів, яка постійно працює над вдосконаленням системи та надає підтримку користувачам. Багато навчальних закладів у світі використовують Moodle для забезпечення онлайн-навчання та підвищення якості освіти. (Рисунок 2.2).

My second course			¢
В начало / Курсы / Miscellaneous / My second course			 Редактировать настройки Завершить редактирование Фильтом
🕂 🖷 News forum 🌶	Редактировать - Редактировать - 🌲 + Добавить элемент или ресурс	Поиск Расшири	 Настройка журнала оценок Резервное копирование Восстановить Импорт Очистка
июль - 2 Август	Редактировать – + Добавить элемент или ресурс	Ф Больше Послед Добавить новую тему (Пока объявлений нет)	
	Редактировать - + Добавить элемент или ресурс	Предстоящие события 🕂	
+ 10 Август - 16 Август	Редактировать = + Добавить элемент или ресурс	Нег предстоящих событий Перейти к календарю	
+ 17 Август - 23 Август 🖌	Редактировать - + Добавить элемент или ресурс	 Последние действия Действия с Вторник, 5 Ноябрь 201 Полный отчет о последних дейс Со времени Вашего последнего вход не произошло 	
+ 24 Август - 30 Август ≠	Редактировать - + Добавить элемент или ресурс		

Рисунок 2.2 Інтерфейс платформа дистанційного навчання «Moodle» Типова функціональність Moodle включає:

- здача завдань;
- дискусійні форуми;
- завантаження файлів;
- оцінювання;
- обмін повідомленнями;
- календар подій;
- новини та анонси подій;
- онлайн тестування.

Основні можливості системи Moodle.

Для студентів:

 мати доступ до курсів, де можуть переглядати навчальний матеріал, виконувати завдання та інтерактивні вправи;

 відстежувати свій прогрес в навчанні та переглядати свої оцінки за завдання, тести та інші види робіт;

- виконувати завдання та тести, створені викладачами, та надсилати їх для оцінювання; обговорювати матеріал курсів, задавати запитання викладачам та іншим студентам, обмінюватися думками та досвідом, використовуючи форуми та чати;

- використовувати мобільний додаток Moodle для отримання доступу до курсів;

 налаштовувати свій профіль в Moodle, включаючи зміну пароля, зміну ім'я та інформації про себе;

- можливість завантажувати свої документи до системи.

Для викладачів:

 створювати курси та додавати до них матеріали для навчання, а саме: лекції, завдання, тести, відео, аудіо;

 додавати нових користувачів, призначати ролі та права доступу, контролювати активність користувачів;

- створювати завдання та тести, щоб перевірити знання студентів;

- створювати форуми, щоб сприяти дискусіям та спілкуванню між студентами;

- оцінювати роботи студентів та стежити за їх успішністю в навчанні;

створювати події та зустрічі в календарі;

- спілкуватися зі студентами через систему повідомлень [2].

2.2. Платформа дистанційного навчання Blackboard

Blackboard – це платформа для дистанційного навчання, яка надає навчальні матеріали та інструменти для спілкування між студентами та викладачами в онлайн-середовищі. Blackboard використовується в багатьох вищих навчальних закладах та школах по всьому світу.

Вlackboard дозволяє викладачам завантажувати матеріали до курсів, такі як лекції, вправи, тестові завдання, розділи підручників та інші ресурси. Студенти мають доступ до цих матеріалів з будь-якого місця та в будь-який час, якщо вони мають доступ до Інтернету (Рисунок 2.3).

Blackboard	Courses		🛅 Course Catalog
f Institution Page	- Demonstration Courses	Current Courses 🔻	Upcoming Courses $ ightarrow$
Activity Stream	Q Search your courses	Filter All Courses ¥	25 🔹 items per page
Courses	Favorites		
	- september	Winite a biskin the line is the	
🛗 Calendar			H and the second se
Messages	Astronomy 212	Chemistry 101	English Composition
🖶 Grades	Multiple Instructors 👘 🛈	••• Multiple Instructors 🔶 🛈 •••	• Multiple Instructors 🛊 🛈 …
💥 Tools		2	
	THE CONTRACTOR	B Stall	
			and and a second se
	Food Science ID: Wade, Food	Fundamentals of Artificial Intelligence ID: Wade_102	Introduction to Anatomy ID: AST_101_wfields
	Multiple Instructors 🔶 😨	••• Multiple Instructors 🛊 🛈 •••	• Multiple Instructors 🛊 🛈 …
	2018 Spring		
	and the second s		
			D

Рисунок 2.3 – Інтерфейс платформи дистанційного навчання «Blackboard»

Система Blackboard має багато можливостей для викладачів та студентів: – створювати курси в системі та керувати ними, додавати та видаляти матеріали, створювати завдання та тести, встановлювати дедлайни та інші параметри;

можливість легко знайти та отримати доступ до матеріалів курсів,
 включаючи лекції, вправи, підручники та інші ресурси;

 - інструменти для взаємодії між викладачами та студентами, такі як форуми для дискусій, завдання для роботи в групах та інші інтерактивні завдання;

автоматизована підготовка звітів;

- сучасний мобільний додаток, що дозволяє отримувати доступ до курсів та матеріалів на смартфонах та планшетах;

 можливість змінювати зображення курсу, робити курс приватним або приховати його;

- функція попереднього перегляду курсу;

- швидка технічна підтримка для викладачів та студенів;

 можливість відстежувати прогрес студентів та покращувати процеси навчання.

Вартість залежить від кількості користувачів і становить від 38 000 \$ [3].

2.3. Платформа дистанційного навчання Edmodo

Edmodo – це платформа для навчання, яка дозволяє викладачам та студентам спілкуватися та співпрацювати над завданнями в режимі онлайн. За допомогою Edmodo викладачі можуть створювати курси, додавати матеріали та давати завдання студентам, а студенти можуть взаємодіяти між собою та з викладачами, обговорювати матеріали та виконувати завдання (Рисунок 2.4).



Рисунок 2.4 – Платформа дистанційного навчання «Edmodo»

Еdmodo була створена у 2008 році як платформа для забезпечення спілкування та співпраці в межах класу, але з того часу стала популярною серед вчителів та учнів у всьому світі. Платформа пропонує безкоштовну та платну версії, в залежності від потреб користувачів. У безкоштовній версії доступні основні функції, такі як створення курсів, завдань та обговорень, а також інструменти для спілкування між вчителем та учнем. У платній версії доступні додаткові функції, такі як інтеграція з системами управління навчанням, інструменти для аналізу даних та звітності, а також можливості для батьків та адміністраторів.

Платформа також надає інструменти для моніторингу прогресу та оцінювання студентів. Edmodo дуже популярна серед вчителів та учнів у всьому світі, особливо в початковій та середній школі. Edermo можна використовувати, як у десктопні версії, так і в додатку, доступному для Android та IOS. Перш ніж створювати класи на Edmodo, вчителі повинні створити обліковий запис, який може бути пов'язаний з їхніми існуючими обліковими записами від Google або Facebook. (Рисунок 2.5).



Рисунок 2.5 - Створення облікового запису в «Edmodo»

Однією з найбільших переваг Edmodo є те, що учням і батькам не потрібно ділитися своєю особистою інформацією або електронною адресою для створення облікового запису. Все, що їм потрібно зробити, це ввести своє ім'я, прізвище, пароль і ввести код класу, який повинен надати вчитель. Цей код автоматично генерується Edmodo при створенні нової групи.

За допомогою свого облікового запису є можливість створювати групи і запрошувати до них учнів та батьків. У групах є можливість ділитися ресурсами, фотографіями та статтями, щоб потім учні змогли бачити у форматі стрічки новин. Є також можливість надсилати та отримувати приватні повідомлення і створювати папки, куди учні можуть завантажувати завдання. Всі матеріали автоматично зберігаються в розділі «Моя бібліотека».

Переваги системи:

моніторинг прогресу;

- безпека та конфіденційність;

- зручність використання;

- заохочення співпраці між учнями;

- надання індивідуального зворотного зв'язку [4].

2.4. Платформа дистанційного навчання Canvas

Canvas – це безкоштовна платформа управлінням навчанням (LMS). Курси легко інтегрувати у змішане навчання або використовувати як самостійний освітній продукт. Ця платформа надає вільний хостинг та зручні інструменти для розробки уроків та курсів (Рисунок 2.6). Canvas розробила компанія Instructure, і на даний момент вона є одним з найпопулярніших LMS у світі.



Рисунок 2.6 – Інтерфейс платформи дистанційного навчання «Canvas» Canvas LMS – це відкрита та надійна платформа, яка дозволяє навчальним закладам керувати цифровим навчанням, викладачам створювати та презентувати навчальні матеріали та оцінювати знання студентів, а студентам – брати участь у курсах та отримувати зворотній зв'язок про розвиток навичок та навчальних досягнень.

Платформа Canvas є досить простою та інтуїтивно зрозумілою у використанні, а також підтримує широкий спектр інтеграцій з іншими програмними продуктами та сервісами, такими як Google Docs, Zoom, Microsoft Teams та інші.

Основні переваги Canvas:

– додаток Canvas є доступний для завантаження на iOS та Android;

складати тести з різними налаштуваннями;

 легко здавати завдання та тести, включаючи можливість завантажувати файли, відповіді на запитання та отримувати миттєві результати тестів;

- додавати документи та файли;

 підтримує багато мов, що дозволяє використовувати його в різних країнах та для різних аудиторій;

- дивитися аналітику за курсом та за окремими студентами;

- проводити конференції;

- легко інтегрувати з зовнішніми LTI-застосунками.

Для зручності користувачів платформа оснащена вбудованим календарем, поштовою скринькою для внутрішньої комунікації, вітриною курсів. У разі виникнення будь-яких питань технічного характеру є можливість запросити допомогу або знайти відповідь, використовуючи вбудований інструмент для пошуку потрібної інформації.

Окрім можливості створення своїх авторських курсів на платформі є можливість скористатися курсами інших вчителів [5].

РОЗДІЛ З. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Опис проектних рішень, інструментів та підходів до розробки.

За результатами аналізу систем аналогічного призначення, проведеного в першій частині нашої роботи, нами було обрано для розробки системи дистанційного навчання:

- сервіс для розробки інтерфейсів та прототипування Figma;
- мову програмування ТуреScript;
- бібліотеки: React, Redux, Redux-Saga, Zod, Ant Design;
- Css-препроцесор Sass;
- редактор коду Visual Studio Code;
- система контролю версіями Git.

Figma – це веб-платформа та програма для розробки інтерфейсів користувача та векторної графіки (Рисунок 3.1). Даним сервісом можна користуватись як онлайн через браузер, так і встановивши програму на ПК.



Рисунок 3.1 – Сервіс для розробки інтерфейсів та прототипів «Figma»

Figma є відкритою програмою, що дозволяє користувачам робити спільну роботу над проектами в реальному часі. Вона використовується для розробки веб-сайтів, мобільних додатків, а також графічного дизайну. Вона використовується для розробки веб-сайтів, мобільних додатків, а також графічного дизайну.

Figma доступна як онлайн-платформа, а також як програма для Windows та MacOS. Її інтерфейс є досить інтуїтивно зрозумілим та легким у використанні, що дозволяє швидко розпочати роботу з нею. Figma також інтегрується з іншими популярними інструментами для розробки, такими як Sketch, Adobe XD та Zeplin, що дозволяє легко імпортувати та експортувати проекти.

За допомогою Figma можна створювати:

макети веб-сторінок, включаючи різні елементи, такі як: заголовки, кнопки, іконки, форми та інші;

макети мобільних додатків для iOS і Android, включаючи елементи, такі як:
 панелі навігації, кнопки, форми вводу даних та інші;

 графічних елементів, таких як логотипи, іконки, банери, пости в соціальних мережах.

Можливості Figma:

 можливість створювати дизайн-макети веб-сторінок, мобільних додатків та інтерфейсів користувача для різних пристроїв;

– змінювати розмір, обрізати, редагувати та оптимізувати зображення в Figma, щоб вони відповідали вимогам дизайну;

 спільно працювати над проектами з колегами, ділитися файлами та коментувати їх, що дозволяє ефективно координувати роботу команди;

 створювати інтерактивні прототипи веб-сторінок та мобільних додатків, що дозволяє тестувати їх функціональність перед розробкою;

 створювати і зберігати стилі та компоненти, що дозволяє швидко та легко змінювати дизайн елементів у всьому проекті;

підтримка інтеграцій з іншими інструментами, такими як Sketch, Adobe XD,
 Jira, Trello та інші.;

 можливість керування версіями проекту, що дозволяє відслідковувати зміни та повертатися до попередніх версій проекту в разі потреби; можливість експортувати проекти в різні формати, такі як PNG, JPG, SVG,
 PDF та інші, що дозволяє легко використовувати дизайн у різних контекстах
 [6].

ТуреScript – це мова програмування, яка є розширенням мови JavaScript. Вона була розроблена компанією Microsoft з метою полегшення розробки великих та складних проектів (Рисунок 3.2). Позиціонується, як засіб розробки веб-застосунків, що розширює можливості JavaScript.



Рисунок 3.2 – Логотип мови програмування «TypeScript»

Однією з головних переваг TypeScript є те, що вона підтримує строгу типізацію, тобто вона дозволяє вказувати типи даних для змінних, функцій та інших об'єктів. Це забезпечує більшу безпеку та стабільність коду, оскільки під час розробки помилки в типах даних будуть виявлятися вже на етапі компіляції, а не під час виконання програми.

Також TypeScript має багато інших корисних функцій, таких як: розширення мови JavaScript, що дозволяє використовувати класи, інтерфейси, декоратори та інші елементи ООП; можливість підключення до будь-якої JSбібліотеки та фреймворка; автоматичну генерацію документації та багато іншого.

Основна різниця між TypeScript та JavaScript полягає у тому, що TypeScript є надмножиною JavaScript з додатковими функціональними можливостями.

Переваги TypeScript:

- статична типізація;

підтримка ООП;

- крос-платформенна та крос-браузерна сумісність;
- раннє виявлення помилок;
- передбачуваність під час компіляції кода;
- швидкий рефакторинг;
- підтримка найновіших функції JavaScript [7].

React – це JavaScript-бібліотека для розробки користувацьких інтерфейсів. Головною перевагою React є його компонентний підхід. Всі елементи інтерфейсу можуть бути розбиті на невеликі, самодостатні компоненти, які можна повторно використовувати в різних частинах додатку (Рисунок 3.3).



Рисунок 3.3 – Логотип JavaScript-бібліотеки «React»

За допомогою React можна створювати різні типи веб-додатків, включаючи:

 створення статичних та динамічних веб-сайтів з високим рівнем інтерактивності та розширюваності;

 односторінкових додатків (SPA), які забезпечують швидку відповідь на дії користувачів та покращення користувацького досвіду;

- нативні мобільні додатки для iOS та Android;

- додатків для роботи з базами даних та іншими джерелами даних.

- дашбордів (статистика з елементами інфографіки).

- React найкраще підходить для:
- адмін-панелей;
- дашбордів (статистика з елементами інфографіки);

- CRM-систем;

- додатків зі складною логікою;

- інтернет-магазинів і маркетплейсів;

- соціальних мереж.

Основні особливості розробки web-додатків за допомогою бібліотеки React:

– Ефективне відображення. React використовує віртуальні DOM (Document Object Model), що дозволяє ефективно оновлювати лише ті частини вебсторінки, які потребують змін, замість повного оновлення всієї сторінки.

 Компонентний підхід. React використовує компонентний підхід до розробки інтерфейсу, що дозволяє розбити складний інтерфейс на невеликі, повторно використовувані компоненти. Це спрощує розробку, тестування та підтримку веб-додатків.

– Багато екосистема. React має велику екосистему інструментів, бібліотек та пакетів, таких як Redux, React Router, Material-UI та багато інших, які допомагають розширити можливості додатку і спростити розробку.

– Широке застосування. React може бути використаний для розробки різноманітних веб-додатків, від простих односторінкових сайтів до складних веб-додатків.

 Активна спільнота. React має велику та активну спільноту розробників, що забезпечує постійний розвиток фреймворку, підтримку, документацію та розширення функціональності.

– Підтримка SEO. React надає можливості для серверного рендерингу (Serverside Rendering, SSR), що допомагає в покращенні індексації веб-сторінок пошуковими системами [8].

Redux – це бібліотека управління станом для JavaScript додатків, яка забезпечує простий та передбачуваний спосіб управління станом додатку в одному місці (Рисунок 3.4). Використовується разом з React або Angular для побудови інтерфейсів.



Рисунок 3.4 – Логотип JavaScript-бібліотеки «Redux»

Redux спеціально створений для управління станом додатку великого масштабу, такого як веб-додатки. Це робить його особливо корисним для додатків, які мають складну логіку взаємодії між компонентами та велику кількість даних, що потребують управління.

Redux дозволяє керувати станом додатку в єдиному місці і робить зміни в ньому більш передбачуваними і відстежуваними (Рисунок 3.5). Це полегшує міркування про зміни, що відбуваються у вашому додатку. Але всі ці переваги мають свої недоліки та обмеження. Комусь може здатися, що це додає шаблонного коду, роблячи прості речі трохи перевантаженими, але це залежить від архітектурних рішень.



Рисунок 3.5 – Основна концепція роботи «Redux»

Переваг використання Redux для управління станом додатку:

– Однозначність та передбачуваність управління станом додатку. Усі зміни в стані додатку відбуваються через дії (actions), що відправляються до стору.

– Можливість легко розширювати функціональність додатку за допомогою middleware, що дозволяє виконувати додаткові дії при обробці дій.

Легко тестувати функціональність додатку, оскільки весь стан зберігається в одному місці.

– Можливість обробки асинхронних запитів, що є необхідним у багатьох додатках.

- Підтримка для React, що робить його ідеальним для використання з Reactдодатками [9].

Redux-saga – це бібліотека для управління побічними ефектами в Redux додатках, таких як асинхронні запити на сервер, обробка таймерів, маніпулювання браузерними подіями тощо (Рисунок 3.6).



Рисунок 3.6 – Логотип бібліотеки «Redux-Saga»

Використовує генератори ES6 для створення – саг це функції, які слідкують за певними діями, що відправляються в Redux та запускають побічні ефекти, коли відповідна дія відбувається.

Redux-Saga можна розглядати як організований спосіб допомогти додатку, створеному за допомогою Redux, спілкуватися і синхронізуватися із зовнішніми залежностями, такими як API. Загалом, це бібліотека проміжного програмного забезпечення, яку можна використовувати для HTTP-запитів до зовнішніх сервісів, виконання операцій вводу/виводу та доступу до сховища браузерів. Сила Redux-Saga полягає в організації та управлінні побічними ефектами таким чином, щоб розробнику було легше ними керувати [10].

Переваги використання Saga:

- підтримка асинхронних операцій;

- просте тестування;
- зручне управління помилками;
- керування складними послідовностями;
- паралельне виконання завдань.

Zod – це бібліотека для валідації даних в JavaScript. Вона дозволяє описувати схеми даних та перевіряти, чи відповідає конкретний об'єкт або значення цим схемам. (Рисунок 3.7). За допомогою Zod можна легко визначати схеми даних, а потім перевіряти дані на відповідність цим схемам, щоб переконатися, щоб вони відповідають вимогам.



Рисунок 3.7 – Логотип TypeScript-бібліотеки «Zod»

Zod має безліч можливостей, які роблять його популярним вибором серед розробників TypeScript. Ось лише деякі з його найпомітніших можливостей:

- статична типізація;
- перевірка схеми;
- користувацькі повідомлення про помилки;
- асинхронна перевірка;
- підтримка вкладених об'єктів та масивів [11].

Ant Design – це бібліотека компонентів інтерфейсу користувача для розробки веб-додатків (Рисунок 3.8). Бібліотека написана на TypeScript, в якій визначені всі необхідні типи. Ant Design був розроблений компанією Alibaba Group. Дизайн Ant базується на популярному стеку React, Redux, React-Router, який активно використовується в проектах Alibaba [12].



Рисунок 3.8 – Бібліотека компонентів «Ant Design»

Ant Design містить в собі багато готових компонентів, таких як: кнопки, форми, таблиці, меню, модальні вікна, картки та інші. Кожен компонент Ant Design має зручний API та може бути налаштований залежно від потреб проекту.

Ant Design має багато переваг:

- готові компоненти;
- підтримка інтернаціоналізації для десятків мов;
- адаптивний дизайн;
- підтримка 24/7.

Sass – це мова для збільшення можливостей звичайних CSS-файлів. Sass надає програмістам можливість використовувати змінні, функції, оператори та інші функції, які допомагають покращити організацію та повторне використання стилів (Рисунок 3.9).



Рисунок 3.9 - Логотип метамови «Sass»

Основні переваги Sass:

- Sass повністю сумісна з CSS.
- Стабільніший, потужніший за CSS.

– Заснований на JavaScript і є підмножиною CSS.

- Має власний синтаксис і компілюється в читабельний CSS.

– Підтримує розширення мови, такі як змінні, вкладеність та міксини.

– Надає багато корисних функцій для маніпулювання кольорами та іншими значеннями.

 Надає багато розширених можливостей, таких як керуючі директиви для бібліотек [13].

Visual Studio Code – це безкоштовний текстовий редактор, створений компанією Microsoft. Він має відкритий вихідний код та працює на різних платформах, таких як Windows, macOS та Linux (Рисунок 3.10). VS Code є дуже популярним серед розробників, оскільки він є легким, швидким та має багато корисних функцій, які допомагають розробникам працювати з кодом більш ефективно.



Рисунок 3.10 - Текстовий редактор «Visual Studio Code»

VS Code надає користувачам можливість працювати з різними мовами програмування, такими як JavaScript, Python, PHP, Java, C#, а також з іншими мовами за допомогою розширень. Він має багато корисних функцій, таких як

автодоповнення, рефакторинг коду, дебагер, вбудований термінал та можливість інтеграції з різними інструментами розробки, такими як Git [14].

Git – це система керування версіями файлів, яка дозволяє зберігати та відстежувати зміни в коді проекту. Git був створений Лінусом Торвальдсом для розробки ядра Linux, а зараз використовується у багатьох інших проектах (Рисунок 3.11).



Рисунок 3.11 – Логотип системи керування версіями файлів «git»

Git використовується в основному розробниками для управління своїм вихідним кодом. Git записує зміни у файлах з плином часу, забезпечуючи цілісність цих змін на кожному етапі обробки. Користувачі можуть повертатися до попередніх версій і порівнювати різні версії файлів (Рисунок 3.12). Вони також можуть розгалужувати та об'єднувати файли для підтримки незалежної розробки та нелінійних робочих процесів. Git можна використовувати для скриптів бази даних, документів LaTeX, конфігураційних файлів, даних управління контентом та інших типів файлів [15].



Рисунок 3.12 – Принцип роботи системи керування версіями файлів «git»

Переваги Git над іншими системами:

безпеку даних за допомогою механізмів шифрування та автентифікації;

- інтеграція з багатьма іншими інструментами розробки, такими як CI / CD системи;

простий та логічний інтерфейс командного рядка;

- масштабованість;

- відкритий вихідний код;

 дозволяє розробникам переглядати та відновлювати будь-яку попередню версію коду;

відновлення коду з локальної копії.

3.2. Опис методології створення програмного забезпечення.

Для розробки системи дистанційного навчання було обрано спіральну модель. Спіральна модель є однією з методологій розробки програмного забезпечення. Вона була запропонована Беррі Боем та є комбінацією ітераційної та інкрементної моделей розробки.

Спіральна модель базується на ідеї поступового розроблення програмного забезпечення через повторювані ітерації. Кожна ітерація складається з чотирьох основних етапів: планування, аналіз ризиків, розробка та оцінка. Після кожної ітерації клієнт має можливість перевірити та оцінити робочий продукт, що дозволяє вносити зміни та коригувати плани на майбутнє.

Основною перевагою спіральної моделі є те, що вона дозволяє розробникам та клієнтам зосередитися на ключових аспектах розробки ПЗ та уникнути багатьох проблем, які зазвичай виникають в каскадній моделі. Також ця модель дозволяє знизити ризики, пов'язані з процесом розробки ПЗ, тому що на кожній ітерації виконується аналіз ризиків та їх зниження [16].

Основні етапи спіральної моделі наступні (Рисунок 3.13) :

 Планування. На цьому етапі визначаються основні метрики проекту, такі як вартість, терміни, обсяг робіт. – Аналіз ризиків. На цьому етапі аналізуються потенційні ризики, які можуть виникнути під час розробки ПЗ, та визначаються шляхи їх зменшення.

 Розробка. На цьому етапі проводиться розробка програмного забезпечення, включаючи створення прототипів, дизайн і реалізацію.

– Оцінка. На цьому етапі оцінюється продукт, який було розроблено на попередніх етапах. Клієнт оцінює роботу, віддає свої коментарі та рекомендації, що допомагає покращувати продукт і проводити коригування до наступної ітерації.



Рисунок 3.13 – Модель спірального процесу

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис та побудова діаграми прецедентів роботи з системою.

Для опису роботи системи дистанційного навчання було побудовано діаграму Прецедентів (Рисунок 4.1). Прецедент – це все те, що може робити система, або що можна робити з нею.



Рисунок 4.1 – Діаграма прецедентів

ПРЕЦЕДЕНТ: АВТОРИЗАЦІЯ.

Ектор: Неавторизований користувач.

Передумова: Користувач не авторизований в системі.

Післяумова: Користувач авторизований в системі. Сценарій:

- 1. Користувач вводить свій етаіl та пароль.
- 2. Користувач натискає кнопку «Увійти».
- 3. Користувач авторизований.

ПРЕЦЕДЕНТ: ПЕРЕГЛЯД ВЛАСНИХ КУРСІВ.

Ектор: Авторизований користувач.

Передумова: Користувач авторизований у системі.

Післяумова: Користувач переглянув курси.

Сценарій:

- 1. Користувач переходить до вкладки «Курси».
- 2. На екрані з'являється список курсів.
- 3. Користувач вибирає потрібний курс.
- 4. Користувач переглянув курс.
 - 4.1. Розширення «Вибір потрібного курсу»
 - 4.2. Розширення «Здати практичну роботу»

ПРЕЦЕДЕНТ: ПЕРЕГЛЯД ЖУРНАЛУ З ОЦІНКАМИ.

Ектор: Авторизований користувач.

Передумова: Користувач авторизований у системі.

Післяумова: Користувач переглянув свої оцінки з відповідного курсу. Сценарій:

- 1. Користувач переходить до вкладки «Журнал оцінок».
- 2. Вибирає курс.
- 3. Отримує оцінки за результатом вибору курс
 - 3.1. Розширення «Вибір потрібного курсу»
 - 3.2. Включення «Перегляд результату»

ПРЕЦЕДЕНТ: РОЗПОЧАТИ ЗУСТРІЧ.

Ектор: Авторизований користувач.

Передумова: Користувач авторизований у системі.

Післяумова: Користувач розпочав зустріч.

Сценарій:

- 1. Користувач переходить до вкладки «Зустрічі».
- 2. На екрані з'являється посилання на кнопка «Розпочати».
- 3. Користувач розпочав зустріч.

3.1. Розширення «Поділитися посиланням» ПРЕЦЕДЕНТ: ПЕРЕГЛЯД НАЛАШТУВАННЯ. Ектор: Користувач.

Передумова: Користувач авторизований у системі.

Післяумова: Користувач змінив фото профілю.

Сценарій:

- 1. Користувач переходить до вкладки «Налаштування».
- 2. Користувач переглянув налаштування профілю.
- 3. Користувач змінив фото профілю.
 - 3.1. Розширення «Редагувати дані»

ПРЕЦЕДЕНТ: ПЕРЕГЛЯД ВЛАСНИХ КУРСІВ.

Ектор: Викладач.

Передумова: Викладач авторизований в системі.

Післяумова: Викладач переглянув перелік курсів, які викладає. Сценарій:

- 1. Викладач переходить до вкладки «Курси».
- 2. На екрані з'являється список курсів.
- 3. Викладач переглянув курси.
 - 3.1. Розширення «Додати курс»
 - 3.2. Розширення «Вибір потрібного курсу»
 - 3.3. Розширення «Вибір потрібної практичної роботи»
- 3.4. Розширення «Виставити оцінку за практичну роботу»

ПРЕЦЕДЕНТ: ПЕРЕГЛЯД НАЛАШТУВАННЯ КУРСУ.

Ектор: Викладач.

Передумова: Викладач авторизований в системі.

Післяумова: Викладач переглянув налаштування курс.

Сценарій:

- 1. Викладач переходить на сторінку «Налаштування курсу».
- 2. На екрані з'являється інформація.
- 3. Викладач переглянув налаштування курсу.
 - 3.1. Розширення «Вибір вкладки з матеріалами»
 - 3.2. Розширення «Вибір вкладки з студентами»
 - 3.3. Розширення «Вибір вкладки з журналом оцінок»
 - 3.4. Розширення «Вибір вкладки з налаштуванням курсу»
 - 3.5. Розширення «Додати, редагувати, видалити тему або матеріал»
 - 3.6. Розширення «Додати або видалити студента»
 - 3.7. Розширення «Перегляд журналу з оцінками»
 - 3.8. Розширення «Редагувати або видалити курс»

ПРЕЦЕДЕНТ: ПЕРЕГЛЯД ЗУСТРІЧЕЙ.

Ектор: Викладач.

Передумова: Викладач авторизований в системі.

Післяумова: Викладач переглянув зустрічі.

Сценарій:

- 1. Викладач переходить на сторінку «Головна».
- 2. На екрані з'являється віджет «Зустрічі».
- 3. Викладач переглянув зустріч.
 - 3.1. Розширення «Вибір потрібного курсу»
 - 3.2. Розширення «Розпочати або закінчити зустріч»

4.2. Інструкція з використання системи

1. Після успішного входу, викладач потрапляє до головної сторінки. На головній сторінці є можливість перегляду навігаційного меню, логотипу, карток з інформацією про курс (назва курсу, група, зображення курсу), розкладу на сьогодні та проведення зустрічей. (Рисунок 4.2).



Рисунок 4.2 – Головна сторінка платформи дистанційного навчання «Learnity»

2. Перегляд власних курсів. Вибравши вкладку «Курси» користувач потрапляє на сторінку з власними курсам, де є можливість перегляду таблиці з відповідними заголовками (назва курсу, група, спеціальність, дата створення та дата оновлення) та кнопка «Додати курс» (Рисунок 4.3).

Мої курси	1				
	Назва	Група	Спеціальність	Дата створення	Дата оновлення
	Цифровий бізнес	А КН 6-41	Комп'ютерні науки	07.04.2023	17.04.2023
	Маркетинг	Я КН 6-41	Комп'ютерні науки	14.04.2023	14.04.2023
	Алгоритми	R KH 6-41	Комп'ютерні науки	14.04.2023	24.04.2023
	Інформатика	Я КН 6-41	Комп'ютерні науки	14.04.2023	17.04.2023
	Граматика	Я КН 6-41	Комп'ютерні науки	15.04.2023	18.04.2023
					< 1 >

Рисунок 4.3 - Сторінка «Мої курси»

3. Створення курсу. Для створення курсу необхідно натиснути кнопку «Створити курс». У відкритому вікні необхідно заповнити наступні поля: назва

курсу, опис та вибрати групу. Після цього потрібно натиснути на кнопку «Створити» (Рисунок 4.4).

^{вагліty} Мої куро	си			 Ілля Комар Додати кур
•	Назва	Додати курс	Дата створення	Дата оновлення
	Цифровий бізнес	Назва курсу	07.04.2023	17.04.2023
8	Маркетинг		14.04.2023	14.04.2023
	Алгоритми	Виберіть групу 🗸	14.04.2023	24.04.2023
2	Інформатика	Опис курсу	14.04.2023	17.04.2023
\$	Граматика	07260	15.04.2023	18.04.2023
		Створити		

Рисунок 4.4 – Створення курсу

4. Перегляд сторінки з курсом. Після того, як викладач створив курс він має можливість потрапити на сторінку з курсом. На сторінці є можливість перегляду шапки курсу (назва курсу, викладач, група, опис курсу, кнопка «Налаштування» та зображення курсу), навігаційного меню з темами (Рисунок 4.5).

象 KH 6-41			
Метою навчальної дисципліни є засвоєння щодо налаштування роботи користувачів в моделювання процесів управління функціс	знань щодо принципів організації локальни дротових та бездротових комп'ютерних мер нування інформаційної мережі. Гарних успіз	х та глобальних інформаційних мереж, придбання навичок иежах, діагностика роботи комп'ютерної мережі, ів у навчанні!	
🔅 Налаштування			Л
Тема 1. Натуральні числа	v		
Тема 2. Ріанлиня	v		
		Виберіть матеріал	

Рисунок 4.5 – Сторінка курсу

5. Перегляд матеріалу. Користувач вибирає тему, після цього відкривається список всіх матеріалів (лекція або практична) та натискає на неї.

На екрані з'являється вибраний матеріал: назва матеріалу, кнопка «Збільшення», кнопка «Зменшення», кнопка «Завантажити файл» та самий файл з навігацією (Рисунок 4.6).



Рисунок 4.6 – Перегляд матеріалу

6. Перегляд налаштування курсу. Для перегляду налаштування курсу необхідно натиснути кнопку «Налаштування». У відкритій сторінці викладач має можливість перегляду відповідних вкладок: матеріали, студенти, журнал оцінок, розклад, налаштування (Рисунок 4.7).

earnity	< Налаштування курсу
	🗄 Матеріали
	糸 Студенти
-	🛱 Журнал оцінок
	🛱 Розклад
	🕸 Налаштування
٢	

Рисунок 4.7 – Налаштування курсу

7. Перегляд матеріалів курсу. Для перегляду матеріалів курсу необхідно вибрати вкладку «Матеріали». У відкритій вкладці викладач має можливість

перегляду двох таблиць. Перша таблиця дозволяє додавати, редагувати, видаляти тему. Друга таблиця дозволяє додавати, редагувати, видаляти матеріали (Рисунок 4.8).

earnity	< Налаштування курсу				
_	🗄 Матеріали	Ma	теріали		
	条 Студенти	Теми +			
	 御 Журнал оцінок Розклад 	Назва	Дата створення	Дата оновлення	Дії
_		Тема 1. Натуральні числа	14.05.2023	14.05.2023	20
	🕸 Налаштування	Тема 2. Рівняння	14.05.2023	14.05.2023	∠ Ū
•					< 1 >
٢					
		Матеріали +			
		Назва	Тип Дата створення	Дата оновлення	Дії
		Лекція 1. Числа	Лекція 14.05.2023	14.05.2023	∠ ⊡
		Практична робота 1. Числа	Практична 14.05.2023	14.05.2023	20

Рисунок 4.8 – Матеріали

8. Додавання теми. Для додавання теми необхідно натиснути кнопку «Теми +». У відкритому вікні необхідно заповнити поле «назва теми». Після цього потрібно натиснути на кнопку «Додати» (Рисунок 4.9).

earnity	< Налаштування курсу				
	🗏 Матеріали	Μ	Іатеріали		
	条 Студенти 問 Жирная оцінок	Теми +			
	Розклад	Hasea	Дата створення ×	Дата оновлення	Ді
	🕲 Налаштування	Тема 2. Рівняння Назва том	35.2023	14.05.2023	20
2					
\$		Додати	_		
		Назва	Тип Да	та створення Дата оновлення	Дії
		Лекція 1. Числа	Лекція 14.	05.2023 14.05.2023	20
		Практична робота 1. Числа	Практична 14.	05.2023 14.05.2023	20



9. Додавання матеріалу. Для додавання матеріалу необхідно вибрати тему, а потім натиснути кнопку «Матеріали +». У відкритому вікні необхідно

заповнити наступні поля: тип матеріалу, назва та файл у форматі Pdf. Після цього потрібно натиснути на кнопку «Додати» (Рисунок 4.10).



Рисунок 4.10 – Додавання матеріалу

10. Перегляд студентів курсу. Для перегляду студентів курсу необхідно вибрати вкладку «Студенти». У відкритій вкладці викладач має можливість перегляду таблиці зі студентами. Є можливість додавати та видаляти студента з курсу (Рисунок 4.11).

earnity	< Налаштування курсу				
	🗄 Матеріали		Студенти		
	冬 Студенти	Додати студента на курс			
-	а Журнал оцінок	Введіть призвіще та імя студента \vee			
•	🛱 Розклад	П.І.Б		Email	Дія
	🕸 Налаштування	Крутий Іван Топовий		student3@gmail.com	0
					< 1 >
•					
-					

Рисунок 4.11 – Студенти

11. Перегляд журналу з оцінками. Для перегляду журналу з оцінками необхідно вибрати вкладку «Журнал оцінок». У відкритій вкладці викладач має можливість перегляду таблиці з оцінками вибраного студента (Рисунок 4.12).

earnity	< Налаштування курсу			
	🗄 Матеріали	Журнал оцінок		
	Я Студенти	Pulation manufacture		
-	🕮 Журнал оцінок	инория судини Метний Даня Крутой V		
•	🛱 Розклад	Назва	Дата здачі	Оцінка
	🕸 Налаштування	Practical	08.04.2023	80
		Practical 3	10.04.2023	0
0				

Рисунок 4.12 – Журнал оцінок

12. Перегляд налаштування. Для перегляду налаштування курсу необхідно вибрати вкладку «Налаштування». У відкритій вкладці викладач має можливість змінити наступні поля: назва курсу, опис курсу та оформлення курсу. Після цього потрібно натиснути на кнопку «Зберегти» (Рисунок 4.13).

earnity	< Налаштування курсу	
	🗐 Матеріали	Налаштування
	条 Студенти	Hasa sypcy
	🛱 Журнал оцінок	Цифровий бізнес
	Розклад	Метою навчальної дисципліни є засвоєння знань цодо принципів організації покальних та побальних інформаціїних нерен.
	🕲 Налаштування	придбання навичок щодо налаштування роботи користувачив в
		Зберегти
•		

Рисунок 4.13 – Налаштування

13. Перегляд зданих робіт. Для перегляду зданих робіт, викладачу необхідно вибрати практичну роботу. У відкритій вкладці викладач має можливість переглянути наступні поля: П.І.Б студента, дата здачі, оцінка та статус роботи (Рисунок 4.14).

earnity	Tema 1	^	^ Практична 1		
	Лекця І. Числа Практична 1		🗐 Кількість зданих робіт: 1	🗐 Кількість пер	евірених робіт: О
	🛱 Практична 2				
	🕼 Практична 3		П.І.Б	Дата здачі Оцінка	Статус
	Тема4	Ŷ	Д Демченко Даня Ігорович	10.04.2023 0 / 100	Здано
					< 1 >
_					
2					
\$					
-					

Рисунок 4.14 – Здані роботи

14. Перевірка надісланої роботи. Для перевірки зданої роботи, викладачу необхідно вибрати здану роботу. У відкритому вікні викладачу необхідно заповнити поле з оцінкою. Після цього потрібно натиснути на кнопку «Опублікувати» (Рисунок 4.15).



Рисунок 4.15 – Перевірка роботи

15. Розпочати зустріч. Щоб розпочати зустріч, викладачу необхідно перейти до «Головної сторінки». На головній сторінці у картці «Зустрічі» викладачу необхідно вибрати курс та натиснути на кнопку «Розпочати зустріч». Після цього викладач може натиснути на кнопку «Приєднатися» або «Завершити» (Рисунок 4.16).



Рисунок 4.16 – Зустрічі

16. Здати роботу. Щоб здати роботу, студенту необхідно перейти до курсу та вибрати практичну роботу. У відкритій вкладці студент повинен натиснути кнопку «Здати роботу», прикріпити файл та натиснути «Відправити» (Рисунок 4.17).



Рисунок 4.17- Здати роботу

17. Налаштування профілю. Для перегляду налаштування профілю необхідно вибрати вкладку «Налаштування». У відкритій вкладці користувач може змінити фото профілю (Рисунок 4.18).



Рисунок 4.18- Налаштування профілю

ВИСНОВКИ

Дистанційне навчання з використанням всіх новітніх технічних досягнень в сфері телекомунікаційних технологій є невід'ємною складовою сучасного освітнього процесу. В ході виконання завдання мною були вдосконалені навички для розробки системи дистанційного навчання. В рамках кваліфікаційного проекту ставилося завдання по розробці програмного забезпечення системи дистанційного навчання, а саме веб-інтерфейсу. Робота пройшла апробацію, по ній було написано тези та опубліковано статтю.

Основні результати роботи:

 сформульовані основні вимоги до програмного забезпечення системи дистанційного навчання, що розробляється;

проведено огляд переваг та недоліків існуючих систем дистанційного навчання;

виконано опис проектних рішень, інструментів та підходів до розробки системи дистанційного навчання;

- розроблено веб-інтерфейс системи дистанційного навчання;

відпрацьовано інструкцію з використання системи;

- проведено опис та побудовано діаграму прецедентів роботи з системою.

СПИСОК ЛІТЕРАТУРИ

- Етапи створення веб сайтів [Електронний ресурс]. Режим доступу до ресурсу: URL: https://webtune.com.ua/statti/web-rozrobka/etapy-stvorennyaveb-sajtiv/ – Назва з екрану.
- Що таке Moodle [Електронний ресурс]. Режим доступу до ресурсу: URL: https://moodle.org/mod/page/view.php?id=8174 – Назва з екрану.
- What is Blackboard Learn? [Електронний ресурс]. Режим доступу до pecypcy:URL:https://utlv.screenstepslive.com/s/faculty/m/BlackboardLearn/l/187 320-what-is-blackboard-learn Назва з екрану.
- What is Edmodo? [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.whatiselt.com/single-post/2018/01/28/what-is-edmodo – Назва з екрану.
- What is Canvas? [Електронний ресурс]. Режим доступу до ресурсу: URL: https://community.canvaslms.com/t5/Canvas-Basics-Guide/What-is-Canvas/tap/45 – Назва з екрану.
- 6. What Is Figma and What Is It Used For? [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.makeuseof.com/what-is-figma-used-for/ Назва з екрану.
- The Top 10 Benefits of TypeScript: Why You Should Switch from JavaScript Today [Електронний ресурс]. – Режим доступу до ресурсу: URL: https://www.linkedin.com/pulse/top-10-benefits-typescript-why-you-shouldswitch-from-nesat-nayem – Назва з екрану.
- 8. React [Електронний ресурс]. Режим доступу до ресурсу: URL: https://react.dev/ – Назва з екрану.
- Understanding Redux: A tutorial with examples [Електронний ресурс]. Режим доступу до ресурсу: URL: https://blog.logrocket.com/understandingredux-tutorial-examples/ – Назва з екрану.

- 10. What is Redux-Saga? [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.javatpoint.com/what-is-redux-saga Назва з екрану.
- Zod For Data Validation In TypeScript [Електронний ресурс]. Режим доступу до ресурсу: URL: https://codedamn.com/news/javascript/zod-gettingstarted – Назва з екрану.
- 12. What is Ant Design and How to Use It? [Електронний ресурс]. Режим доступу до ресурсу: URL: https://mockitt.wondershare.com/ant-design/ant-design.html Назва з екрану.
- 13. What is Sass CSS? Advantages & disadvantages of using Sass CSS. [Електронний ресурс]. – Режим доступу до ресурсу: URL: https://mobiosolutions.medium.com/what-is-sass-css-advantages-disadvantagesof-using-sass-css-61a753d63a01 – Назва з екрану.
- 14. What is Visual Studio Code? [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.educative.io/answers/what-is-visual-studio-code Назва з екрану.
- 15. Git [Електронний ресурс]. Режим доступу до ресурсу: URL: https://www.techtarget.com/searchitoperations/definition/Git – Назва з екрану.
- 16. What is Spiral Model and How is it Used? ТесhTarget [Електронний ресурс].
 Режим доступу до ресурсу: URL: https://www.techtarget.com/searchsoftwarequality/definition/spiral-model – Назва з екрану.

додаток а. вихідні коди

КОД ГОЛОВНОЇ СТОРІНКИ

import { Empty, Layout } from 'antd';

import { useEffect } from 'react';

import { useNavigate } from 'react-router';

import './Home.scss';

import { useAppSelector, useAppDispatch } from '../../hooks/reduxhooks';

import { SagaAction, UserRoleEnum } from '../../common/types';

import HeaderPage from '../../components/header/HeaderPage';

import Schedule from '../../components/schedule/Schedule';

import Course from '../../components/course/Course';

import CurrentActivity from '../../components/currentActivity/CurrentActivity';

import Meetings from '../../components/meetings/Meetings';

```
const Home = () => {
```

```
const { courses } = useAppSelector((state) => state.coursesReducer);
```

```
const { user } = useAppSelector((state) => state.authReducer);
```

```
const dispatch = useAppDispatch();
```

```
const navigate = useNavigate();
```

```
const onCourseClick = (id: string) => navigate(/main/courses/${id});
```

```
useEffect(() => \{
```

dispatch({ type: SagaAction.COURSES_GET });

}, [dispatch]);

```
const renderedCourses = courses?.map((course) => (
```

<Course onClick={() => onCourseClick(course.id)} key={course.id} data={course} />

));

return (

<Layout>

<HeaderPage />

<div className='home-page'>

<div className='home-page_name-container'>

Головна сторінка

</div>

<div className='home-page__content-container'>

<div className='home-page__course'>

{courses?.length?(

<div className='course-card'>{renderedCourses}</div>

```
) ( <Empty description={<span className='empty-title'>Курси відсутні</span>} />
```

)}

</div>

```
<div className='home-page__activity'>
```

```
{user?.role === UserRoleEnum.TEACHER ? <Meetings /> : <CurrentActivity />}
```

<Schedule />

</div>

</div>

```
</div>
```

```
</Layout>
```

);

};

export default Home;

СТОРІНКА МОЇ КУРСИ

import { useEffect, useState } from 'react';

import { Button, Layout, Table } from 'antd';

import { PlusCircleOutlined } from '@ant-design/icons';

import './Courses.scss';

import { useAppDispatch, useAppSelector } from '../../hooks/reduxhooks';

import { courseStudentColumns, courseTeacherColumns } from './constants';

import { SagaAction, UserRoleEnum } from '../../common/types';

import HeaderPage from '../../components/header/HeaderPage';

import CourseModal from '../../components/modals/course/Course';

import AppLoader from '../../components/AppLoader';

const Courses = () => {

```
const { take } = useAppSelector((state) => state.paginatedDataReducer);
```

const { courses } = useAppSelector((state) => state.coursesReducer);

const { user } = useAppSelector((state) => state.authReducer);

const [show, setShow] = useState(false);

const handleClose = () => setShow(false);

const handleShow = () => setShow(true);

```
const dispatch = useAppDispatch();
```

useEffect(() => {

dispatch({ type: SagaAction.COURSES_GET });

}, [dispatch]);

if (!courses) return <AppLoader />;

return (

<Layout>

<HeaderPage />

```
<div className='course-page'>
```

<div className='course-page__name-container'>

Moï курси

</div>

```
{user?.role === UserRoleEnum.TEACHER && (
<div className='course-page_button-container'>
<Button
type='primary'
shape='round'
icon={<PlusCircleOutlined className='icon' />}
className='course-page__button-connect'
onClick={handleShow}
>
Додати курс
</Button>
<CourseModal onStart={show} handleClose={handleClose} />
</div>
)}
</div>
<div className='course-page_table-container'>
<div className='course-page__table'>
{user?.role === UserRoleEnum.TEACHER ? (
<Table
pagination={{ defaultPageSize: take }}
columns={courseTeacherColumns}
dataSource={courses}
\geq
):(
<Table
pagination={{ defaultPageSize: take }}
columns={courseStudentColumns}
```

```
dataSource={courses}
\geq
)}
</div>
</div>
</Layout>
};
export default Courses;
СТОРІНКА ДЕТАЛІ КУРСУ
import { Layout, Empty, Menu } from 'antd';
import { useEffect, useMemo, useState } from 'react';
import { useParams } from 'react-router';
import { BookOutlined } from '@ant-design/icons';
import type { MenuProps } from 'antd';
import { SagaAction } from '../../common/types';
import { useAppDispatch, useAppSelector } from '../../hooks/reduxhooks';
import AppLoader from '../../components/AppLoader';
import CourseHeader from '../../../components/courseHeader/CourseHeader';
import CardMaterial from '../../components/cardMaterial/CardMaterial';
```

```
import { getIconActivity } from '../../common/helpers';
```

```
import './CourseDetails.scss';
```

```
const Course = () => {
```

```
const { course, courseTopics, courseTopic, courseActivities } = useAppSelector(
(state) => state.coursesReducer,
```

```
);
```

);

const [openKeys, setOpenKeys] = useState(["]);

```
const { id } = useParams();
const dispatch = useAppDispatch();
const onTopicOpen = (id: string | undefined) => {
dispatch({ type: SagaAction.COURSES_TOPIC_GET, payload: id });
dispatch({ type: SagaAction.COURSES_TOPICS_ACTIVITIES_GET, payload: id
});
};
const onActivitySelect = (id: string | undefined) => {
dispatch({ type: SagaAction.COURSES_TOPICS_ACTIVITY_GET, payload: id });
};
const onOpenChange: MenuProps['onOpenChange'] = (keys) => {
const latestOpenKey = keys.find((key) => openKeys.indexOf(key) === -1);
if (latestOpenKey) {
onTopicOpen(latestOpenKey);
}
setOpenKeys(latestOpenKey ? [latestOpenKey] : []);
};
useEffect(() => \{
dispatch({ type: SagaAction.COURSE_GET, payload: id });
dispatch({ type: SagaAction.COURSES_TOPICS_GET, payload: id });
}, [dispatch, id]);
useEffect(() => \{
return () => {
dispatch({ type: SagaAction.COURSES_RESET });
};
}, []);
const topicsChilder = (topicId: string) => {
```

55

```
const activitiesData = courseActivities?.map(({ id, title, type }) => ({
key: id,
label: title,
icon: getIconActivity(type),
}));
return activitiesData ? activitiesData.filter(() => courseTopic?.id === topicId) : [];
};
const topicsData: MenuProps['items'] = useMemo(
() =>
courseTopics?.map(({ id, title }) => ({
key: id,
label: title,
icon: <BookOutlined />,
children: topicsChilder(id),
})),
[courseTopics, courseActivities],
);
if (!course) return <AppLoader />;
return (
<Layout>
<CourseHeader />
{courseTopics?.length === 0 ? (
<Empty description={<span className='empty-title'>Теми відсутні</span>} />
):(
<div className='course-page__container'>
<div className='course-page__sidebar'>
```

<div className='smartphone-menu-trigger' />

<Menu

openKeys={openKeys}

onOpenChange={onOpenChange}

mode='inline'

items={topicsData}

inlineIndent={40}

onSelect={({ key }) => onActivitySelect(key)}

 $\!\!\!/\!\!\!>$

</div>

```
<div className='course-page_task'>
```

<CardMaterial />

</div>

</div>

```
)}
```

</Layout>

);

};

export default Course;