

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

_____ Оксана ЧЕРНЕНКО

(підпис)

«___» _____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

«ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСОБІВ ЕЛЕКТРОННОГО НАВЧАННЯ»

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

Виконавець роботи Долін Даніїл Володимирович

_____ «___» _____ 202_ р.
(підпис)

Науковий керівник к. ф.-м. н., доц., Черненко Оксана Олексіївна

_____ «___» _____ 202_ р.
(підпис)

Рецензент _____

ПОЛТАВА 2024

ЗАТВЕРДЖУЮ

Завідувач кафедри ____ Олена ОЛЬХОВСЬКА

« ____ » вересня

2024р.

ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему Програмна реалізація засобів електронного навчання

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки» ступеня

бакалавр

Прізвище, ім'я, по батькові Долін Данііл Володимирович

Затверджена наказом ректора № 144-Н від «_» вересня 202_р.

Термін подання студентом роботи «__» _____ 202_ р.

Вихідні дані до кваліфікаційно роботи: публікації з теми, розробка рейтингування.

Зміст пояснювальної записки

ВСТУП

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Аналіз систем електронного навчання

1.2 Система Moodle

1.3 Гугл клас

1.4 Постановка задачі

2. ТЕОРИТИЧНА ЧАСТИНА

2.1 Вибір інструментальних засобів

2.2 Структура програми

2.3 Алгоритм роботи програми

2.4 Вибір та обґрунтування бази даних

3. ПРАКТИЧНА ЧАСТИНА

3.1 Розробка бази даних

3.2 Програмний інтерфейс користувача

3.3 Тестування функціоналу

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

Перелік графічного матеріалу: 7 ілюстрацій.

Консультанти розділів кваліфікаційної роботи

Розділ	ПІБ, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постанова задачі	Черненко О. О.		
Інформаційний огляд	Черненко О. О.		
Теоретична частина	Черненко О. О.		
Практична реалізація	Черненко О. О.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій і стандартів та звіт керівнику		
3. Постановка завдання		
4. Інформаційний огляд джерел бібліотек та Інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доопрацювання (за необхідності), рецензування		

Дата видачі завдання «__» __202_ р.

Здобувач вищої освіти Долін Данііл Володимирович

Науковий керівник к. ф.-м. н., Черненко О. О.

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ЄКТС)

Протокол засідання ЕК № _____ від «_____» _____202_ р.

Секретар ЕК _____
(підпис) (ініціал та прізвище)

Затверджую

Зав. кафедрою _____

к. ф.-м. н. Олена ОЛЬХОВСЬКА

«__» _____202_ р.

Погоджено

Науковий керівник _____

к. ф.-м. н. Оксана ЧЕРНЕНКО

«__» _____202_ р.

План

кваліфікаційної роботи на тему

Програмна реалізація засобів електронного навчання

зі спеціальності 122 Комп'ютерні науки

освітня програма 122 «Комп'ютерні науки» ступеня бакалавр

Прізвище, ім'я, по батькові Долін Данііл Володимирович**ВСТУП****1. ІНФОРМАЦІЙНИЙ ОГЛЯД***1.1 Аналіз систем електронного навчання**1.2 Система Moodle**1.3 Гугл клас**1.4 Постановка задачі***2. ТЕОРИТИЧНА ЧАСТИНА***2.1 Вибір інструментальних засобів**2.2 Структура програми**2.3 Алгоритм роботи програми**2.4 Вибір та обґрунтування бази даних***3. ПРАКТИЧНА ЧАСТИНА***3.1 Розробка бази даних**3.2 Програмний інтерфейс користувача**3.3 Тестування функціоналу***ВИСНОВКИ****СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ****ДОДАТКИ**

Здобувач вищої освіти _____Данііл ДОЛІН

«__» _____202_ р.

РЕФЕРАТ

Записка: 52 с., 7 рис., 5 таблиць, 1 додаток, 20 джерел.

ЕЛЕКТРОННЕ НАВЧАННЯ, АДАПТИВНЕ НАВЧАННЯ, ЯКІСТЬ ОСВІТИ, АЛГОРИТМІЗАЦІЯ, ПРОЄКТУВАННЯ НАВЧАЛЬНИХ КУРСІВ, ІНДИВІДУАЛЬНИЙ ПІДХІД

Об'єкт розробки – Процеси проєктування електронного навчання та їх вплив на якість освіти

Мета роботи – проєктування і програмна реалізація системи електронного навчання

Методи дослідження – методи теорії систем числення, середовище візуальної розробки програм PyCharm та об'єктно-орієнтована мова програмування Python.

Зроблено огляд відомого програмного забезпечення для електронного навчання студентів, виявлено позитивні та негативні сторони. Розроблено алгоритм системи електронного навчання, побудовано його блок-схему. Програмно реалізовано систему електронного навчання.

Здійснено програмну реалізацію для електронного навчання.

У результаті тестування виявлено, що система працює коректно, та допоможе студентам адаптуватись до дистанційного навчання.

ЗМІСТ

ВСТУП	7
1. ІНФОРМАЦІЙНИЙ ОГЛЯД	9
1.1 Аналіз систем електронного навчання.....	9
1.2 Система Moodle.....	10
1.3 Гугл клас	13
1.4 Постановка задачі.....	16
2. ТЕОРИТИЧНА ЧАСТИНА	18
2.1 Вибір інструментальних засобів	18
2.2 Структура програми.....	20
2.3 Типи UML діаграм та їх застосування у розробці програмного забезпечення.....	25
2.4 Вибір та обґрунтування бази даних	32
3. ПРАКТИЧНА ЧАСТИНА	34
3.1 Розробка бази даних	34
3.2 Програмний інтерфейс користувача	35
3.3 Тестування функціоналу.....	44
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТКИ	49

ВСТУП

У сучасному цифровому світі швидкі та інноваційні технології знаходять широке застосування у сферах освіти та навчання. Проектування та алгоритмізація електронного навчання є ключовими аспектами у вдосконаленні та оптимізації процесу освіти. Ця тема вивчається з точки зору розробки та застосування інноваційних педагогічних технологій, які базуються на використанні електронних засобів та програмних рішень з метою підвищення ефективності та доступності навчального процесу.

Основна мета проектування електронного навчання полягає у створенні інтерактивних та інноваційних навчальних матеріалів, які сприяють залученню студентів та стимулюють їх активну участь у навчальному процесі. Це включає в себе розробку електронних курсів, веб-платформ для навчання, мультимедійних навчальних ресурсів та інших цифрових засобів, спрямованих на підвищення якості освіти.

У процесі алгоритмізації електронного навчання велика увага приділяється розробці та впровадженню адаптивних алгоритмів, які забезпечують індивідуалізований підхід до кожного студента з урахуванням його потреб, навичок та інтересів. Це дозволяє оптимізувати навчальний процес, забезпечуючи краще засвоєння матеріалу та підвищення мотивації до навчання.

Дослідження у галузі проектування та алгоритмізації електронного навчання спрямовані на вдосконалення методів та підходів до створення цифрових навчальних середовищ, що відповідають сучасним вимогам до якості освіти. Розробка новаторських технологій та методик в цій галузі є важливим напрямом розвитку освіти, спрямованим на підвищення її доступності, ефективності та рівня якості.

Мета дослідження полягає у вивченні та аналізі сучасних підходів до проектування та алгоритмізації електронного навчання з метою розробки ефективних методик та інструментів, спрямованих на підвищення якості освіти та забезпечення індивідуалізованого підходу до кожного студента.

Предметом дослідження є процеси проектування та алгоритмізації електронного навчання, включаючи розробку навчальних матеріалів, створення веб-платформ для навчання, реалізацію адаптивних алгоритмів та інші аспекти цифрового навчання.

Об'єктом дослідження є сучасні технології та методики, що застосовуються у сфері електронного навчання, а також процеси навчання та взаємодії учасників навчального процесу.

Актуальність дослідження обумовлена стрімким розвитком інформаційних технологій та постійними змінами у суспільстві, що вимагають постійного вдосконалення системи освіти. Інноваційні підходи до електронного навчання стають все більш важливими для забезпечення доступності, ефективності та якості навчального процесу.

Новизна дослідження полягає у впровадженні сучасних технологій та методик в сферу освіти, а також у розробці інноваційних підходів до проектування та реалізації електронного навчання з урахуванням індивідуальних потреб та особливостей кожного учня.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Аналіз систем електронного навчання

Аналіз систем електронного навчання є важливим етапом у дослідженні сучасних тенденцій у сфері освіти та визначенні їх ефективності та придатності для впровадження в практику. Системи електронного навчання представляють собою комплекси програмних та апаратних засобів, спрямованих на організацію та здійснення навчального процесу з використанням інформаційних технологій.

Першим аспектом аналізу є оцінка функціональних можливостей системи. Це включає аналіз доступних інструментів для створення, редагування та розповсюдження навчального контенту, можливостей для спілкування та взаємодії між учасниками навчального процесу, а також системи оцінювання та контролю успішності студентів[1].

Другим аспектом є аналіз інтерфейсу користувача та зручності використання системи. Оцінка здійснюється з урахуванням простоти та інтуїтивності інтерфейсу, зручності навігації, доступності функцій та можливостей персоналізації.

Третім аспектом є аналіз педагогічних принципів та методик, що лягли в основу системи. Це включає перевірку відповідності системи сучасним педагогічним тенденціям, наявність інтерактивних та адаптивних методик навчання, а також можливості індивідуалізації навчального процесу.

Четвертим аспектом є аналіз технічної підтримки та безпеки системи. Важливими є стабільність та швидкодія системи, її сумісність з різноманітними пристроями та операційними системами, а також заходи забезпечення конфіденційності та захисту персональних даних користувачів[2].

У підсумку, аналіз систем електронного навчання дозволяє здійснити об'єктивну оцінку їхньої ефективності та придатності для використання у навчальних закладах. Врахування усіх зазначених аспектів дозволяє здійснити інформований вибір та впровадження найбільш відповідних та ефективних систем електронного навчання.

1.2 Система Moodle

Інтерфейс головної сторінки системи представлено на рисунку 1.1.

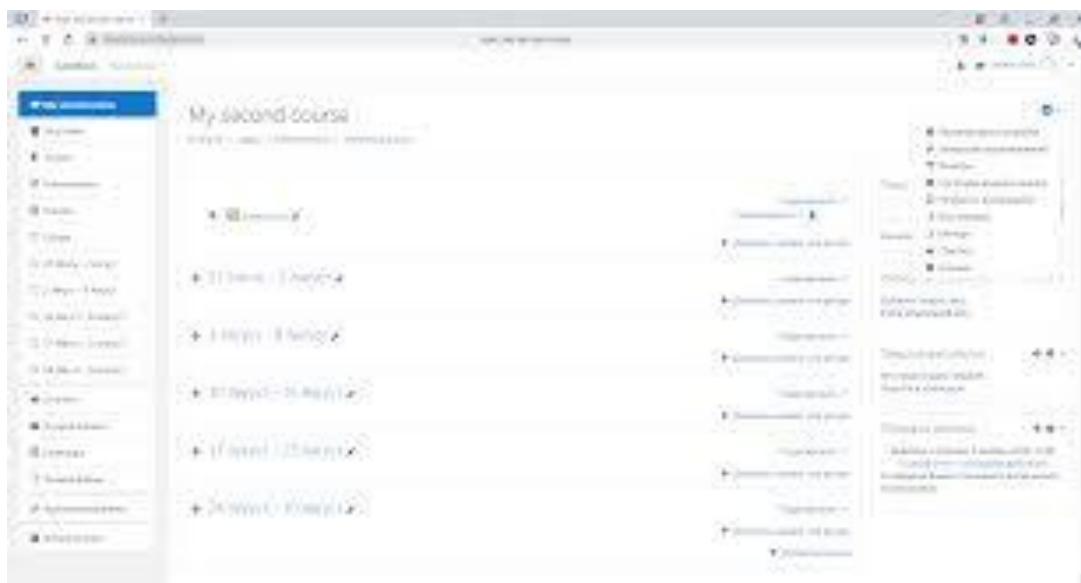


Рисунок 1.1 – Інтерфейс системи Moodle

Цільове призначення системи Moodle полягає у забезпеченні ефективного управління навчанням та сприянні активному навчальному процесу в освітніх та корпоративних середовищах. Moodle є відкритою та гнучкою платформою, яка дозволяє створювати онлайн-курси, управляти навчальним матеріалом, спілкуватися та співпрацювати між учасниками навчального процесу, а також оцінювати навчальні досягнення студентів.

Система розроблена з урахуванням сучасних педагогічних концепцій та методик, спрямованих на створення стимулюючого та інтерактивного навчального середовища. Вона дозволяє вчителям та тренерам ефективно керувати процесом навчання, а студентам та учням - активно взаємодіяти з навчальним матеріалом та отримувати якісний освітній досвід[3].

Завдяки своїй гнучкості та модульності, Moodle може бути успішно використаний у найрізноманітніших навчальних та професійних контекстах, від шкіл і університетів до корпоративних навчальних центрів. Він сприяє

індивідуалізації навчання та врахуванню потреб кожного учня чи співробітника, що робить його важливим інструментом для розвитку сучасної освіти та навчання.

Функціональні можливості Moodle, системи управління навчанням, орієнтовані на забезпечення широкого спектру інструментів для організації та ведення навчальних курсів. Серед основних можливостей можна відзначити можливість створення та керування різними типами курсів, включаючи традиційні, онлайн, гібридні та інші форми навчання. Moodle надає можливість завантажувати різноманітні навчальні матеріали, такі як тексти, відео, аудіо, презентації, тестові завдання тощо, що дозволяє розширити інструментарій навчання та підвищити його ефективність.

Крім того, Moodle надає можливості для організації співпраці та взаємодії між учасниками навчального процесу, зокрема за допомогою форумів для обговорення, чатів для спілкування в реальному часі, а також спільного доступу до навчальних матеріалів та завдань. Такий підхід сприяє активній участі студентів у навчанні, обміну знаннями та досвідом, а також сприяє розвитку комунікативних навичок.

Крім того, Moodle забезпечує можливість проведення та оцінювання різних видів контрольних заходів, таких як тести, онлайн-завдання, заліки тощо. Система має інструменти для створення та налаштування цих контрольних заходів, а також можливості для автоматичної або ручної оцінки результатів студентів. Це дозволяє вчителям ефективно вести моніторинг та оцінку успішності студентів та забезпечує об'єктивність та надійність процесу оцінювання.

Інтерфейс користувача системи електронного навчання відіграє ключову роль у взаємодії між користувачами та навчальними ресурсами. Він є основним засобом сприйняття та взаємодії користувача з навчальною платформою[4]. Зручність та ефективність інтерфейсу безпосередньо впливають на якість навчального процесу та задоволення користувачів від використання системи.

Інтерфейс користувача системи електронного навчання повинен бути спроектований з урахуванням потреб та характеристик цільових аудиторій. Це вимагає детального аналізу потреб користувачів, їхніх навичок та вподобань щодо способів сприйняття та взаємодії з інформацією.

Основними вимогами до інтерфейсу користувача є зручність навігації, легкість використання, інтуїтивність та естетичний вигляд. Користувач повинен легко знаходити необхідні ресурси та функції, швидко орієнтуватися в системі та виконувати потрібні дії без зайвих зусиль.

Ефективність інтерфейсу також залежить від його адаптивності до різних типів пристроїв та розмірів екранів. Інтерфейс повинен оптимально працювати як на настільних комп'ютерах, так і на мобільних пристроях, забезпечуючи однакову зручність використання для всіх користувачів.

Педагогічні принципи та методики, що лягли в основу систем електронного навчання, відображають сучасні підходи до організації навчального процесу з використанням інформаційно-комунікаційних технологій. Одним з основних принципів є активне навчання, яке передбачає залучення студентів до активної участі у процесі засвоєння знань. Це може бути досягнуто за допомогою інтерактивних завдань, дискусій, групової роботи та інших форм активної взаємодії.

Інший важливий принцип – індивідуалізація навчання. Системи електронного навчання надають можливість створювати персоналізовані навчальні шляхи для кожного студента з урахуванням його потреб, навичок та інтересів. Це дозволяє кожному студентові працювати у власному темпі та зосереджуватися на темах, які для нього найбільш актуальні та цікаві[4].

Одним із методів, що використовуються у системах електронного навчання, є проблемне навчання. Цей підхід передбачає вирішення реальних або модельних проблем, що ставляться перед студентами, що сприяє розвитку їхніх аналітичних та критичного мислення.

До інших педагогічних методик, що застосовуються у системах електронного навчання, відносяться проектне навчання, колективне навчання, диференційоване навчання та інші. Використання цих методик дозволяє створити стимулююче та продуктивне навчальне середовище, сприяючи ефективному засвоєнню знань та розвитку навичок учнів.

Узагальнюючи, педагогічні принципи та методики у системах електронного навчання спрямовані на створення інтерактивного, індивідуалізованого та

ефективного середовища для засвоєння знань, що відповідає сучасним вимогам до якості освіти.

1.3 Гугл клас

Інтерфейс системи гугл клас представлено на рисунку 1.2.

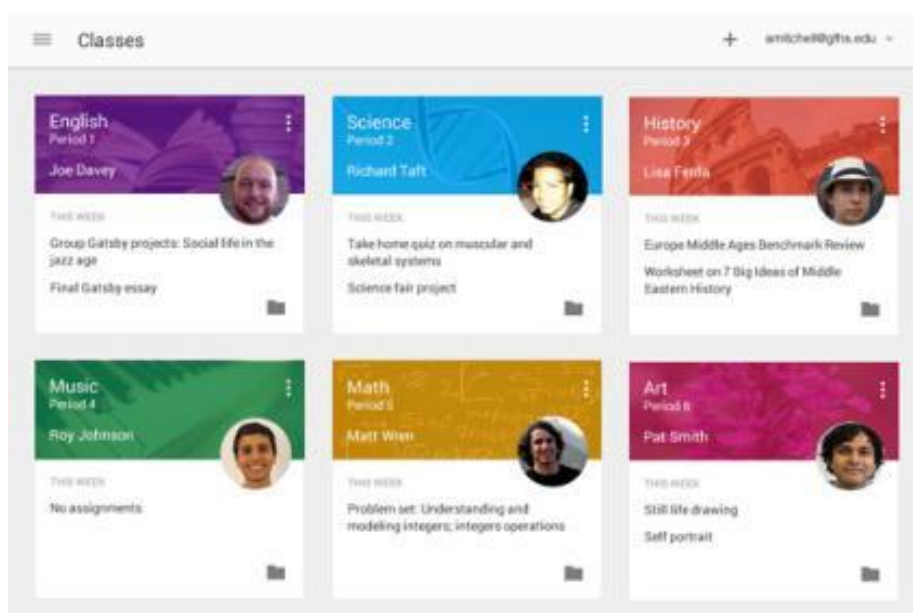


Рисунок 1.2 – Інтерфейс системи «гугл клас»

Функціональні можливості системи Google Classroom представляють собою комплекс програмних інструментів, спрямованих на організацію та здійснення навчального процесу у віртуальному середовищі. Ці можливості включають у себе широкий спектр функцій, спрямованих на забезпечення ефективної комунікації між вчителем та учнями, організацію навчального матеріалу, виконання та оцінювання завдань.

Система надає засоби для створення та редагування різноманітних навчальних матеріалів, таких як завдання, лекції, тести тощо, що дозволяє вчителям ефективно структурувати навчальний процес та передавати інформацію студентам у доступній формі. Крім того, в системі реалізовані засоби для створення обговорень та форумів,

що сприяє активній взаємодії між учасниками навчального процесу та обміну думками та ідеями[7].

Однією з ключових функцій є можливість виставлення завдань та контролю їх виконання. Вчителі мають можливість створювати завдання різної складності та обсягу, призначати їх учням та відстежувати прогрес у їх виконанні. Такий підхід дозволяє забезпечити індивідуальний підхід до кожного учня та відповідати його потребам та можливостям.

Також в системі реалізовані засоби для оцінювання та звітності. Вчителі можуть створювати тести та оцінювати знання учнів, а також генерувати звіти про успішність учнів та їх прогрес у навчанні. Це дозволяє вчителям ефективно відстежувати успішність кожного учня та вчасно реагувати на можливі проблеми.

Загалом, функціональні можливості системи Google Classroom спрямовані на створення ефективного та зручного навчального середовища, що сприяє активному взаємодії між вчителем та учнями, організації навчального процесу та ефективному контролю над ним.

Інтерактивність у контексті систем електронного навчання визначається як здатність забезпечити активну взаємодію між учасниками навчального процесу, що сприяє залученню учнів до навчання та стимулює їх активність та зацікавленість у процесі освоєння матеріалу. Інтерактивність включає в себе не лише можливість обміну інформацією між учасниками, але й можливість активної участі учнів у вирішенні навчальних завдань, обговоренні матеріалу та спільному конструюванні знань.

В контексті систем електронного навчання інтерактивність забезпечується за допомогою різноманітних інструментів та можливостей, таких як форуми для обговорення, веб-конференції для віртуального спілкування, відео- та аудіоматеріали для візуалізації матеріалу, інтерактивні завдання та тести для перевірки знань. Ці інструменти дозволяють створювати атмосферу взаємодії та співпраці між учасниками навчального процесу навіть у віддаленому режимі.

Здатність до інтерактивності у системах електронного навчання сприяє не лише покращенню засвоєння матеріалу, але й розвитку критичного мислення,

творчих навичок та здатності до колективної роботи. Інтерактивність є важливим елементом в удосконаленні навчального процесу, оскільки вона стимулює активність та самостійність учнів, сприяє поглибленню розуміння навчального матеріалу та розвитку ключових компетентностей.

Адаптивність системи Google Classroom виявляється у здатності забезпечити індивідуалізований підхід до навчання кожного учня з урахуванням його особистих потреб, здібностей та темпу освоєння матеріалу. Ця адаптивність реалізується через різноманітні функціональні можливості та інструменти платформи, які дозволяють вчителям індивідуалізувати навчальний процес для кожного учня.

Платформа надає можливість створювати та призначати індивідуальні завдання для учнів, враховуючи їхні потреби та рівень знань. Крім того, вчителі можуть надавати додаткові матеріали або додаткові пояснення для учнів, які потребують додаткової підтримки.

Також система Google Classroom дозволяє вчителям моніторити прогрес кожного учня окремо, аналізуючи їхні досягнення та слабкі сторони. На основі цієї інформації вчителі можуть вносити корективи в навчальний процес, пропонуючи додаткові завдання для поглибленого вивчення або індивідуальну підтримку[8].

Крім того, Google Classroom надає можливість для інтерактивної комунікації між вчителями та учнями, що стимулює активну участь та співпрацю кожного учня у навчальному процесі. Це сприяє більш ефективному засвоєнню матеріалу та збільшенню мотивації до навчання.

У підсумку, адаптивність системи Google Classroom полягає у здатності забезпечити індивідуальний підхід до навчання кожного учня, що сприяє покращенню якості освіти та залученню учнів до активної навчальної діяльності.

Google Classroom дозволяє вчителям створювати та розміщувати різноманітні завдання для учнів, такі як тести, домашні завдання, проекти тощо, і встановлювати критерії оцінювання для кожного завдання. Після виконання завдання учні можуть здати його через платформу, де вчителі мають можливість оцінювати роботу студентів, встановлювати оцінки та залишати коментарі.

Крім того, Google Classroom забезпечує зручні засоби для відстеження активності учнів у межах платформи, такі як перегляд статистики виконання завдань, часу, витраченого на розв'язання завдань, а також відвідування курсу та участь у дискусіях. Це дозволяє вчителям аналізувати заангажованість учнів у навчальний процес та вчасно реагувати на можливі труднощі або проблеми, що виникають під час навчання.

Загалом, відстеження прогресу у Google Classroom забезпечує ефективний механізм контролю та оцінювання успішності учнів у навчанні, сприяючи підвищенню ефективності навчального процесу та забезпеченню індивідуального підходу до кожного учня.

Інтеграція з іншими сервісами Google є важливою характеристикою системи Google Classroom, що сприяє зручності та ефективності навчального процесу. Зокрема, ця інтеграція дозволяє вчителям та учням здійснювати безперервну роботу з різними сервісами Google без необхідності переключення між різними платформами[10].

1.4 Постановка задачі

Постановка задачі полягає у визначенні конкретних цілей та завдань, спрямованих на досягнення мети дослідження у галузі проектування та алгоритмізації електронного навчання.

Однією з ключових задач є аналіз сучасних тенденцій у сфері електронного навчання та виявлення переваг та недоліків існуючих підходів. Це дозволить з'ясувати напрямки подальших досліджень і вдосконалення методик.

Другою важливою задачею є вивчення та узагальнення наукової літератури з питань проектування та алгоритмізації електронного навчання, що сприятиме усвідомленню актуальних концепцій та технологій у цій області.

Третьою задачею є розробка інноваційних підходів та методик до створення електронних навчальних матеріалів, що враховують специфіку предметної області та індивідуальні особливості студентів[11].

Крім того, необхідно провести експериментальне тестування розроблених методик з метою перевірки їх ефективності та визначення можливостей їх подальшого вдосконалення.

Завершальною задачею є аналіз отриманих результатів дослідження та формулювання висновків і рекомендацій щодо подальших напрямків розвитку в галузі проектування та алгоритмізації електронного навчання.

2. ТЕОРИТИЧНА ЧАСТИНА

2.1 Вибір інструментальних засобів

Мова програмування Python є однією з найпопулярніших та широко використовуваних мов у сфері програмування. Вона відома своєю простотою синтаксису, високою читабельністю коду та широким спектром функціональних можливостей. Python використовується у багатьох областях, включаючи веб-розробку, аналіз даних, штучний інтелект, наукові дослідження та інші[12].

Ця мова програмування надає розробникам зручний та ефективний інструментарій для вирішення різноманітних завдань. Завдяки своїй простоті та лаконічності, Python дозволяє швидко створювати програми та розвивати їх функціонал без великих зусиль.

У світлі стрімкого розвитку сфери штучного інтелекту та машинного навчання, Python займає особливе місце, оскільки він надає потужні бібліотеки та фреймворки для розробки та реалізації алгоритмів машинного навчання. Його екосистема, що включає бібліотеки, такі як TensorFlow, PyTorch та Scikit-learn, дозволяє розробникам ефективно створювати та впроваджувати складні моделі штучного інтелекту.

Python також відомий своєю активною та дружньою спільнотою розробників, яка постійно розвивається та поповнюється новими інструментами та рішеннями. Це сприяє швидкому вирішенню проблем, обміну досвідом та поширенню кращих практик у використанні мови програмування Python.

Загалом, Python є потужним інструментом для розвитку програмного забезпечення у різних областях та відкриває широкі можливості для творчості та інновацій у сучасному програмуванні.

Розглянемо середовище розробки PyCharm з наукової перспективи. PyCharm є інтегрованим середовищем розробки (Integrated Development Environment, IDE) для програмування на мові програмування Python(рис.2.1). Це програмне забезпечення

розроблене компанією JetBrains та призначене для підтримки різних аспектів програмування, таких як редагування коду, відлагодження, взаємодія з системами контролю версій та інші.

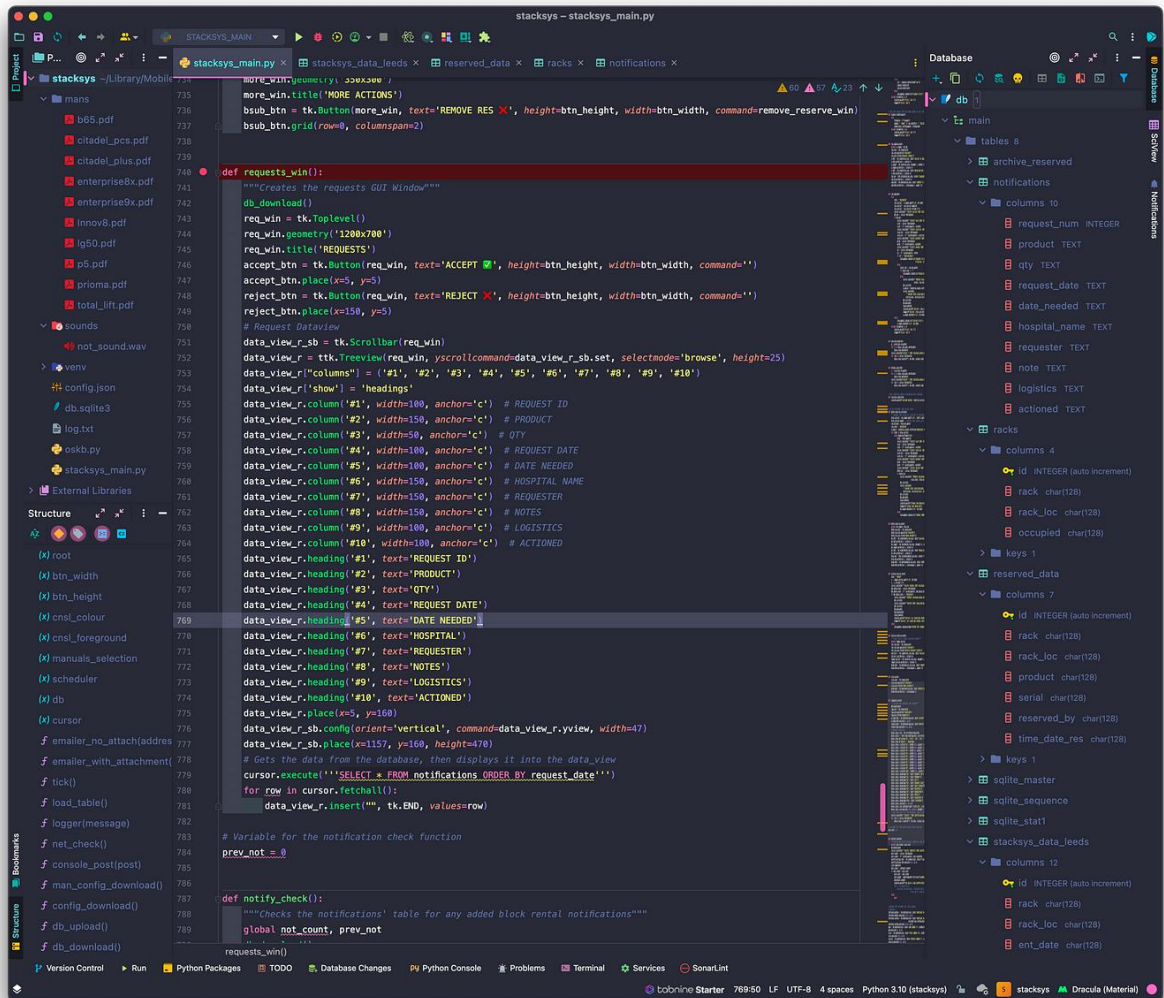


Рисунок 2.1 – Середовище PyCharm

Однією з ключових особливостей PyCharm є його можливості розширення та налаштування відповідно до потреб користувача. Інтегровані інструменти дозволяють забезпечити ефективність розробки проектів різної складності, починаючи від невеликих скриптів до великих програмних продуктів[13].

У середовищі PyCharm реалізовані такі важливі можливості, як автодоповнення коду, перевірка синтаксису, відлагодження за допомогою інтерактивного дебагера та інші. Ці можливості допомагають підвищити

продуктивність розробників та зменшити ймовірність виникнення помилок у програмному коді.

З огляду на поширеність мови програмування Python у різних галузях інформатики та індустрії, середовище розробки PyCharm має велике значення для розробників усіх рівнів кваліфікації. Його використання сприяє підвищенню якості програмного забезпечення та прискорює процес розробки проектів на мові Python.

Науковий аспект вивчення середовища розробки PyCharm полягає в аналізі його можливостей та ефективності в контексті різних сценаріїв розробки програмного забезпечення. Дослідження може спрямовуватися на вдосконалення інструментів розробки, оптимізацію робочого процесу та підвищення продуктивності розробників.

2.2 Структура програми

Структура програми представлена на рисунку 2.2.

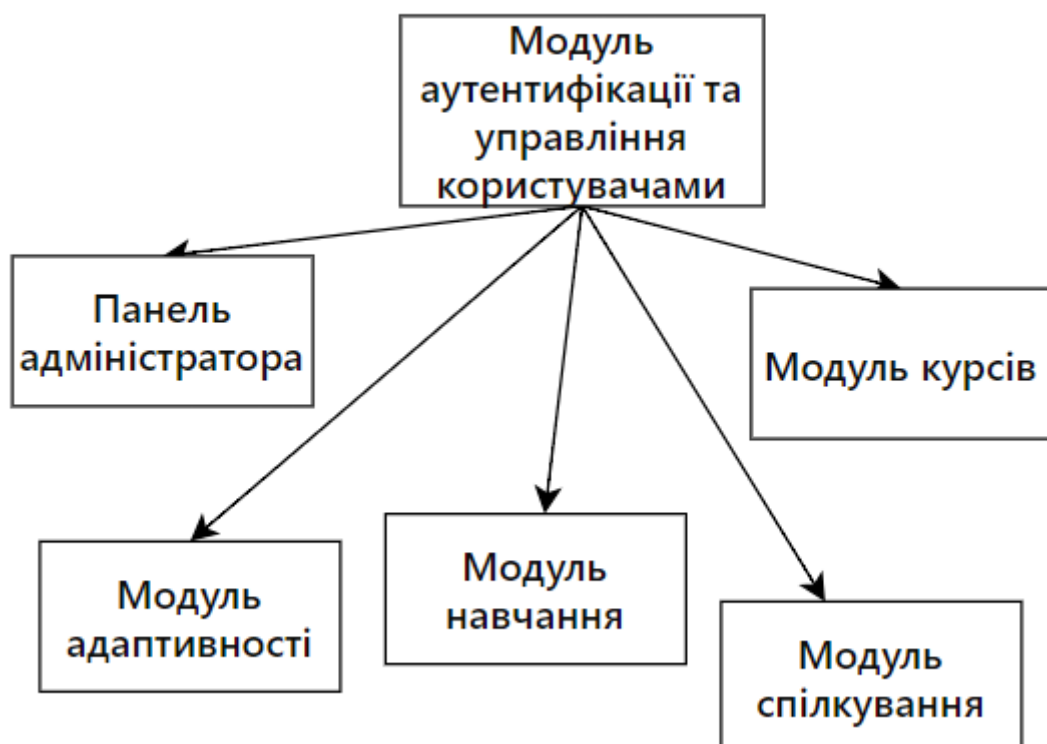


Рисунок 2.2 – Структура програми

Модуль аутентифікації та управління користувачами в програмному застосунку електронного навчання виконує ключову роль у забезпеченні безпеки та ідентифікації користувачів. Він включає в себе комплекс функцій, спрямованих на перевірку ідентичності користувачів та керування їхніми правами доступу до різних ресурсів системи. Завданням модуля є забезпечення автентифікації користувачів на основі їхніх облікових даних, таких як ім'я користувача та пароль, а також забезпечення заходів безпеки для уникнення несанкціонованого доступу до системи[13].

Цей модуль включає в себе процеси реєстрації нових користувачів, підтвердження їхньої ідентичності та надання прав доступу до різних функціональних можливостей системи в залежності від їхніх ролей та повноважень. Він також забезпечує можливість управління обліковими записами користувачів, включаючи скасування, блокування та видалення облікових записів за необхідності.

Одним з важливих аспектів цього модуля є забезпечення безпеки облікових даних користувачів. Це включає в себе застосування різноманітних методів шифрування та захисту паролів, а також реалізацію заходів для запобігання атакам на облікові записи, таких як перехоплення сесій, SQL-ін'єкції та інші види кіберзлочинності.

Крім того, модуль аутентифікації та управління користувачами забезпечує логування дій користувачів в системі, що дозволяє вести журнали активності та аудиту, спрощуючи виявлення та реагування на потенційні проблеми безпеки та порушення. Такий підхід до управління користувачами в електронних навчальних системах допомагає забезпечити високий рівень безпеки та конфіденційності даних користувачів, що є важливим аспектом у сучасному інформаційному середовищі.

Панель адміністратора є ключовим компонентом програмного застосунку електронного навчання, який забезпечує адміністративний доступ до системи для управління різними аспектами навчального процесу. Цей модуль включає в себе широкий набір функцій та інструментів, які дозволяють адміністраторам ефективно керувати ресурсами, користувачами та контентом системи.

Однією з основних функцій панелі адміністратора є управління користувачами, що включає створення нових облікових записів, надання прав доступу, редагування профілів та вирішення проблем з авторизацією. Крім того, адміністратор може блокувати або видаляти облікові записи користувачів в разі потреби.

Додатково, панель адміністратора надає можливість керування навчальним контентом, що включає створення нових курсів, додавання матеріалів, редагування існуючих модулів та видалення застарілого контенту. Це дозволяє адміністраторам підтримувати актуальність та релевантність навчальної інформації у системі.

Панель адміністратора також включає в себе засоби аналітики та звітності, які дозволяють збирати та аналізувати дані про активність користувачів, успішність навчання, використання ресурсів та інші ключові показники. Це допомагає адміністраторам приймати обґрунтовані рішення щодо оптимізації навчального процесу та підвищення ефективності системи в цілому[14].

Загалом, панель адміністратора є важливим інструментом управління програмним застосунком електронного навчання, який забезпечує адміністраторам потрібний рівень контролю, доступу та аналітики для ефективного функціонування системи.

Модуль курсів у системі електронного навчання є ключовим компонентом, що забезпечує структуру та організацію навчальних матеріалів для користувачів. Цей модуль відповідає за управління навчальними програмами, створення та підтримку курсів, надання доступу до навчального контенту для студентів та викладачів. Він включає в себе функціонал для розгортання нових курсів, редагування існуючих, а також вивчення та аналізу активності користувачів.

Основними функціями модуля курсів є створення структурованого навчального плану, що включає в себе розділи, теми та завдання, організація навчального матеріалу у вигляді лекцій, відеоуроків, тексти тощо, а також надання можливості викладачам взаємодіяти зі студентами шляхом надання завдань, виставлення оцінок та зворотнього зв'язку.

Модуль курсів також забезпечує можливість встановлення прав доступу до курсів для різних категорій користувачів, забезпечуючи конфіденційність та безпеку навчального процесу. Крім того, він може включати функціонал для автоматизації процесів, таких як генерація звітів про виконання курсів, нагадування про дедлайни та інші аспекти управління навчальним процесом.

У контексті педагогічної науки модуль курсів є інструментом для реалізації основних принципів навчання, таких як структурованість, систематичність, інтерактивність та індивідуалізація. Він дозволяє викладачам ефективно організовувати та проводити навчальний процес, забезпечуючи студентам доступ до необхідних знань та навичок у зручний та доступний спосіб.

Модуль навчання у програмному застосунку електронного навчання є ключовим компонентом, що забезпечує взаємодію між студентами та навчальним контентом. Його основна функція полягає в наданні студентам доступу до матеріалів курсів, проведенні навчальних заходів та оцінці їхнього прогресу[15].

Передбачається, що модуль навчання буде включати інтерактивність, що дозволить студентам взаємодіяти з навчальними матеріалами, такими як відеоуроки, текстові матеріали, завдання та тести. Інтерактивність може виявлятися у формі можливості розв'язування завдань прямо у системі, коментування матеріалів або спілкування з викладачами та студентами.

Модуль навчання також передбачає систему оцінювання, що дозволяє студентам здавати тести, виконувати завдання та отримувати оцінки за свою роботу. Викладачам надається можливість перегляду та оцінювання результатів студентів, що дозволяє їм відстежувати прогрес кожного учасника курсу та надавати необхідну підтримку.

Крім того, модуль навчання може бути розширений функціоналом адаптивності, що дозволить системі враховувати індивідуальні особливості та потреби кожного студента. Застосування алгоритмів машинного навчання дозволить персоналізувати навчальний процес, пропонуючи студентам індивідуальні рекомендації та завдання відповідно до їхнього рівня знань та індивідуальних потреб.

Модуль спілкування у програмному застосунку електронного навчання відіграє важливу роль у забезпеченні ефективної комунікації між учасниками навчального процесу. Цей модуль призначений для сприяння взаємодії між студентами та викладачами, стимулювання обговорень, вирішення питань та спільної роботи над навчальними завданнями.

Однією з основних функцій модуля спілкування є створення віртуального простору для обміну ідеями та думками, який допомагає студентам отримувати підтримку від колег та викладачів, а також ділитися своїми знаннями та досвідом з іншими учасниками навчального процесу.

Крім того, модуль спілкування може включати в себе різноманітні інструменти для організації комунікації, такі як форуми, чати, системи обміну повідомленнями тощо. Це дозволяє користувачам вибирати найзручніший спосіб спілкування залежно від контексту та потреб.

Додатково, модуль спілкування може включати інструменти для організації спільних робочих груп, де студенти можуть спільно працювати над проєктами, виконувати завдання або обговорювати конкретні теми.

У плані організації, модуль спілкування повинен забезпечувати можливість модерації та контролю за обговореннями, щоб забезпечити безпеку та відповідність правилам етики спілкування[16].

Таким чином, модуль спілкування у програмному застосунку електронного навчання є необхідним елементом для підтримки взаємодії між учасниками навчального процесу, сприяючи створенню сприятливого середовища для навчання, обміну знаннями та співпраці.

Модуль адаптивності в системі електронного навчання є ключовим елементом, спрямованим на індивідуалізацію навчального процесу та оптимізацію навчального досвіду для кожного користувача. Цей модуль базується на принципах адаптивного навчання, яке враховує потреби, вміння, знання та індивідуальні характеристики кожного учня з метою максимізації його успішності та ефективності навчання.

В основі модуля адаптивності лежить використання алгоритмів машинного навчання та штучного інтелекту, які аналізують поведінку, відповіді на завдання, успішність у навчанні та інші параметри користувача для створення персоналізованих навчальних траєкторій. Ці алгоритми можуть враховувати рівень знань, швидкість засвоєння матеріалу, особливості виконання завдань та інші фактори для пропозиції навчальних матеріалів та завдань, які найбільш відповідають потребам конкретного користувача.

Один з важливих аспектів модуля адаптивності – це постійне оновлення та вдосконалення алгоритмів на основі зібраних даних та відгуків користувачів. Це дозволяє підтримувати високу точність адаптивної системи та забезпечувати постійне покращення навчального процесу.

Модуль адаптивності може також включати інтерфейс для взаємодії користувачів з системою, де вони можуть вказувати свої переваги, цілі та очікування від навчання. Ця взаємодія дозволяє системі краще розуміти потреби користувачів та надавати їм більш ефективні навчальні матеріали та завдання.

У цілому, модуль адаптивності в електронному навчанні є важливим інструментом, який сприяє покращенню якості навчання та забезпечує індивідуалізований підхід до кожного учня, що допомагає зростанню їхнього академічного успіху.

2.3 Типи UML діаграм та їх застосування у розробці програмного забезпечення

UML діаграми, або діаграми мови моделювання об'єктно-орієнтованих систем, є важливим інструментом для візуалізації структури та поведінки системи. Ці діаграми використовуються для моделювання різних аспектів програмного забезпечення, від структури класів до взаємодії між об'єктами під час виконання програми.

Одним із видів UML діаграм є діаграма класів, яка відображає структуру системи через відображення класів, їх атрибутів та методів, а також взаємозв'язків між класами. Ця діаграма дозволяє програмістам та аналітикам краще зрозуміти організацію програмного коду та його компонентів.

Ще одним типом UML діаграм є діаграма взаємодії, яка моделює взаємодію між об'єктами під час виконання програми[17]. Ці діаграми відображають обмін повідомленнями між об'єктами та порядок їх виконання, допомагаючи розібратися в тому, як система працює в реальному часі.

Крім того, UML діаграми включають діаграми станів, діаграми активностей, діаграми компонентів та діаграми розгортання, які кожна відображає певний аспект системи. Ці діаграми допомагають узагальнити інформацію про систему та забезпечити зручний інструмент для аналізу та вдосконалення програмного забезпечення.

UML діаграми є важливим етапом у процесі розробки програмного забезпечення, оскільки вони дозволяють команді розробників спілкуватися та спільно працювати над проектом, визначаючи його архітектуру та функціональність. Вони є потужним інструментом для розуміння та аналізу складних систем, що допомагає забезпечити якість та ефективність розробки програмного забезпечення.

Умовна модель варіантів використання (Use Case Diagram) в рамках методології UML (Unified Modeling Language) відображає функціональність системи та її взаємодію з користувачами або зовнішніми системами. Ця діаграма дозволяє ідентифікувати основні можливості системи та ролі, які взаємодіють з цими можливостями.

Умовна модель варіантів використання включає в себе набір акторів та варіантів використання. Актори представляють ролі, які взаємодіють з системою, тоді як варіанти використання описують конкретні дії або сценарії, які можуть виконувати ці актори.

Актори включають в себе користувачів, зовнішні системи, або будь-які інші елементи, що мають взаємодіяти з системою. Кожен актор має свої унікальні ролі та

пов'язані з ними варіанти використання. Варіанти використання описують конкретні сценарії взаємодії між акторами та системою, включаючи послідовність дій та реакції на події.

Умовна модель варіантів використання допомагає визначити функціональні вимоги до системи, визначити структуру системи та її взаємодію з користувачами та іншими системами. Ця модель є важливим інструментом для аналізу та проектування системи, що допомагає уточнити потреби користувачів та визначити ключові можливості системи перед її розробкою.

Діаграма варіантів використання застосунку представлена на рисунку 2.3.

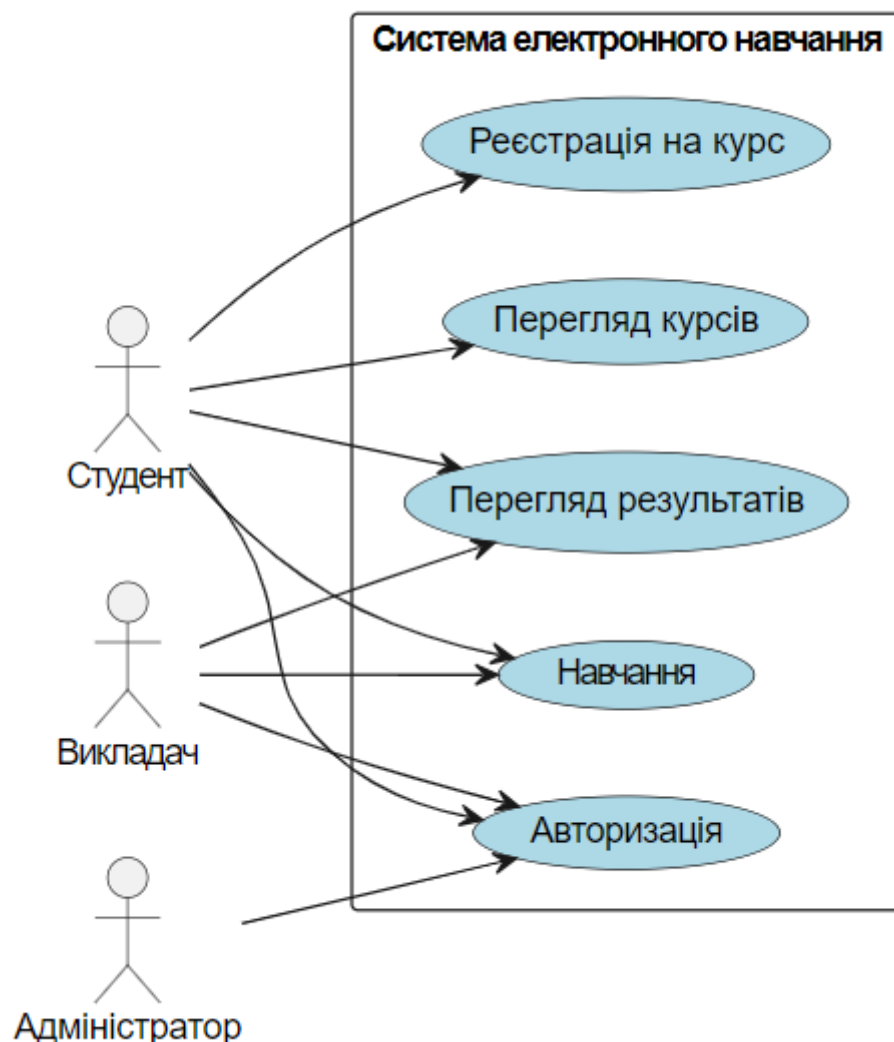


Рисунок 2.3 – Діаграма варіантів використання системи

У створеній діаграмі варіантів використання для системи електронного навчання використовується мова UML (Unified Modeling Language), що є стандартом для моделювання програмних систем. Діаграма відображає основні функціональні можливості системи та взаємодію з користувачами.

У діаграмі присутні три основні актори: "Студент", "Викладач" та "Адміністратор". Кожен з цих акторів представляє роль, яка має доступ до системи електронного навчання та взаємодіє з нею. Наприклад, "Студент" може переглядати доступні курси, реєструватися на курси та навчатися, "Викладач" може додавати навчальний матеріал та оцінювати студентів, а "Адміністратор" має доступ до адміністративних функцій системи.

Варіанти використання включають в себе такі дії, як авторизація користувачів у системі, перегляд доступних курсів, реєстрація на курси, процес навчання, перегляд результатів та інші функціональні можливості, які доступні користувачам.

Структура діаграми розподіляється на відповідні пакети, що відображають різні аспекти функціональності системи електронного навчання. Кожен варіант використання має свою назву та опис, що дозволяє зрозуміти, яку конкретну дію він представляє.

Діаграма варіантів використання є важливим інструментом для аналізу та проектування системи, оскільки вона дозволяє ідентифікувати основні функціональність та взаємодію з користувачами, що допомагає визначити вимоги до системи та структуру її реалізації.

UML (Unified Modeling Language) діаграма послідовності використовується для візуалізації взаємодії між об'єктами у системі в конкретний момент часу. Ця діаграма дозволяє моделювати послідовність повідомлень, що передаються між об'єктами під час виконання певної функції або сценарію. Вона корисна для розуміння порядку виконання операцій та взаємодії між елементами системи[18].

Діаграма послідовності складається з вертикальних ліній, які представляють різні об'єкти чи учасники системи, і горизонтальних стрілок, що показують передачу повідомлень між ними. Кожна стрілка має написання, яке вказує на тип повідомлення та параметри, які передаються.

У процесі створення UML діаграми послідовності спочатку визначаються учасники системи, а потім моделюється послідовність взаємодії між ними під час виконання певної дії чи функції. Це може включати послідовність викликів методів, передачу даних та зміну стану об'єктів.

Діаграма послідовності є потужним інструментом для аналізу та проектування систем, оскільки дозволяє візуалізувати взаємодію між об'єктами та ілюструвати порядок виконання операцій у системі. Вона допомагає розуміти логіку програми та виявляти потенційні проблеми в проєкті ще на етапі розробки.

Діаграма послідовності представлена на рисунку 2.4.

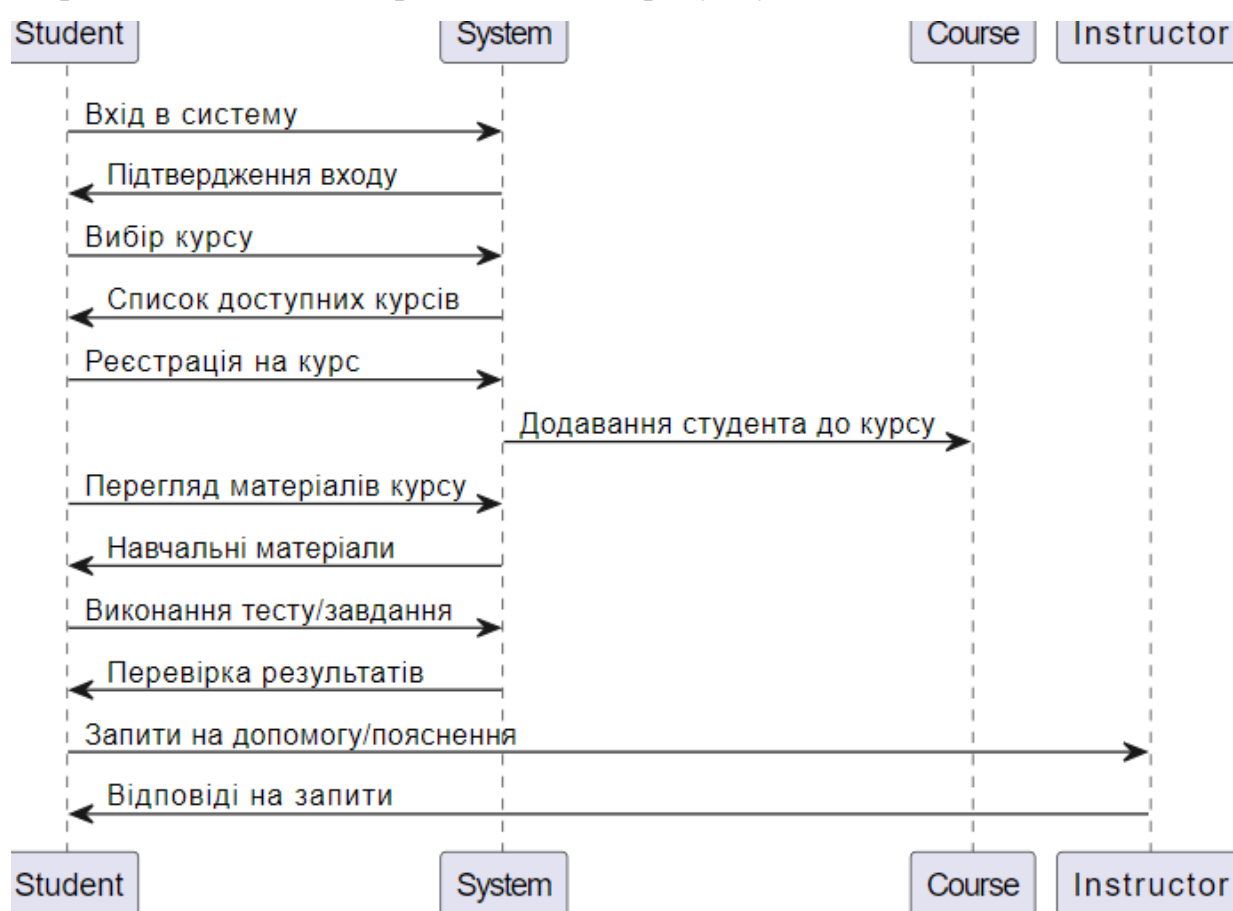


Рисунок 2.4 – Діаграма послідовності системи

Надана UML діаграма послідовності для системи електронного навчання відображає взаємодію різних учасників та компонентів системи під час виконання різних дій та сценаріїв користування. У діаграмі представлені чотири основних учасника: студент, система, курс та інструктор.

Взаємодія починається з входу студента в систему, після чого система підтверджує його вхід та надає список доступних курсів. Студент обирає бажаний курс та реєструється на нього, що спричиняє додавання студента до курсу у системі. Після цього студент може переглядати навчальні матеріали курсу та виконувати тести або завдання[19].

Система відповідає на запити студента, надаючи необхідні навчальні матеріали, та перевіряє результати виконаних тестів або завдань. У разі необхідності студент може звертатися до інструктора з запитом на допомогу або пояснення. Інструктор, у свою чергу, надає відповіді на запити студента.

Ця діаграма послідовності створена з метою візуалізації та уточнення логіки взаємодії між учасниками системи електронного навчання, що допомагає розуміти та аналізувати процес навчання з більшою чіткістю та ефективністю.

У проектуванні програмного забезпечення UML-діаграми класів використовуються для моделювання структури системи та відображення класів, їх взаємозв'язків і властивостей. Діаграма класів визначає архітектуру програми, показуючи, як класи пов'язані між собою та як вони взаємодіють.

На діаграмі класів кожен клас представлений у вигляді прямокутника з трьома відокремленими секціями: назвою класу, його атрибутами та методами. Зв'язки між класами представлені стрілками, які вказують на відношення між ними, такі як асоціація, агрегація чи композиція.

Кожен клас на діаграмі може мати різноманітні атрибути, що відображають його характеристики та властивості, а також методи, які визначають його поведінку та функціональність. Асоціації між класами вказують на взаємозв'язок між об'єктами цих класів, а композиції та агрегації відображають структурні залежності між класами, вказуючи на те, як вони організовані та взаємодіють між собою.

Діаграма класів є потужним інструментом для аналізу та проектування програмних систем, оскільки вона дозволяє розглядати систему на високому рівні абстракції та визначати її структуру та компоненти. Вона допомагає розуміти взаємодію між класами, визначати їхні взаємозв'язки та групувати їх у логічні секції для кращого розуміння та керованості кодом програми.

Діаграма класів представлена на рисунку 2.5.

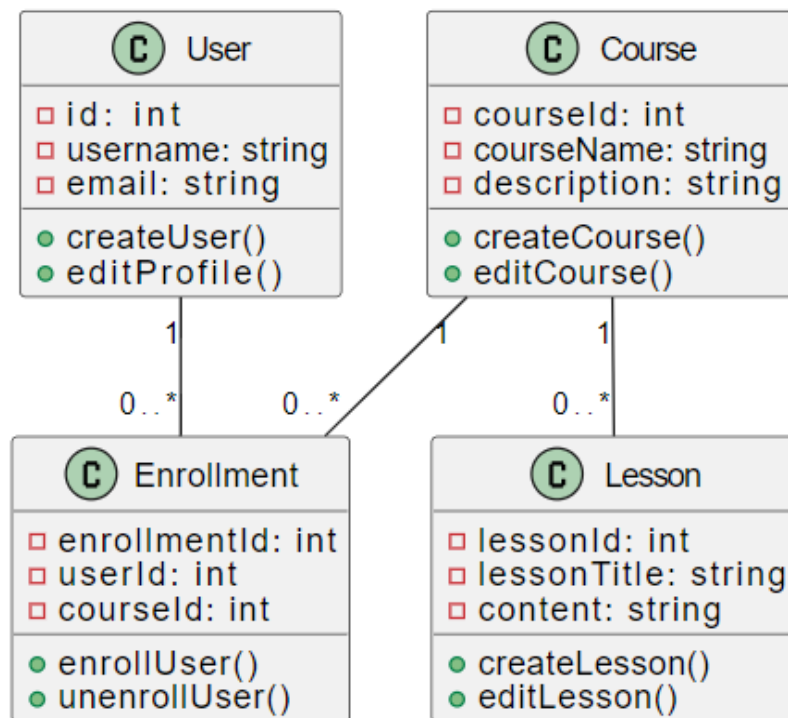


Рисунок 2.5 – Діаграма класів системи

Розроблена UML-діаграма класів відображає структуру системи електронного навчання та визначає основні компоненти, їх взаємозв'язки та властивості. Діаграма складається з чотирьох основних класів: `User`, `Course`, `Lesson` та `Enrollment`. Кожен з цих класів представляє певну сутність в системі та має визначені атрибути та методи, які відображають характеристики та функціональність цих сутностей.

Клас ``User`` відображає користувачів системи та має атрибути, такі як ``id``, ``username`` та ``email``, які відображають ідентифікатор, ім'я користувача та електронну пошту відповідно. Методи цього класу, такі як ``createUser()`` та ``editProfile()``, дозволяють створювати нових користувачів та редагувати їх профіль.

Клас ``Course`` представляє курси, які доступні в системі, та має атрибути, такі як ``courseId``, ``courseName`` та ``description``, що відображають ідентифікатор курсу, назву та опис. Методи ``createCourse()`` та ``editCourse()`` надають можливість створювати та редагувати інформацію про курси.

Клас ``Lesson`` відображає уроки, що належать до певного курсу, та має атрибути, такі як ``lessonId``, ``lessonTitle`` та ``content``, які відображають ідентифікатор уроку, назву та вміст відповідно. Методи ``createLesson()`` та ``editLesson()`` дозволяють додавати нові уроки та редагувати їх вміст.

Клас ``Enrollment`` відображає записи студентів на курси та має атрибути, такі як ``enrollmentId``, ``userId`` та ``courseId``, які відображають ідентифікатор запису, ідентифікатор користувача та ідентифікатор курсу відповідно. Методи ``enrollUser()`` та ``unenrollUser()`` дозволяють студентам записуватися та відписуватися від курсів.

У діаграмі також показані асоціації між класами, що вказують на взаємозв'язки між ними. Наприклад, асоціація між класами ``User`` та ``Enrollment`` показує, що один користувач може мати багато записів на курси, тоді як кожен курс може мати багато студентів. Таким чином, діаграма класів визначає структуру та компоненти системи електронного навчання та вказує на їхні взаємозв'язки для подальшого проектування та розробки програмного забезпечення.

2.4 Вибір та обґрунтування бази даних

SQLite є вбудовуваною, серверно-менеджерською, архітектурною базою даних, яка використовується для зберігання та керування реляційними даними. Ця система управління базами даних (СУБД) надає набір функцій для створення, оновлення та оптимізації баз даних без необхідності розгортання окремого сервера.

SQLite використовується у широкому спектрі додатків, від малих мобільних додатків до великих веб-проектів, завдяки своїй простоті використання та високій продуктивності.

Ця СУБД відрізняється від інших систем зберігання даних, таких як MySQL чи PostgreSQL, оскільки вона не працює як окремий процес або сервер, а замість цього вбудовується безпосередньо в додаток. Це робить SQLite ідеальним вибором для мобільних додатків та інших проектів, які вимагають локального збереження даних без необхідності підключення до віддаленого сервера.

Однією з ключових особливостей SQLite є його нульова конфігурація, що означає, що для початку використання не потрібно виконувати складні налаштування або конфігурувати сервер. SQLite автоматично створює базу даних у вигляді одного файлу на диску, що спрощує роботу з даними та робить його легко переносимим.

Ще однією перевагою SQLite є його швидкість та ефективність. Вона має низький рівень накладних витрат завдяки простому файловому формату та оптимізації операцій зчитування/запису даних[20]. Це робить SQLite ідеальним вибором для додатків з великим обсягом операцій збереження та доступу до даних.

Крім того, SQLite підтримує широкий спектр типів даних, включаючи текстові, числові, дати, бінарні дані та багато інших. Це дозволяє розробникам зберігати різноманітні дані у своїх додатках з високою ефективністю та гнучкістю.

3. ПРАКТИЧНА ЧАСТИНА

3.1 Розробка бази даних

Структура бази даних представлена на рисунку 3.1.

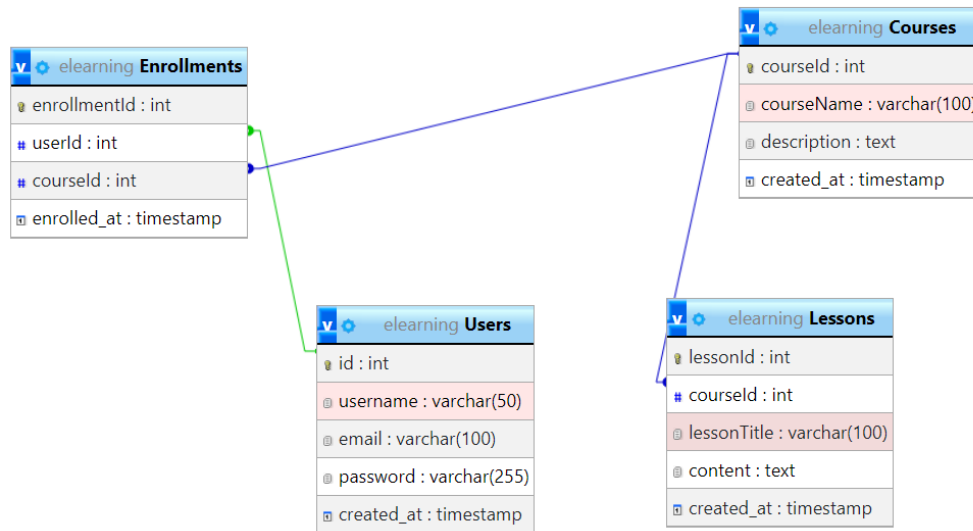


Рисунок 3.1 – Структура бази даних

Створена структура бази даних для системи електронного навчання складається з чотирьох основних таблиць: 'Users', 'Courses', 'Lessons' та 'Enrollments'. Кожна таблиця відповідає певній сутності в системі та містить атрибути, які відображають основні характеристики цих сутностей. Цілісність даних та взаємозв'язки між таблицями забезпечуються за допомогою зовнішніх ключів та індексів.

Таблиця 'Users' містить інформацію про користувачів системи. Вона включає такі поля, як 'id', 'username', 'email', 'password' та 'created_at'. Поле 'id' є первинним ключем та автоматично збільшується для кожного нового користувача. Поля 'username' та 'email' зберігають унікальні ідентифікатори користувачів, а поле 'password' містить зашифровані паролі. Поле 'created_at' автоматично заповнюється поточним часовим значенням під час створення запису.

Таблиця 'Courses' зберігає інформацію про курси, що пропонуються в системі. Вона включає поля 'courseId', 'courseName', 'description' та 'created_at'. Поле

`courseId` є первинним ключем та автоматично збільшується. Поле `courseName` містить назву курсу, а поле `description` — його опис. Поле `created\_at` автоматично заповнюється під час створення курсу.

Таблиця `Lessons` відображає уроки, що належать до курсів. Вона містить поля `lessonId`, `courseId`, `lessonTitle`, `content` та `created\_at`. Поле `lessonId` є первинним ключем та автоматично збільшується. Поле `courseId` є зовнішнім ключем, що посилається на `courseId` у таблиці `Courses`, забезпечуючи зв'язок між уроками та курсами. Поле `lessonTitle` містить назву уроку, а поле `content` — його вміст. Поле `created\_at` автоматично заповнюється під час створення уроку.

Таблиця `Enrollments` зберігає записи про студентів, які підписалися на курси. Вона включає поля `enrollmentId`, `userId`, `courseId` та `enrolled\_at`. Поле `enrollmentId` є первинним ключем та автоматично збільшується. Поле `userId` є зовнішнім ключем, що посилається на `id` у таблиці `Users`, а поле `courseId` є зовнішнім ключем, що посилається на `courseId` у таблиці `Courses`. Це забезпечує зв'язок між користувачами та курсами. Поле `enrolled\_at` автоматично заповнюється під час створення запису.

Для оптимізації пошуку та підвищення продуктивності запитів створені індекси на полях `username` у таблиці `Users`, `courseName` у таблиці `Courses`, `courseId` у таблиці `Lessons` та `userId` і `courseId` у таблиці `Enrollments`. Ці індекси дозволяють швидко знаходити необхідні записи та підвищують ефективність роботи бази даних.

Таким чином, розроблена структура бази даних забезпечує надійне зберігання інформації про користувачів, курси, уроки та записи на курси, а також забезпечує цілісність даних та ефективність їх обробки в системі електронного навчання.

3.2 Програмний інтерфейс користувача

Навігаційна панель та структура меню є критично важливими компонентами інтерфейсу користувача системи електронного навчання, які забезпечують

інтуїтивний доступ до основних функцій та розділів системи. Ефективна навігаційна панель дозволяє користувачам легко орієнтуватися у великій кількості інформації, що сприяє підвищенню продуктивності та задоволеності користувачів.

Навігаційна панель повинна бути розміщена у видимій частині інтерфейсу, зазвичай у верхній або лівій частині екрану, щоб забезпечити зручний доступ до неї в будь-який момент часу. Вона повинна містити основні розділи, такі як "Головна", "Курси", "Мій профіль", "Повідомлення", "Налаштування" та "Вихід". Кожен розділ повинен бути чітко позначений та логічно організований, що сприяє швидкому розумінню та використанню системи навіть новими користувачами.

Структура меню повинна бути багаторівневою, що дозволяє користувачам переходити від загальних категорій до більш детальних підкатегорій. Наприклад, у розділі "Курси" можуть бути підрозділи "Мої курси", "Всі курси", "Категорії курсів" та "Мої оцінки". Така ієрархічна структура забезпечує гнучкість та дозволяє зберігати порядок навіть при значному обсязі інформації.

Інтерактивні елементи навігаційної панелі, такі як випадаючі меню та значки сповіщень, повинні бути інтегровані безпосередньо у панель для забезпечення швидкого доступу до важливої інформації, такої як нові повідомлення, сповіщення про нові завдання або оновлення у курсах. Це дозволяє користувачам залишатися в курсі подій та своєчасно реагувати на них.

Ергономіка та адаптивність навігаційної панелі також відіграють важливу роль. Навігаційна панель повинна адаптуватися до різних розмірів екранів та пристроїв, забезпечуючи зручність використання як на великих моніторах, так і на мобільних пристроях. Для цього можуть використовуватися такі технології, як адаптивний дизайн та мобільні версії інтерфейсу, що дозволяють зберігати функціональність та зручність незалежно від платформи.

Кольорова схема та візуальні елементи навігаційної панелі повинні бути ретельно підібрані для забезпечення читабельності та візуального комфорту. Використання контрастних кольорів для основних елементів навігації, таких як активні розділи та сповіщення, сприяє легкому сприйняттю інформації. Крім того,

важливо забезпечити відповідність кольорової схеми загальному стилю інтерфейсу, що підвищує естетичну привабливість системи.

Таким чином, навігаційна панель та структура меню є фундаментальними елементами інтерфейсу користувача, які визначають зручність та ефективність взаємодії користувачів з системою електронного навчання. Оптимальне поєднання логічної організації, інтерактивності, адаптивності та візуальної привабливості забезпечує позитивний досвід користування та сприяє успішному впровадженню системи в освітній процес.

Панель адміністратора у системі електронного навчання є ключовим компонентом, що забезпечує ефективне управління та моніторинг усіх аспектів системи. Вона слугує інструментом для адміністраторів, які відповідають за організацію та підтримку навчального процесу, а також за забезпечення надійності та безпеки системи. Панель адміністратора повинна бути розроблена з урахуванням потреб користувачів, які виконують адміністрування, забезпечуючи інтуїтивно зрозумілий інтерфейс та багатофункціональність.

Однією з основних функцій панелі адміністратора є управління користувачами. Це включає реєстрацію нових користувачів, надання їм відповідних ролей та прав доступу, а також моніторинг їхньої активності в системі. Адміністратори мають можливість редагувати профілі користувачів, змінювати їхні права доступу та видаляти користувачів у разі необхідності. Такий контроль за користувачами дозволяє підтримувати безпеку системи та забезпечувати відповідність її використання політикам конфіденційності та безпеки.

Управління курсами є ще однією важливою функцією панелі адміністратора. Адміністратори мають можливість створювати, редагувати та видаляти курси, а також призначати викладачів до конкретних курсів. Вони можуть організовувати структуру курсів, додаючи модулі та уроки, забезпечуючи тим самим логічну послідовність навчальних матеріалів. Також адміністратори можуть контролювати доступ до курсів, визначаючи, хто з користувачів може записатися на певний курс.

Моніторинг та звітність є критично важливими аспектами роботи адміністратора. Панель адміністратора повинна надавати детальні звіти про

активність користувачів, успішність студентів, відвідуваність курсів та інші ключові показники. Це дозволяє адміністраторам аналізувати ефективність навчального процесу, виявляти проблемні зони та приймати обґрунтовані рішення щодо покращення якості освіти. Важливим є також можливість генерації та експорту звітів у різних форматах для подальшого аналізу.

Панель адміністратора повинна включати засоби для управління контентом, що використовується в навчальних курсах. Це передбачає можливість завантаження, редагування та організації різних типів навчальних матеріалів, таких як тексти, відео, аудіо та інтерактивні вправи. Адміністратори повинні мати можливість легко інтегрувати нові навчальні ресурси та забезпечувати їх доступність для викладачів та студентів.

Безпека та конфіденційність є ще одним важливим аспектом панелі адміністратора. Вона повинна включати інструменти для управління безпекою системи, такі як налаштування політик паролів, контроль доступу, ведення журналу подій та моніторинг підозрілої активності. Адміністратори повинні мати можливість швидко реагувати на потенційні загрози та забезпечувати захист особистих даних користувачів.

Отже, панель адміністратора в системі електронного навчання є багатофункціональним інструментом, що забезпечує ефективне управління всіма аспектами системи. Вона включає управління користувачами, курсами, контентом, моніторингом та звітністю, а також забезпеченням безпеки та конфіденційності. Така панель дозволяє адміністраторам підтримувати високу якість навчального процесу та забезпечувати стабільність і надійність системи.

Інтерфейс створення та управління курсами в системі електронного навчання є критично важливим компонентом, що забезпечує викладачів необхідними інструментами для розробки, організації та адміністрування навчальних програм. Цей інтерфейс повинен бути інтуїтивно зрозумілим і функціональним, дозволяючи викладачам ефективно створювати навчальні матеріали, структуровані за темами та модулями, а також забезпечувати контроль за їх доступністю та актуальністю.

Процес створення курсу починається з надання можливості викладачу ввести основні дані про курс, такі як назва, опис, цільова аудиторія та очікувані результати навчання. Ця інформація допомагає студентам зрозуміти зміст та мету курсу перед його вибором. Інтерфейс повинен також забезпечувати можливість завантаження додаткових матеріалів, таких як силлабуси, графіки занять та інші документи, які допомагають структурувати навчальний процес.

Однією з важливих функцій інтерфейсу є можливість додавання та організації уроків. Викладачі повинні мати змогу створювати нові уроки, вказуючи їхні назви, зміст та необхідні матеріали. Інтерфейс повинен підтримувати різні типи контенту, включаючи текстові матеріали, відеолекції, інтерактивні завдання та мультимедійні елементи. Це забезпечує багатогранність та різноманітність навчальних ресурсів, що сприяє кращому засвоєнню матеріалу студентами.

Для ефективного управління курсами, інтерфейс повинен включати інструменти для контролю доступу до курсів. Викладачі мають змогу визначати, хто може підписуватися на курс, встановлювати умови доступу, такі як попереднє проходження певних модулів або наявність необхідних знань. Це дозволяє підтримувати належний рівень підготовки студентів та забезпечувати поступовий прогрес у навчанні.

Окремим аспектом є функціональність моніторингу та звітності. Інтерфейс повинен надавати можливість викладачам відстежувати активність студентів, їхній прогрес у навчанні та результати виконання завдань. Аналітичні інструменти допомагають оцінювати ефективність навчальних матеріалів та своєчасно виявляти проблемні моменти. Це дозволяє викладачам оперативно реагувати на потреби студентів та коригувати навчальний процес відповідно до їхніх вимог.

Загалом, інтерфейс створення та управління курсами має бути розроблений таким чином, щоб максимально спростити роботу викладачів та забезпечити їм усі необхідні інструменти для ефективного ведення навчальних програм. Це включає інтуїтивно зрозумілий дизайн, багатофункціональність та гнучкість, що дозволяє адаптувати курси до різноманітних освітніх потреб та методик.

Інтерфейс управління уроками є ключовим компонентом системи електронного навчання, призначеним для викладачів, що дозволяє їм ефективно створювати, редагувати та організовувати навчальні матеріали. Цей інтерфейс забезпечує зручність і гнучкість у роботі з контентом, що сприяє підвищенню якості освітнього процесу та задоволенню потреб як викладачів, так і студентів.

Основні функції інтерфейсу управління уроками включають можливість додавання нових уроків до курсу, редагування існуючих матеріалів та їх структурування. Викладачі можуть створювати уроки з різноманітним контентом, таким як текстові матеріали, відео, інтерактивні елементи та зовнішні ресурси. Для цього інтерфейс повинен підтримувати широкий спектр форматів файлів та інтеграцію з іншими системами, що дозволяє збагачувати навчальний процес.

Редагування уроків передбачає надання викладачам інструментів для зміни змісту, оновлення матеріалів та додавання нових розділів. Інтерфейс має бути інтуїтивно зрозумілим і забезпечувати простоту виконання цих завдань, що мінімізує час, витрачений на адміністративні функції, і дозволяє зосередитися на розробці якісного навчального контенту. Крім того, важливою є можливість попереднього перегляду уроків, що дозволяє викладачам бачити матеріали так, як їх бачитимуть студенти, та вносити необхідні корективи до публікації.

Організація уроків включає структурування матеріалів у логічні модулі та секції, що сприяє покращенню навігації та розумінню контенту студентами. Інтерфейс має забезпечувати можливість легкого переміщення уроків, їх групування за темами та встановлення послідовності вивчення. Це дозволяє створювати чітку та логічну структуру курсу, що підвищує його ефективність та сприяє кращому засвоєнню матеріалу студентами.

Однією з ключових вимог до інтерфейсу управління уроками є його адаптивність та інтерактивність. Інтерфейс має підтримувати адаптивний дизайн, що забезпечує зручне використання на різних пристроях, включаючи комп'ютери, планшети та смартфони. Це особливо важливо в сучасних умовах, коли студенти можуть мати доступ до навчальних матеріалів у різних ситуаціях і з різних пристроїв. Крім того, інтерактивність інтерфейсу дозволяє включати в уроки

елементи взаємодії, такі як тести, опитування, форуми та чати, що сприяє активному залученню студентів до навчального процесу.

Таким чином, інтерфейс управління уроками є невід'ємною частиною системи електронного навчання, яка забезпечує викладачам зручні інструменти для створення, редагування та організації навчальних матеріалів. Високий рівень функціональності та зручності цього інтерфейсу сприяє покращенню якості освітнього процесу, забезпечує адаптивність та інтерактивність навчання, що в свою чергу підвищує задоволеність студентів та ефективність їхнього навчання.

Інтерфейс для студентів у системі електронного навчання є ключовим елементом, який забезпечує доступ до навчальних матеріалів, комунікацію з викладачами та однокурсниками, а також виконання завдань і тестів. Ефективність інтерфейсу для студентів безпосередньо впливає на їхній досвід навчання, мотивацію та успішність у навчальному процесі.

Основним завданням інтерфейсу для студентів є забезпечення зручного та інтуїтивно зрозумілого доступу до курсів і навчальних матеріалів. Інтерфейс повинен мати добре організовану структуру, яка дозволяє студентам легко знаходити необхідні матеріали, переходити між модулями та уроками, а також отримувати актуальну інформацію про стан курсу, завдання та оцінки. Важливим аспектом є наявність зрозумілої навігаційної панелі, яка дозволяє швидко орієнтуватися в системі та знаходити потрібні розділи.

Інтерфейс для студентів також повинен включати інтерактивні елементи, які сприяють активній взаємодії з навчальним контентом. Це можуть бути відеолекції, інтерактивні вправи, тестові завдання та форуми для обговорення. Важливо, щоб всі ці елементи були легко доступні та мали інтуїтивно зрозумілий інтерфейс, який не викликає труднощів у користуванні. Інтерактивність і адаптивність навчальних матеріалів допомагають студентам глибше зрозуміти тему, забезпечуючи більш ефективне засвоєння знань.

Комунікація з викладачами та іншими студентами є ще одним важливим аспектом інтерфейсу. Система повинна забезпечувати зручні інструменти для обміну повідомленнями, проведення дискусій та консультацій. Це можуть бути

вбудовані чати, форуми, а також можливість проведення відеоконференцій. Така функціональність сприяє створенню спільноти, де студенти можуть обмінюватися знаннями, задавати питання та отримувати підтримку як від викладачів, так і від однокурсників.

Окрему увагу слід приділити інтерфейсу для виконання завдань і тестів. Студенти повинні мати можливість легко переглядати завдання, виконувати їх і завантажувати відповіді. Інтерфейс повинен забезпечувати зрозуміле відображення дедлайнів та інших важливих деталей щодо завдань. Також важливо забезпечити швидкий доступ до результатів тестів і зворотного зв'язку від викладачів, що дозволяє студентам своєчасно коригувати свої помилки та вдосконалювати знання.

Зрештою, інтерфейс для студентів повинен бути адаптований для використання на різних пристроях, включаючи мобільні телефони та планшети. Це забезпечує гнучкість і доступність навчання, дозволяючи студентам використовувати систему в будь-який час і в будь-якому місці. Мобільна адаптація інтерфейсу повинна зберігати всі основні функціональні можливості та забезпечувати зручний доступ до навчальних матеріалів та комунікаційних інструментів.

Таким чином, інтерфейс для студентів у системі електронного навчання повинен бути інтуїтивно зрозумілим, інтерактивним, зручним для комунікації та адаптивним до різних пристроїв. Всі ці аспекти сприяють покращенню навчального процесу, підвищенню мотивації студентів та їх успішності у навчанні.

Інтерфейс курсу в системі представлено на рисунку 3.2.



Рисунок 3.2 – Інтерфейс курсу «Технології виявлення кібератак»

Курс включає в себе набір лекцій для ознайомлення з темою. Набір тестів до лекцій для визначення рівня знань. Набір лабораторних для покращення практичних знань.

Загалом, інтерфейс застосунку є зручним та інтуїтивно зрозумілим користувачам.

3.3 Тестування функціоналу

Тестування функціоналу представлено на рисунках 3.3 – 3.5.

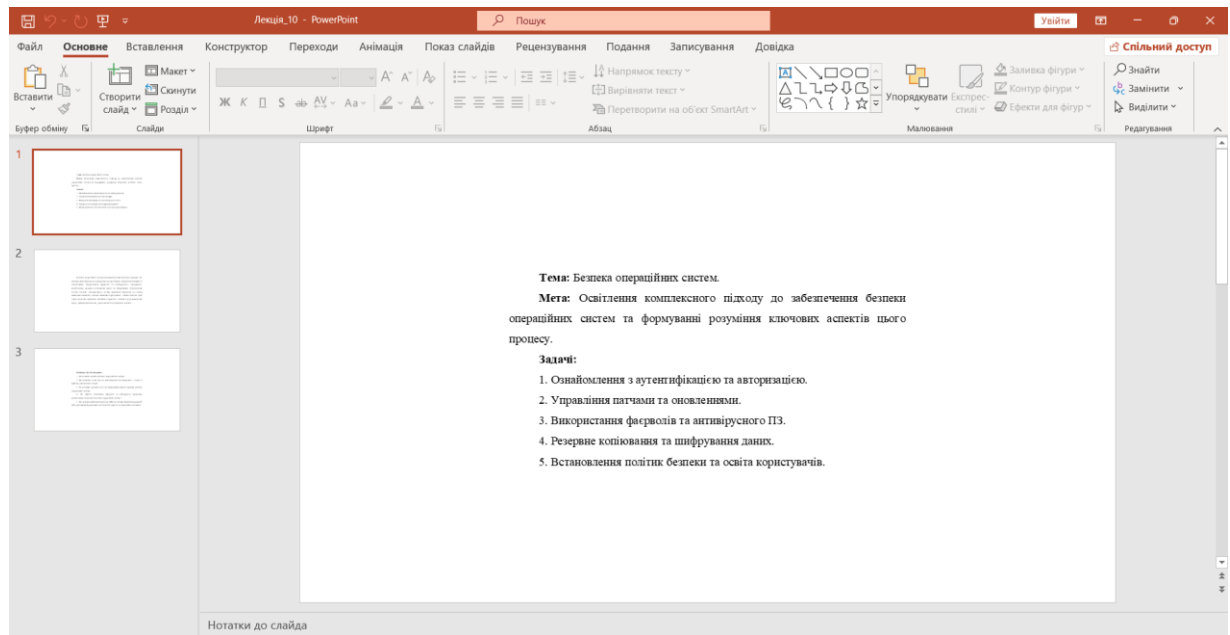


Рисунок 3.3 – Відкриття лекції

Тест №3. Основи захисту від вірусів та шкідливих програм

5708052@stud.nau.edu.ua [Змінити обліковий запис](#)

Спільно не використовується

Зірочка (*) указує, що запитання обов'язкове

ПІБ *

Ваша відповідь

Що таке шкідливе програмне забезпечення (malware)? * 1 бал

- ☐ а) Програми для оптимізації роботи комп'ютера;
- ☐ б) Програми для створення резервних копій;
- ☐ в) Програми, які завдають шкоди та отримують несанкціонований доступ;
- ☐ г) Програми для аналізу мережевого трафіку;
- ☐ д) Програми для автоматизації рутинних завдань;
- ☐ е) Програми для криптографічного захисту файлів;

Рисунок 3.4 – Відкриття тесту

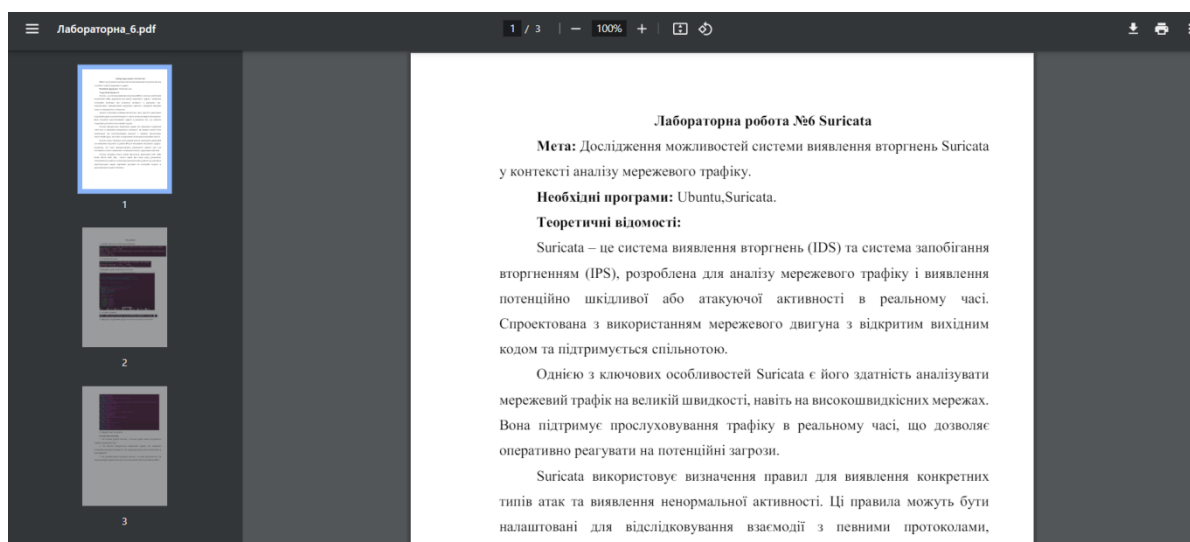


Рисунок 3.5 – Відкриття лабораторної роботи

У результаті тестування було зроблено висновок про те, що система дозволяє ефективно навчатись студентам та учням засобами електронного навчання.

ВИСНОВКИ

На основі проведеного дослідження створення системи електронного навчання можна зробити наступні висновки. Перш за все, розроблення програмного забезпечення для електронного навчання є актуальним завданням в сучасному освітньому середовищі, оскільки воно надає можливість доступу до освітніх матеріалів з будь-якого місця та в будь-який час. Така система сприяє зручності навчання та розвитку дистанційної освіти, що стає все більш важливим у сучасному світі.

Дослідження показало, що структура та функціональність програмного інтерфейсу користувача є критичними для успішного впровадження системи електронного навчання. Ефективна навігаційна панель, інтуїтивний інтерфейс реєстрації та входу, а також зручний доступ до курсів та навчального контенту сприяють покращенню користувацького досвіду та підвищенню зацікавленості користувачів у використанні системи.

Додатково, важливо звернути увагу на функціональність адміністративного інтерфейсу для ефективного управління системою, включаючи керування користувачами, курсами та навчальним контентом. Оптимізація процесів створення та управління курсами, а також можливість взаємодії між викладачами та студентами, також є важливими аспектами успішного впровадження системи.

Нарешті, важливо розглянути мобільний інтерфейс як необхідний компонент системи електронного навчання. Адаптивність та зручність використання на мобільних пристроях є ключовими факторами для покращення доступності та зручності навчання.

Отже, створення системи електронного навчання вимагає комплексного підходу та уважного розгляду різних аспектів розробки програмного забезпечення. Забезпечення зручності, ефективності та доступності є ключовими для успішного впровадження та подальшого розвитку системи електронного навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Guido van Rossum. "Python Programming for the Absolute Beginner". No Starch Press. 2010. 464 p.
2. Eric Matthes. "Python Crash Course: A Hands-On, Project-Based Introduction to Programming". No Starch Press. 2019. 544 p.
3. Mark Lutz. "Learning Python". O'Reilly Media. 2013. 1648 p.
4. Allen Downey. "Think Python: How to Think Like a Computer Scientist". O'Reilly Media. 2012. 292 p.
5. Al Sweigart. "Automate the Boring Stuff with Python: Practical Programming for Total Beginners". No Starch Press. 2015. 504 p.
6. Charles Severance. "Python for Everybody: Exploring Data in Python 3". CreateSpace Independent Publishing Platform. 2016. 244 p.
7. Jose Portilla. "Complete Python Bootcamp: Go from zero to hero in Python 3". UdeMy. 2020. Онлайн-курс.
8. Wes McKinney. "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython". O'Reilly Media. 2017. 544 p.
9. Jake VanderPlas. "Python Data Science Handbook: Essential Tools for Working with Data". O'Reilly Media. 2016. 548 p.
10. Joel Grus. "Data Science from Scratch: First Principles with Python". O'Reilly Media. 2015. 330 p.
11. John V. Guttag. "Introduction to Computation and Programming Using Python: With Application to Understanding Data". MIT Press. 2016. 472 p.
12. Al Sweigart. "Invent Your Own Computer Games with Python". CreateSpace Independent Publishing Platform. 2010. 366 p.
13. Luciano Ramalho. "Fluent Python: Clear, Concise, and Effective Programming". O'Reilly Media. 2015. 792 p.
14. Tariq Rashid. "Make Your Own Neural Network". CreateSpace Independent Publishing Platform. 2016. 222 p.

15. Joel Grus. "Ten Essays on FizzBuzz: Meditations on Python, mathematics, science, engineering, and design". CreateSpace Independent Publishing Platform. 2016. 104 p.
16. Dan Bader. "Python Tricks: A Buffet of Awesome Python Features". Dan Bader. 2017. 302 p.
17. Eric Matthes. "Python Flash Cards: Syntax, Concepts, and Examples". No Starch Press. 2019. 101 p.
18. Allen B. Downey. "Think Complexity: Complexity Science and Computational Modeling". O'Reilly Media. 2012. 142 p.
19. Jason Cannon. "Python Programming Tutorial for Beginners". Udemy. 2019. Онлайн-курс.
20. Sweigart, Al. "Beyond the Basic Stuff with Python: Best Practices for Writing Clean Code". No Starch Press. 2020. 328 p.

ДОДАТКИ

Додаток А «Лістинг програмного коду»

```
import os

import tkinter as tk

from tkinter import messagebox

class LectureApp:

    def __init__(self, root):

        self.root = root

        self.root.title("Навчально-методичний комплекс з дисципліни «Технології  
виявлення та попередження кібератак»")

        self.root.geometry("600x450") # Фіксований розмір вікна

        self.root.resizable(False, False) # Заборона зміни розмірів вікна

        self.root.configure(bg="red")

        # Новий колір для кнопок

        button_color = "#333"

        # Frame для лекцій

        frame_lectures = tk.Frame(self.root, bg="red")

        frame_lectures.grid(row=0, column=0, padx=(10, 5), pady=(10, 10), rowspan=3)

        # Додаємо елемент Label для підпису "Лекції"

        label_lectures = tk.Label(frame_lectures, text="Лекції", fg="white", bg="red",
font=("Helvetica", 18))

        label_lectures.grid(row=0, column=0, columnspan=2, pady=(10, 5))

        # Додаємо 16 кнопок для лекцій у дві колонки

        for i in range(1, 17):
```

```

lecture_button = tk.Button(frame_lectures, text=f"Лекція {i}",
                           command=lambda i=i: self.show_lecture(i),
                           font=("Helvetica", 14), bg=button_color, fg="white",
activebackground=button_color)

        column = 0 if i % 2 == 1 else 1
        lecture_button.grid(row=(i + 1) // 2, column=column, pady=5, padx=5)


# Frame для тестів
frame_tests = tk.Frame(self.root, bg="red")
frame_tests.grid(row=0, column=1, padx=(5, 5), pady=(10, 10), rowspan=3)


# Додаємо елемент Label для підпису "Тести"
label_tests = tk.Label(frame_tests, text="Тести", fg="white", bg="red",
font=("Helvetica", 18))
label_tests.grid(row=0, column=0, columnspan=2, pady=(10, 5))


# Додаємо 16 кнопок для тестів у дві колонки
for i in range(1, 17):

    test_button = tk.Button(frame_tests, text=f"Тест {i}",
                           command=lambda i=i: self.show_test(i),
                           font=("Helvetica", 14), bg=button_color, fg="white",
activebackground=button_color)

        column = 0 if i % 2 == 1 else 1
        test_button.grid(row=(i + 1) // 2, column=column, pady=5, padx=5)


# Frame для лабораторних
frame_lab = tk.Frame(self.root, bg="red")
frame_lab.grid(row=0, column=2, padx=(5, 10), pady=(10, 10))

```

```

# Додаємо елемент Label для підпису "Лабораторні"
label_lab = tk.Label(frame_lab, text="Лабораторні", fg="white", bg="red",
font=("Helvetica", 18))

label_lab.grid(row=0, column=0, columnspan=2, pady=(10, 5))


# Додаємо 8 кнопок для лабораторних у одну колонку
for i in range(1, 9):
    lab_button = tk.Button(frame_lab, text=f"Лабораторна {i}",
                           command=lambda i=i: self.show_lab(i),
                           font=("Helvetica", 14), bg=button_color, fg="white",
activebackground=button_color)
    lab_button.grid(row=i, column=0, pady=5, padx=5)


def show_lecture(self, lecture_number):
    presentation_path = os.path.abspath(f"Лекції/Лекція_{lecture_number}.pptx")
    if os.path.exists(presentation_path):
        os.startfile(presentation_path)
    else:
        messagebox.showinfo("Помилка", f"Презентація для Лекції {lecture_number}
не знайдена.")


def show_lab(self, lab_number):
    lab_path = os.path.abspath(f"Лабораторні/Лабораторна_{lab_number}.pdf")
    if os.path.exists(lab_path):
        os.startfile(lab_path)
    else:
        messagebox.showinfo("Помилка", f"Файл лабораторної {lab_number} не
знайдено.")

```

```
def show_test(self, test_number):  
    html_path = os.path.abspath(f"Тести/Тест_{test_number}.html")  
    if os.path.exists(html_path):  
        os.startfile(html_path)  
    else:
```

```
        messagebox.showinfo("Помилка", f"HTML-файл для Тесту {test_number} не  
        знайдено.")
```

```
if __name__ == "__main__":  
    root = tk.Tk()  
    app = LectureApp(root)  
    root.mainloop()
```