

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)

Студента 2м курсу, 5 групи,
спеціальності
105 «Програмна інженерія»
спеціалізації
«Інженерія програмного
забезпечення»

підпис студента

Яремича Валентина
Романовича

Науковий керівник
кандидат юридичних наук,
доцент

підпис керівника

Демедюк Сергій
Васильович

Гарант освітньої програми
доктор технічних наук,
професор

підпис керівника

Криворучко Олена
Володимирівна

КИЇВ – 2019

[illegible]

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 СИСТЕМНИЙ АНАЛІЗ ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ ЗАКЛАДУ ВИЩОЇ ОСВІТИ

1.1 Управління освітнім процесом

1.2 Формування розкладу занять

1.3 Висновки до розділу 1

РОЗДІЛ 2 МЕТОДИ РЕАЛІЗАЦІЇ БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)

2.1 Постановка задачі

2.2 Вибір методу реалізації завдання

2.3.1 Система керування реляційними базами даних MySQL

2.3.2 Мова програмування JavaScript

2.3.3 Мова програмування PHP

2.3.4 Програмний каркас CodeIgniter

2.3.5 Бібліотека Grocery CRUD

2.3 Висновки до розділу 2

РОЗДІЛ 3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)

3.1 Опис структури таблиць і схема даних

3.2 Висновки до розділу 3

РОЗДІЛ 4 СТВОРЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)

4.1 Створення адміністративної частини

4.2 Створення користувацької частини

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

7. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	21.09.2018	
2	<i>Розробка та затвердження завдання на проект магістра</i>	07.11.2018	
3	<i>Вступ та перелік літературних джерел</i>	25.01.2019	
4	<i>Розробка технічного завдання</i>	28.02.2019	
5	<i>Розділ 1.</i>	22.03.2019	
6	<i>Розділ 2.</i>	18.04.2019	
7	<i>Розділ 3.</i>	24.05.2019	
8	<i>Розділ 4.</i>	21.06.2019	
9	<i>Розробка програми та методики тестування</i>	20.09.2019	
10	<i>Написання наукової статті</i>	27.09.2019	
11	<i>Керівництво користувача</i>	18.10.2019	
12	<i>Висновки та пропозиції</i>	30.10.2019	
13	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>	08.11.2019	
14	<i>Підготовка автореферату та презентації доповіді</i>	15.11.2019	
15	<i>Попередній захист випускної кваліфікаційної роботи</i>	29.11.2019	
16	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	01.12.2019	
17	<i>Підготовка до публічного захисту випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання " ____ " _____ 20 ____ р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Демедюк С.В.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Криворучко О.В.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Яремич В.Р.

(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

(підпис, дата)

Випускна кваліфікаційна робота (проект) студента Яремича В.Р.
(прізвище, ініціали)
може бути допущена до захисту екзаменаційній комісії.

Криворучко О.В.
(підпис, прізвище, ініціали)

Криворучко О.В.
(підпис, прізвище, ініціали)

" " 20 p.

Анотація

Яремич В.Р. Розробка бази даних аудиторного фонду вищого закладу освіти (на прикладі КНТЕУ, корпус А)

У випускній кваліфікаційній роботі розглядається розробка бази даних аудиторного фонду вищого закладу освіти (на прикладі КНТЕУ, корпусу А), за допомогою програмного каркасу CodeIgniter та бібліотеки Grocery CRUD, особливості розробки за допомогою даних технологій, їх переваги та недоліки.

Ключові слова: база даних аудиторного фонду, CodeIgniter, Grocery CRUD.

Annotation

Yaremych V.R. Development of a classroom database for institution of higher education (on the example of KNUTE, building A)

The final qualification examines the development of a database for institution of higher education (on the example of KNUTE, building A) using the CodeIgniter software framework and Grocery CRUD library, features of development using these technologies, their advantages and disadvantages.

Keywords: classroom database, CodeIgniter, Grocery CRUD.

Зміст

ВСТУП.....	5#
РОЗДІЛ 1 СИСТЕМНИЙ АНАЛІЗ ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ ЗАКЛАДУ ВИЩОЇ ОСВІТИ.....	8#
1.1# Управління освітнім процесом	8#
1.2# Формування розкладу занять	10#
1.3# Висновки до розділу 1.....	12#
РОЗДІЛ 2 МЕТОДИ РЕАЛІЗАЦІЇ БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А).....	13#
2.1# Постановка задачі.....	13#
2.2# Вибір методу реалізації завдання	13#
2.3.1# Система керування реляційними базами даних MySQL	14#
2.3.2# Мова програмування JavaScript.....	15#
2.3.3# Мова програмування PHP	15#
2.3.4# Програмний каркас CodeIgniter.....	15#
2.3.5# Бібліотека Grocery CRUD	18#
2.3# Висновки до розділу 2.....	19#
РОЗДІЛ 3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)	20#
3.1# Опис структури таблиць і схема даних.....	21#
3.2# Висновки до розділу 3.....	26#

					КНТЕУ 121-05-25.МР			
					Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Зміст	2	41
Зав. каф.	Криворучко О.В.							
Керівник	Демедюк С.В.							
Гарант	Криворучко О.В.				Зміст	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Розроб.	Яремич В.Р.							

РОЗДІЛ 4 СТВОРЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)	27#
4.1# Створення адміністративної частини.....	28#
4.2# Створення користувацької частини.....	34#
4.3# Висновки до розділу 4.....	37#
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	38#
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	40#
ДОДАТКИ.....	42#

					<i>КНТЕУ 121-05-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		3

Перелік умовних позначень та термінів

КНТЕУ	– Київський національний торговельно-економічний університет.
ЗВО	– Заклад вищої освіти.
БД	– База даних.
СКБД	– Система керування базами даних.
EER	– Розширена модель сутностей-зв'язків (англ. Extended Entity-Relationship Model)
CRUD	– Акронім, що позначає чотири базові функції, які використовуються при роботі з базами даних: створення (англ. create), читання (read), модифікація (update), видалення (delete).
ПЗ	– Програмне забезпечення (програмні засоби).
MVC	– Патерн розробки Model-View-Controller («Модель-Відображення-Контролер»)
API	– Набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

ВСТУП

Швидкий доступ до інформації на сьогодні є одним з ключових факторів ефективності роботи будь-якої установи. На пошук необхідної інформації без використання інформаційних систем, створених для її видачі, може бути витрачено у десятки разів більше часу ніж з нею. Така втрата часу є неприпустимою в цифрову еру.

Автоматизація будь-яких процесів, які можуть бути автоматизовані, повинна бути пріоритетним завданням для установ. Час та кошти витрачені на автоматизацію процесів виправдовують ціль, надаючи суттєву економію того самого часу та коштів у майбутньому.

Незалежно від об'єкта автоматизації, в закладі вищої освіти такі системи впроваджують, переслідуючи кінцеву мету – підвищення якості освіти.

Однією з найважливіших проблем якісної організації освітнього процесу у закладі вищої освіти є завдання створення навчального розкладу. Правильно і точно складений розклад забезпечує рівномірне завантаження студентських груп і професорсько-викладацького складу.

Тому розробка бази даних аудиторного фонду з користувацьким веб-інтерфейсом для оптимізації розподілу аудиторного фонду є актуальною задачею. Швидкий доступ до інформації про аудиторії повинен значно пришвидшити час складання навчального розкладу.

Кожна аудиторія – це фізична кімната в приміщенні навчального закладу, обладнана спеціальним чином для проведення занять. Кожна з

					КНТЕУ 121-05-25.МР			
					Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		В	5	41
Зав. каф.	Криворучко О.В.							
Керівник	Демедюк С.В.							
Гарант	Криворучко О.В.				Вступ	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Розроб.	Яремич В.Р.							

кімнат облаштована під конкретний вид занять: лекційні аудиторії – для лекцій, лабораторії – для лабораторних. Також аудиторія містить певну кількість місць, на яку вона розрахована, специфічне оснащення, як-от: інтерактивна дошка, мультимедійний проектор, телевізійна панель, трибуна з акустичною системою. У деяких аудиторіях можуть проводитися декілька типів занять. Але такі аудиторії будуть використовуватись тільки тоді, коли відповідні будуть зайняті.

Розробка дозволить працівникам університету швидко отримувати інформацію про аудиторію або знаходити необхідну за певними параметрами. Таким чином, можна оптимізувати швидкість створення навчальних розкладів.

Актуальність полягає у дослідженні роботи Київського національного торговельно-економічного університету, та розробки бази даних аудиторного фонду КНТЕУ (корпус А), яка б допомогла персоналу в роботі, спростивши пошук аудиторій з необхідними параметрами.

Об’єкт дослідження: освітня діяльність КНТЕУ.

Предмет дослідження: процес створення бази даних аудиторного фонду КНТЕУ (корпус А) та інтерфейсу користувача з використанням СКБД MySQL, програмного каркаса CodeIgniter, генератору CRUD-інтерфейсів Grocery CRUD та мов програмування PHP і JavaScript.

Мета роботи: створення бази даних аудиторного фонду КНТЕУ (корпус А) для оптимізації розподілу аудиторного фонду у відповідності з освітнім процесом.

Методи і технології розробки: в роботі використовуються методи об’єктно-орієнтованого програмування, програмування на мові PHP, робота з сховищами даних, розробка веб-інтерфейсу користувача, робота з програмними каркасами.

					<i>КНТЕУ 121-05-25.MP</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

Завдання: дослідження роботи КНТЕУ, аналіз нормативної і технічної документації КНТЕУ, проектування бази даних аудиторного фонду КНТЕУ (корпус А), розробка веб-інтерфейсу користувача.

Наукова новизна дослідження полягає у цілеспрямованому створенні бази даних аудиторного фонду КНТЕУ (корпус А), яка дасть змогу користувачеві отримати та зібрати всю необхідну інформацію в єдиній системі, мати доступ до її редагування та перегляду, не встановлюючи додаткового програмного забезпечення.

Результат роботи: база даних аудиторного фонду КНТЕУ (корпус А) з інтуїтивно зрозумілим користувацьким веб-інтерфейсом.

					<i>КНТЕУ 121-05-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		7

РОЗДІЛ 1

СИСТЕМНИЙ АНАЛІЗ ОРГАНІЗАЦІЇ ОСВІТНЬОГО ПРОЦЕСУ

ЗАКЛАДУ ВИЩОЇ ОСВІТИ

1.1 Управління освітнім процесом

Основним нормативним документом, що визначає перелік навчальних дисциплін і логічну послідовність їх вивчення є навчальний план.

Навчальний план – це нормативний документ, за допомогою якого здійснюється організація освітнього процесу. Навчальний план складається вищим закладом освіти на підставі освітньо-професійної програми та структурно-логічної схеми підготовки, і який визначає перелік та обсяг нормативних і вибіркового навчальних дисциплін, послідовність їх вивчення, конкретні форми проведення навчальних занять (лекції, лабораторні, практичні, семінарські, індивідуальні заняття, консультації, навчальні й виробничі практики) та їхній обсяг, графік навчального процесу, форми і засоби поточного й підсумкового контролю. У навчальному плані відображається також обсяг часу, передбачений на самостійну роботу [1].

Розглядаючи навчальний план у контексті організації та управління навчальним процесом, можна також констатувати, що навчальний план, крім студентів, стосується викладачів, а також розподілу ресурсів навчального закладу (приміщення, матеріали для навчання), тобто усієї сфери навчальної діяльності ЗВО (Рис. 1.1).

					<i>КНТЕУ 121-05-25.МР</i>			
					Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		РІ	8	41
Зав. каф.	Криворучко О.В.				Розділ 1. Системний аналіз організації освітнього процесу закладу вищої освіти	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Керівник	Демедюк С.В.							
Гарант	Криворучко О.В.							
Розроб.	Яремич В.Р.							

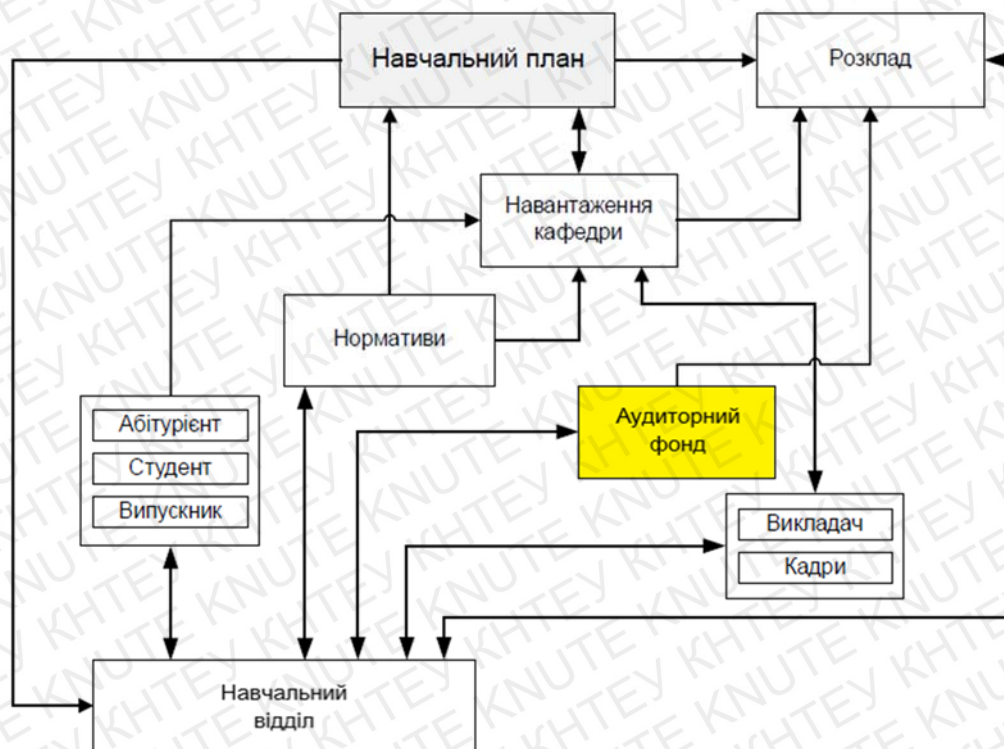


Рис. 1.1. Схема зв'язків інформаційної моделі управління освітнім процесом

У результаті аналізу предметної області була розроблена контекстна діаграма потоків даних (Рис. 1.2).



Рис. 1.2. Контекстна діаграма потоків даних при управлінні освітнім процесом

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121-05-25.МР

Аркуш

9

У контекстній діаграмі наявний основний процес і три зовнішні сутності, що взаємодіють з ним: “Навчальний відділ”, “Кафедра”, “Деканат”.

Зовнішня сутність “Навчальний відділ” є відповідним підрозділом ЗВО, що здійснює облік і контроль навчальних планів за спеціальностями, складання та корегування розкладу навчальних занять, облік і контроль розподілу навчального навантаження професорсько-викладацького складу кафедр, контроль фактичного виконання цього навантаження та інші види діяльності, дає змогу на основі єдиного довідника дисциплін, кафедр, спеціальностей та інших елементів навчання уніфікувати елементи навчального плану, і у результаті отримувати інтегровані показники навантаження за спеціальностями, кафедрами і загалом за ЗВО.

Зовнішня сутність “Кафедра” бере участь в організації освітнього процесу на рівні цього підрозділу.

Зовнішня сутність “Деканат” – це відповідний підрозділ ЗВО, який бере участь в організації навчального процесу на рівні деканату.

1.2 Формування розкладу занять

Розклад занять – основний документ, що регламентує академічну роботу студентів і викладачів, передбачає проведення лекційних, лабораторних, семінарських занять відповідно до робочих навчальних планів, ухвалених вченою радою та затверджених ректором.

Диспетчерам потрібно завжди вчасно надавати зручний розклад занять, розрахований і спроектований на основі даних офіційних навчальних планів кожного напрямку навчання.

Для цього важливо бути в курсі стану кожної аудиторії, що знаходиться в розпорядженні, а також володіти інформацією про їх місткість і наявне обладнання.

					<i>КНТЕУ 121-05-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		10

Характеристики аудиторного фонду також є важливим параметром при вирішенні задач планування навчального розкладу.

Всі заняття можна розділити на два види:

- лекційні;
- спеціалізовані.

Якщо в разі лекційних занять основним параметром аудиторії є її місткість, то обмеження, що накладаються спеціалізованими заняттями, досить великі і складність їх вимог може варіюватися від простих до вимог надати аудиторію з обладнанням високої потужності яке володіє конкретними характеристиками.

В умовах розвитку сфери інформаційних технологій і все більшого застосування комп'ютерної техніки в навчанні найбільш поширеним типом спеціалізованих аудиторій є комп'ютерні класи.

У загальному випадку вимоги до комп'ютерних класів можна розділити на дві групи:

- вимога особливих технічних характеристик;
- вимога до встановленого програмного забезпечення.

У зв'язку з цим для даних аудиторій можна виділити профільні та непрофільні заняття. У загальному випадку, якщо йде мова про непрофільних заняттях, то вимоги до характеристик обладнання та встановленим програмному забезпеченню мінімальні. У випадку ж з профільними заняттями потрібно говорити про більш конкретні вимоги. В якості найпростішого рішення може служити повне оснащення аудиторій необхідним обладнанням, але, з точки зору раціонального використання матеріальних ресурсів, дане рішення є одним з найгірших.

Таким чином, актуальною стає автоматизація ведення обліку стану аудиторного фонду та його оснащеності.

					<i>КНТЕУ 121-05-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

1.3 Висновки до розділу 1

Основним нормативним документом, що визначає перелік навчальних дисциплін і логічну послідовність їх вивчення є навчальний план. На основі навчального плану складається розклад занять. Він регламентує академічну роботу студентів і викладачів, передбачає проведення лекційних, лабораторних, семінарських занять.

За складання навчальних планів та розкладів занять відповідає навчальний відділ закладу вищої освіти.

Для складання зручного розкладу занять, розрахованого і спроектованого на основі даних офіційних навчальних планів кожного напрямку навчання, враховуючи його специфіку та залежність від оснащення аудиторій, необхідно бути в курсі стану кожної аудиторії, що знаходиться в розпорядженні, а також володіти інформацією про їх місткість і наявне обладнання.

Таким чином, актуальною стає автоматизація ведення обліку стану аудиторного фонду та його оснащення.

					<i>КНТЕУ 121-05-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

РОЗДІЛ 2

МЕТОДИ РЕАЛІЗАЦІЇ БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)

Призначення бази даних аудиторного фонду КНТЕУ (корпус А) – збереження інформації про аудиторії у вищому закладі освіти; надання користувачеві зручного інтерфейсу пошуку, редагування, додавання та видалення інформації про аудиторії.

Створена база даних в купі з веб-інтерфейсом керування повинна значно оптимізувати розподіл аудиторного фонду у відповідності з індивідуальними навчальними планами здобувачів вищої освіти.

2.1 Постановка задачі

Інтерфейс бази даних аудиторного фонду КНТЕУ (корпус А) повинен підтримувати наступні функції:

- додавання нових аудиторій;
- редагування інформації про наявні аудиторії;
- видалення аудиторій;
- перегляд інформації про аудиторії;
- можливість фільтрувати аудиторії за різними критеріями;
- завантаження фотографії аудиторії.

2.2 Вибір методу реалізації завдання

Базу даних аудиторного фонду КНТЕУ (корпус А) та веб-інтерфейс керування було вирішено розробляти за допомогою відкритого програмного забезпечення, що є оптимальним для використання закладом

					<i>КНТЕУ 121-05-25.МР</i>			
					<i>Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>Р2</i>	<i>13</i>	<i>41</i>
<i>Зав. каф.</i>	<i>Криворучко О.В.</i>							
<i>Керівник</i>	<i>Демедюк С.В.</i>							
<i>Гарант</i>	<i>Криворучко О.В.</i>				<i>Розділ 2. Методи реалізації бази даних аудиторного фонду кнтеу (корпус а)</i>	<i>Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група</i>		
<i>Розроб.</i>	<i>Яремич В.Р.</i>							

вищої освіти і не потребує виділення додаткових коштів для реалізації програмного застосунку.

Для створення бази даних аудиторного фонду КНТЕУ (корпус А) та веб-інтерфейсу користувача, були використані СКБД MySQL, програмний каркас CodeIgniter, CRUD генератор Grocery CRUD та мови програмування PHP і Javascript.

2.3.1 Система керування реляційними базами даних MySQL

MySQL – вільно доступна система керування реляційними базами даних з відкритим вихідним кодом, яка використовує структуровану мову запитів (SQL). MySQL може розповсюджуватися відповідно до умов ліцензії GPL або комерційною ліцензією Oracle [2].

MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Система MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопотоковості, що підвищує продуктивність системи в цілому [3].

Для баз даних MySQL створено інструмент візуального проектування баз даних MySQL Workbench, що інтегрує проектування, моделювання, створення й експлуатацію БД в єдине безкоштовне оточення.

Переваги системи керування базами даних MySQL:

- простота у встановленні та використанні;
- підтримка необмеженої кількості користувачів, що одночасно працюють із БД;

- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої й ефективної системи безпеки.

2.3.2 Мова програмування JavaScript

JavaScript – це нетипізована інтерпретована мова програмування з об'єктно-орієнтованими можливостями.

Зазвичай JavaScript застосовується в веб-браузерах, а розширення її можливостей за рахунок введення об'єктів дозволяє організувати взаємодію з користувачем, управляти веб-браузером і змінювати вміст документа, що відображається в межах вікна веб-браузера. Ця вбудована версія JavaScript запускає сценарії, впроваджені в HTML-код веб-сторінок. Як правило, ця версія називається клієнтською, щоб підкреслити, що сценарій виконується на клієнтському комп'ютері, а не на веб-сервері [12].

2.3.3 Мова програмування PHP

PHP – скриптова мова програмування, створена для генерації HTML-сторінок на стороні веб-сервера. Одна з найпоширеніших мов, що використовуються у сфері веб-розробок, проект відкритого програмного забезпечення.

Мова PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий HTML-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок [4].

2.3.4 Програмний каркас CodeIgniter

CodeIgniter – програмний каркас, створений на PHP, для розробки веб-систем та застосунків. В перше публічно представлений EllisLab у 2006 році, а 6 червня 2013 року переданий Технологічному Інституту Британської Колумбії (Канада). CodeIgniter поширюється за групою

					<i>КНТЕУ 121-05-25.MP</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

ліцензій, розроблених Массачусетським технологічним інститутом для розповсюдження вільного програмного забезпечення [10].

CodeIgniter швидко набрав популярність завдяки своїй простоті та швидкодії. Він використовує MVC архітектуру, що дозволяє підтримувати структурування при розробці. Має підтримку сторонніх бібліотек та систему кешування.

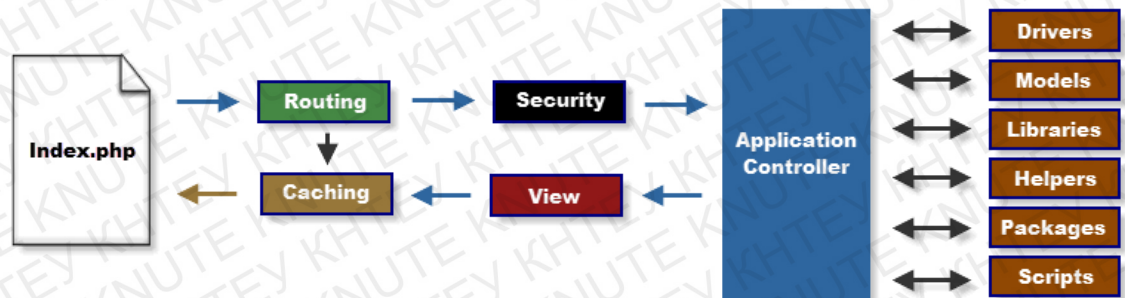


Рис.2.1 Архітектура CodeIgniter

Основою стеку використаних технологій є програмний каркас CodeIgniter, який має свої переваги і недоліки.

Переваги програмного каркаса CodeIgniter:

- Відмінна і зрозуміла документація. У CodeIgniter дуже багата і зрозуміла документація, яку приємно читати. Все підкріплено прикладами коду, який можна просто скопіювати та використовувати у себе.
- Не вимагає великої кількості ресурсів. CodeIgniter останньої стабільної версії може працювати майже на всіх веб-серверах з підтримкою PHP не нижче 5.6 версії. Завдяки своїй продуманій структурі CodeIgniter не навантажує систему і дуже швидко відпрацьовує.
- Дуже простий у використанні. Як новачки, так і досвідчені програмісти PHP можуть використовувати CodeIgniter без

витрачання великої кількості часу і зусиль на опанування програмного каркаса.

- Швидкість генерації сторінок. CodeIgniter дуже швидкий в роботі у порівнянні з іншими програмними каркасами.
- Підтримка великої кількості баз даних. За допомогою своїх вбудованих драйверів CodeIgniter може працювати з різними базами даних, таким як: MySQL, PostgreSQL, MSSQL, SQLite, Oracle. В CodeIgniter реалізований шаблон проектування ActiveRecord для роботи з базами даних.
- Підтримка кешування веб-сторінок і баз даних. Крім того, що CodeIgniter швидше, ніж інші програмні каркаси PHP, він дозволяє програмістам налаштувати кешування веб-сторінок і баз даних. Це зменшує навантаження на сервер, дозволяючи розробникам кешувати веб-сторінки в повному візуалізованому стані. Веб-розробники мають можливість ввімкнути кешування лише для певних сторінок та встановити тривалість кешування. Водночас, програмісти можуть зменшити навантаження на базу даних, кешуючи запити бази даних як прості текстові файли через Database Caching Class.
- Безліч ресурсів по CodeIgniter. В інтернеті існує величезна кількість ресурсів, присвячених CodeIgniter. Так що знайти бібліотеки сторонніх розробників або знайти відповідь на своє питання по CodeIgniter не складає великих труднощів.

Недоліки програмного каркаса CodeIgniter:

- Відсутня підтримка модульного розділення коду за замовчуванням. Сьогодні підприємства вимагають від програмістів писати чистий, читаний і структурований код. Високоякісний код полегшує

					<i>КНТЕУ 121-05-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

підприємствам підтримку, зміну та оновлення веб-додатків у майбутньому. На відміну від інших сучасних програмних каркасів PHP, CodeIgniter не підтримує модульне розділення коду за замовчуванням. Отже, програмісти повинні докласти додатковий час і зусилля для підтримки або оновлення коду.

- Не вистачає вичерпних бібліотек. Як і інші програмні каркаси, CodeIgniter також прискорює розробку власних веб-додатків, надаючи ряд бібліотек. Вона також дозволяє розробникам встановлювати лише бібліотеки, необхідні веб-додатку. Програмісти можуть навіть розширювати та налаштовувати рідні бібліотеки, просто розширюючи рідні класи. Але бібліотеки, що надаються CodeIgniter, не є настільки вичерпними, як бібліотеки, надані іншими новішими програмними каркасами PHP.

2.3.5 Бібліотека Grocery CRUD

Grocery CRUD – це бібліотека з відкритим кодом для CodeIgniter, яка полегшує життя веб-розробників. За допомогою кількох простих рядків коду Grocery CRUD дозволяє створити повний стабільний CRUD-інтерфейс, який до того має гарний вигляд [11].

Переваги бібліотеки Grocery CRUD:

- Підтримує відносини «один до багатьох» і «багато до багатьох» з коробки.
- Перекладено на 34 мови у тому числі на українську.
- Підтримує всі основні браузери, Chrome, Safari, Firefox, Opera та навіть Internet Explorer.
- Існує можливість змінити стандартну поведінку бібліотеки за допомогою зворотних викликів.
- Має добре написану API документацію та живе ком'юніті.

					<i>КНТЕУ 121-05-25.MP</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

2.3 Висновки до розділу 2

Стек використаних технологій, що має в основі програмний каркас CodeIgniter, бібліотеку Grocery CRUD та СКБД MySQL, добре підходить для реалізації бази даних аудиторного фонду КНТЕУ (корпус А), він не потребує закупівлі платних ліцензій зі сторони закладу вищої освіти, має чудову масштабованість.

Зрозуміла документація до програмного каркасу CodeIgniter та бібліотеки Grocery CRUD дозволить підтримувати у майбутньому розроблену базу даних аудиторного різними програмістами.

					<i>КНТЕУ 121-05-25.MP</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		19

РОЗДІЛ 3

ПРОЕКТУВАННЯ БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)

Перед усім, при проектуванні бази даних повинна бути спланована структура кожної таблиці та зв'язки між ними. Структура таблиць обумовлюється вмістом запитів та результатів, які повинні бути отримані при роботі з базою даних.

Таблиці – це важливі об'єкти в базі даних. Саме в них зберігаються всі дані. Інші об'єкти бази даних сильно залежать від структури таблиць, тому розробку бази даних завжди слід починати зі створення власне таблиць і лише після цього можна переходити до будь-яких інших об'єктів.

Умовно таблиці, що складають базу даних аудиторного фонду КНТЕУ (корпус А), можна поділити на два типи: таблиці-словники, та таблиці з інформацією, що може часто змінюватися, або доповнюватися.

Таблиці-словники найчастіше містять в собі статичну інформацію, яка не змінюється протягом довгого періоду часу. Прикладом такої таблиці в даній базі даних будуть співвідношення «ключ – розшифрування». Вони використовуються для визначення певних даних, як наприклад, назва кафедри, чи призначення аудиторії.

Таблиці з даними, що змінюються загалом містять посилання на таблиці-словники.

Зміст таблиць бази даних аудиторного фонду КНТЕУ (корпус А) представлений у таблиці 3.1.

					<i>КНТЕУ 121-05-25.МР</i>			
					<i>Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>Р4</i>	<i>20</i>	<i>41</i>
<i>Зав. каф.</i>	<i>Криворучко О.В.</i>				<i>Розділ 4. Проектування бази даних аудиторного фонду КНТЕУ (корпус А)</i>	<i>Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група</i>		
<i>Керівник</i>	<i>Демедюк С.В.</i>							
<i>Гарант</i>	<i>Криворучко О.В.</i>							
<i>Розроб.</i>	<i>Яремич В.Р.</i>							

Таблиця 3.1.

ЗМІСТ ТАБЛИЦЬ БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)

Назва таблиці	Зміст таблиці
Auditorium	Інформація про аудиторії.
AuditoriumSoftwares	Інформація про встановлене в аудиторіях програмне забезпечення.
Software	Перелік програмного забезпечення.
AuditoriumFeatures	Інформація про встановлене в аудиторії обладнання.
Feature	Перелік обладнання.
ResponsiblePerson	Інформація про відповідальних за аудиторії осіб.
Function	Перелік призначень аудиторій.
Department	Перелік кафедр ЗВО.

3.1 Опис структури таблиць і схема даних

Таблиця «Auditorium» відображає основну сутність – аудиторію. Таблиця містить загальну інформацію про аудиторію, таку як номер аудиторії, призначення, належність до кафедри, кількість навчальних місць (Таблиця 3.2).

Таблиця 3.2.

ТАБЛИЦЯ «AUDITORIUM»

Поле	Тип	Опис
Id	Int	Числове значення, первинний ключ, що однозначно визначає аудиторію.
Area	Decimal	Числове значення, площа аудиторії у м ² .
PictureUrl	Varchar	Строкове значення, посилання на фото.

Продовження таблиці 3.2.

StudyPlaces	Int	Числове значення, кількість навчальних місць в аудиторії.
Computers	Int	Числове значення, кількість персональних комп'ютерів в аудиторії.
FunctionId	Int	Числове значення, вторинний ключ до таблиці-словника «Function», що містить перелік призначень аудиторій.
DepartmentId	Int	Числове значення, вторинний ключ до таблиці-словника «Department», що містить перелік кафедр.
ResponsiblePersonId	Int	Числове значення, вторинний ключ до таблиці-словника «ResponsiblePerson», що містить перелік відповідальних за аудиторії осіб.

Таблиця «AuditoriumFeatures» призначена для організації зв'язку «Many-to-many» між таблицями «Auditorium» та «Feature». Суть полягає у тому, що аудиторії можуть мати певне обладнання необхідне для проведення заняття, наприклад, проектор або телевізійну панель. Наявність або відсутність такого обладнання суттєво впливає на підбір аудиторії для проведення занять. Таким чином, для того, щоб зв'язати відповідні записи в таблицях «Auditorium» та «Feature» вводиться допоміжна таблиця, що встановлює цей зв'язок. Також в даному випадку у таблицю зв'язку додано поле, що зберігає додаткову інформацію про наявне обладнання, наприклад, таку як розмір телевізійної панелі, наявність підключеного персонального комп'ютера до неї (Таблиця 3.3).

					КНТЕУ 121-05-25.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		22

Таблиця 3.3.

ТАБЛИЦЯ «AUDITORIUMFEATURES»

Поле	Тип	Опис
AuditoriumId	Int	Числове поле, вторинний ключ до таблиці з записами щодо аудиторій.
SoftwareId	Int	Числове поле, вторинний ключ до таблиці з записами щодо програмного забезпечення.
AdditionalInfo	Varchar	Строкове значення, містить додаткову інформацію про наявне обладнання.

Таблиця «AuditoriumSoftware» призначена для організації зв'язку «Many-to-many» між таблицями «Auditorium» та «Software». Суть полягає у тому, що персональні комп'ютери у різних комп'ютерних класах мають різне програмне забезпечення, що впливає на підбір класу для проведення занять. Таким чином, для того, щоб зв'язати відповідні записи в таблицях «Auditorium» та «Software» вводиться допоміжна таблиця, що встановлює цей зв'язок (Таблиця 3.4).

Таблиця 3.4.

ТАБЛИЦЯ «AUDITORIUMSOFTWARES»

Поле	Тип	Опис
AuditoriumId	Int	числове поле, вторинний ключ до таблиці з записами щодо аудиторій.
SoftwareId	Int	числове поле, вторинний ключ до таблиці з записами щодо програмного забезпечення.

Таблиця «Department» – таблиця словник, використовуються в першу чергу для відокремлення в БД, конкретно в таблиці «Auditorium», інформації про кафедру (Таблиця 3.5). Таким чином, інформація, що часто повторювалася б у таблиці «Auditorium», замінюється вторинним ключем на запис в окремій таблиці.

Таблиця 3.5.

ТАБЛИЦЯ «DEPARTMENT»

Поле	Тип	Опис
Id	Int	Числове значення, первинний ключ.
Denotation	Varchar	Строкове значення, містить назву кафедри.

Таблиця «Function» – таблиця словник, слугує для тих самих цілей що і таблиця «Department», містить інформацію про типи аудиторій (Таблиця 3.6).

Таблиця 3.6.

ТАБЛИЦЯ «FUNCTION»

Поле	Тип	Опис
Id	Int	Числове значення, первинний ключ.
Denotation	Varchar	Строкове значення, містить тип аудиторії.

Таблиця «Software» – таблиця словник, використовуються для відокремлення в таблиці «AuditoriumSoftwares» інформації про програмне забезпечення, що використовується в аудиторіях (Таблиця 3.7).

Таблиця 3.7.

ТАБЛИЦЯ «SOFTWARE»

Поле	Тип	Опис
Id	Int	Числове значення, первинний ключ.

Denotation	Varchar	Строкове значення, містить назву ПЗ.
------------	---------	--------------------------------------

Таблиця «Feature» – таблиця словник, використовуються для відокремлення в таблиці «AuditoriumFeatures» інформації про спеціальне обладнання, що використовується в аудиторіях (Таблиця 3.8).

Таблиця 3.8.

ТАБЛИЦЯ «FEATURE»

Поле	Тип	Опис
Id	Int	Числове значення, первинний ключ.
Denotation	Varchar	Строкове значення, містить назву обладнання.

Таблиця «ResponsiblePerson» містить інформацію про відповідальних за аудиторії осіб (Таблиця 3.9).

Таблиця 3.9.

ТАБЛИЦЯ «RESPONSIBLEPERSON»

Поле	Тип	Опис
Id	Int	Числове значення, первинний ключ, що однозначно визначає відповідальну особу.
FullName	Varchar	Строкове значення, містить прізвище, ім'я, по батькові відповідальної особи.
InternalPhoneNumber	Varchar	Строкове значення, містить внутрішній номер телефону відповідальної особи.
MobilePhoneNumber	Varchar	Строкове значення, містить мобільний номер телефону відповідальної особи.
EmailAdress	Varchar	Строкове значення, містить робочу електронну адресу відповідальної особи.
CabinetNumber1	Int	Числове значення, містить перший номер

		кабінету відповідальної особи.
CabinetNumber2	Int	Числове значення, містить другий номер кабінету відповідальної особи.

На основі структури бази даних у програмі MySQL Workbench було створено EER-діаграму бази даних з усіма зв'язками (Рис. 3.1).

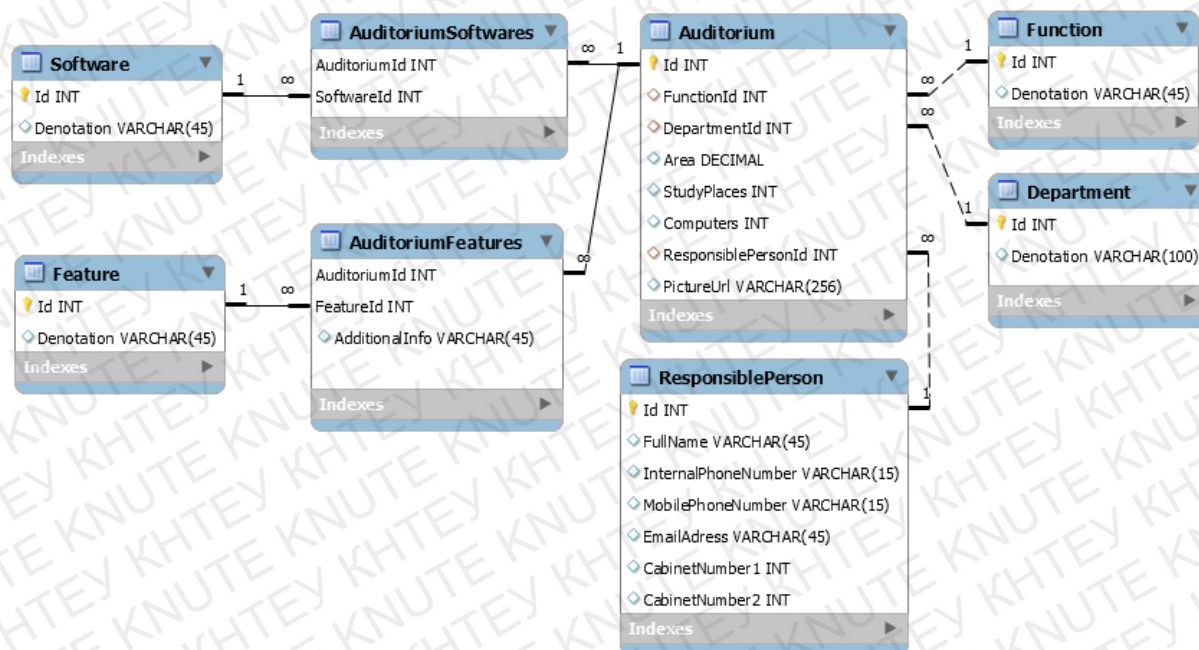


Рис. 3.1. EER-діаграма бази даних аудиторного фонду КНТЕУ (корпус А)

3.2 Висновки до розділу 3

Перед усім, при проектуванні бази даних повинна бути спланована структура кожної таблиці та зв'язки між ними. Структура таблиць обумовлюється вмістом запитів та результатів, які повинні бути отримані при роботі з базою даних.

Інші об'єкти бази даних сильно залежать від структури таблиць, тому розробку бази даних завжди слід починати зі створення власне таблиць і тільки після цього переходити до створення інших частин.

РОЗДІЛ 4

СТВОРЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА БАЗИ ДАНИХ АУДИТОРНОГО ФОНДУ КНТЕУ (КОРПУС А)

Відображення, редагування, додавання та видалення записів з бази даних аудиторного фонду КНТЕУ (корпус А) здійснюватиметься засобами програмного каркасу CodeIgniter. Для розширеної фільтрації записів буде використовуватися відкрита бібліотека TableFilter [9], яка написана на мові програмування Javascript.

CodeIgniter заснований на патерні розробки Model-View-Controller («Модель-Відображення-Контролер»). На практиці це дозволяє веб-сторінкам містити мінімальну кількість коду в поданні, відокремленим від PHP-скриптинга.

Модель відображає структуру даних. Зазвичай класи моделей містять функції, які допомагають отримувати, вставляти і оновлювати інформацію у вашій базі даних.

Відображення це інформація, яка буде відправлена користувачеві. Відображення це звичайні веб-сторінки, але в CodeIgniter відображення можуть бути також фрагментами сторінок, наприклад шапкою і підвалом.

Контролер працює як посередник між моделлю, відображенням і іншими ресурсами, необхідними для обробки HTTP-запиту і генерації веб-сторінки.

					КНТЕУ 121-05-25.МР			
					Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Р4	27	41
Зав. каф.	Криворучко О.В.							
Керівник	Демедюк С.В.							
Гарант	Криворучко О.В.				Розділ 4. Створення інтерфейсу користувача бази даних аудиторного фонду КНТЕУ (корпус А)	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Розроб.	Яремич В.Р.							

Використовуючи патерн MVC, данні з бази даних MySQL оброблюються за допомогою PHP-запиту який опрацьовується на веб-сервері (Рис. 4.1).

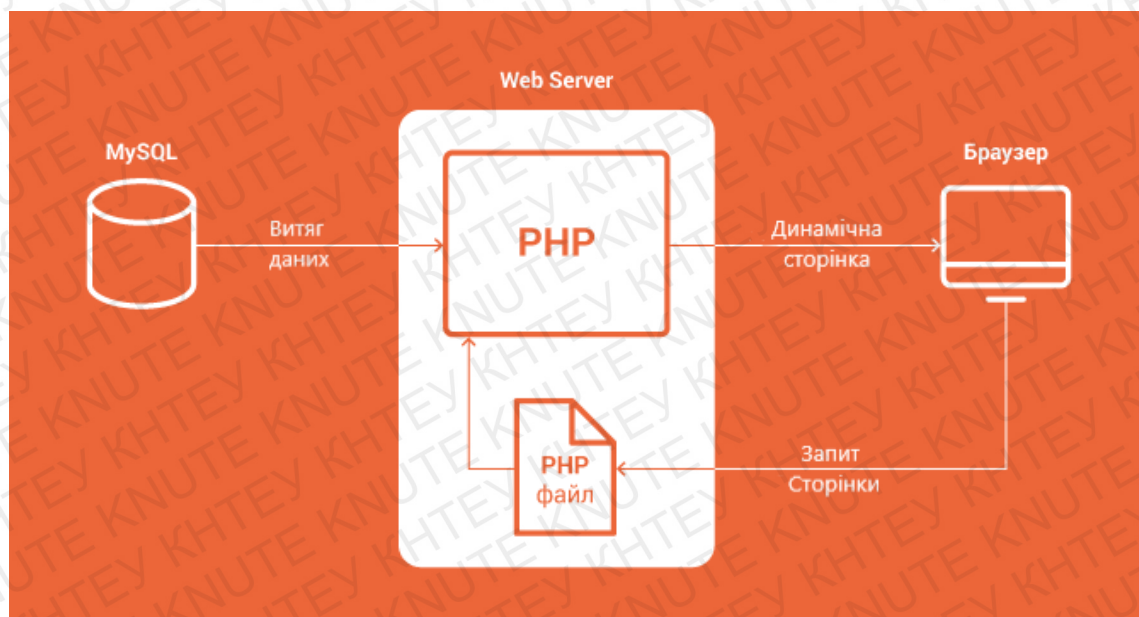


Рис. 4.1 Схема генерації користувацької веб-сторінки

4.1 Створення адміністративної частини

Дотримуючись патерну MVC, створимо для адміністративної частини окреме відображення **management.php** та контролер **manage.php** (модель представлена Grocery CRUD).

За допомогою цих двох файлів відбувається генерація адміністративної частини.

Одразу після створення файлів можна побачити результат своєї роботи. За допомогою запитів від браузера до веб-сервісу генеруються окремі сторінки для перегляду, редагування, додавання та видалення записів з певної таблички бази даних.

Згенерований інтерфейс являє собою табличку з налаштованими у контролері полями, панеллю фільтрації, та кнопками, що дозволяють редагувати, видаляти або додавати нові записи. Так, контролер дозволяє

прибирати певні поля з виводу, забороняти певні операції із записами в базі даних, налаштовувати зв'язки між таблицями для виводу пов'язаної інформації замість ідентифікаторів.

Для редагування в адміністративній частині доступні наступні таблиці:

- «Auditorium» (Аудиторія);
- «Software» (Програмне забезпечення);
- «Feature» (Обладнання);
- «AuditoriumFeatures» (Обладнання аудиторій);
- «Function» (Призначення);
- «Department» (Кафедра);
- «ResponsiblePerson» (Відповідальна особа).

Додавання нових записів та видалення старих у таблицях «AuditoriumFeatures» та «AuditoriumSoftwares» відбувається автоматично при виборі з переліку на сторінці редагування інформації про аудиторії:

Рис. 4.2 Перелік обладнання на сторінці редагування аудиторії

Як видно зі списку до редагування додана таблиця «AuditoriumFeatures». Це було зроблено для надання можливості введення додаткової інформації про наявне обладнання.

Окрім в адміністративній частині наявне посилання на перехід до користувацької частині, яка надає лише можливість перегляду інформації, без змоги її редагувати, видаляти чи додавати нову.

Вікно перегляду аудиторій в адміністративній частині виглядає наступним чином:

Кабінети | Кафедри | Призначення | Програмне забезпечення | Обладнання | Аудиторне обладнання | Відповідальні особи | Перегляд

Додати Кабінет

№ Кабінету	Призначення	Кафедра	Площа	Навчальні місця	ПК	Обладнання	Програмне забезпечення	Відповідальна особа	Фото аудиторії
119	Навчальна аудиторія	Управління безпеністю та якістю	73	55	0			Іванов Іван Іванович	
177	Комп'ютерний клас	Інженерно-технічних дисциплін	68	25	25		Autocad	Петренко Іван Федорович	
222	Навчальна лабораторія	Інженерно-технічних дисциплін	68	30	0	Інтерактивна дошка	Excel 2007		

Експорт Друж

Рис. 4.2 Вікно перегляду інформації про аудиторії (адміністративна частина)

Вікно редагування аудиторної інформації в адміністративній частині виглядає наступним чином:

Кабінети | Кафедри | Призначення | Програмне забезпечення | Обладнання | Аудиторне обладнання | Відповідальні особи | Перегляд

Редагувати Кабінет

№ Кабінету : 177

Призначення : Комп'ютерний клас X

Кафедра : Інженерно-технічних дисциплін X

Площа : 68

Навчальні місця : 25

ПК : 25

Обладнання : Вибрати Обладнання

Програмне забезпечення : Autocad X Excel 2007 X

Відповідальна особа : Петренко Іван Федорович X

Фото аудиторії : Видалити

Зберегти зміни Зберегти зміни та повернутися до списку Скасувати

Рис. 4.3 Вікно редагування інформації про аудиторію

Вікно перегляду обладнання в адміністративній частині виглядає наступним чином:

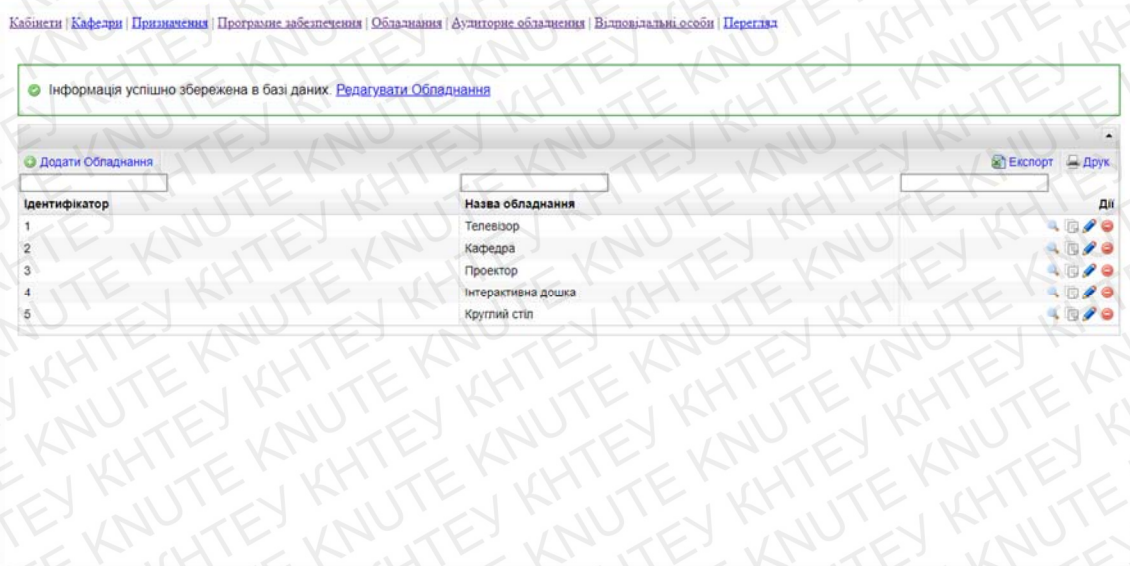


Рис. 4.4 Вікно перегляду обладнання

Вікно перегляду програмного забезпечення в адміністративній частині виглядає наступним чином:

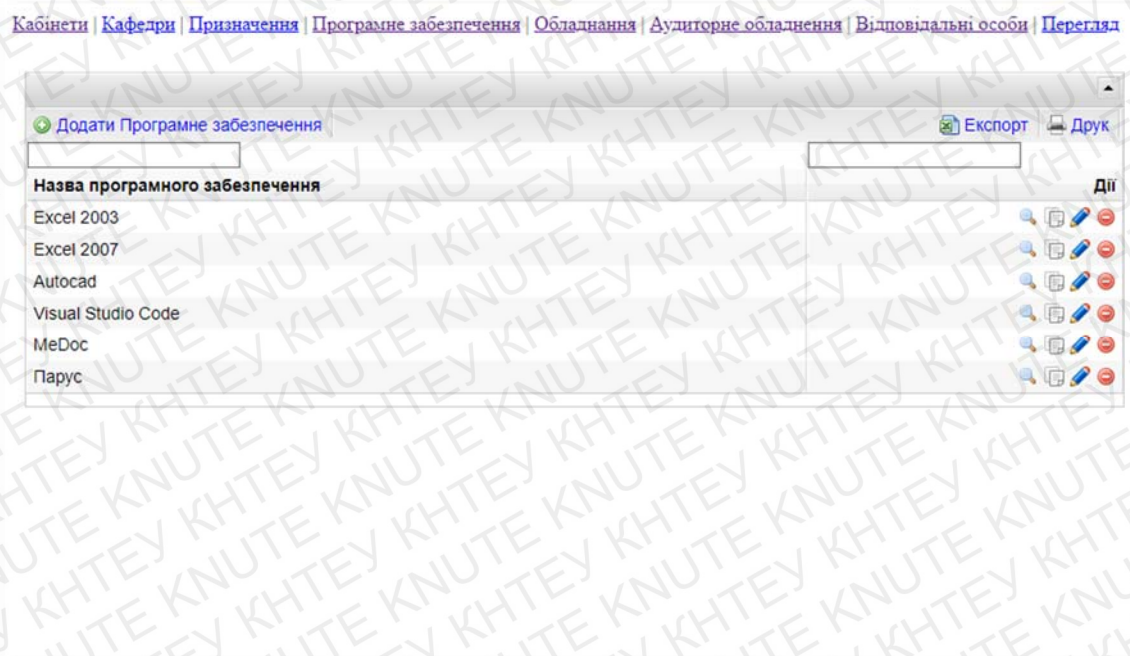


Рис. 4.5 Вікно перегляду програмного забезпечення

Вікно перегляду кафедр в адміністративній частині виглядає наступним чином:

[Кабінети](#) | [Кафедри](#) | [Призначення](#) | [Програмне забезпечення](#) | [Обладнання](#) | [Аудиторне обладнання](#) | [Відповідальні особи](#) | [Перегляд](#)

✓ Інформація успішно збережена в базі даних. [Редагувати Кафедра](#)

[Додати Кафедра](#) [Експорт](#) [Друк](#)

Назва кафедри	Дії
Товарознавства, управління безпекою та якістю	
Іноземної філології та перекладу	
Банківської справи	
Фінансів	
Інженерно-технічних дисциплін	
Менеджменту	
Статистики та економетрії	

Рис. 4.6 Вікно перегляду кафедр

Вікно перегляду призначень а в адміністративній частині виглядає наступним чином:

[Кабінети](#) | [Кафедри](#) | [Призначення](#) | [Програмне забезпечення](#) | [Обладнання](#) | [Аудиторне обладнання](#) | [Відповідальні особи](#) | [Перегляд](#)

[Додати Призначення](#) [Експорт](#) [Друк](#)

Призначення	Дії
Навчальна аудиторія	
Комп'ютерний клас	
Навчальна аудиторія	

Рис. 4.7 Вікно перегляду призначень аудиторії

Вікно перегляду відповідальних осіб а в адміністративній частині виглядає наступним чином:

[Кабінети](#) | [Кафедри](#) | [Призначення](#) | [Програмне забезпечення](#) | [Обладнання](#) | [Аудиторне обладнання](#) | [Відповідальні особи](#) | [Перегляд](#)

✓ Інформація успішно збережена в базі даних. [Редагувати Відповідальна особа](#)

➕ Додати Відповідальна особа [Експорт](#) [Друк](#)

ПІБ	Внутрішній телефон	Мобільний телефон	Електронна адреса	1й номер кабінету	2й номер кабінету	Дії
Іванов Іван Іванович	100	(068)888-88-88	test@knute.edu.ua	101	102	 

Рис. 4.8 Вікно перегляду відповідальних осіб
(адміністративна частина)

Вікно редагування інформації про відповідальних осіб в адміністративній частині виглядає наступним чином:

[Кабінети](#) | [Кафедри](#) | [Призначення](#) | [Програмне забезпечення](#) | [Обладнання](#) | [Аудиторне обладнання](#) | [Відповідальні особи](#) | [Перегляд](#)

Редагувати Відповідальна особа

ПІБ :

Внутрішній телефон :

Мобільний телефон :

Електронна адреса :

1й номер кабінету :

2й номер кабінету :

✓ Інформація успішно оновлена. [Повернутися до списку](#)

Рис. 4.9 Вікно редагування відповідальної особи

Вікно перегляду аудиторного обладнання в адміністративній частині виглядає наступним чином:

№ Кабінету	Обладнання	Додаткова інформація	Дії
2	Телевізор	50"	
2	Кафедра		

Рис. 4.10 Вікно перегляду аудиторного обладнання

З попереднього рисунка можна побачити, що на сторінці відсутні кнопки для додавання та видалення записів з таблички, а наявна лише кнопки перегляду та редагування. Як було описано раніше це зроблено, щоб була можливість введення додаткового опису обладнання встановленого в аудиторії. На рисунку можна побачити, що для телевізору вказана додаткова характеристика, яка відображає діагональ екрану.

4.2 Створення користувацької частини

Аналогічно створимо для адміністративної частини окреме відображення **views.php**. та контролер **view.php** (модель представлена Grocery CRUD).

На відміну від адміністративної частини, користувацька частина не має можливості користувачу робити будь-які операції із записами в середині бази даних, окрім перегляду та фільтрації.

Для перегляду в користувацькій частині доступні до перегляду лише аудиторії та відповідальні особи. Окрім того наявне посилання на адміністративну частину. Відповідальних осіб додано як довідник через неможливість об'єднання інформації з двох таблиць в одній.

[Аудиторії](#) | [Відповідальні особи](#) | [Регістрація](#)

№ Кабінету	Призначення	Кафедра	Площа	Начальні місця	ПК	Відповідальна особа	Фото аудиторії	Програмне забезпечення	Обладнання
119	Начальна аудиторія	Управління безпеністю та якістю	73	55	0	Іванов Іван Іванович			
177	Комп'ютерний клас	Інженерно-технічних дисциплін	66	25	25	Петренко Іван Федорович		Autocad, Excel 2007	
222	Начальна лабораторія	Інженерно-технічних дисциплін	66	30	0				Інтерактив

[Експорт](#) [Друк](#)

Рис. 4.11 Вікно перегляду інформації про аудиторії (користувацька частина)

З рисунка можна побачити, що на сторінці відсутні кнопки для операцій із записом та наявна лише панель фільтрації. Аналогічним чином виглядає сторінка перегляду інформації про відповідальних осіб:

[Кабінети](#) | [Відповідальні особи](#) | [Регістрація](#)

ПІБ	Внутрішній телефон	Мобільний телефон	Електронна адреса	1й номер кабінету	2й номер кабінету	Дія
Іванов Іван Іванович	100	(068)888-88-88	test@knu-te.edu.ua	101	102	

[Експорт](#) [Друк](#)

Рис. 4.12 Вікно перегляду відповідальних осіб (користувацька частина)

Засоби CodeIgniter та Grocery CRUD надають можливість виводу зображення з сервера по збереженому у таблиці посиланні. Виглядає це наступним чином:



Рис. 4.13 Перегляд фотографії аудиторії

Окрім того, інтерфейс користувача дозволяє друкувати дані та експортувати в ексел файл наступному вигляді:

The screenshot shows an Excel spreadsheet with the following data:

№ Кабінету	Призначення	Кафедра	Площа	Навчальні місця	ПК	Відповідальна особа	Фото аудиторії	Програмне забезпечення	Обладнання
119	Навчальна аудиторія	Управління безпеністю та якістю	73	55	0	Іванів Іван Іванович			
177	Комп'ютерний клас	Інженерно-технічних дисциплін	68	25	25	Петренко Іван Федорович		Autocad, Excel 2007	
222	Навчальна лабораторія	Інженерно-технічних дисциплін	68	30	0				Інтерактивна дошка, Проектор

Рис. 4.14 Експортовані в ексел файл дані

4.3 Висновки до розділу 4

Використання програмного каркасу CodeIgniter та бібліотеки для створення CRUD інтерфейсів Grocery CRUD, що розширює можливості CodeIgniter, дозволяє створити користувацькі інтерфейси для редагування та перегляду записів в базі даних створивши лише декілька невеликих файлів.

Окрім того бібліотека Grocery CRUD дозволяє легко налаштовувати зв'язки «один до багатьох» та «багато до багатьох» використовуючи третю табличку для зберігання зв'язку між пов'язаними таблицями.

Такі можливості надають змогу легко додавати нові таблички до бази даних та встановлювати зв'язки з вже існуючими табличками.

					<i>КНТЕУ 121-05-25.MP</i>	Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

При написанні даної випускної кваліфікаційної роботи, було проведено аналіз програмних засобів для роботи з базами даних. Метою аналізу було знаходження “легких” в роботі та освоєні програмістом програмних засобів. Таким засобом виявився програмний каркас CodeIgniter та бібліотека для створення CRUD інтерфейсів Grocery CRUD, що розширює можливості CodeIgniter.

Створена база даних аудиторного фонду КНТЕУ (корпус А) керується СКБД MySQL. Взаємодія між веб-браузером, програмним каркасом CodeIgniter та СКБД MySQL відбувається за допомогою веб-сервісу.

Також, було досліджено, як працює університет всередині, та які процеси налагодженні між його відділами. Чого не вистачає, та який функціонал потребується у веб-інтерфейсі бази даних.

На даному етапі, база даних та веб-інтерфейс доступу до бази знаходиться в розробці, і в роботі представлено частину можливого функціоналу, який можна додати.

Оскільки обрані засоби для розробки база даних аудиторного фонду КНТЕУ (корпус А) дозволяють легко вводити нові частини в систему, то при необхідності є можливість додавання нових сутностей в базу даних та їх зв'язування з уже існуючими.

Планується розробка системи авторизації, яка дозволить користуватися адміністративною частиною лише особам, що мають право на редагування інформації про аудиторний фонд.

					КНТЕУ 121-05-25.МР			
					Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		ВП	38	40
Зав. каф.	Криворучко О.В.				Висновки та пропозиції	Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група		
Керівник	Демедюк С.В.							
Гарант	Криворучко О.В.							
Розроб.	Яремич В.Р.							

У майбутньому можливе розгортання бази даних та веб-інтерфейсу керування на сервері університету, та надання доступу до бази працівникам навчального відділу.

Розраховано, що розробка повинна оптимізувати розподіл аудиторного фонду університету та прискорити пошук аудиторій з певними критеріями.

					<i>КНТЕУ 121-05-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		39

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Болюбаш Я.Я. Організація навчального процесу у вищих закладах освіти: Навч. посібник для слухачів закладів підвищення кваліфікації системи вищої освіти. – К.: ВВП “КОМПАС”, 1997. – 64 с.
2. Веллинг Люк, Томсон, Лора. Разработка Web-приложений с помощью PHP и MySQL / Л.Веллинг, Л. Томсон. – 4-е изд. – М. : Вильямс, 2010. – 837 с.
3. Бьюли А. Изучаем SQL / А. Бьюли – СПб. : Символ-Плюс, 2007. – 312 с.
4. Дмитрий Котеров, Алексей Костарев PHP. В подлиннике. / Д. Котеров, А.Костарев – СПб.: «БХВ-Петербург», 2005. – 1120 с.
5. Хольцнер С. PHP в примерах / Стивен Холцнер; пер. с англ. – М.: ООО «Бином-пресс», 2007. – 352 с.
6. Ульман Л. Основы программирования на PHP / Л. Ульман; пер. с англ. – М.: ДМК Пресс, 2001. – 288 с.
7. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Николай Прохоренок – 3-е изд – СПб.: БХВ-Петербург, 2010. – 900 с.
8. Криворучко О.В. Сучасні тенденції інформатизації освітнього процесу / О.В. Криворучко, М.А. Костюк // Science and Education a new Dimension. Natural and Technical Sciences, VI(20), Issu: 172, Budapest 2018. – 68 с

					<i>КНТЕУ 121-05-25.МР</i>			
					<i>Розробка бази даних аудиторного фонду закладу вищої освіти (на прикладі КНТЕУ, корпус А)</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>СД</i>	<i>40</i>	<i>41</i>
<i>Зав. каф.</i>	<i>Криворучко О.В.</i>							
<i>Керівник</i>	<i>Демедюк С.В.</i>							
<i>Гарант</i>	<i>Криворучко О.В.</i>				<i>Список використаних джерел</i>	<i>Факультет обліку, аудиту та інформаційних систем, 2м курс, 5 група</i>		
<i>Розроб.</i>	<i>Яремич В.Р.</i>							

9. Офіційний сайт бібліотеки TableFilter. [Електронний ресурс]. — Режим доступу: <https://www.tablefilter.com>
10. Офіційний сайт програмного каркасу CodeIgniter. [Електронний ресурс]. — Режим доступу: <https://codeigniter.com>
11. Офіційний сайт бібліотеки Grocery CRUD. [Електронний ресурс]. — Режим доступу: <https://www.grocerycrud.com>
12. Флэнаган Д. JavaScript. Подробное руководство, 6 е издание. — Пер. с англ. — СПб: Символ Плюс, 2012. — 1080 с.
13. Яремич В.Р. Огляд функціональних можливостей бази даних аудиторного фонду КНТЕУ (корпус А) // Software engineering комп'ютерних систем — Київ : Київ. нац торг.-екон. ун-т, 2019. — 135 с.

					<i>КНТЕУ 121-05-25.МР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		41

ДОДАТКИ

Додаток А

Програма та методика тестування

Для тестування даного проекту, була обрана методика «Manual testing», тобто метод ручного тестування. Цей спосіб не потребує додаткової розробки та спеціальних навиків, проте дає хороші результати.

Для тестування веб-інтерфейсу керування додатку, були використанні наступні види тестування:

1. Функціональне тестування.
2. Регресивне тестування.

Під час функціонального тестування були перевірені всі функції веб-інтерфейсу керування базою даних, таких як вставка нових записів, видалення та редагування існуючих записів, щоб бути упевненими у тому, що вони працюють коректно.

Після того, як відбувалися глобальні зміни в коді веб-інтерфейсу керування базою даних або в структурі бази даних, проводилося повторне регресивне тестування всіх функцій та компонентів, щоб забезпечитися в тому, що після змін та виправлення помилок, функціональність працюватиме як було заплановано.

Щоб забезпечити якість проекту, були розроблені тест кейси, які покривають весь функціонал додатку. Завдяки їм, навіть людина без досвіду в тестуванні, може слідувати написаним тест кейсам, та провести перевірку всього функціоналу, після внесення змін при розробці.

Тест кейс має наступні складові:

1. Порядковий номер тест кейсу
2. Описання тестового кейсу
3. Кроки виконання

4. Отриманий результат
5. Очікуваний результат

Тест кейси

1. ТК_01
2. Перевірка завантаження веб-інтерфейсу керування базою даних
3. Кроки виконання:
 - 3.1. Перейти за посиланням до веб-інтерфейсу.
 - 3.2. Відкрити сторінку будь-якої сутності.
 - 3.3. Повторити для кожної сутності.
4. Згенерувалася табличка з CRUD інтерфейсом обраної сутності.
5. При переході на сторінку будь-якої сутності повинна згенеруватися відповідна табличка з CRUD інтерфейсом.
1. ТК_02
2. Перевірка функцій додавання нових записів
3. Кроки виконання:
 - 3.1. Перейти за посиланням до веб-інтерфейсу.
 - 3.2. Відкрити сторінку будь-якої сутності.
 - 3.3. Натиснути на кнопку «Додати <Назва сутності>».
 - 3.4. Ввести дані.
 - 3.5. Натиснути кнопку «Зберегти та повернутися до списку».
 - 3.6. Повторити для кожної сутності.
4. На сторінці сутності в табличці сформувався новий запис.
5. Після додавання нового запису сутності на відповідній сторінці повинен сформуватися новий запис.
1. ТК_03
2. Перевірка функцій редагування записів
3. Кроки виконання:

- 3.1. Перейти за посиланням до веб-інтерфейсу.
- 3.2. Відкрити сторінку будь-якої сутності.
- 3.3. Натиснути на кнопку «Редагувати».
- 3.4. Змінити дані.
- 3.5. Натиснути кнопку «Зберегти та повернутися до списку».
- 3.6. Повторити для кожної сутності.
4. На сторінці сутності в таблиці було оновлено запис.
5. Після редагування запису сутності на відповідній сторінці повинен оновитися редагований запис.

1. ТК_04
2. Перевірка функцій видалення записів
3. Кроки виконання:
 - 3.1. Перейти за посиланням до веб-інтерфейсу.
 - 3.2. Відкрити сторінку будь-якої сутності.
 - 3.3. Натиснути на кнопку «Видалити».
 - 3.4. Повторити для кожної сутності.
4. Запис відповідної сутності видалився і не відображається на сторінці в таблиці.
5. Після видалення відповідний запис сутності повинен зникнути з таблиці.

Керівництво користувача

Для того, щоб перейти до веб-інтерфейсу керування базою даних аудиторного фонду КНТЕУ (корпус А), необхідно перейти по відповідному посиланню, використовуючи будь-який сучасний браузер. Після того можна перейти за наступними посиланнями:

- Кабінети.
- Відповідальні особи.
- Редагування.

При переході за посиланням «Кабінети» буде відкрито вікно перегляду інформації про наявні аудиторії, відповідно для посилання «Відповідальні особи» відкриється вікно перегляду інформації про відповідальних осіб.

На цих сторінках для користувача доступні функції перегляду, пошуку, друку та експорту даних в Excel.

Для експорту даних в Excel користувачу необхідно натиснути кнопку «Експорт», а для друку — «Друк».

Для фільтрацій даних користувачеві необхідно ввести пошукове значення у комірку що представлена над кожним стовпчиком таблиці. Для фільтрації користувачеві доступні наступні оператори:

Таблиця Б.1.

ОПЕРАТОРИ ФІЛЬТРАЦІЇ

Оператор	Опис	Приклад
<	Значення менші ніж критерій пошуку	<20
<=	Відповідні або менші значення	<=20
>	Значення більші ніж критерій пошуку	>20
>=	Відповідні або більші значення	>=20

=	Значення точно відповідає критерію пошуку	=20
---	---	-----

Продовження таблиці Б.1.

*	Значення містить критерій пошуку	*Моніт
!	Значення не містить критерій пошуку	!Фінансів
{	Значення починається з критерію пошуку	{М
}	Значення закінчується критерієм пошуку	}а
	Значення містить що найменше один із шуканих критеріїв	Телевізор Проектор
&&	Значення відповідає двом критеріям пошуку	>20 && <30
[empty]	Значення порожнє	[empty]
[nonempty]	Значення не порожнє	[nonempty]
rgx:	Пошук з використанням регулярного виразу	rgx:top\$

Для можливості редагування, користувачеві необхідно перейти за посиланням «Редагування». Нове вікно буде містити наступні посилання:

- Кабінети.
- Кафедри.
- Призначення.
- Програмне забезпечення.
- Обладнання.
- Аудиторне обладнання.
- Відповідальні особи.
- Перегляд.

Кожне посилання веде до сторінки редагування відповідної сутності. Посилання «Перегляд» повертає користувача до інтерфейсу перегляду аудиторій без можливості операцій над записами.

Кожна сторінка відповідної сутності надає користувачеві можливість додавати нові записи натисканням кнопки «Додати <Назва сутності>», редагувати — кнопка «Редагування» та видаляти — кнопка «Видалити».

ТЕКСТ ПРОГРАМИ

Лістинг файлу manage.php

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Manage extends CI_Controller {

    public function __construct()
    {
        parent::__construct();

        $this->load->database();
        $this->load->helper('url');

        $this->load->library('grocery_CRUD');
    }

    public function _example_output($output = null)
    {
        $this->load->view('management.php',(array)$output);
    }

    public function index()
    {
        $this->_example_output((object)array('output' => " , 'js_files' => array() , 'css_files' =>
array()););
    }

    public function auditorium_management()
    {
        $scrud = new grocery_CRUD();

        $scrud->set_table('auditorium');

        $scrud->set_relation('FunctionId','function','Denotation');
        $scrud->set_relation('DepartmentId','department','Denotation');
        $scrud->set_relation('ResponsiblePersonId','responsibleperson','FullName');

        $scrud->set_field_upload('PictureUrl','assets/uploads/files');

        $scrud->set_relation_n_n('Softwares', 'auditoriumsoftwares', 'software', 'AuditoriumId',
'SoftwareId', 'Denotation');
        $scrud->set_relation_n_n('FeaturesView', 'auditoriumfeatures', 'feature', 'AuditoriumId',
'FeatureId', '{Denotation} {AdditionalInfo}');
        $scrud->set_relation_n_n('Features', 'auditoriumfeatures', 'feature', 'AuditoriumId', 'FeatureId',
'Denotation');

        $scrud->display_as('Id','№ Кабінету');
        $scrud->display_as('FunctionId','Призначення');
        $scrud->display_as('DepartmentId','Кафедра');
        $scrud->display_as('Area','Площа');
        $scrud->display_as('StudyPlaces','Навчальні місця');
    }
}

```

```

$crud->display_as('Computers','ПК');
$crud->display_as('Features','Обладнання');
$crud->display_as('FeaturesView','Обладнання');
$crud->display_as('Softwares','Програмне забезпечення');
$crud->display_as('ResponsiblePersonId','Відповідальна особа');
$crud->display_as('PictureUrl','Фото аудиторії');

$crud->columns('Id','FunctionId','DepartmentId','Area','StudyPlaces','Computers',
'FeaturesView','Softwares','ResponsiblePersonId','PictureUrl');
$crud->fields('Id','FunctionId','DepartmentId','Area','StudyPlaces','Computers',
'Features','Softwares','ResponsiblePersonId','PictureUrl');

$crud->set_subject('Кабінет');

$output = $crud->render();
$this->_example_output($output);
}

public function software_management()
{
    $crud = new grocery_CRUD();

    $crud->set_table('software');
    $crud->set_subject('Програмне забезпечення');

    $crud->display_as('Denotation','Назва програмного забезпечення');

    $output = $crud->render();
    $this->_example_output($output);
}

public function feature_management()
{
    $crud = new grocery_CRUD();

    $crud->set_table('feature');
    $crud->set_subject('Обладнання');

    $crud->columns('Id','Denotation');
    $crud->display_as('Id','Ідентифікатор');
    $crud->display_as('Denotation','Назва обладнання');

    $output = $crud->render();
    $this->_example_output($output);
}

public function auditoriumfeatures_management()
{
    $crud = new grocery_CRUD();

    $crud->set_table('auditoriumfeatures');
    $crud->set_subject('Аудиторне обладнання');

    $crud->unset_add();
    $crud->unset_clone();
    $crud->unset_delete();
}

```

```

$crud->columns('AuditoriumId','FeatureId','AdditionalInfo');
$crud->fields('AuditoriumId','FeatureId','AdditionalInfo');

$crud->display_as('AuditoriumId','№ Кабінету');
$crud->display_as('FeatureId','Ідентифікатор обладнання');
$crud->display_as('AdditionalInfo','Додаткова інформація');

$output = $crud->render();
$this->_example_output($output);
}

public function department_management()
{
    $crud = new grocery_CRUD();

    $crud->set_table('department');
    $crud->set_subject('Кафедра');
    $crud->display_as('Denotation','Назва кафедри');

    $output = $crud->render();
    $this->_example_output($output);
}

public function function_management()
{
    $crud = new grocery_CRUD();

    $crud->set_table('function');
    $crud->set_subject('Призначення');
    $crud->display_as('Denotation','Призначення');

    $output = $crud->render();
    $this->_example_output($output);
}

public function responsibleperson_management()
{
    $crud = new grocery_CRUD();

    $crud->set_table('responsibleperson');
    $crud->set_subject('Відповідальна особа');

    $crud->display_as('FullName','ПІБ');
    $crud->display_as('InternalPhoneNumber','Внутрішній телефон');
    $crud->display_as('MobilePhoneNumber','Мобільний телефон');
    $crud->display_as('EmailAdress','Електронна адреса');
    $crud->display_as('CabinetNumber1','1й номер кабінету');
    $crud->display_as('CabinetNumber2','2й номер кабінету');

    $output = $crud->render();
    $this->_example_output($output);
}
}

```

Лістинг файлу view.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class View extends CI_Controller {

    public function __construct()
    {
        parent::__construct();

        $this->load->database();
        $this->load->helper('url');

        $this->load->library('grocery_CRUD');
    }

    public function _example_output($output = null)
    {
        $this->load->view('views.php',(array)$output);
    }

    public function index()
    {
        $this->_example_output((object)array('output' => " , 'js_files' => array() , 'css_files' =>
array()));
    }

    public function auditorium_view()
    {
        $crud = new grocery_CRUD();

        $crud->set_table('auditorium');
        $crud->set_subject('Кабінет');

        $crud->unset_add();
        $crud->unset_clone();
        $crud->unset_edit();
        $crud->unset_delete();

        $crud->set_relation('FunctionId','function','Denotation');
        $crud->set_relation('DepartmentId','department','Denotation');
        $crud->set_relation('ResponsiblePersonId','responsibleperson','FullName');

        $crud->set_field_upload('PictureUrl','assets/uploads/files');

        $crud->set_relation_n_n('Softwares', 'auditoriumsoftwares', 'software', 'AuditoriumId',
'SoftwareId', 'Denotation');
        $crud->set_relation_n_n('FeaturesView', 'auditoriumfeatures', 'feature', 'AuditoriumId',
'FeatureId', '{Denotation} {AdditionalInfo}');
```

```

$crud->display_as('Id','№ Кабінету');
$crud->display_as('FunctionId','Призначення');
$crud->display_as('DepartmentId','Кафедра');
$crud->display_as('Area','Площа');
$crud->display_as('StudyPlaces','Навчальні місця');
$crud->display_as('Computers','ПК');
$crud->display_as('Features','Обладнання');
$crud->display_as('FeaturesView','Обладнання');
$crud->display_as('Softwares','Програмне забезпечення');
$crud->display_as('ResponsiblePersonId','Відповідальна особа');
$crud->display_as('PictureUrl','Фото аудиторії');

$crud->fields('Id','FunctionId','DepartmentId','Area','StudyPlaces','Computers','Features',
'Softwares','ResponsiblePersonId','PictureUrl');

$output = $crud->render();

$this->_example_output($output);
}

public function responsibleperson_view()
{
    $crud = new grocery_CRUD();

    $crud->set_table('responsibleperson');
    $crud->set_subject('Відповідальна особа');

    $crud->display_as('FullName','ПІБ');
    $crud->display_as('InternalPhoneNumber','Внутрішній телефон');
    $crud->display_as('MobilePhoneNumber','Мобільний телефон');
    $crud->display_as('EmailAdress','Електронна адреса');
    $crud->display_as('CabinetNumber1','1й номер кабінету');
    $crud->display_as('CabinetNumber2','2й номер кабінету');

    $output = $crud->render();

    $this->_example_output($output);
}
}

```

Лістинг файлу management.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
<?php
foreach($scss_files as $file): ?>
    <link type="text/css" rel="stylesheet" href="<?php echo $file; ?>" />
<?php endforeach; ?>
</head>

<body>
    <div>
        <a href='<?php echo site_url('Manage/auditorium_management')?>'>Кабінети</a> |
        <a href='<?php echo site_url('Manage/department_management')?>'>Кафедри</a> |
        <a href='<?php echo site_url('Manage/function_management')?>'>Призначення</a> |
        <a href='<?php echo site_url('Manage/software_management')?>'>Програмне
забезпечення</a> |
        <a href='<?php echo site_url('Manage/feature_management')?>'>Обладнання</a> |
        <a href='<?php echo site_url('Manage/auditoriumfeatures_management')?>'>Аудиторне
обладнання</a> |
        <a href='<?php echo site_url('Manage/responsibleperson_management')?>'>Відповідальні
особи</a> |
        <a href='<?php echo site_url('View/')?>'>Перегляд</a>
    </div>
    <div style='height:20px;'></div>
    <div style="padding: 10px">
        <?php echo $output; ?>
    </div>
    <?php foreach($js_files as $file): ?>
        <script src="<?php echo $file; ?>"></script>
    <?php endforeach; ?>
</body>
<script src="/fond/Js/TableFilter/tablefilter.js"></script>
<script data-config>
    var tf = new TableFilter(document.getElementsByTagName('table')[0], {
        base_path: '/fund/Js/TableFilter/'
    });
    tf.init();
</script>
</html>
```

Лістинг файлу views.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" /><meta name="viewport" content="width=device-width, initial-scale=1.0">
    <?php foreach($css_files as $file): ?>
        <link type="text/css" rel="stylesheet" href="<?php echo $file; ?>" />
    <?php endforeach; ?>
</head>

<body>
    <div>
        <a href='<?php echo site_url('View/auditorium_view')?'>'>Аудиторії</a> |
        <a href='<?php echo site_url('View/responsibleperson_view')?'>'>Відповідальні особи</a> |
        <a href='<?php echo site_url('Manage/')?'>'>Редагування</a>
    </div>
    <div style='height:20px;'></div>
    <div style="padding: 10px">
        <?php echo $output; ?>
    </div>
    <?php foreach($js_files as $file): ?>
        <script src="<?php echo $file; ?>"></script>
    <?php endforeach; ?>
</body>
<script src="/fond/Js/TableFilter/tablefilter.js"></script>
<script data-config>
    var tf = new TableFilter(document.getElementsByTagName('table')[0], {
        base_path: '/fund/Js/TableFilter'
    });
    tf.init();
</script>
</html>
```