

Факультет інформаційних технологій
Кафедра мережевих та інтернет технологій

ЗАТВЕРДЖУЮ

завідувач кафедри

мережевих та інтернет технологій

Ю.В. Кравченко

« _____ » _____ 2020 року

**КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА**

галузі знань 17 «Електроніка та телекомунікації»
за спеціальністю 172 «Телекомунікації та радіотехніка»

на тему:

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АКТИВІЗАЦІЇ
СТУДЕНТІВ ПІД ЧАС ДИСТАНЦІЙНОГО ПРОВЕДЕННЯ
ЛЕКЦІЙНИХ ЗАНЯТЬ**

Виконав: студент групи МІТ -41

Тимошенко Денис Володимирович

(прізвище ім'я по-батькові)

(підпис)

Керівник: доцент кафедри мережевих та інтернет технологій

д.т.н., Миколайчук Р.А.

(прізвище ім'я по-батькові)

(підпис)

Київ 2021

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра мережевих та інтернет технологій

ЗАТВЕРДЖУЮ

завідувач кафедри
мережевих та інтернет технологій

_____ Ю.В. Кравченко

«_____» _____ 2020 року

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ

Здобувачу вищої освіти

Тимошенко Денис Володимирович

(прізвище, ім'я, по батькові)

1. Тема роботи:

«Інформаційна технологія активізації студентів під час дистанційного проведення лекційних
занять»

затверджена на засіданні кафедри МІТ «.» ... 202. р. протокол № ..

2. Термін здачі закінченої роботи

«.» ... 2021 р.

3. Вихідні дані до проекту (роботи)

...

4. Зміст пояснювальної записки (перелік питань, що їх потрібно розробити, обсяг – 35-50 стор.)

5. Перелік графічного матеріалу 8-10 слайдів

Дата видачі завдання

Керівник роботи

ді.т.н., доцент кафедри МІТ Миколайчук Р.А.

(підпис)

(посада, прізвище, ім'я, по батькові)

Завдання прийняв до виконання

(підпис)

(прізвище, ім'я, по батькові)

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Підготовчий		
2	Розділ 1		
3	Розділ 2		
4	Розділ 3		
5	Доповідь та слайди		
6	Пояснювальна записка		

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

Керівник _____
(підпис) (прізвище, ім'я, по батькові)

РЕФЕРАТ

Пояснювальна записка: 46 с., 4 рис., 5 табл., 17 джерел.

Об'єкт дослідження: інформаційна технологія оптимізації процесів дистанційного навчання та активізації студентів під час дистанційного проведення лекційних занять.

Мета роботи (проекту): розробити веб-додаток для тестування студентів під час дистанційного проведення лекційних занять, та програму для аналізу даних запису чату веб-конференції Zoom.

Методи дослідження: системний підхід, методи порівняння, індексний метод, структурний аналіз, кореляційно-регресійний аналіз.

У спеціальній частині дана характеристика сучасного стану технологій інтернет-навчання.

В роботі проведено аналіз існуючих рішень у сфері інтернет-навчання, та їх ефективності.

Запропоновано розробити систему опитування студентів, та оптимізувати проведення дистанційних лекційних занять.

Побудовано логічну та фізичну моделі даних бази даних.

Розроблено веб-додаток для опитування студентів та програму аналізу даних запису чату веб-конференції Zoom.

Практичне значення роботи полягає у наданні викладачам нового інструменту для оптимізації дистанційного навчально процесу.

Результати здійснених у дипломному проекті досліджень можуть бути використані педагогами для покращення результатів успішності студентів.

Галузь використання – навчальні заклади з дистанційною або гібридною формою навчання.

Напрямки подальшого розвитку роботи включають інтеграцію розробленої

інформаційної технології у навчальній процес, аналіз впливу систему на успішність студентів.

Ключові слова:

ІНФОРМАЦІЙНА СИСТЕМА, БАЗА ДАНИХ, СИСТЕМА УПРАВЛІННЯ
БАЗАМИ ДАНИХ, РЕЛЯЦІЙНА БАЗА ДАНИХ, РЕЛЯЦІЙНА СИСТЕМА
УПРАВЛІННЯ БАЗАМИ ДАНИХ, MYSQL, ПРОЕКТУВАННЯ БАЗИ ДАНИХ,
РОЗГОРТАННЯ БАЗИ ДАНИХ, ВЕБ-ДОДАТОК, ФРЕЙМВОРК, PYTHON, FLASK,
ZOOM

ЗМІСТ

ВСТУП	9
РОЗДІЛ I. АНАЛІЗ ЗАДАЧІ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АКТИВІЗАЦІЇ СТУДЕНТІВ ПІД ЧАС ДИСТАНЦІЙНОГО ПРОВЕДЕННЯ ЛЕКЦІЙНИХ ЗАНЯТЬ	9
1.1 Дослідження підходів та інструментів які використовуються під час дистанційного навчання.	9
1.1.1. Відеоконференції як засіб проведення онлайн-лекцій.	10
1.1.2. Застосування інструментів оцінки та опитування	12
1.1.3. Нагальні проблеми у сфері дистанційного навчання	13
1.2 Опис та аналіз предметної області	15
1.3 Вибір технологій та інструментів для реалізації інформаційної технології ...	16
1.3.1 Бази даних	16
1.3.2 Засоби веб розробки	20
1.3.3 Засоби аналізу даних	23
РОЗДІЛ II. ПРОЕКТУВАННЯ ТЕХНОЛОГІЇ АКТИВІЗАЦІЇ СТУДЕНТІВ В ПРЕДМЕТНІЙ ОБЛАСТІ.....	25
2.1 Проектування бази даних.....	25
2.1.1 Вибір СУБД	25
2.1.2 Проектування моделей.....	26
2.2 Проектування інтерфейсу	31
2.2.1 Вибір фреймворку	31
2.2.2 Функції додатку.....	32

	10
2.3 Проектування скрипту для аналізу проведеної лекції.....	32
РОЗДІЛ III. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АКТИВІЗАЦІЇ	
СТУДЕНТІВ.....	34
3.1 Створення бази даних.....	34
3.1.1 Створення таблиць	35
3.2 Розробка програмного додатку	37
3.3 Створення програми аналізу даних чату відео-конференції.....	41
ВИСНОВОК	42
ПЕРЕЛІК ПОСИЛАНЬ	45

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІТ – інформаційні технології.

ІС – інформаційна система.

БД- база даних.

СУБД – система управління базами даних.

РБД – реляційна база даних.

РСУБД – реляційна система управління базами даних.

DML – Data Manipulation Language (мова маніпулювання даними).

DDL – Data Definition Language (мова опису даних).

ООМД – об'єктно-орієнтована модель даних.

ООП – об'єктно-орієнтоване програмування.

ООБД – об'єктно-орієнтована база даних.

ОРСУБД – об'єктно-реляційна СУБД.

ОРБД – об'єктно-реляційна база даних.

СУРБД – системи управління розподіленої базою даних.

ПК – персональний комп'ютер.

ОС – операційна система.

ВСТУП

Пандемія COVID-19 вразила увесь світ. У глобальному масштабі все зупинилося. Проекти затримувались, закривалися дитячі садки, школи та університети. Однак студенти продовжують навчання онлайн. Наразі ця модель є найкращою альтернативою, оскільки відвідування навчальних закладів створює ризик для безпеки учнів.

Наразі нагальним питанням є продовження впровадження онлайн навчання, чи буде надалі освіта залишатися «постпандемічною», та який вплив це буде мати на світовий ринок освіти.

Навіть до COVID-19 вже спостерігався високий ріст впровадження нових освітніх технологій, глобальні інвестиції в «EdTech» досягли 18,66 млрд. доларів США в 2019 році, а загальний ринок онлайн-освіти досягне 350 мільярдів доларів до 2025 року. Спостерігається значне збільшення використання мовних програм, засобів віртуального навчання, інструментів для відеоконференцій або програмне забезпечення для онлайн-навчання.

Чи є навчання в Інтернеті настільки ефективним? Хоча від вчителів вимагається інтегрувати технології в клас, реальність може бути зовсім іншою. Незважаючи на значні ресурси, що виділяються на інтеграцію технологій, багато вчителів стикаються з перебоями, які можуть спричинити пристрої, але, що навіть важливіше, виникли проблеми з педагогічною адаптацією, оскільки традиційні моделі лекцій потребують вдосконалення для використання їх у віддаленому навчальному середовищі. Незалежно від типу використовуваної технології, викладачі повинні адаптувати свою практику та проявляти творчі здібності щоб залучати учнів. Пандемія підкреслила необхідність гнучкості та важливості взаємодії студент-викладач.

Не дивлячись на те, що пройшло вже досить часу, багато вчителів все ще відмовляються адаптуватися до нових умов і в повній мірі використовувати весь спектр доступних засобів для оптимізації навчального процесу. В основному це стосується викладачів у віці, у яких великий досвід у викладанні і їм може бути складно так кардинально змінити свою, вибудовану роками, систему.

Технології не стоять на місці, і розробники додають в свої продукти все більше можливостей, що мало б спростити завдання педагогам, але насправді, тим самим отримують прямо протилежний ефект. Сервіси стають настільки громіздкими, що недосвідчений користувач просто розгублюється.

У поточних умовах все ще залишається потреба в простих в освоєнні продуктів, які будуть вирішувати базові потреби викладачів.

Розробка інформаційної технології покликана спростити адаптацію недосвідчених у використанні інтернет-засобів педагогів, а досвідченим пришвидшити оцінювання активності студентів.

РОЗДІЛ I. АНАЛІЗ ЗАДАЧІ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АКТИВІЗАЦІЇ СТУДЕНТІВ ПІД ЧАС ДИСТАНЦІЙНОГО ПРОВЕДЕННЯ ЛЕКЦІЙНИХ ЗАНЯТЬ

1.1 Дослідження підходів та інструментів які використовуються під час дистанційного навчання.

Освіта перейшла на вищий рівень за допомогою технологій. Це робить навчальний процес простішим та доступнішим. Викладачі можуть викладати з власного місця та взаємодіяти зі студентами, які перебувають у будь-якій частині світу. Особливо під час пандемії або будь-якої іншої природної кризи, коли ніхто не може вийти з дому, онлайн-освіта відіграє життєво важливу роль. Дистанційне навчання має багато переваг, таких як гнучкість, практичність, зручність тощо. Останнім часом інформаційно-комунікаційні технології керують галуззю освіти, легко підключаючи учнів та викладачів та надаючи безліч спеціальних функцій для вчителів для участі в онлайн-класі ефективно. Отже, як викладати в Інтернеті за допомогою технологій? На даний час технології які застосовуються поділяють на наступні категорії:

- Інструменти оцінки та опитування (Quiz Builder, StudyMate, Google Forms);
- Інструменти асинхронної комунікації (Електронна пошта, форуми);
- Цифрові сховища (Google Scholar, ePortfolio, Youtube);
- Інструменти управління та адміністрування (Gradebook, iGoogle);
- Інструменти обміну фотографіями (Flickr, Picasa);
- Подкасти та стримінг (iTunesU, ePodcast);

- Спільні документи (Google Docs);
- Соціальний нетворкінг (Facebook, LinkedIn);
- Інструменти синхронної комунікації (Google Talk, Zoom, Skype);
- Віртуальна реальність (eSimulations);
- Блоги та мікроблоги (Wordpress, Twitter);

Для проведення педагогом онлайн-заняття необхідні базові навички використання інтернету та технічні знання про програмне та апаратне забезпечення. Щоб полегшити навчання в Інтернеті, багато компаній пропонують навчальні рішення зі спеціальними функціями для ефективного проходження онлайн-уроків. Популярніший спосіб навчання в Інтернеті - це відеоконференції або пряма трансляція.

1.1.1. Відеоконференції як засіб проведення онлайн-лекцій.

Відеоконференції - це зручна платформа для викладачів для зв'язку з групами студентів та взаємодії з ними під час проведення лекцій у прямому ефірі. На ринку доступно багато програмного забезпечення та мобільних додатків з такими функціями, як групові відеодзвінки, спільний доступ до екрану та інші необхідні функції, необхідні для навчання. Отже, які найкращі програми для групових відеодзвінків для навчання в онлайні?

1) Zoom.

Zoom став однією з найкращих платформ для проведення конференцій. Саме Zoom користуються такі відомі навчальні заклади, як UC Berkeley, Арізонський університет, Пітсбурзький університет, Принстон, та інші. Також відома як, мабуть,

найбільш зручна платформа, отже має ряд ресурсів для вчителів, які вперше користуються відеоконференціями. Темпи зросту кількості користувачів Zoom зображено на рис. 1.1.

Date	Number of daily meeting participants
December 31, 2019	10 million
March 31, 2020	200 million
April 21, 2020	300 million

Рисунок 1.1 – Щоденна кількість користувачів Zoom (джерело: Business Insider)

2) Microsoft Teams.

Ще однією платформою, яку обрали багато навчальних закладів, таких як Техаський університет Даллас, Університет Денвера, Університет Бейлора та Університет Маямі став Microsoft Teams. Teams - це інструмент для онлайн-співпраці, який не обмежується лише відеодзвінками. Ця платформа пропонує чудовий варіант для спільної роботи, включаючи обмін файлами, чат тощо. Викладачі обирають цю платформу як простий спосіб спілкуватися зі студентами, проводити віртуальні години роботи та проводити конференц-дзвінки переважно в менших групах. Відеоконференції Google Meet для навчальних закладів

3) Google Meet.

Google Meet (раніше Google Hangout) - ще одне програмне забезпечення, яке педагоги впроваджують у школах. Такі школи як Карнегі Меллон, Чепмен, UC Riverside, Бостонський університет та інші, використовують її як частину платформи G-Suite, яка дозволяє студентам мати легкий доступ до відеоконференцій та інші прості у використанні інструменти групової співпраці на одній платформі.

4) Webex.

Наступним варіантом є платформа Webex від Cisco. Webex частіше використовується для корпоративних та ділових відеоконференцій, але він потрапив

до освітнього простору як ще один інструмент дистанційного навчання. Багато вчителів у таких школах, як Університет Джорджії, Університет Пердью, Університет Рутгерса та інші звертаються до Webex задля підвищення рівня безпеки.

1.1.2. Застосування інструментів оцінки та опитування

Педагоги повинні знати, наскільки добре студенти засвоюють матеріал, щоб нікого не залишати позаду. Також треба оцінити кожного студента наприкінці чверті, семестру та поставити остаточну оцінку, яка відображає як їх розуміння, так і рівень їхніх зусиль.

Інструменти онлайн-оцінювання для вчителів є необхідною частиною дистанційного навчання, адже використовуючи ці інструменти є можливість аналізувати рівень успішності засвоєння матеріалу студентами. Якщо в результаті аналізу буде знайдено прогалини в навчанні, тоді є можливість підлаштуватися та скоригувати навчальний план.

Різні інструменти оцінювання, які використовуються вчителем, можуть мати формуючий або підсумковий характер, або для формування поточних занять, або для оцінювання, коли навчальний план буде завершено.

Формативні інструменти оцінювання - це вікторини, завдання та запитання на обговорення, які вчителі використовують для оцінки та керівництва (або формування) навчального процесу своїх учнів.

Інструменти підсумкового оцінювання - це підсумкові есе та тести, які даються в кінці проекту, курсу, семестру, одиниці, програми чи навчального року. Вчителі використовують їх для оцінки навчання студентів.

Найпоширенішими прикладами інструментів оцінки та опитування є наступні:

1) Socrative.

Socrative згідно з онлайн-оглядами має найвищу оцінку серед викладачів та професійних рецензентів. Це інтерактивний цифровий інструмент, який дозволяє проводити вікторини. Вчителі можуть вибирати серед швидких запитань для миттєвого зворотного зв'язку, або об'ємних вікторин. Вікторини оцінюються в режимі реального часу, та дозволяють зберігати їх для повторного використання з іншими групами.

2) Google Forms.

Google Forms набув популярності завдяки своїй швидкості і простоті. Є можливість створювати вікторини з декількома варіантами або з короткими відповідями.

3) Mentimeter.

Mentimeter постачається із попередньо завантаженими навчальними шаблонами для класу, такими як оцінка навичок аудіювання, опитування після лекцій.

4) Kahoot.

Студенти люблять ігровий підхід Kahoot до навчання та оцінювання. Вчителі можуть вибрати з понад 40 мільйонів готових навчальних ігор або створити власні за лічені хвилини.

1.1.3. Нагальні проблеми у сфері дистанційного навчання

Попри існуюче різноманіття доступних технологій, питання оптимізації навчального процесу все ще стоїть гостро.

За даними дослідницького порталу educationdata.org, 42% студентів зазначили, що залишатися мотивованим впродовж онлайн-занять є основною проблемою для

них, причому 37% зазначили, що це незначна проблема. Також, за даними дослідницького центру RAND Corporation педагоги повідомляють про високий рівень прогулів серед студентів та значно менший обсяг виконаної студентами роботи.

У жовтні 2021 року RAND опитав 1082 викладача та 1147 керівників шкіл з американських виховательських організацій, запитуючи про їхній досвід у навчальному році 2020-21.

Результати охоплюють широкий спектр питань, але дані про навчання, зокрема, натякають на довгостроковий вплив пандемії на навчання студентів.

Згідно з повідомленням, заняття йдуть не звичними темпами:

- дві третини викладачів сказали, що більшість їх учнів були менш підготовленими до роботи на рівні класу, ніж у той час минулого року;
- 56 відсотків вчителів сказали, що вони охопили лише половину або менше половини змісту навчальної програми, яку вони мали б досягти до цього часу минулого року, лише приблизно 1 з 5 викладачів сказав, що вони працювали за тим же графіком, що і минулі роки;

Як повідомив «Тиждень освіти», цієї осені прогули були постійною проблемою у всіх формах навчання. Але вчителі в опитуванні RAND, які працювали у аудиторіях, сказали, що їх учні з'являлися більш послідовно:

Вчителі, які працювали за очною формою відвідування сказали, що 91 відсоток їх учнів був присутній щодня, а викладачі з гібридним типом викладання повідомляли, що 85 відсотків їх учнів відвідували щодня. За оцінками, у викладачів з повністю дистанційною формою відвідування було на рівні 84 відсотків.

Рівень виконання завдань також змінювався залежно від навчального середовища. Вчителі з очною формою, сказали, що 82 відсотки учнів виконали свою роботу, чи, принаймні, більшу її частину, а от у викладачів з повністю дистанційною формою статистика була значно гірша, лише 62 відсотки.

1.2 Опис та аналіз предметної області

Зазвичай інтерес людини направлений в якусь одну конкретно сферу суспільного життя. Інформаційна система зберігає інформацію, яка відноситься до конкретної сфери. В ІС інформація про навколишній світ зберігається у вигляді даних.

Предметна область – це частина навколишнього світу, яка досліджується і дані якої відображаються у БД.

Модель предметної області – це існуючі знання про предметну область. Модель описує процеси, які відбуваються в предметній області, а також дані, які представляють предметну область. Від правильного опису предметної області залежить успіх розробки.

Як правило, модель предметної області будується на трьох рівнях:

- зовнішній рівень. На цьому рівні визначають вимоги, що саме потрібно від цієї системи. Визначаються елементи системи – об'єкти;
- концептуальний рівень. На цьому рівні визначають як саме взаємодіють компоненти і об'єкти системи, як функціонує ця система;
- внутрішній рівень. На цьому рівні потрібно вирішити як усі вимоги повинні реалізовуватися, за допомогою яких програмних і технічних засобів.

Першою частиною предметної області є інформаційна система тестування студентів під час проведення лекційних занять. Інформаційна система повинна бути простою в освоєнні недосвідченим у використанні подібних систем викладачам, мати можливість швидко створювати тести або вікторини і зберігати результати їх проходження, та надавати можливість педагогу переглянути відповіді кожного студента. В процесі створення вікторини додається будь-яка кількість питань, а вже після створення у користувача є можливість поділитися посиланням на проходження

створеного тесту. Результати успішності проходження тесту кожного учасника вираховується автоматично, але у користувача який створив цю вікторину або тест є можливість змінити остаточну оцінку.

Другою частиною предметної області є аналізатор активності студентів під час проведення лекційних занять засобами веб-конференцій Zoom. Інформаційна система повинна отримувати дані з файлу, в якому збережений запис усіх повідомлень які були відправлені під час лекції, ідентифікувати лектора, ідентифікувати запитання лектора та відповіді учасників на ці питання. В результаті викладач має отримати упорядковану таблицю з відповідями студентів, та робити висновок про активність участі студента у проведеній лекції.

Головним чином система повинна допомагати педагогам мати уяву про успішність студентів, та наскільки добре вони засвоюють матеріал.

1.3 Вибір технологій та інструментів для реалізації інформаційної технології

1.3.1 Бази даних

Дані - це сукупність окремої одиниці інформації. Ці "дані" використовуються в різних формах тексту, чисельних значень та багатьох інших. Говорячи з точки зору обчислень, дані - це в основному інформація, яка може бути перекладена в певну форму для ефективного переміщення та обробки. Наприклад: ім'я, вік, вага, зріст тощо.

База даних - це організована колекція структурованих даних, що робить її легкодоступною, керованою та оновлюваною. Простими словами, база даних це місце де зберігаються дані. Найкраща аналогія - бібліотека. Бібліотека містить величезну колекцію книг різних жанрів, тут бібліотека - це база даних, а книги - це дані.

Виділяють три основні типи баз даних, а саме: ієрархічні, мережеві та реляційні. Ієрархічна база даних суттєво відрізняється від решти: для довільного зв'язку один елемент вважають головним (його називають безпосереднім предком, батьком), інший – підлеглим (його називають безпосереднім нащадком, сином). Об'єкт може не мати жодного нащадка або мати їх кілька, проте, має не більше одного предка. Близнюками називають об'єкти зі спільним предком.

Відмінність мережевої моделі від ієрархічної полягає у відсутності підпорядкування. У мережевій моделі елементи різних рівнів вільно зв'язані. Перевагою мережевої моделі є найповніше відображення структури деяких задач, наприклад, мережевого планування в економіці.

Реляційна модель побудована на взаємовідношенні частин, що її складають. У найпростішому випадку реляційна модель є двовимірним масивом (таблицею), а при створенні складних інформаційних моделей – сукупністю взаємопов'язаних таблиць, кожна з яких містить набір однотипних об'єктів. Кожен рядок такої таблиці називають записом, кожний стовпчик – полем. Таблиці реляційної моделі даних, як правило, зв'язані між собою ключами. Ключ (ключове поле) – це поле таблиці, яке однозначно визначає запис. Значення ключа не може повторюватись у таблиці. Ключі у базах даних відіграють вкрай важливу роль, адже саме за їх допомогою відбувається ідентифікація об'єктів. Переваги реляційної моделі:

- простота і доступність для розуміння користувачем. Єдиною використовуваною інформаційною конструкцією є «таблиця»;
- суворі правила проектування, які базуються на математичному апараті;

- повна незалежність даних. Зміни в прикладній програмі при зміні реляційної БД мінімальні;
- для організації запитів і написання прикладного ПЗ немає необхідності знати конкретну організацію БД у зовнішній пам'яті.

Для створення баз даних і маніпулювання ними використовують спеціальне програмне забезпечення – системи управління базами даних (СУБД).

Сьогодні відомо більше двох десятків серверних систем управління базами даних. Найпопулярнішими є Oracle, MySQL, PostgreSQL. Проведемо критичний аналіз перелічених СУБД.

Oracle – об'єктно-реляційна система керування базами даних від Oracle Corporation.

До переваг даної СУБД належать:

- централізована система управління та контролю;
- стандартизація та послідовність декількох різних реалізацій.

Oracle має також і свої недоліки:

- неможливість реалізації рекурсивної обробки;
- обмежена функціональність;
- висока вартість ліцензії. Oracle Database 11g Standard Edition – орієнтовно 6000 доларів США.

MySQL – це найпоширеніша повноцінна серверна СУБД. MySQL є дуже функціональною та вільнорозповсюджуваною, успішно працює з різними сайтами і веб додатками. Перевагами MySQL є:

- простота в роботі, легке встановлення;
- багатий функціонал – MySQL підтримує більшість функціоналу SQL;
- безпека – велика кількість функцій забезпечують безпеку, яка підтримується за замовчуванням;
- масштабованість – MySQL легко працює з великими обсягами даних і легко масштабується;

- швидкість – спрощення деяких стандартів дозволяє MySQL значно збільшити продуктивність.

До недоліків даної СУБД належать:

- багатопоточність – паралельні операції читання-запису можуть спричиняти проблеми у роботі СУБД;
- недолік функціоналу – деякі движки MySQL, наприклад, не підтримують повнотекстовий пошук;
- повільна розробка.

PostgreSQL є найбільш професійною з усіх трьох розглянутих СУБД. Вона вільно розповсюджується і максимально відповідає стандартам SQL. PostgreSQL або Postgres намагаються повною мірою використовувати ANSI / ISO SQL стандарти своєчасно з виходом нових версій.

Переваги PostgreSQL:

- відкрите ПЗ відповідає стандарту SQL PostgreSQL – безкоштовне ПЗ з відкритим вихідним кодом;
- велике співтовариство – існує велика спільнота в якому ви запросто знайдете відповіді на свої питання;
- велика кількість доповнень - існує дуже багато доповнень, що дозволяють розробляти дані для цієї СУБД і управляти ними;
- розширення – існує можливість розширення функціоналу за рахунок збереження своїх процедур;
- об'єктна орієнтованість – PostgreSQL це не тільки реляційна СУБД, але також і об'єктно-орієнтована з підтримкою успадкування і багато іншого.

Недоліки PostgreSQL:

- можливе уповільнення швидкості роботи СУБД;
- складний процес налаштування перед початком роботи із СУБД.

1.3.2 Засоби веб розробки

Веб-фреймворк - це програмна платформа для розробки веб-додатків та веб-сайтів. Зазвичай має колекцію бібліотек та модулів, які підтримують процес розробки. Надають можливість створювати налаштовуванні веб-програми, змінюючи існуючі компоненти. Фреймворки веб-розробки можуть бути використані для розробки веб-служб, веб-API (Інтерфейс програмування додатків) та інших веб-ресурсів.

Фреймворки пропонують широкий спектр заздалегідь написаних компонентів, фрагментів коду або цілих шаблонів програм. Дозволяють змінювати ці функції за допомогою додаткового коду, що дозволяє створювати спеціальне програмне забезпечення, розроблене для задоволення конкретних потреб.

Це дозволяє повністю зосередитися на додатку, в той час як фреймворк ефективно дбає про всі фонові деталі, такі як прив'язка даних та конфігурація, тому за допомогою існуючих допоміжних функцій розробники можуть заощадити багато часу. Багато фреймворків також мають стандартизовані практики кодування та конвенції для структур коду, які систематизують процес програмування та дозволяють уникати помилок.

Як правило, існує два типи фреймворків - на стороні клієнта та на стороні сервера. Фреймворки на стороні клієнта використовуються для роботи з інтерфейсом користувача, фреймворк на стороні сервера працює у фоновому режимі, щоб забезпечити безперебійне функціонування веб-сайту.

Веб-сайти використовують бази даних для зберігання інформації. Веб-фреймворки часто забезпечують рівень бази даних, який абстрагує операції читання,

запису, запиту та видалення бази даних. Цей шар абстракції називається об'єктно-реляційним відображенням (ORM). Використання ORM має дві переваги:

- Дають можливість замінити базу даних без необхідності змінювати код, який її використовує. Це дозволяє розробникам оптимізувати характеристики різних баз даних на основі їх використання;
- Валідація даних може бути реалізована в рамках фреймворку. Це робить простішою та безпечнішою перевірку даних, чи зберігаються вони у правильному типі, полі бази даних, мають правильний формат (наприклад, адресу електронної пошти) і жодним чином не є шкідливими (зловмисники можуть використовувати певні шаблони коду, щоб причинити шкоду, таку як, наприклад, видалення записів бази даних).

Веб-фреймворк Django, наприклад, працює з ORM і посилається на об'єкт, що використовується для визначення структури запису як моделі. Модель визначає типи полів, які слід зберігати, що може забезпечити перевірку рівня поля щодо того, яку інформацію можна зберігати (наприклад, поле електронної пошти дозволить лише дійсні адреси електронної пошти). Визначення полів можуть також визначати їх максимальний розмір, значення за замовчуванням, параметри списку вибору, довідковий текст для документації, текст міток для форм тощо. Модель не містить жодної інформації про основну базу даних, оскільки це параметр конфігурації, який можна змінити окремо від нашого коду. Проведемо аналіз існуючих веб-фреймворків:

1. Django

Django - це високоякісний веб-фреймворк Python, який стимулює швидкий розвиток та чистий, прагматичний дизайн. Побудований досвідченими розробниками, опікується великою частиною клопоту при веб-розробці, тому дозволяє зосередитись на написанні програми. Безкоштовний та з відкритим кодом.

Django дотримується філософії "Батарей в комплекті" і надає майже все, що може знадобитися більшості розробників. Оскільки все включено, все це працює

разом, дотримується послідовних принципів проектування та має велику та сучасну документацію. Це також швидко, безпечно і дуже добре масштабується. Популярні сайти, що використовують Django: Instagram, Mozilla, National Geographic.

2. Flask

Flask - це мікрофреймворк для Python. Незважаючи на свій мінімалізм, Flask може створювати серйозні веб-сайти «прямо з коробки». Він містить сервер розробки та налагоджувач, а також включає підтримку шаблонування Jinja2, безпечні файли cookie, модульне тестування та відправлення запитів RESTful. Він має гарну документацію, активну спільноту та велику кількість доступних бібліотек. Flask став надзвичайно популярним, особливо для розробників, яким потрібно надавати веб-сервіси в невеликих системах з обмеженими ресурсами (наприклад, запуск веб-сервера на Raspberry Pi, контролерах Drone тощо)

3. Express

Express - це швидкий, непроникний, гнучкий та мінімалістичний веб-фреймворк для Node.js (node js- це середовище без браузера для запуску JavaScript). Він надає надійний набір функцій для веб і мобільних додатків, а також забезпечує корисні методи утиліти HTTP та проміжне програмне забезпечення.

Express надзвичайно популярний, частково тому, що полегшує міграцію веб-програмістів JavaScript на стороні клієнта у розробку на стороні сервера, а частково тому, що він ефективно використовує ресурси (базове середовище вузла використовує полегшену багатозадачність у потоці, а не породжує окремі процеси для кожного нового запиту). Оскільки Express - це мінімалістичний веб-фреймворк, він не включає всі компоненти, які можуть знадобитися (наприклад, доступ до бази даних та підтримка користувачів та сеансів надаються через незалежні бібліотеки). Є багато чудових незалежних компонентів, але іноді буває важко розібратися, який найкращий для певної мети. Express використовують такі компанії, як Uber, Accenture, IBM тощо.

4. Rails

Rails (зазвичай його називають "Ruby on Rails") - це веб-фреймворк, написаний для мови програмування Ruby. Має дуже подібну до Django філософію дизайну. Як і Django, він забезпечує стандартні механізми для маршрутизації URL-адрес, доступу до даних з бази даних, генерації HTML із шаблонів та форматування даних як JSON або XML. Це також заохочує використання патернів дизайну, таких як DRY ("не повторюйся" - напиши код лише один раз, якщо це можливо), MVC (model-view-controller) та ряд інших. Rails використовують для гучних сайтів, серед яких: GitHub, Shopify, Airbnb, Twitch та інші.

5. ASP.NET

ASP.NET - це веб-фреймворк з відкритим кодом, розроблений корпорацією Майкрософт для створення сучасних веб-додатків та служб. За допомогою ASP.NET можна швидко створювати веб-сайти на основі HTML, CSS та JavaScript, масштабувати їх для використання мільйонами користувачів і легко додавати більш складні можливості, такі як веб-API або спілкування в реальному часі.

Одним із особливостей ASP.NET є те, що він побудований на Common Language Runtime (CLR), що дозволяє програмістам писати код ASP.NET, використовуючи будь-яку підтримувану мову .NET (C #, Visual Basic та ін.). Як і багато інших продуктів Microsoft, він користується чудовими інструментами, активною спільнотою розробників та добре написаною документацією. ASP.NET використовується Microsoft, Xbox.com, Stack Overflow та багатьма іншими.

1.3.3 Засоби аналізу даних

Для створення програми аналізу буде використано стандартні засоби мови Python, та розширення «csv», яке дозволить зберігати результати аналізу у таблиці формату CSV.

РОЗДІЛ II. ПРОЕКТУВАННЯ ТЕХНОЛОГІЇ АКТИВІЗАЦІЇ СТУДЕНТІВ В ПРЕДМЕТНІЙ ОБЛАСТІ

2.1 Проектування бази даних

2.1.1 Вибір СУБД

Серед інших СУБД вибрана MySQL. MySQL — вільна система керування реляційними базами даних. Як зазначено у [4]: «MySQL є рішенням для малих і середніх додатків. Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Більш того, СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць». MySQL працює на Linux, Windows, OSX, FreeBSD і Solaris. Можна почати працювати з безкоштовним сервером, а потім перейти на комерційну версію. Ліцензія GPL з відкритим вихідним кодом дозволяє модифікувати ПЗ MySQL.

MySQL Workbench - це уніфікований візуальний інструмент для архітекторів баз даних та розробників. Як зазначено у [5]: «MySQL Workbench забезпечує моделювання даних, розробку SQL та комплексні інструменти адміністрування для конфігурації сервера, адміністрування користувачів, резервного копіювання та багато іншого. MySQL Workbench доступний у Windows, Linux та Mac OS X. MySQL

Workbench дозволяє адміністратору баз даних, розробнику або архітектору даних візуально проектувати, моделювати, генерувати та керувати базами даних. Він включає все, що потрібно модельатору даних для створення складних моделей ER, прямого та зворотного проектування, а також забезпечує ключові функції для виконання складних завдань управління змінами та документування, які зазвичай вимагають багато часу та зусиль».

2.1.2 Проектування моделей

ER-модель (entity-relationship model), так звана модель «сутність–зв’язок». Як зазначено у [6]: «Це модель даних високого рівня, яка визначає елементи даних та їх взаємозв’язок для певної програмної системи. Модель ER використовується для представлення реальних об’єктів.

Сутність - це річ або об’єкт у реальному світі, який можна відрізнити від навколишнього середовища. Наприклад, кожен працівник організації є окремим суб’єктом». Нижче наведено деякі основні характеристики суб’єктів:

- суб’єкт має набір властивостей;
- властивості сутності можуть набувати значення.

Під час проектування бази даних: ER-модель перетворюють в конкретну схему бази даних на основі моделі даних (реляційна, об’єктно-орієнтована і т.д.).

Інформаційна система буде орієнтована на педагогів та тести.

Етапи створення концептуальної моделі:

1. Визначення сутностей.
2. Визначення атрибутів.
3. Визначенні унікальних ідентифікаторів.

4. Визначення відношень та їх властивостей.

Базові сутності предметної області:

1. Педагог.
2. Тест.
3. Запитання.
4. Спроби.
5. Відповіді.

Атрибути сутностей:

1. Педагог.
 - ID
 - Ім'я
 - Електронна пошта
 - Пароль
2. Тест
 - ID
 - Педагог
 - Веб-адреса
 - Назва
 - Дата
 - Статус
3. Запитання
 - ID
 - Тест
 - Текст запитання
 - Правильна відповідь
4. Спроба
 - ID

- Тест
- Ім'я студента
- Дата
- 5. Відповіді
 - ID
 - Спроба
 - Питання
 - Відповідь
 - Правильність відповіді

На рисунку 2.1 представлена ER діаграма для даної предметної області.

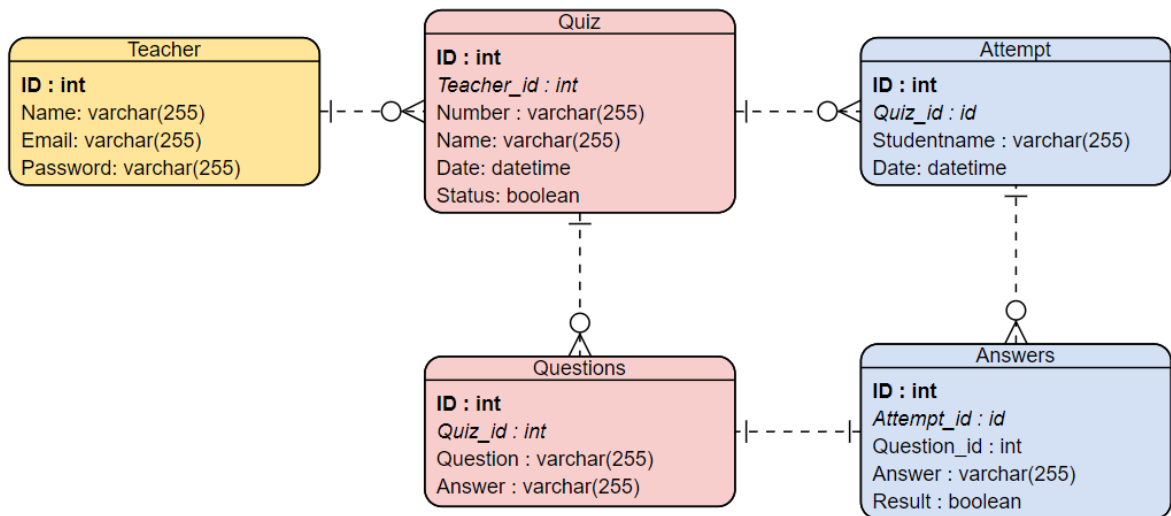


Рисунок 2.1 – ER діаграма інформаційної системи опитування студентів

Логічна модель розширяє інфологічну завдяки визначенню для кожної сутності атрибутів, описує та уточнює склад сутностей та відношення між ними (рис. 2.2).

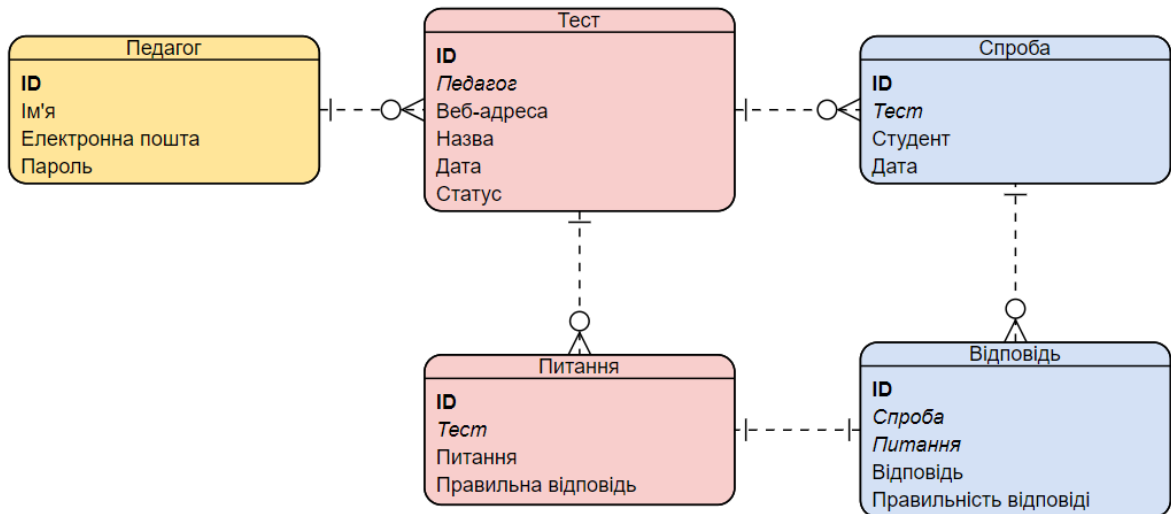


Рисунок 2.2 – Логічна модель

Фізична модель даних описує реалізацію об'єктів логічної моделі на рівні об'єктів конкретної бази даних у вибраній СУБД. У логічній моделі не має значення, який тип даних має атрибут. А у фізичній, навпаки, важливо визначити всю інформацію про конкретні фізичні об'єкти – таблиці, колонки, індекси і т.д. Нижче буде приведено фізичну схему таблиць БД.

Таблиця 2.1 – Фізична схема таблиці «Teacher/Педагог»

Атрибут	Властивість
ID	Int, AI, PK
Name	Varchar(255)
Email	Varchar(255), UNIQUE
Password	Varchar(255)

Таблиця 2.2 – Фізична схема таблиці «Quiz/Тест»

Атрибут	Властивість
ID	Int, AI, PK
Teacher_ID	Int, FK

Number	Varchar(10), UNIQUE
Name	Varchar(255)
Date	Datetime
Status	Boolean

Таблиця 2.3 – Фізична схема таблиці «Question/Питання»

Атрибут	Властивість
ID	Int, AI, PK
Quiz_ID	Int, FK
Quest	Varchar(255)
Answer	Varchar(255)

Таблиця 2.4 – Фізична схема таблиці «Attempt/Спроба»

Атрибут	Властивість
ID	Int, AI, PK
Quiz_ID	Int, FK
Studentname	Varchar(255)
Date	Datetime

Таблиця 2.5 – Фізична схема таблиці «Answer/Відповідь»

Атрибут	Властивість
ID	Int, AI, PK
Attempt_ID	Int, FK
Question_ID	Int, FK
Answ	Varchar(255)
Result	Boolean

При переході від логічної моделі до фізичної були розкриті зв'язки типу «один-до-багатьох» між відношеннями «Педагог» і «Тест», «Тест» і «Спроба», «Тест» і «Питання», «Спроба» та «Відповідь».

2.2 Проектування інтерфейсу

2.2.1 Вибір фреймворку

В даний час доступна велика кількість фреймворків веб-розробки, і цей різноманітний перелік іноді бентежить навіть найдосвідченіших розробників. Майбутній веб-сайт і спосіб його роботи та зовнішнього вигляду значною мірою залежить від того, що саме обрано в якості основи.

Flask є одним з найпопулярніших фреймворків, що використовується професійними розробниками. Це один з класичних веб-фреймворків для роботи з веб-додатками на різних пристроях і має технічне обслуговування на базі Unicode та чудову документацію. Фреймворк використовують такі відомі сервіси, як Pinterest, LinkedIn та їх власна веб-сторінка спільноти. У Flask немає рівня абстракції бази даних, але він підтримує розширення для додавання функцій програми, як якщо б вони були виконані в Flask. Існують розширення, доступні для об'єктно-реляційних баз даних, і вони оновлюються набагато частіше порівняно з основною програмою Flask. Завдяки тому, що фреймворк розроблений на базі Python, використовуючи бібліотеки SQLAlchemy, Flask-SQLAlchemy чи інші, можна працювати з даними як з об'єктами, а це значно процес обробки та аналізу даних.

2.2.2 Функції додатку

Створений веб-додаток має надавати можливість створювати тести чи вікторини з довільною кількістю питань, та поширювати посилання на створений тест. Змінювати питання, видаляти питання. Активувати чи вимикати обраний тест. Переглядати відповіді студентів та зараховувати, чи не зараховувати відповіді. Переглядати створені тести та дізнаватись оцінки.

Тому веб-додаток буде мати такі сторінки:

- Сторінка авторизації;
- Сторінка перегляду створених тестів;
- Сторінка створення нового тесту;
- Сторінка з можливістю зручно поширити посилання на тест;
- Сторінка додання питання;
- Сторінка перегляду питань у створеному тесті;
- Сторінка, що відображає результати проходження тесту студентами;
- Сторінка перегляду відповідей студентів;
- Сторінка проходження тесту для студентів;

2.3 Проектування скрипту для аналізу проведеної лекції

Створений скрипт має отримувати назву файлу в якому збережені дані чату проведеної у Zoom веб-конференції, та назву файлу у який буде сформований результат аналізу. Програма має ідентифікувати лектора, знайти питання, які задавав

лектор, та відповіді студентів на ці питання. В результаті потрібно отримати таблицю з питаннями та відповідями.

РОЗДІЛ III. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АКТИВІЗАЦІЇ СТУДЕНТІВ

3.1 Створення бази даних

У MySQL існує декілька способів створення бази даних.

1. Створення через командну строку. Для створення бази даних через команду строку потрібно зробити наступне:

- Ввійти у систему MySQL як root користувач;
- Створити нового користувача командою: «GRANT ALL PRIVILEGES»;
- Ввійти у систему MySQL зі створеного аккаунту;
- Створити базу даних командою: «CREATE DATABASE db_name»;

2. Створити базу даних засобами MySQL Workbench. Для створення бази даних треба зробити наступне:

- Створити нове з'єднання або увійти як root користувач (рис. 3.1);
- Натиснути кнопку «Create new schema»;
- Ввести назву бази даних та, якщо потрібно, вибрати тип кодування, і натиснути кнопку “Apply”.

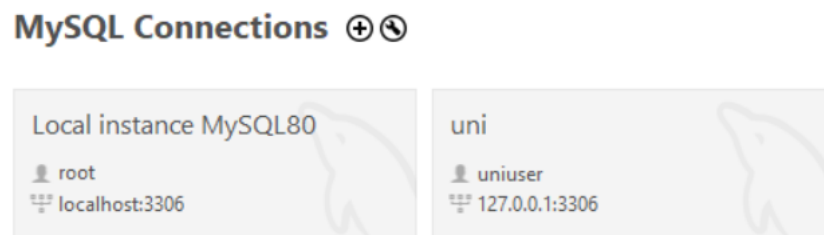


Рисунок 3.1 – З'єднання з MySQL засобами MySQL Workbench

3. Створити базу даних використовуючи файл із SQL скриптом. Для цього потрібно:

- Створити текстовий файл з розширенням «.sql»;
- Ввести команду «CREATE DATABASE dbname»;
- Через командний рядок операційної системи ввести команду «mysql -u username -p < example.sql»

3.1.1 Створення таблиць

Процес створення таблиць за допомогою мови SQL включає наступні дії:

1. Задання ім'я таблиці.
2. Перелік стовпчиків із зазначенням типів даних атрибутів.
3. Обмеження для кожного стовпчика:
 - приналежність первинному ключу;
 - допустимість невизначених значень;
 - деякі обмеження на значення стовпчика;
 - задання значення за замовчуванням – те, що буде введено, якщо в операторі введення відсутнє конкретне значення для даного поля;
 - приналежність для зовнішнього ключа.

Враховуючи попередні умови, створимо таблиці для бази даних інформаційної системи. Нижче наведений приклад створення однієї таблиці.

MySQL Workbench пришвидшує створення таблиць, адже дозволяє надавати функції первинного або зовнішнього ключа, інкримінування та інших одним або двома натисканнями у меню. Створення таблиці засобами MySQL Workbench (рис. 3.2).

Table Name:

Charset/Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
attempt_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
question_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
answer	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
result	TINYINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Рисунок 3.2 – Створення таблиці засобами MySQL Workbench

Також існує можливість створення таблиць через командний рядок або SQL файл. Приклад коду для створення таблиці:

```
CREATE TABLE `attempt` (
  `id` int NOT NULL AUTO_INCREMENT,
  `quiz_id` int NOT NULL,
  `name` varchar(255) NOT NULL,
  `date` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  KEY `fk_quiz_id_idx` (`quiz_id`),
  CONSTRAINT `fk_quiz_id` FOREIGN KEY (`quiz_id`) REFERENCES `quiz` (`id`)
```

Після створення таблиць слід з'єднати їх, задавши зв'язок ключами. Також створимо індекси для ключів та полів таблиць, до яких найчастіше будуть відбуватися звернення запитів

3.2 Розробка програмного додатку

Створення веб-додатку у фреймворку Flask розпочинається із налаштування файлу ініціалізації «__init__.py». Тут буде підключено бібліотеки, а саме:

- SQLAlchemy буде відповідати за взаємодію додатку з базою даних та дасть можливість працювати с даними як з Python об'єктами;
- Wcrypt буде використано для захисту паролів. Wcrypt шифрує дані типом «one-way encryption», що не передбачає розшифрування отриманих даних. При вводі паролю Wcrypt шифрує отримані дані та порівнює їх з записом у базі. Тип шифрування «Blowfish cipher» який використовується у Wcrypt дозволяє отримати дуже високий рівень захисту;
- LoginManager забезпечує управління сеансами користувача. Він обробляє загальні завдання входу, виходу та запам'ятовування сеансів користувачів протягом тривалого періоду часу. Зберігає ідентифікатор активного користувача у сеансі та дозволяє легко входити та виходити з системи, обробляє функціональність "запам'ятай мене", допомагає захистити сеанси користувачів від викрадення cookie.

Дані для підключення до бази даних, та ключ який буде відповідати за безпечне збереження сеансів користувачів винесено у окремий файл, що покращить рівень безпеки веб-додатку.

Для роботи SQLAlchemy створено файл «models.py». Тут будуть створені класи, які відповідають таблицям бази даних, та відношення «один-до-одного», «один-до-багатьох». Приклад створення класу, який відповідає таблиці «Quiz» у бази даних:

```
class Quiz(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    user_id = db.Column(db.Integer, db.ForeignKey('users.id'))
```

```

number = db.Column(db.VARCHAR(10))
name = db.Column(db.VARCHAR(200), nullable=False)
date = db.Column(db.DateTime, default=datetime.utcnow)
status = db.Column(db.Integer)
quests = db.relationship('Questions', backref='quiz', lazy='dynamic',
cascade="all, delete, delete-orphan")
attempts = db.relationship('Attempt', backref='quiz', lazy='dynamic',
cascade="all, delete, delete-orphan")

```

Для обробки форм знадобиться бібліотека WTForms. Це гнучка бібліотека для перевірки та рендеру форм у фреймворку Flask. Може працювати з будь-якою веб-структурою та механізмом шаблонів. Підтримує перевірку даних, захист CSRF, інтернаціоналізацію (I18N) тощо.

У веб-додатку будуть наступні форми:

- Форма авторизації;
- Форма реєстрації;
- Форма для додання до тесту питання;
- Форма для відповідей на питання.

Усі потрібні форми винесено у окремий файл «Forms.py». Приклад форми тестування:

```

class CheckQuestForm(FlaskForm):
    i = IntegerField('int')
    question = StringField('Question', validators=[DataRequired(), Length(max=199)])
    true_answer = StringField('True Answer', validators=[DataRequired(),
Length(max=99)])
    answer = StringField('Answer', validators=[DataRequired(), Length(max=99)])
    tr = RadioField('True', choices=[(True, 'Correct'), (False, 'Wrong')])

```



```
class Meta:
    # No need for csrf token in this child form
    csrf = False
```

```
class CheckQuestionForm(FlaskForm):
    input = FieldList(FormField(CheckQuestForm), min_entries=1)
    submit = SubmitField('Submit')
```

Основним файлом, файлом маршрутів та обробки даних буде «routes.py». Сюди підключаються усі, створені попередньо, класи, та описане реагування на отримані HTTP запити. Для прикладу розглянемо частину коду, яка відповідає за сторінку, на яку, за отриманим посиланням, будуть переходити студенти, або, якщо на сторінку переходить користувач який створив тест, буде відображено поточний стан тесту:

```
@app.route("/quiz/<string:l>", methods=['GET', 'POST'])
def quiz(l):
    q = Quiz.query.filter_by(number=l).first_or_404()
    if current_user.is_authenticated:
        if q.user_id == current_user.id:
            i = Attempt.query.filter_by(quiz_id=q.id).count()
            return render_template('userquiz.html', q=q, i=i, l=l)
    x = q.quests.all()
    if q.status != 1 or not x:
        return redirect(url_for('home'))
    form = StudentNameForm()
    if form.validate_on_submit():
```

```

return redirect(url_for('quizattempt', l=l, n=form.sname.data))
return render_template('quiz.html', form=form, q=q)

```

Маршрутизація здійснюється доданням декоратору:

«@app.route("/quiz/<string:l>", methods=['GET', 'POST'])».

“/{path}” – шлях, за яким буде запускатися наступна функція. В цьому випадку «string:l» відповідає за номер тесту.

«methods=['GET', 'POST']» - методи доступні даному маршруту.

«def quiz(l)» - функція.

«q = Quiz.query.filter_by(number=l).first_or_404()» - шукаємо тест із отриманим номером. Якщо такого нема в системі, повертаємо помилку «404».

«if current_user.is_authenticated:» - перевіряє чи ввійшов користувач у систему.

«x = q.quests.all()» знаходить усі питання які належать даному тесту.

«if q.status != 1 or not x:» - перевіряє статус тесту.

«return redirect(url_for('home'))» - переадресація, де «home» назва методу.

«form = StudentNameForm()» створює екземпляр класу форми.

«if form.validate_on_submit():» - «якщо натиснуте поле «Submit» у формі (метод «POST»), то: ».

«return render_template('quiz.html', form=form, q=q)» - повернення даних:

- «render_template()» – функція приймає ім'я файлу html, який буде доданий до шаблону «layout.html» и змінні, які будуть передаватися а потім оброблюватися. Функція «render_template()» визиває механізм шаблонів «Jinja2». Jinja2 замінює блоки {{...}} у html файлах відповідними значеннями, вказаними у «render_template()»;
- «form=form» – передання аргументу «form» зі значенням «form» заданим вище у коді.

3.3 Створення програми аналізу даних чату відео-конференції

Відкрити отриманий файл з даними дозволяє Python-модуль «codecs». Відкривши файл, зберігаємо отримані дані.

Наступний крок це пошук лектора. Знаючи, що приватні повідомлення зберігаються тільки в тому випадку, якщо вони відправлені учаснику, або учасником, який створив веб-конференцію, приймаємо цього учасника як лектора. Пошук питань, заданих лектором, здійснюється перебором повідомлень лектора, та ті повідомлення, які мають у собі знак питання, враховуються як питання. Після цього, всі отримані лектором приватні повідомлення рахуються відповідями на останнє, до цього повідомлення, питання. Зустрівши повідомлення лектора, яке складається тільки зі слова «stop», питання закривається, і всі, отримані після цього, приватні повідомлення не враховуються.

Формування таблиці з результатами здійснюється за допомоги модуля «csv».

РОЗДІЛ IV. ВИКОРИСТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ТА ОЦІНКА ЇЇ ЕФЕКТИВНОСТІ

4.1 Використання інформаційної системи

При всій різноманітності доступних інструментів взаємодії між викладачем і студентом вони все ще не досконалі. Використовуючи засоби Zoom у педагога немає гарантії, що студенти дійсно присутні на лекції і уважно її слухають. Впроваджуючи розроблену систему з'являється можливість перевіряти присутність студента і стимулювати його активно брати участь в лекції. За бажанням викладача, можна заохочувати активних студентів додатковими балами.

Пропонується наступний алгоритм використання інформаційної технології: в процесі проведення лекції задавати студентам питання по темі, отримані відповіді буде проаналізовано розробленою програмою, в кінці лекції у розробленому додатку провести тестування з вивченого матеріалу. В результаті буде отримано інформацію про активність студентів, а також на скільки добре ними був засвоєний пройдений матеріал. Ці дані допоможуть у подальшій оптимізації навчального плану, а також будуть корисні під час виставлення оцінок після проходження курсу.

4.2 Оцінка ефективності інформаційної системи

Для оцінки ефективності інформаційної системи її було частково впроваджено на кафедрі.

Таблиця 4.1 – Результати впровадження ІС на кафедрі

	Без використання ІС	З використанням ІС
Всього лекцій	30	20
Всього повідомлень у чаті	210	360
Відповіді	-	305
У середньому	7	18

В результаті впровадження ІС середня кількість повідомлень у веб-конференції збільшилася у 3 рази, та більшість цих повідомлень були відповідями на запитання, тобто стосувалися теми лекцій.

ВИСНОВОК

У даному проекті було проаналізовано предметну область і створено письмовий опис об'єктів предметної області. Проведено аналіз структури і функцій інформаційної технології активізації студентів під час дистанційного проведення лекційних занять, та взаємодії між педагогом та студентом.

Був проведений аналіз наукових та практичних підходів до вдосконалення та оптимізації процесу дистанційного навчання. В результаті аналізу зроблений висновок, що попри існуюче різноманіття доступних технологій, питання оптимізації навчального процесу все ще стоїть гостро. А отже, прийнято рішення спробувати вирішити це питання створенням нового, простого в освоєнні сервісу, який покликаний спростити адаптацію недосвідчених у використанні інтернет-засобів педагогів, або швидко вирішувати базові потреби, вже досвідчених у використанні веб-сервісів, викладачів.

На основі аналізу предметної області було створено ERD-модель, визначено базові сутності та атрибути сутностей предметної області. Між сутностями визначені і встановлені типи відносин. Створені логічна та фізична моделі.

При проектуванні фізичної моделі також була створена перевірка відношень за допомогою правил нормалізації. Визначено первинний ключ і зовнішній ключ для кожної сутності.

Виконано проектування та створення бази даних на основі СУБД MySQL, на основі Python та фреймворку Flask створено веб-додаток для тестування студентів, та програма аналізу даних чату відео-конференції Zoom.

Запропоновано алгоритм використання створеної інформаційної системи, та проаналізовано її ефективність.

ПЕРЕЛІК ПОСИЛАНЬ

1. Education during COVID-19; moving towards e-learning. [Електронний ресурс]. – Режим доступу: <https://data.europa.eu/en/impact-studies/covid-19/education-during-covid-19-moving-towards-e-learning>
2. UNESCO. COVID-19 Educational Disruption and Response. [Електронний ресурс]. – Режим доступу: <https://en.unesco.org/covid19/educationresponse>
3. Dennis, M. (2019). Higher education opportunities after COVID-19. [Електронний ресурс]. – Режим доступу: <https://www.universityworldnews.com/post.php?story=20200507152524762>
4. Іванов К. Є. Розробка програмних та апаратних засобів системи управління контентом факультетського рівня [Електронний ресурс] / Іванов Костянтин Євгенійович – Режим доступу: <http://masters.donntu.org/2009/fvti/ivanovk/diss/indexu.htm>
5. Демієн А. MySQL Workbench, візуальний інструмент для проектування баз даних. [Електронний ресурс]. – Режим доступу: <https://ubunlog.com/uk/%D0%B1%D0%B0%D0%B7%D0%B8-%D0%B4%D0%B0%D0%BD%D0%B8%D1%85-%D1%80%D0%BE%D0%B1%D0%BE%D1%87%D0%BE%D0%B3%D0%BE-%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%BE%D0%B2%D0%B8%D1%89%D0%B0-mysql/>
6. Що таке ER Modeling? Дізнайтеся на прикладі. [Електронний ресурс]. – Режим доступу: <https://uk.csstricks.net/8224980-what-is-er-modeling-learn-with-example>
7. Карпова И.П. Базы данных. Учебное пособие для вузов. – СПб: «Питер», 2013. – 240 с.

8. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика = Database Systems: A Practical Approach to Design, Implementation, and Management. — 3-е изд. — М.: Вильямс, 2003. — 1436 с.
9. Бураков П.В., Петров В.Ю. Введение в системы баз данных: учебное пособие. — СПб, 2010. — 129 с.
10. Классификация СУБД. Типы и виды СУБД. [Электронный ресурс]. — Режим доступа: <https://www.sqlhome.org.ua/klassifikaciya-tipy-vidy-subd/>
11. Карепова Е.Д. Реляционная модель данных. — Красноярск, 2004 — 45 с.
12. Базы данных. [Электронный ресурс]. — Режим доступа: <https://siblec.ru/informatika-i-vychislitel'naya-tehnika/bazy-dannykh#3>
13. Хомоненко А.Д. Цыганков В.М. Мальцев М.Г. Базы данных. Учебник для высших учебных заведений. — Санкт-Петербург: КОРОНА-Век, 2009 — 736 с.
14. Карпова И.П. Базы данных. Учебное пособие. — М., 2009. — 131 с.
15. Избачков Ю. С. Информационные системы: учебник : — 2-е изд. — СПб: Питер, 2008. — 656 с.
16. Что такое NoSQL? [Электронный ресурс]. — Режим доступа: <https://aws.amazon.com/ru/nosql/>
17. Основы проектирования реляционных баз данных. [Электронный ресурс]. — Режим доступа: https://www.intuit.ru/studies/professional_retraining/953/courses/191/info