

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

Кафедра _____ **Комп'ютерних систем та мереж** _____

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
комп'ютерних систем та мереж

_____ (Жуков І.А.)

« ____ » _____ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”
спеціальність 123 «Комп'ютерна інженерія»
Освітньо-професійна програма «Комп'ютерні системи та мережі»

Тема: Інформаційна комп'ютерна система для обліку абітурієнтів

Виконавець: _____ Баденкова А.С.

Керівник: _____ Лукашенко В.В.

Нормоконтролер: _____ Журавель С.В.

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютерних систем та мереж

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

комп'ютерних систем та мереж

_____ (Жуков І.А.)

« ____ » _____ 2023 р.

ЗАВДАННЯ

на виконання дипломного проекту

_____ Баденсковій Аріні Сергіївні

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема проекту (роботи): Інформаційна комп'ютерна система для обліку абітурієнтів

затверджена наказом ректора від " 26 " квітня 2023 року № 591/ст.

2. Термін виконання проекту (роботи): 22.05.2023 р. по 25.06.2023 р.

3. Вихідні дані до проекту (роботи): Інформація про абітурієнтів, такі як ПІБ, факультет, спеціальність, сертифікати, квоти, контактні дані (телефон, електронна пошта) та інші відомості, що стосуються процесу вступу.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):
Аналіз існуючих систем, процес вступу до вищого навчального закладу, системи управління навчання; Вибір засобів розробки, мови та платформи програмування, бази даних, веб-сервери, стек; Розробка програмного продукту, розробка front-end частини системи, створення бази даних

5. Перелік обов'язкового графічного матеріалу:

Презентація PowerPoint

РЕФЕРАТ

Пояснювальна записка до дипломного проекту “ Інформаційна комп'ютерна система для обліку абітурієнтів ”: 48 с., 15 рис., 16 літературних джерел.

ОБЛІК, АБІТУРІЄНТ, АДМІНІСТРУВАННЯ, КОМП'ЮТЕРНА СИСТЕМА.

Мета дипломного проекту – є розробка та реалізація комп'ютерної системи, яка спрощує та автоматизує процес обліку та управління даними абітурієнтів у вищому навчальному закладі.

Об'єкт проектування – інформаційна комп'ютерна система для обліку абітурієнтів.

Предмет проектування – інформаційна комп'ютерна система для обліку абітурієнтів, система служить інструментом для зберігання, обробки та управління даними абітурієнтів, що подають заявки на вступ до вищого навчального закладу.

Метод проектування- розробка сучасної інформаційної комп'ютерної системи

Прогнози припущення щодо розвитку об'єкта дослідження – Зростання популярності та використання онлайн-систем для подачі заявок: З урахуванням технологічного розвитку та зручності для абітурієнтів, ймовірно, буде спостерігатись зростання популярності та використання онлайн-систем для подачі заявок на вступ до вищих навчальних закладів. Це може призвести до збільшення попиту на системи, які забезпечують ефективний облік та управління цим процесом.

Результати створення системи для обліку абітурієнтів можливі в використанні вищих навчальних закладів для покращення та вдосконалення їх роботи

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ.....	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ДЛЯ УНІВЕРСИТЕТІВ.....	8
1.1. Процес вступу до вищого навчального закладу.....	8
1.2. Системи управління навчанням.....	11
1.2.1. Moodle	13
1.2.2. Oracle PeopleSoft	15
1.2.3. Система обліку «Кадри Плюс Україна».....	17
Висновки до розділу	19
РОЗДІЛ 2 ВИБІР ЗАСОБІВ РОЗРОБКИ.....	20
2.1. Засоби розробки	20
2.2. Мови та платформи програмування.....	21
2.3. Бази даних	23
2.4. Веб-сервери.....	24
2.5. Стек.....	25
Висновки до розділу	27
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	29
3.1. Розробка front-end частини системи.....	29
3.2. Створення бази даних	34
3.3. Розробка back-end частини системи.....	37
Висновки до розділу	44
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47

Кафедра КСМ

НАУ 21 38 79 000 ПЗ

<i>Виконав</i>	<i>Іванов І.І.</i>			<i>Тема дипломного проекту</i>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Сидоров А.О.</i>					<i>5</i>	<i>73</i>
<i>Консульт.</i>					<i>123 КС-421Б</i>		
<i>Норм. контр.</i>	<i>Журавель С.В.</i>						
<i>Зав. Каф.</i>	<i>Жуков І.А.</i>						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

<i>ВНЗ</i>	–	<i>вищий навчальний заклад</i>
<i>ЗНО</i>	–	<i>зовнішнє незалежне оцінювання</i>
<i>LMS</i>	–	<i>Система дистанційного навчання</i>
<i>CRM</i>	–	<i>Customer Relationship Management</i>
<i>CMS</i>	–	<i>система управління курсами</i>
<i>LAN</i>	–	<i>локальна мережа (ЛОМ)</i>
<i>VLE</i>	–	<i>віртуальне навчальне середовище</i>
<i>PHP</i>	–	<i>Hypertext Preprocessor</i>
<i>UI</i>	–	<i>користувацький інтерфейс</i>
<i>SQL</i>	–	<i>Structured Query Language (структурована мова запитів)</i>
<i>БД</i>	–	<i>база даних</i>

ВСТУП

У сучасному світі інформаційні технології відіграють важливу роль у всіх сферах життя людини, включаючи освіту. Університети та вищі навчальні заклади часто стикаються з великою кількістю абітурієнтів, які подають заявки на вступ на освітні програми. Ефективний облік та управління цими абітурієнтами є ключовим аспектом успішного функціонування Університету.

З метою покращення процесу реєстрації та полегшення заявникам і керівникам, у цьому документі пропонується розробити комп'ютерну систему для інформації про реєстрацію заявників. Система дозволить заявникам зручно подавати заявки в Інтернеті та надасть адміністраторам інструменти для ефективного керування цим процесом.

Першим кроком у впровадженні системи є аналіз існуючої системи реєстрації абітурієнтів та виявлення проблем і недоліків у цій сфері. Після цього розробляються цілі та завдання дипломної проекту, вибираються методи та технології, розробляється архітектура системи відповідно до потреб університету та абітурієнта.

Метою даного дипломного проекту є розробка комп'ютеризованої системи яка містить інформацію для реєстрації абітурієнтів, яка забезпечить зручне та ефективно оформлення заяв про вступ до ВНЗ. Для досягнення поставленої мети передбачається вирішення таких завдань, як аналіз існуючих систем реєстрації конкурентів, формулювання вимог до системи, вибір методів і технологій, розробка архітектури системи та реалізація необхідних функцій.

Під час аналізу буде розглянуто різні системи, які університети та вищі навчальні заклади використовують для реєстрації абітурієнтів. Буде досліджено їхню функціональність, інтерфейс користувача, архітектуру та продуктивність. Зокрема, розглядатимуться системи обліку студентів, системи реєстрації та оцінювання, системи електронного вступу тощо.

На основі аналізу будуть виділені ключові проблеми та недоліки існуючої системи, такі як відсутність онлайн-подання заявок, незручний інтерфейс та

недостатня кількість функцій. Це дасть змогу визначити вимоги до нової інформаційно-комп'ютерної системи реєстрації абітурієнтів, яка враховуватиме проблеми та потреби ВНЗ та абітурієнта.

Результатом дипломного проекту стане надійна та ефективна комп'ютеризована інформаційна система для реєстрації абітурієнтів, яка покращить процес вступу до університету та полегшить абітурієнтам та роботу співробітникам приймальної комісії.

АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ДЛЯ УНІВЕРСИТЕТІВ

1.1.Процес вступу до вищого навчального закладу

Процес вступу до українських вишів зазвичай складається з кількох етапів.

Основні кроки:

- Вибір університетів і програм:

Спочатку абітурієнт визначає свої інтереси та цілі, а потім вибирає університет, який пропонує програму, яка підходить. Рейтинги дослідницьких університетів, навчальні плани та програми та вимоги до вступу можуть містити сайти університетів.

- Підготовка необхідних документів:

Кожен університет має свої вимоги до документів для вступу. Зазвичай це передбачає заповнення заяви, надання атестата про загальну середню освіту (або його еквівалента), результатів зовнішнього незалежного оцінювання (ЗНО) чи іншого вступного іспиту, а також подання медичної довідки та іншої необхідної документації.

- Заява на ЗНО/вступний іспит:

Вступ до вищих навчальних закладів України зазвичай передбачає здачу ЗНО. ЗНО – це національний іспит, який оцінює знання з різних предметів. Абітурієнт має зареєструватися на ЗНО та підготуватися до іспитів. Деякі університети мають власні вступні іспити та співбесіди.

Кафедра КСУ

НАУ 21 38 79 000 ПЗ

<i>Виконав</i>	<i>Іванов І.І.</i>			<i>Назва роділу 1</i>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Сидоров А.О.</i>					9	60
<i>Консульт.</i>					<i>123 КС-421Б</i>		
<i>Норм. контр.</i>	<i>Журавель С.В.</i>						
<i>Зав. Каф.</i>	<i>Жуков І.А.</i>						

- Проходження вступного конкурсу:

Після завершення професійної підготовки або вступних іспитів абітурієнт бере участь у вступному конкурсі (наприклад, у конкурсі на соціальне забезпечення чи спеціальні квоти), який враховує оцінки та інші критерії.

Університети рахуватимуть конкурсні бали відповідно до квот і кількості заяв.

- Реєстрація та прийом:

Після складання вступних іспитів і отримання рекомендації, необхідно вступити до університету та подати необхідні документи. Зазвичай існує кінцевий термін, до якого ви повинні зареєструватися та підтвердити своє місце

Українські абітурієнти мають змогу зареєструвати електронний кабінет вступника, для подачі заявок в вищі навчальні заклади.

Електронний кабінет вступника є онлайн-платформою, де вступники можуть здійснювати різні дії та отримувати інформацію щодо свого вступу до університету. Такий електронний кабінет забезпечує зручність та доступність для вступників, дозволяючи їм виконувати різні завдання онлайн.

Основні можливості електронного кабінету вступника можуть включати:

1. Реєстрація та заповнення аплікаційної форми

Вступники можуть створити обліковий запис та зареєструватися в електронному кабінеті. Вони можуть заповнити аплікаційну форму, надавши особисті дані, академічну інформацію та інші вимоги.

2. Завантаження документів

Вступники можуть завантажити необхідні документи, такі як свідоцтво про повну загальну середню освіту, результати ЗНО або інші документи, які можуть бути вимагані для реєстрації та вступу. Моніторинг статусу програми:

Учасники можуть переглядати статус свого запиту, включаючи інформацію про схвалення, відхилення або очікування рішення. Абітурієнт може отримувати сповіщення та оновлення на електронну пошту.

3. Зв'язок з університетами

Абітурієнти можуть зв'язатися з працівниками університету через електронний кабінет. Ставити запитання, отримувати відповіді та дізнавайтись більше про процес вступу.

4. Отримання інформації

Електронний кабінет надає доступ до розкладів вступних випробувань, дат вступної кампанії, інформації приймальної комісії, правил прийому та іншої корисної інформації. Електронний кабінет абітурієнта спрощує процес вступу та забезпечує зручний спосіб зв'язку абітурієнта з університетом. Це дозволяє абітурієнтам легко виконувати необхідні завдання, отримувати інформацію та контролювати процес вступу до університету.

Після зарахування до вищого навчального закладу, абітурієнт має надати оригінали документів до приймальної комісії, зазвичай це відбувається за місяць до початку навчального року. На сьогоднішній день не всі абітурієнти мають змогу подати документи вчасно. Для прикладу, абітурієнт який буде навчатись в іншому місті не має змоги подати оригінали документів в зазначений термін. Виникає потреба в створенні комп'ютерної системи окремих університетів для полегшення абітурієнту вступу та покращення роботи приймальної комісії.

Комп'ютерна система має містити змогу абітурієнта завантажити необхідні документи, зазначені вищим навчальним закладом, а приймальній комісії доступ до цих документів. А самі оригінали документів, абітурієнт зможе надати під час початку навчання, що полегшує та покращує роботу приймальної комісії, та облегшує вступ для абітурієнта.

1.2 Системи управління навчанням

Система управління навчанням, також система дистанційного навчання (Learning management system) — система управління навчальною діяльністю, яка використовується для розробки, управління та поширення навчальних онлайнматеріалів із забезпеченням спільного доступу [1].

Сучасна LMS відрізняється зручним і гнучким інтерфейсом, відмінною функціональністю і дозволяє вивести дистанційне навчання на якісно новий рівень. Системи дистанційного навчання (learning management systems) дозволяють

організувати навчальний процес «з нуля», створювати онлайн-курси та віртуальні класи, які доступні в будь-який час і в будь-якій точці світу з підключенням до Інтернету. При цьому всі навчальні матеріали зберігаються в одному місці та можуть легко коригуватися та переглядатися відповідно до цілей навчання та сфер діяльності організації. LMS часто порівнюють із віртуальними класами, де студентів можна навчати з будь-якої точки світу та відслідковувати їх успішність. Вони можуть містити модулі для підтримки процесу прийому абітурієнтів, такі як реєстрація, подача заявок та взаємодія зі заявниками.

- Системи керування прийомом (Admission Management Systems): Ці системи спеціально розроблені для управління процесом прийому абітурієнтів. Вони надають інструменти для подачі заявок, оцінки, відстеження статусу заявок та комунікації зі заявниками. Приклади таких систем включають TargetX, Ellucian Recruiter, Hobsons ApplyYourself.
- Системи управління студентськими записами (Student Information Systems): Ці системи забезпечують зберігання та обробку основної інформації про студентів, включаючи абітурієнтів. Вони можуть містити модулі для управління прийомом, реєстрації на курси, фінансового обліку та інших аспектів студентського життя. Приклади таких систем включають Banner by Ellucian, PeopleSoft Campus Solutions, Jenzabar.
- Системи електронного документообігу: Ці системи дозволяють університетам керувати документами, пов'язаними з прийомом абітурієнтів. Вони дозволяють зберігати, обробляти, відстежувати та обмінюватися документами, такими як заявки, рекомендаційні листи, академічні документи тощо.
- CRM-системи (Customer Relationship Management): Деякі університети використовують CRM-системи для управління взаємодією зі заявниками та управління відносинами з потенційними студентами. Ці системи дозволяють збирати, аналізувати та використовувати дані про заявників для покращення процесу прийому та персоналізованої комунікації.

Такі системи спеціалізуються на освітніх або бізнес-звітних процесах і можуть не відображати всю необхідну інформацію про студентів і співробітників університету. Тому облік університетів та їх структурних підрозділів зазвичай

ведеться за допомогою офісних програм, паперових або спеціальних програмних рішень. Розглянемо програмне рішення, яке може виконувати облік для університету.

1.2.1 Moodle

Moodle (Modular Object-Oriented Dynamic Learning Environment, вимовляється «Мудл») - це модульне об'єктно-орієнтоване динамічне навчальне середовище, яке називають також системою управління навчанням (LMS), системою управління курсами (CMS), віртуальним навчальним середовищем (VLE) або просто платформою для навчання, яка надає викладачам, учням та адміністраторам дуже розвинутий набір інструментів для комп'ютеризованого навчання, в тому числі дистанційного. Moodle можна використовувати в навчанні школярів, студентів, при підвищенні кваліфікації, бізнес-навчанні, як в комп'ютерних класах навчального закладу, так і для самостійної роботи вдома, головну сторінку можемо побачити на Рис.1[2].

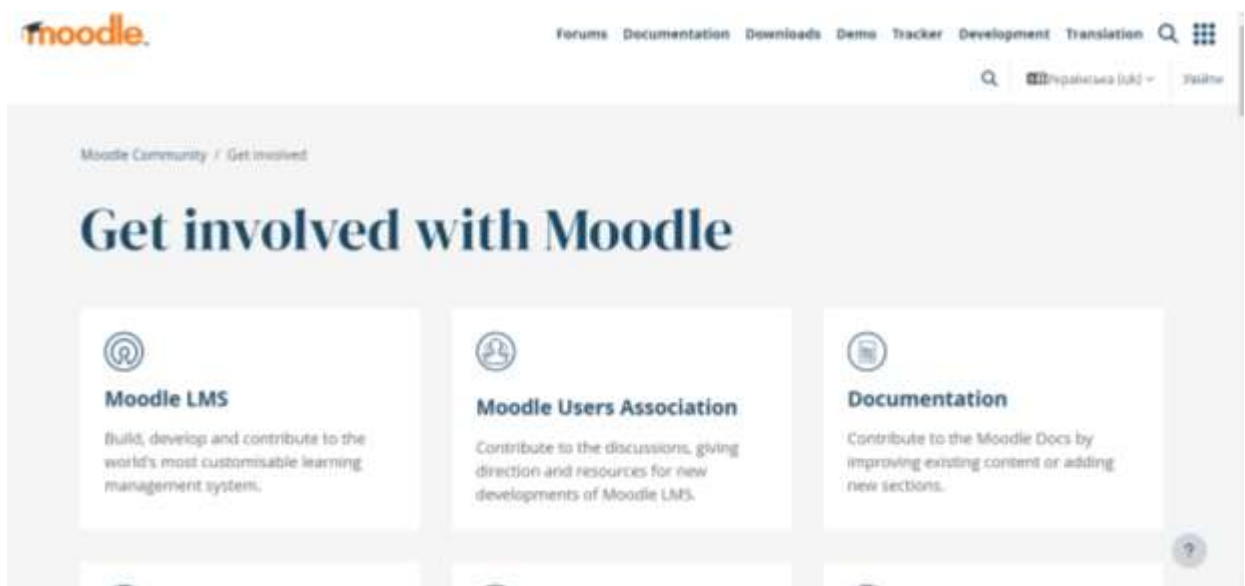


Рис.1.1-головна сторінка

Moodle — це безкоштовна та відкрита система. Він не лише безкоштовний сам по собі, але й не потребує платного програмного забезпечення для роботи. Це означає, що будь-яка установа може мати безкоштовну, повну та повністю ліцензовану систему, не витрачаючи ні копійки на придбання програмного забезпечення.

Moodle дозволяє викладачам створювати та організовувати курси, використовуючи різні типи матеріалів, включаючи текстові файли, відео, аудіо, зображення, презентації тощо. Це гарантує, що студенти завжди мають доступ до навчальних матеріалів, які їм підходять. Платформа має вбудовані засоби спілкування, такі як форуми, чат і відеоконференції. Це сприяє жвавому обміну інформацією та співпраці між студентами та викладачами. Великою перевагою є те що, Moodle дозволяє вчителям відстежувати діяльність учнів, прогрес навчання та результати оцінювання. Учні також можуть перевірити свій прогрес і результати.

Щодо розширень та плагінів то, Moodle підтримує широкий спектр розширень і плагінів, які можуть розширити функціональність платформи. Це дозволяє адаптувати Moodle до конкретних потреб і вимог.

Серед переваг можна зазначити гнучкість платформи: Moodle пропонує гнучкі параметри налаштування, які дозволяють адаптувати систему реєстрації студентів до конкретних потреб вашого університету. Інтерактивність: Moodle має вбудовані інструменти для взаємодії учень-викладач. В. Форуми, чати та спільні завдання.

Щодо недоліків системи можна зазначити складність системи, а саме: встановлення та налаштування Moodle може вимагати певного рівня технічних знань. Залежність від ресурсів сервера: щоб забезпечити швидкий і надійний доступ для студентів і викладачів, використання Moodle для систем реєстрації студентів вимагає потужних серверів і достатньої пропускної здатності мережі. Підтримка та оновлення: підтримка та оновлення Moodle може зайняти час і зусилля, особливо якщо потрібні спеціальні функції або виправлення помилок.

Однак ця система не дозволяє створювати та редагувати файли даних в режимі онлайн. Тому нам потрібно розробити новий застосунок, який може створювати та редагувати файли в режимі онлайн

1.2.2 Oracle PeopleSoft

Oracle PeopleSoft є комплексним програмним рішенням, спеціально розробленим для управління персоналом, фінансами та студентським обліком у вищих навчальних закладах. Ця система надає рішення для різних аспектів управління університетом,

Управління студентами: Oracle PeopleSoft надає інструменти для управління реєстрацією студентів, планування розкладу занять, контролю академічного прогресу, оцінювання та зарахування кредитів. Вона також дозволяє студентам переглядати свої академічні дані та графік занять через онлайн-портал.

Управління персоналом: система дозволяє керувати інформацією про викладацький персонал, включаючи управління кадровими даними, контроль оплати праці, розрахунок заробітної плати, планування кадрів та інші аспекти управління персоналом.

Фінансовий облік: Oracle PeopleSoft має модулі для фінансового обліку, бюджетування, звітності, контролю витрат та інших фінансових процесів. Це дозволяє університету вести облік фінансових операцій, контролювати бюджети та складати фінансові звіти. Система також забезпечує такі функції управління ресурсами, як управління закупівлями, облік запасів, управління активами та управління запасами в університеті.

Головний аспект це цінова політика. Ціни на Oracle PeopleSoft можуть значно відрізнятись залежно від набору функцій, розміру розгортання, кількості користувачів та інших факторів. Оскільки Oracle PeopleSoft є корпоративним рішенням, ціна зазвичай визначається індивідуально для кожного клієнта.

Ціна включає ліцензування, установку, налаштування, навчання, підтримку та вартість додаткових модулів або функцій. Оскільки Oracle PeopleSoft — це рішення корпоративного рівня, вартість може бути значною залежно від обсягу та складності. Для загального розуміння цінової політики можемо переглянути загальний документ з цінами на рис.1.2:

Prices in USA (Dollar)

	Component License Price	Software Update License & Support	License Metric	Minimum
Customer Relationship Management				
HelpDesk for Employee Self Service	30.00	6.60	Employee	All Employees
HelpDesk for Human Resources	120.00	26.40	Employee	All Employees
Multichannel Communications	1,950.00	429.00	Application User	5
Supply Chain Management				
eProcurement	80.00	17.60	Application User	5
eSupplier Connection	9,195.00	2,022.90	Application User	5
Inventory	4,595.00	1,010.90	Application User	5
Option: Mobile Inventory Management	1,725.00	379.50	Application User	5
Order Management	5,100.00	1,122.00	Application User	5
Purchasing	4,595.00	1,010.90	Application User	5
Strategic Sourcing	9,195.00	2,022.90	Application User	5
Supplier Contract Management	6,895.00	1,516.90	Application User	5
Asset Lifecycle Management				
Lease Administration	4,595.00	1,010.90	Application User	5
Maintenance Management	4,595.00	1,010.90	Application User	5
Option: Self-Service Work Requests	575.00	126.50	Application User	5
ESA				
Contracts	6,895.00	1,516.90	Application User	5
Expenses	6.00	1.32	Expense Report	1,000
Grants	7,125.00	1,567.50	Application User	5
Program Management	2,895.00	636.90	Application User	10
Project Costing	4,595.00	1,010.90	Application User	5

Рис.1.2 Цінова політика Oracle PeopleSoft

До переваг Oracle PeopleSoft можна віднести-великий функціонал: Oracle PeopleSoft надає широкий спектр функціональності для управління університетськими процесами, включаючи управління студентами, фінансовий облік, управління персоналом, управління ресурсами та інші аспекти управління університетом. Щодо гнучкості та налаштовуваності :Oracle PeopleSoft дозволяє налаштовувати систему відповідно до конкретних потреб та вимог університету. Вона може бути адаптована до унікальних бізнес-процесів та структури університетської організації. Інтеграція з іншими системами: Oracle PeopleSoft має здатність інтегруватися з іншими програмними рішеннями, що використовуються університетом, такими як системи електронного навчання, бібліотечні системи, системи управління дослідженнями та інші.

Недоліки Oracle PeopleSoft це-високі витрати: Вартість впровадження та підтримки Oracle PeopleSoft може бути високою, особливо для менших університетів з обмеженими бюджетами. Це може бути перешкодою для прийняття рішень у деяких університетах. Складність реалізації: Oracle PeopleSoft є складною системою, яка потребує спеціального впровадження та усунення несправностей. Це

може вимагати значних зусиль, ресурсів і досвіду ІТ-команди університету або залучення зовнішніх експертів.

Тому через високі витрати Oracle PeopleSoft, не доцільно використовувати в даному проєкті

1.2.3. Система обліку «Кадри Плюс Україна»

Кадри Плюс Україна - програма, створена компанією AnDeeSoft, для автоматизації обліку і ведення документообігу у відділах кадрів. Додаток дозволяє створювати звіти, заяви, накази, відстежувати персонал і здійснювати облік робочого часу. Є можливість систематизації груп і посад, використання класифікатора посад, ведення бази персоналу, що включає всю інформацію про співробітників, їхніх дітей і наявних соціальних пільг. [2].

Бухгалтерська система «Кадри Плюс Україна» – популярне в Україні програмне забезпечення для ведення кадрового обліку та оплати праці.

«Кадри Плюс Україна» пропонує широкий спектр функцій, необхідних для розрахунку заробітної плати, таких як управління персоналом, реєстрація співробітників, контроль робочого часу, облік заробітної плати та звітність. Це дозволить університету ефективно вести облік кадрів і зарплат.

Система базується на законодавстві України з обліку персоналу та оплати праці. Надають функціональність і звітність, необхідні для дотримання юридичних вимог і уникнення потенційних проблем. Зручний для користувача:

«Кадри Плюс Україна» має зручний та інтуїтивно зрозумілий інтерфейс, який спрощує роботу з системою. Його може використовувати як кваліфікований персонал, так і користувачі з базовими знаннями бухгалтерського обліку.

Провайдери систем підтримують користувачів шляхом консультацій, навчання та усунення несправностей. Також надають оновлення системи, які дозволяють університетам користуватися найновішими функціями та виправляти помилки.

Однак облікова система «Кадри Плюс Україна» має і свої недоліки.

Високі вимоги до обладнання: щоб використовувати цю програму, потрібен потужний комп'ютер і швидке підключення до Інтернету. Налаштування програми

може бути складним і може вимагати додаткових знань і досвіду. Ця програма дорога і може стати проблемою для малого бізнесу з обмеженим бюджетом.

З цінами на ліцензії можна ознайомитись на офіційному сайті, та наведено на рис.1.3.[3]

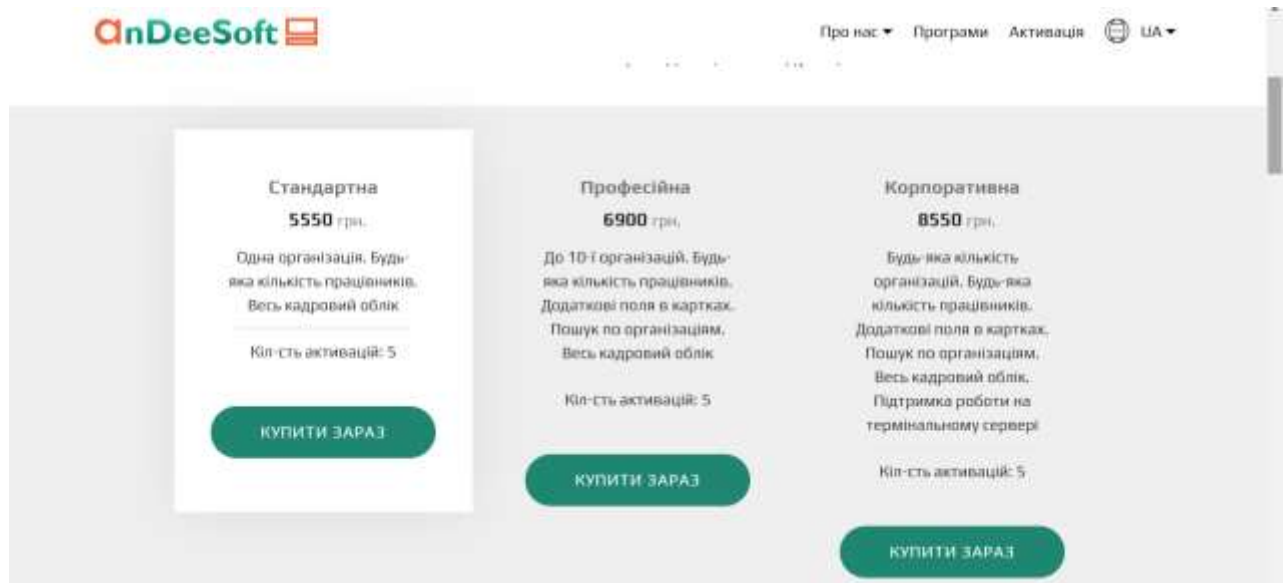


Рис.1.3- Ліцензії

Як видно рис, Andeesoft надає програмне забезпечення. Доступні продукти в різних версіях і різних рівнях функціональності. Ця програма спеціалізується на бухгалтерському обліку. На малюнку 1.4 показаний функціонал програми «Кадри Плюс Україна»



Рис.1.4- функціонал програми «Кадри Плюс Україна»

Як бачимо на малюнку 1.4, програма «Кадри Плюс Україна» дозволяє нам вести облік для організації чи структурованої системи, опис усіх організацій з класифікаторами посад та ведення кадрової статистики. При цьому програма не дозволяє операцію користування онлайн, тому варто створювати новий програмний продукт.

Висновки до розділу

Наведені вище згадки про системи управління навчанням, такі як Moodle, Oracle PeopleSoft, облікова система «Персонал Плюс Україна» тощо, свідчать про наявність різноманітних інструментів та рішень для організації навчального процесу. Основний висновок полягає в тому, що ці системи мають потенціал для покращення управління навчанням у вищих навчальних закладах.

Moodle — це потужна онлайн-навчальна платформа, яку можна використовувати для створення систем входу. Він пропонує широкий спектр інструментів для створення курсів, спілкування зі студентами, публікації матеріалів і виставлення оцінок. Однак використання Moodle для входу може потребувати додаткових модулів або налаштувань для задоволення потреб проекту.

Oracle PeopleSoft — комплексна система навчання та управління людськими ресурсами, яку можна використовувати для створення систем реєстрації. Він пропонує широкі можливості для автоматизації реєстрації студентів, планування курсу, оцінювання та інших процесів. Однак, варто враховувати, що Oracle PeopleSoft є великим та складним рішенням, яке може бути більш підходящим для великих університетів зі складними структурами та потребами управління.

Система обліку "Кадри Плюс Україна" спеціалізується на управлінні персоналом та кадрами. Хоча ця система не є прямим рішенням для створення системи обліку абітурієнтів, вона може бути корисною для обліку та управління персоналом в університеті, включаючи абітурієнтів та їхні дані.

Враховуючи особливості кожної програми, варто розглянути можливість розробки власної системи обліку абітурієнтів, яка буде повністю відповідати нашим потребам та вимогам

2.ВИБІР ЗАСОБІВ РОЗРОБКИ

Розробка є дуже важливим процесом у сучасному світі, тому що веб-технології домінують у більшості аспектів життя та бізнесу. Веб-програми можуть імітувати роль стандартних настільних програм, але з багатьма перевагами:

- Доступ з будь-якого місця
- Додаткова установка не потрібна, оскільки є підключення до Інтернету
- Програмне забезпечення та масштабування для мінливих потреб користувача
- Однією з головних переваг розробки веб-додатків є доступність.
- Веб-програми дозволяють користувачам взаємодіяти з програмами

Головною перевагою систем є простота їх розгортання обслуговування. Користувачеві не потрібно встановлювати програмне забезпечення що зменшує час і зусилля, необхідні для використання програма. Це також значно спрощує технічне обслуговування для розробників та забезпечення його постійного доступності програмного забезпечення.

Веб-додатки можуть працювати на будь-якій операційній системі, що підтримує веб-браузери для легкого доступу для користувачів. Користувачі можуть отримати доступ до веб-програм на будь-якому пристрої з доступом до Інтернету.

Загалом, розробка веб-застосунків дуже важлива та має багато переваг. Веб-застосунки забезпечують доступність, легкість в розгортанні та підтримці, масштабованість та кросплатформеність. Ці переваги дозволяють розробникам створювати додатки, які можуть задовольнити потреби різних користувачів та бізнесів, збільшуючи продуктивність та ефективність роботи. [4,5]

2.1 Засоби розробки

Плюси та мінуси розробника полягають у тому, що Інтернет постійно змінюється. Хоча це захоплююче, це також означає, що розробники завжди повинні бути готові вивчати нові техніки та мови програмування, адаптуватися до змін і приймати нові виклики. Це включає адаптацію існуючих фреймворків до потреб бізнесу, тестування веб-сайтів на технічні проблеми, оптимізацію та розширення веб-сайтів для кращої роботи з серверною

Менеджер пакетів і завдань програми виконання використовуються для автоматизації робочих процесів. Наприклад, можемо створити завдання, яке автоматизує мінімізацію JavaScript. Потім створюємо та комбінуємо завдання, щоб скоротити час розробки. Менеджер пакетів відстежує всі пакети, якими користуєтеся, і переконається, що вони оновлені та мають конкретну версію, яка потрібна.

`npm` (Node Package Manager) — це менеджер пакетів для платформи Node.js. Розробники Node.js можуть легко керувати залежностями, які використовуються в їхніх проектах. Основна мета `npm` — дозволити розробникам легко та швидко встановлювати, оновлювати та видаляти пакети, які використовуються в їхніх проектах. `npm` забезпечує це через файл `package.json`. Цей файл зберігає інформацію про залежності проекту. Назва, версія та інша інформація про пакет, необхідна для належного функціонування проекту. Однією з головних переваг `npm` є його репозиторій пакетів, який містить понад мільйон пакетів з відкритим кодом, створених розробниками з усього світу. Кожен із цих пакетів можна легко встановити у ваш проект за допомогою команди `npm install`. Крім того, `npm` дозволяє розробникам створювати власні пакети та публікувати їх у репозиторіях, щоб інші розробники могли використовувати їх у своїх проектах. Кожен пакет може містити не лише код, але й інформацію про використання, наприклад: документація, тести, зразки коду.

Окрім керування залежностями та публікації пакетів, `npm` також надає інші корисні функції, такі як виявлення пакетів, оновлення пакетів і версії пакетів. Його також можна використовувати для керування проектами та допомагає створювати та запускати проекти.

2.2 Мови та платформи програмування

За кожним інструментом розробки стоїть мова. Мова програмування — це формальна структурована мова, яка використовується для написання програм, які можуть спілкуватися з комп'ютером і контролювати його поведінку. Нижче наведено приклади поширених мов програмування та розмітки, які використовуються для веб-розробки.

На сьогоднішній день популярна мова-PHP, яка розшифровується як «PHP: Hypertext Preprocessor» — це мова сценаріїв, яка використовується для розробки веб-додатків і динамічних веб-сторінок. Він широко використовується у веб-розробці завдяки простоті використання та здатності взаємодіяти з різними базами даних та іншими зовнішніми джерелами даних. PHP є мовою з відкритим кодом. Це означає, що вихідний код доступний кожному, хто хоче його використати або змінити. Це дозволяє розробникам додавати нові функції та вдосконалювати мову.

PHP можна використовувати для створення різноманітних веб-додатків, таких як блоги, форуми, соціальні мережі, електронні магазини та багато іншого.

Особливості PHP:

- Простота використання: Простий синтаксис PHP полегшує його вивчення та використання.

- Динаміка: PHP може створювати динамічні веб-сторінки. Це означає, що вміст сторінки може змінюватися у відповідь на дії користувача чи іншу зовнішню інформацію.

- Розмаїття функцій: PHP має велику кількість вбудованих функцій і бібліотек, що робить його дуже потужним для розробки різних веб-додатків.

- Підтримка бази даних: PHP добре працює з різними базами даних, такими як MySQL, PostgreSQL і Oracle. Це дозволяє розробникам легко зберігати та отримувати дані з баз даних.

- Масштабованість: PHP дозволяє створювати складні веб-додатки та масштабувати їх відповідно до зростаючих потреб користувачів.

- Кросплатформенність: PHP працює в будь-якій операційній системі, включаючи Windows, Linux і macOS, а також може використовуватися з великою кількістю веб-серверів.

- Безпека: PHP має вбудований захист від XSS (міжсайтових сценаріїв), впровадження SQL та інших атак.

- Відкритий код: PHP є мовою з відкритим кодом. Це означає, що він безкоштовний для використання та може бути змінений і вдосконалений спільнотою розробників.

2.3 Бази даних

База даних — це структурований набір даних, що зберігається в комп'ютерній системі, з якого дані можна швидко й ефективно отримувати, оновлювати, видаляти та вставляти. Бази даних використовуються для зберігання великих обсягів даних і забезпечення доступу до них із різних джерел. У базі даних можна зберігати різноманітні дані. Текстові документи, зображення, відео, аудіо тощо.

MySQL - це безкоштовна система керування базами даних з відкритим вихідним кодом, яка дозволяє зберігати, оновлювати та запитувати великі обсяги даних. MySQL створена для використання в сучасних веб-додатках, де необхідно зберігати велику кількість даних, та для інших застосувань, де вимоги до продуктивності дуже високі.

Основні особливості MySQL включають:

- Висока продуктивність:

MySQL забезпечує швидкий доступ до даних за рахунок ефективної організації даних в пам'яті та на диску, оптимізації запитів та інших методів.

- Масштабованість:

MySQL може працювати з великими обсягами даних і може бути розміщений на декількох серверах, що дозволяє забезпечити високу доступність та продуктивність.

- Безпека:

MySQL має вбудовану систему автентифікації та авторизації, що дозволяє забезпечити безпеку даних.

- Простота використання:

MySQL має простий і зрозумілий інтерфейс для розробників і адміністраторів.

- Функціональність:

MySQL має широкий набір функцій, включаючи можливість зберігати дані в різних форматах і підтримувати транзакції та запити для пошуку даних.

- Підтримка:

MySQL має велику спільноту користувачів і розробників, що забезпечує постійний розвиток системи та підтримку користувачів.

MySQL — одна з найпопулярніших систем керування базами даних, яка використовується багатьма веб-додатками, такими як WordPress, Facebook, Twitter

2.4.Веб-сервери

Веб-сервер — це сервер, який отримує HTTP-запити від клієнтів (зазвичай веб-браузерів) і надає HTTP-відповіді клієнтам, зазвичай із HTML-сторінками, зображеннями, файлами, медіа-потокami чи іншими даними. Для прикладу, на Рис.2.4.1 можемо побачити архітектуру серверів фонду Вікімедіа[6]

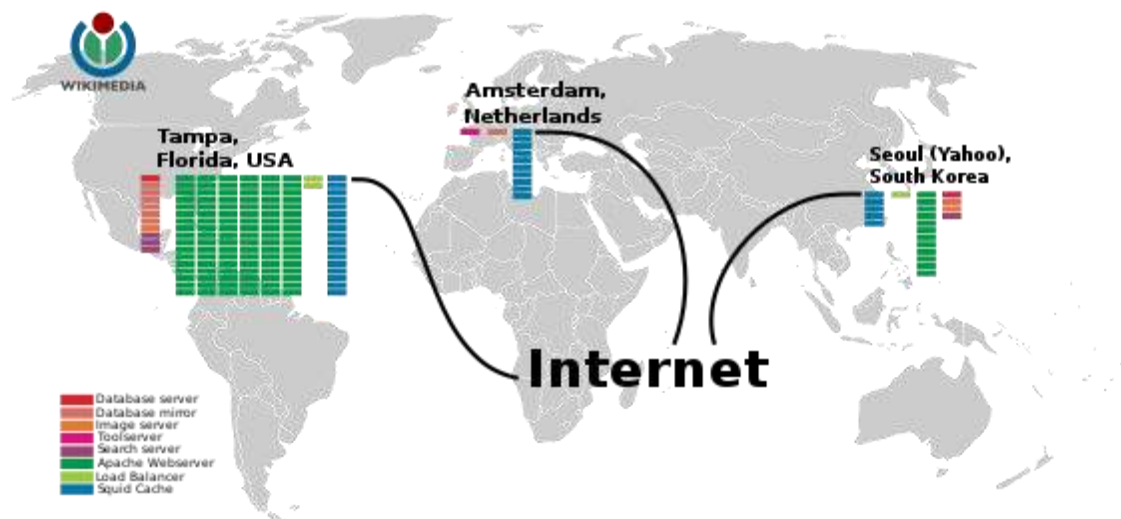


Рис. 2.4.1 Архітектура серверів фонду Вікімедіа[6]

Apache HTTP-сервер — відкритий вебсервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем. Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation.[7]

Є кілька переваг і причин використання сервера Apache у дипломному проєкті. Ось деякі з них:

- Відкрите джерело: Apache є відкритим кодом. Це означає, що можемо використовувати його безкоштовно та налаштувати вихідний код відповідно до своїх потреб. Це дає гнучкість і можливість налаштувати сервер відповідно до потреб.

- Популярність і документи: Apache є одним із найпопулярніших серверів у світі та має велику спільноту користувачів. Це означає, що існує багато документації, навчальних посібників і форумів, які допоможуть вирішити будь-які проблеми, які можуть виникнути під час розробки дипломного проекту.
- Підтримка різних протоколів і функцій: Apache підтримує багато протоколів, таких як HTTP, HTTPS, FTP, CGI та PHP. Він також має безліч функцій, які дозволяють налаштувати безпеку, контроль доступу, журналювання тощо.
- Сумісність з різними операційними системами: Apache доступний для багатьох операційних систем, включаючи Windows, macOS і різні дистрибутиви Linux. Це дозволить використовувати його на будь-якій платформі.
- Підтримка розробки веб-додатків: Apache дозволяє розгорнути веб-додатки, написані на різних мовах програмування, таких як PHP, Python і Ruby.

Отже, Apache - популярний і надійний веб-сервер, який підтримує багато мов програмування і протоколів. Це безкоштовне програмне забезпечення з відкритим вихідним кодом із великою спільнотою користувачів. Apache має широкі можливості, які дозволяють налаштувати безпеку, контроль доступу, журналювання та інші аспекти сервера.

Враховуючи всі ці фактори, використання Apache для проекту може бути вдалим вибором, тому що відповідає нашим конкретним потребам.

2.5.Стек

Стек (англ. stack) в інформатиці та програмуванні — різновид лінійного списку, структура даних, яка працює за принципом (дисципліною) «останнім прийшов — першим пішов» (LIFO, англ. last in, first out). Всі операції (наприклад, видалення елемента) в стеку можна проводити тільки з одним елементом, який міститься на верхівці стека та був уведений в стек останнім. [8]

Вибір відповідного стека програмування – це важливий крок під час розроблення комп'ютерної системи.

Є переваги використання готових стеків програмування. По-перше, стеки забезпечують готові рішення для різних компонентів програми, заощаджуючи час і зусилля на розробку. Крім того, вони зазвичай добре перевірені та мають велику підтримку з боку спільноти розробників. Це означає, що розробники завжди мають можливість звернутися за допомогою до інших досвідчених професіоналів, якщо у них виникнуть проблеми чи запитання.

Ще одна перевага - сумісність. Компоненти стеку зазвичай розроблені таким чином, щоб бути сумісними один з одним, полегшуючи інтеграцію та взаємодію між компонентами. Це дозволяє розробникам зосередитися на створенні функціональних можливостей програми замість вирішення проблем сумісності між компонентами.

Однак стеки програмування також мають обмеження.

По-перше, використання попередньо створених стеків може обмежити вибір певних технологій та інструментів. Розробнику потрібно знайти інші рішення або внести зміни в стек, якщо їм потрібні певні функції, які не надаються компонентами стеку. Крім того, деякі стеки програмування краще підходять для певних типів проектів і завдань.

Проаналізувши багато варіантів було прийнято рішення в використанні відповідних стеків:

- HTML5:

HTML5 — остання версія мови розмітки гіпертексту, яка використовується для створення структури та вмісту веб-сторінок.

- CSS (каскадні таблиці стилів):

CSS використовується для визначення зовнішнього вигляду елементів на веб-сторінках. В. Кольори, шрифти, вирівнювання тощо.

- Bootstrap 5:

Bootstrap — популярна платформа CSS і JavaScript для розробки адаптивних веб-додатків, зручних для мобільних пристроїв. Він надає готові стилі, компоненти та сценарії, які спрощують розробку та покращують вигляд і функціональність ваших програм.

- JavaScript (JS) і jQuery:

JavaScript — це мова програмування, яка використовується для створення динамічного та інтерактивного вмісту на веб-сайтах. jQuery — це популярна бібліотека JavaScript, яка дозволяє легко маніпулювати документами HTML і реалізовувати різні функції.

- PHP 8.1:

PHP — це мова програмування, яка зазвичай використовується для розробки серверних програм для веб-додатків. PHP 8.1 — це остання версія мови з новими функціями та покращеною продуктивністю.

- SQL:

SQL (Structured Query Language) — стандартна мова для керування реляційними базами даних. SQL дозволяє створювати, зберігати та маніпулювати даними в базах

Висновки до розділу

У роботі було визначено, що для створення інформаційної комп'ютерної системи для обліку абітурієнтів обрано наступні компоненти:

Сервер Apache: Apache є одним з найпоширеніших веб-серверів, який забезпечує надійну та стабільну роботу веб-додатків. Використання Apache дозволить забезпечити доступ до системи через мережу Інтернет та ефективно обробляти запити користувачів.

База даних MySQL: MySQL є потужною системою управління базами даних, яка забезпечує надійне зберігання та маніпулювання даними. Використання MySQL дозволить зручно зберігати та керувати інформацією про абітурієнтів, їхні документи та інші деталі.

Стек технологій, що складається з HTML5, CSS з використанням Bootstrap 5, JavaScript з використанням jQuery, PHP 8.1 та SQL, забезпечує потужний та гнучкий розвиток системи. Використання HTML5 та CSS з допомогою Bootstrap 5 дозволить створити естетичний та адаптивний інтерфейс користувача. JavaScript з

використанням jQuery надасть можливості для динамічного та інтерактивного взаємодії з системою. PHP 8.1 використовується для розробки серверної логіки та обробки запитів користувачів. SQL дозволить взаємодіяти з базою даних та виконувати операції збереження, оновлення та отримання даних.

Застосування цього стеку технологій дозволить реалізувати функціональну та ефективну систему обліку абітурієнтів. Вона забезпечить зручний доступ до інформації про абітурієнтів, можливість додавання та оновлення даних

РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1. Розробка front-end частини системи

Підрозділ "Розробка front-end частини комп'ютерної системи для обліку абітурієнтів" присвячений опису процесу створення функціональної та естетично привабливої користувацької інтерфейсу (UI) для комп'ютерної системи, яка призначена для обліку абітурієнтів. Front-end частина системи відповідає за взаємодію з користувачем, надаючи йому зручні інструменти для реєстрації, обробки та відстеження даних абітурієнтів.

У цьому підрозділі розглянуто ключові аспекти розробки front-end частини системи для обліку абітурієнтів, зосередившись на таких елементах, як дизайн інтерфейсу, взаємодія з користувачем та функціональні можливості.

Перш за все, висвітлено процес проектування інтерфейсу, включаючи визначення основних вимог до зовнішнього вигляду системи, структури та розміщення елементів керування.

Далі ми розглянемо аспекти взаємодії з користувачем, зосередившись на важливих елементах, таких як навігація, валідація даних та забезпечення зручності використання. Ми дослідимо різні способи покращення взаємодії, такі як використання анімацій, вказівок для користувачів та інтерактивних елементів.

На завершення розділу ми розглянемо функціональні можливості, які повинна надавати front-end частина системи для обліку абітурієнтів. Ми дослідимо можливості авторизації абітурієнтів, внесення та перегляд файлів, а також відстеження цих файлів адміністратором.

Цей підрозділ надасть всебічний огляд процесу розробки front-end частини комп'ютерної системи для обліку абітурієнтів, а також надасть рекомендації та кращі практики для створення функціонального та привабливого інтерфейсу.

Розробка front-end є важливою складовою процесу створення веб-додатків та комп'ютерних систем. Front-end відповідає за те, що бачить і взаємодіє з користувачем. Це включає в себе дизайн користувацького інтерфейсу, розміщення елементів на сторінці, навігацію та взаємодію з користувачем.

Процес розробки front-end починається зі збору вимог та визначення

функціональності, яку потрібно реалізувати.

Перша сторінка, яку бачить користувач, потрапляючи на сайт — це сторінка авторизації. Простота та зрозумілість дизайну сторінки авторизації є надзвичайно важливими аспектами у розробці front-end. Це перша сторінка, з якою зустрічається користувач, і вона має надати зручний та легкий для розуміння спосіб введення облікових даних.

Користувачі повинні миттєво зрозуміти, які дії потрібно виконати для авторизації. Мінімалістичний та інтуїтивно зрозумілий інтерфейс допоможе забезпечити швидку та зручну авторизацію для користувачів.

Сторінка авторизації повинна містити лише необхідні поля та інформацію, таку як поле для введення електронної пошти та поле для пароля (рис. 3.1.1.). Забагато полів може заплутати користувача і створити перешкоди для успішної авторизації.



The image shows a login form with the following elements:

- Title:** Вхід
- Label:** Електронна пошта
- Input field:** Введіть електронну пошту
- Label:** Пароль
- Input field:** Введіть пароль
- Button:** Увійти

Рис. 3.1.1. Форма авторизації

Якщо користувач зробив помилку під час введення облікових даних, важливо відобразити чіткі повідомлення про помилки і надати відповідні підказки щодо правильного формату вводу (рис.3.1.2.). Це допоможе користувачам легко виправити свої помилки та продовжити процес авторизації.

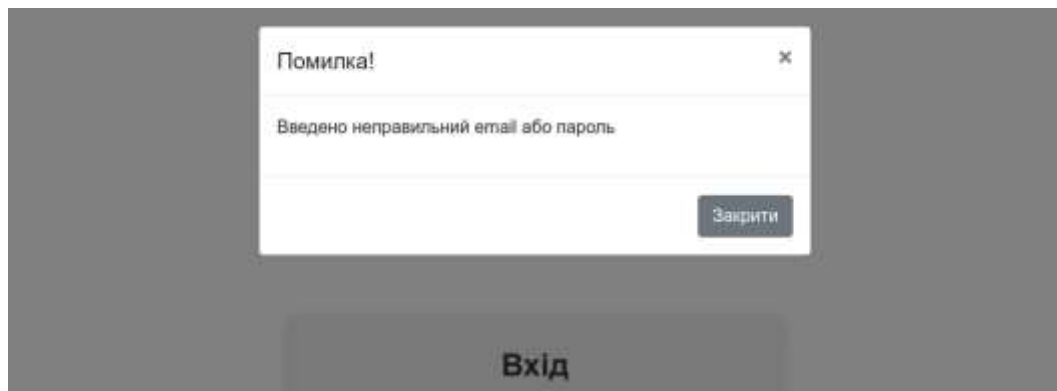


Рис.3.1.2. Повідомлення про помилку авторизації

Дизайн сторінки авторизації повинен бути адаптивним до різних пристроїв та розмірів екранів. Це дозволить користувачам зручно авторизуватися незалежно від того, з якого пристрою вони використовують систему.

Усе це сприяє позитивному враженню користувача, полегшує йому процес авторизації та збільшує загальну використовуваність системи. Простота та зрозумілість дизайну сторінки авторизації є ключовими факторами успіху і задоволення користувачів.

Наступною сторінкою, на яку потрапляє користувач після авторизації є сторінка керування та поширення файлів. Основним елементом цієї сторінки є шапка застосунку (рис.3.1.3.).

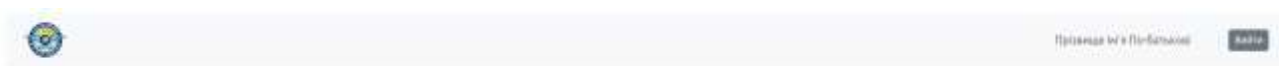


Рис.3.1.3. Шапка застосунку

В лівій частині шапки вирішено було розташувати логотип Національного авіаційного університету. Логотип є активним посиланням, яке веде на офіційний сайт університету.

В правій частині знаходиться ПІБ користувача, який авторизувався та кнопка для виходу з системи.

В залежності від ролі користувача — адміністратор чи студент, інтерфейс комп'ютерної системи для обліку абітурієнтів буде варіюватися. Якщо авторизований користувач є студентом — він побачить блок для завантаження файлів у систему (рис.3.1.4).



Рис.3.1.4. Кнопка додавання файлів

При кліку на кнопку з'являється форма для завантаження файлу (рис.3.1.5).



Рис.3.1.5. Форма завантаження файлів

Цей функціонал було реалізовано на бібліотеці jQuery — популярній бібліотеці JavaScript, яка спрощує взаємодію з HTML-документами, обробку подій, анімацію, виконання асинхронних запитів до сервера та багато інших завдань, пов'язаних з розробкою веб-додатків (рис.3.1.6).

```
<script>
$(document).ready(function () {
    $('#upload-form').hide();
    $('#add-icon').click(function () {
        if ($('#upload-form').is(":visible")) {
            $('#add-icon p').show();
            $('#upload-form').hide();
        } else {
            $('#upload-form').show();
            $('#add-icon p').hide();
        }
    });
});
</script>
```

Рис.3.1.6. Реалізація показу блоку на jQuery

Основним блоком головної сторінки даної комп'ютерної системи є безпосередньо блок управління та перегляду файлів (рис.3.1.7.). У випадку, якщо авторизований користувач є студентом йому доступний буде функціонал перегляду вивантажених ним файлів. Також буде можливість завантажити конкретний файл на свій пристрій чи видалити його з переліку.



Назва файлу	Тип файлу	Розмір файлу	Дата	Файлів: 10	
файл10.pdf	application/pdf	0 B	2023-06-09 00:26:27	Download	Delete
файл9.pdf	application/pdf	0 B	2023-06-09 00:26:20	Download	Delete
файл8.pdf	application/pdf	0 B	2023-06-09 00:28:11	Download	Delete
файл7.pdf	application/pdf	0 B	2023-06-09 00:26:02	Download	Delete
файл6.pdf	application/pdf	0 B	2023-06-09 00:24:54	Download	Delete
файл5.pdf	application/pdf	0 B	2023-06-09 00:24:47	Download	Delete
файл4.pdf	application/pdf	0 B	2023-06-09 00:24:36	Download	Delete
файл3.pdf	application/pdf	0 B	2023-06-09 00:24:24	Download	Delete
файл2.pdf	application/pdf	0 B	2023-06-09 00:24:11	Download	Delete
файл1.pdf	application/pdf	0 B	2023-06-09 00:23:56	Download	Delete

Рис.3.1.7. Блок управління файлами зі сторони студента

Перелік вивантажених файлів являє собою таблицю зі стовпцями, які містять інформацію про файл. У стовпцях зазначено: назву файлу, його тип, розмір та дату завантаження. Також у правому кутку вказано загальну кількість вивантажених файлів користувача.

Якщо ж авторизований користувач має роль адміністратора — він отримує інформацію про студента, який поділився файлом, а саме: факультет, на якому навчається студент, його спеціальність, безпосередньо ПІБ користувача-власника файлу та його сертифікат.

Крім загальнодоступних функцій завантаження та видалення вибраного файлу, з додаткових функцій, адміністратор має можливість переглянути файл не завантажуючи його (рис.3.1.8).

Файли: 10

Назва файлу	Тип файлу	Розмір файлу	Дата	Факультет	Спеціальність	Користувач	Сертифікат			
файл0.pdf	application/pdf	0 B	2023-06-09 00:28:27	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл9.pdf	application/pdf	0 B	2023-06-09 00:28:20	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл8.pdf	application/pdf	0 B	2023-06-09 00:28:11	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл7.pdf	application/pdf	0 B	2023-06-09 00:28:02	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл6.pdf	application/pdf	0 B	2023-06-09 00:28:54	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл5.pdf	application/pdf	0 B	2023-06-09 00:28:47	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл4.pdf	application/pdf	0 B	2023-06-09 00:28:36	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл3.pdf	application/pdf	0 B	2023-06-09 00:28:24	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл2.pdf	application/pdf	0 B	2023-06-09 00:28:11	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete
файл1.pdf	application/pdf	0 B	2023-06-09 00:23:56	ФЮНТ	КІ-123	Баденськова Аріна Сергієвна	456824	Open	Download	Delete

Рис.3.1.8. Блок управління файлами зі сторони адміністратора

Розробка front-end частини комп'ютерної системи для обліку абітурієнтів включає створення важливих елементів, таких як сторінка авторизації та сторінка керування та поширення файлів. Важливість створення зручного та ефективного інтерфейсу полягає в спрощенні взаємодії користувачів з системою, забезпеченні зрозумілих повідомлень про помилки та наданні корисних підказок.

Використання бібліотеки jQuery в процесі розробки front-end дозволяє спростити взаємодію з HTML-документами, обробку подій та анімацію. Це допомагає забезпечити більш швидку та ефективну розробку веб-додатків.

Створення головної сторінки системи, яка містить блок управління та перегляду файлів, є ключовим елементом. Для студентів доступний функціонал завантаження та перегляду вивантажених файлів, тоді як адміністратори можуть переглядати інформацію про студентів та їх файлів.

Унікальний об'ємний висновок до підрозділу "Розробка front-end частини системи" полягає в уважному проектуванні сторінок, забезпеченні зручного та інтуїтивно зрозумілого інтерфейсу для користувачів, а також використанні потужних інструментів, таких як jQuery, для покращення функціональності та взаємодії з веб-додатком.

3.2. Створення бази даних

SQL (Structured Query Language) є мовою програмування, яка використовується для роботи з реляційними базами даних. За допомогою SQL можна створювати таблиці, вставляти, оновлювати та видаляти дані, виконувати запити і багато іншого.

Створення таблиці включає наступні етапи:

- Визначення назви таблиці: Назва таблиці повинна бути унікальною і описувати зміст таблиці, в нашому проєкті назвемо таблицю «users»
- Визначення стовпців: Визначили набір стовпців, які будуть присутні у таблиці. Для кожного стовпця вказали його назву і тип даних. Id ,pid ,faculty, specialty certificate, quotas, phone, email, password, role. (що є відповідно Id-номер ідифікації абітурієнта, pid-Прізвище Ім'я По-батькові, faculty-номер факультету, specialty-спеціальність, certificate-номер сертифікату, quotas-квоти, phone-номер телефону, email-номер електронної пошти, password-пароль, role-роль користувача адміністратор/абітурієнта)
- Визначення обмежень: Обмеження встановлюють правила для даних в таблиці. Наприклад, можна вказати, що певний стовпець є обов'язковим для заповнення (NOT NULL), чи не є обов'язковим такі як faculty,quotas, certificate, specialty.
- Визначення первинного ключа: Первинний ключ ідентифікує унікальний запис у таблиці і забезпечує його ідентифікацію. Це може бути один стовпець, в нашому випадку це id абітурієнта

Архітектура бази даних визначає структуру та організацію даних у базі даних. Це включає в себе спосіб створення таблиць, взаємозв'язки між ними, індексацію, забезпечення цілісності даних та інші аспекти, що стосуються збереження і управління даними які можемо побачити на рис 3.2.1.

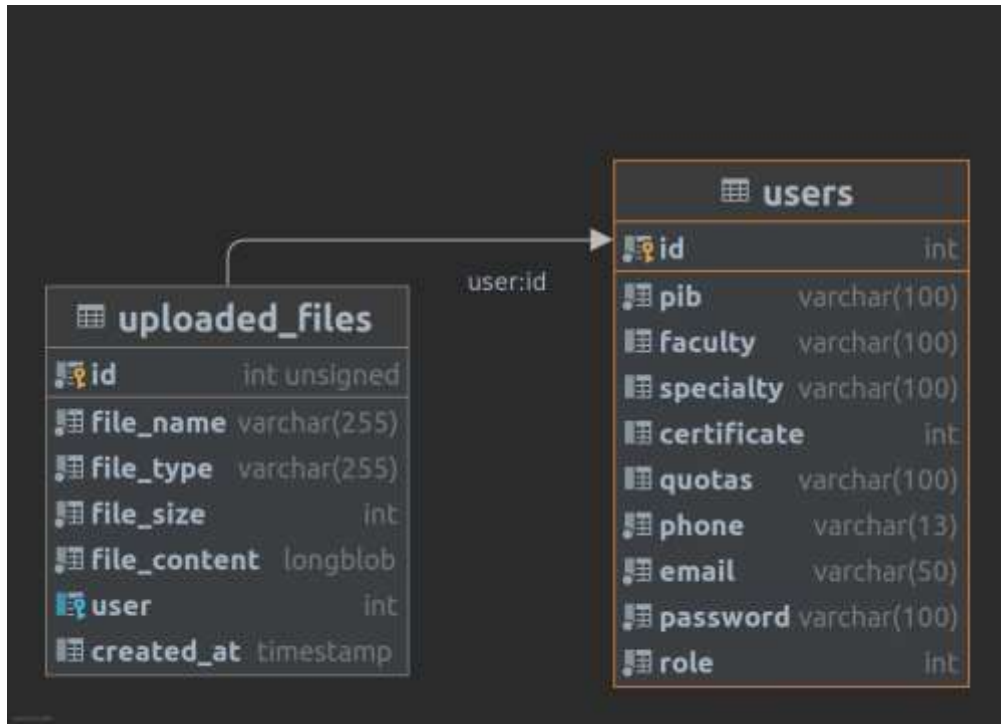


рис3.2.1- Архітектура БД

Запис таблиці: Після визначення структури таблиці і полів можна вставити дані в таблицю за допомогою команди INSERT. Кожен запис вставляється у новий рядок таблиці з відповідними значеннями для кожного стовпця(рис 3.2.2).

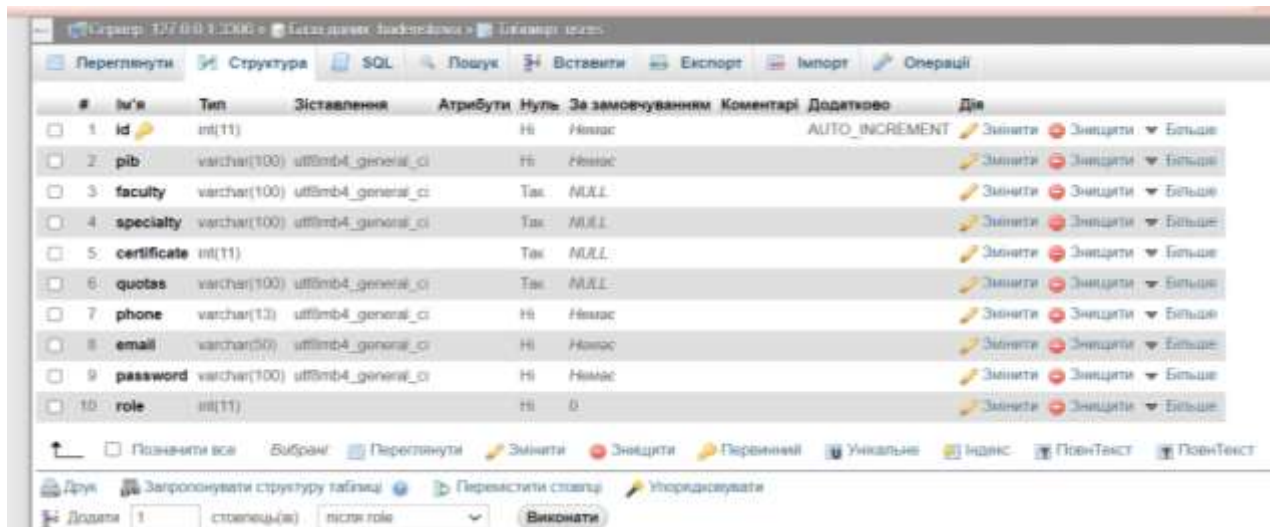


рис 3.2.2 Вигляд таблиці

Маніпулювання даними: Крім створення таблиць, SQL дозволяє виконувати різноманітні операції з даними, такі як оновлення (UPDATE), видалення (DELETE) і вибірка (SELECT). Це дає можливість управляти та отримувати дані

Занесення в таблицю абітурієнта (рис 3.2.3)

```
INSERT INTO `users` (`id`, `pib`, `faculty`, `specialty`, `certificate`, `quotas`, `phone`, `email`, `password`, `role`) VALUES ('45643', 'Петрова Олена Миколаївна', 'ФКНТ', '121', '234563', NULL, '+380950930661', 'petrov@gmail.com', '', '0');
```



рис 3.2.3-Успішне додавання абітурієнта в БД

Перегляд створеної таблиці(рис 3.2.4)

	id	pib	faculty	specialty	certificate	quotas	phone	email	password	role
Розглянути	1	Ваданська Анна Сергійвна	ФКНТ	КІ-123	45624	NULL	+38097777777	anna@gmail.com	a5c0e18232061acc36db743a412b	0
Розглянути	2	Коваленко Анна Сергійвна	ФКНТ	КІ-1	567	NULL	+38098000000	anna@gmail.com	89c3191d40c4932746f9e01794549	1
Розглянути	45633	Ваданська Анна Сергійвна	ФКНТ	122	234567	NULL	+38093344331	anna@gmail.com	9d812194e1acc36db7	0
Розглянути	45643	Петрова Олена Миколаївна	ФКНТ	121	234563	NULL	+380950930661	petrov@gmail.com		0

рис 3.2.5 Загальний вид створеної таблиці

3.3. Розробка back-end частини системи

Підрозділ "Розробка back-end частини комп'ютерної системи для обліку абітурієнтів" присвячений детальній опису процесу створення та розробки back-end компонентів системи обліку абітурієнтів. У даному підрозділі будуть розглянуті ключові аспекти, пов'язані з проектуванням та реалізацією серверної частини системи, що забезпечує ефективне зберігання, обробку та обмін даними з фронтенд-інтерфейсом.

В першій частині підрозділу будуть розглянуті вимоги до back-end частини системи, включаючи функціональні та нефункціональні вимоги. Будуть описані основні функціональності, які система повинна забезпечувати, такі як реєстрація абітурієнтів, зберігання їхніх персональних даних, надання доступу до різних інформаційних ресурсів тощо. Також будуть визначені нефункціональні вимоги, наприклад, надійність, масштабованість, безпека та продуктивність системи.

У другій частині підрозділу буде описано архітектурний підхід до розробки back-end частини системи обліку абітурієнтів. Будуть розглянуті основні компоненти архітектури, включаючи базу даних, серверні модулі, API (Application

Programming Interface) та інші складові системи. Також будуть обговорені питання вибору технологій та інструментів для розробки, зокрема мов програмування, фреймворків, баз даних та інших важливих компонентів.

У третій частині підрозділу будуть розглянуті деталі реалізації back-end частини системи обліку абітурієнтів. Будуть описані основні модулі та функціональні компоненти системи, їх взаємодія та особливості реалізації. Крім того, будуть розглянуті методи забезпечення безпеки та захисту даних, включаючи автентифікацію, авторизацію та захист від зловживань.

У заключній частині підрозділу будуть підведені підсумки розробки back-end частини комп'ютерної системи для обліку абітурієнтів. Будуть висвітлені досягнуті результати, переваги розробленої системи та можливості для подальшого розвитку та удосконалення. Також будуть наведені висновки та рекомендації щодо використання та підтримки back-end частини системи.

Перший етап - авторизація користувачів, виконується на серверній стороні за допомогою мови програмування PHP та використанням бази даних.

Початок PHP-скрипту:

```
session_start();  
include('./db_connect.php');  
global $db;
```

У першому рядку використовується функція `session_start()`, яка розпочинає або відновлює сесію користувача. У другому рядку включається файл `db_connect.php`, який містить код для підключення до бази даних. Глобальна змінна `$db` використовується для доступу до з'єднання з базою даних.

Перевірка, чи користувач надіслав форму:

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    $email = $_POST['email'];  
    $password = $_POST['password'];  
    // ...  
}
```

Звертається до глобального масиву `$_SERVER`, щоб перевірити, чи метод запиту є POST (тобто, чи користувач натиснув кнопку "Вхід" на формі). Змінні

\$email і \$password містять значення, введені користувачем у відповідні поля форми авторизації.

Перевірка правильності електронної пошти та пароля:

```
$stmt = $db->prepare('SELECT id, pib, faculty, specialty, certificate, quotas, phone, email,
role FROM users WHERE email = :email AND password = :password');
$stmt->execute(array(':email' => $email, ':password' => md5($password)));
$result = $stmt->fetch(PDO::FETCH_ASSOC);
```

За допомогою підготовленого запиту (prepared statement) підготовлюється SQL-запит до бази даних. Запит шукає користувача за введеним електронним листом (:email) та хешем пароля (:password). Використовується функція md5() для хешування пароля перед збереженням його в базі даних.

Результат запиту зберігається в змінну \$result за допомогою функції fetch(PDO::FETCH_ASSOC), яка отримує асоціативний масив з даними користувача.

Обробка результату авторизації:

```
if ($result) {
    // User exists, store user ID in session
    $_SESSION['user'] = $result;

    if ($result['role'] === 1) {
        $_SESSION['user']['path'] = 'admin';
    } else if ($result['role'] === 0) {
        $_SESSION['user']['path'] = 'dashboard';
    }

    header('Location: /' . $_SESSION['user']['path']);
    exit();
} else {
    // Display error message
    $_SESSION['error'] = 'Введено неправильний email або пароль';
    header('Location: /login');
    exit();
}
```

```
}
```

Якщо результат запиту не є порожнім (тобто, знайдено користувача з введеними електронною поштою та паролем), то дані користувача зберігаються в сесії \$_SESSION['user']. Також, в залежності від ролі користувача, встановлюється шлях переадресації (admin або dashboard).

Використовується функція header() для переадресації користувача на відповідну сторінку після успішної авторизації. У разі неправильних облікових даних, зберігається повідомлення про помилку в сесії \$_SESSION['error'], та користувач перенаправляється на сторінку авторизації (/login).

Наступний етап, який описаний у кодї, є виведенням списку файлів, які вивантажив користувач, з урахуванням їх ролі в системі (рис.3.3.1.).

```
<?php
include 'controllers/db_connect.php';
global $db;

if (isset($_SESSION['user']) && !empty($_SESSION['user'])) {
    $role = $_SESSION['role'];
    $sql = "SELECT * FROM uploaded_files ORDER BY created_at DESC";
} else {
    $role = null;
    $sql = "SELECT * FROM uploaded_files WHERE user_id = " . $_SESSION['user_id'] . " ORDER BY created_at DESC";
}

$result = $db->query($sql);
echo "<p class='text-end'>Кількість файлів: " . $result->rowCount() . "</p>";

// Display table of files
if ($result->rowCount() > 0) {
    echo "<table class='table table-striped table-hover' id='file-table'>";
    if ($role == 'admin') {
        echo "<thead class='thead-dark'><tr><th class='text-left'>Ім'я файлу</th><th class='text-left'>Тип файлу</th><th class='text-left'>Роль користувача</th><th class='text-left'>Дата завантаження</th><th class='text-left'>Дія</th></tr>";
    } else {
        echo "<thead class='thead-dark'><tr><th class='text-left'>Ім'я файлу</th><th class='text-left'>Тип файлу</th><th class='text-left'>Дата завантаження</th><th class='text-left'>Дія</th></tr>";
    }
    echo "<tbody>";
    while ($row = $result->fetch()) {
        echo "<tr>";
        echo "<td>" . $row['file_name'] . "</td>";
        echo "<td>" . $row['file_type'] . "</td>";
        echo "<td>" . $row['file_size'] . "</td>";
        echo "<td>" . $row['created_at'] . "</td>";

        // Check if admin
        if ($role == 'admin') {
            $userQuery = $db->query("SELECT * FROM users WHERE id = " . $row['user_id']);
            $user = $userQuery->fetch();
            echo "<td>" . $user['faculty'] . "</td>";
            echo "<td>" . $user['specialty'] . "</td>";
            echo "<td>" . $user['id'] . "</td>";
            echo "<td>" . $user['certificate'] . "</td>";
            echo "<td><a href='\" . $row['file_name'] . ">";
        }

        echo "<td><a href='\" . $row['id'] . ">";
        echo "<td><a href='\" . $row['id'] . ">";
        echo "</tr>";
    }
    echo "</tbody></table>";
} else {
    echo "<p class='text-danger text-center'>Файли не знайдені</p>";
}
}
```

Рис.3.3.1. Вивід списку вивантажених листів

Нижче розглянуто деталі кожної частини коду:

Починаємо з підключення до бази даних і визначення змінної \$db, яка представляє з'єднання з базою даних.

Перевіряємо роль користувача, який увійшов до системи, за допомогою

умовного оператора `if`. Якщо роль рівна 1, це означає, що користувач є адміністратором, тому виводимо всі завантажені файли. У зворотньому випадку, якщо роль дорівнює 0, виводимо тільки файли, які належать конкретному користувачу.

Виконуємо запит до бази даних залежно від ролі користувача. Результат запиту зберігається в змінній `$result`.

Виводимо загальну кількість файлів, яку отримали з результату запиту, за допомогою функції `rowCount()`. Ця кількість виводиться у вигляді тексту в елементі `<p>`.

Перевіряємо, чи отримано хоча б один файл з бази даних. Якщо так, тоді виводимо таблицю з файлами. Якщо ні, то виводимо повідомлення про відсутність файлів.

У разі наявності файлів, розпочинаємо виведення таблиці. Залежно від ролі користувача, відображаються відповідні стовпці таблиці.

За допомогою циклу `while`, проходимо через кожен рядок результату запиту. Для кожного рядка виводимо дані про файл, такі як назва, тип, розмір та дату створення.

Якщо роль користувача - адміністратор, додатково отримуємо інформацію про користувача, який завантажив файл, звертаючись до таблиці `users` за допомогою запиту до бази даних. Виводимо дані про факультет, спеціальність, ПІБ користувача та наявність сертифікату.

Для кожного рядка в таблиці, незалежно від ролі користувача, виводимо посилання на відкриття файлу (якщо роль - адміністратор) та завантаження та видалення файлу. Посилання мають відповідні URL-адреси, які спрямовують на відповідні контролери.

Завершуємо таблицю та блок коду.

Отже, цей код виводить список файлів, які були завантажені користувачем, з врахуванням їхньої ролі в системі. Також надається можливість відкриття, завантаження та видалення цих файлів, залежно від прав доступу користувача.

Наступний етап - завантаження файлів, які вивантажив користувач - реалізований у поданому фрагменті PHP-коду. Давайте розглянемо його детально:

Підключення до бази даних:

```
include './db_connect.php';  
global $db;
```

Цей рядок коду включає файл `db_connect.php`, який містить налаштування для підключення до бази даних. Глобальна змінна `$db` використовується для зберігання об'єкту з'єднання з базою даних.

Отримання ідентифікатора файлу для завантаження:

```
$id = $_GET['id'];
```

Цей рядок коду отримує ідентифікатор файлу, який передається як параметр у URL-запиті.

Вибірка файлу з бази даних:

```
$sql = "SELECT * FROM uploaded_files WHERE id = '$id'";  
$result = $db->query($sql);
```

Цей код виконує запит до бази даних, щоб отримати запис про файл з таблиці `uploaded_files`, використовуючи ідентифікатор файлу. Результат запиту зберігається в змінній `$result`.

Перевірка наявності файлу:

```
if ($result->rowCount() == 1) {  
    // Файл знайдено, отримуємо дані про нього  
    $row = $result->fetch();  
    $fileName = $row['file_name'];  
    $fileType = $row['file_type'];  
    $fileSize = $row['file_size'];  
    $fileContent = $row['file_content'];  
    // ...  
} else {  
    echo "File not found";  
}
```

Цей блок коду перевіряє, чи було знайдено файл у базі даних, перевіряючи кількість рядків у результаті запити. Якщо файл знайдено, то змінні \$fileName, \$fileType, \$fileSize та \$fileContent отримують значення відповідних полів запису про файл.

Відправка файлу користувачеві для завантаження:

```
header("Content-Type: $fileType");
header("Content-Length: $fileSize");
header("Content-Disposition: attachment; filename=\"$fileName\"");

echo $fileContent;
exit();
```

Цей блок коду встановлює необхідні HTTP-заголовки для передачі файлу користувачеві. Content-Type встановлює тип файлу, Content-Length встановлює розмір файлу, а Content-Disposition вказує браузеру, що файл має бути завантажений, а не відображений. Код \$fileContent виводить вміст файлу. Після цього використовується exit() для забезпечення завершення виконання скрипту.

Обробка випадку, коли файл не знайдено:

```
else {
    echo "File not found";
}
```

Якщо файл не знайдено у базі даних, виводиться повідомлення "File not found". Цей фрагмент коду дозволяє отримати файл з бази даних і надіслати його користувачеві для завантаження за допомогою необхідних HTTP-заголовків.

В результаті розділу "Розробка back-end частини комп'ютерної системи для обліку абітурієнтів" було детально описано процес створення та розробки серверної частини системи обліку абітурієнтів. На протязі цього розділу були розглянуті ключові аспекти, пов'язані з проектуванням та реалізацією back-end компонентів, необхідних для забезпечення ефективного зберігання, обробки та обміну даними з фронтенд-інтерфейсом.

Перший етап, яким була авторизація користувачів, було успішно реалізовано на серверній стороні за допомогою мови програмування PHP та використання бази даних. Код, який виконує перевірку введених користувачем даних, дозволяє визначити, чи вони є правильними, та надати відповідний доступ до системи або повідомити користувача про помилку.

Далі у розділі було розглянуто виведення списку файлів, які були завантажені користувачем, з урахуванням їх ролі в системі. Це дозволяє забезпечити зручний доступ до файлів залежно від прав доступу користувача і його ролі.

Також було описано етап завантаження файлів, які були вивантажені користувачем. Відповідний PHP-код забезпечує обробку завантажених файлів та їх збереження в системі з дотриманням необхідних прав доступу та безпекових заходів.

У результаті розробки back-end частини системи було досягнуто мети забезпечення ефективного функціонування, зберігання та обробки даних, а також забезпечено відповідність прав доступу та безпеку системи. Реалізація цих етапів дозволяє системі обліку абітурієнтів працювати з оптимальною продуктивністю та забезпечувати зручну та безпечну взаємодію з користувачами.

Завершуючи розділ, можна зазначити, що розробка back-end частини комп'ютерної системи для обліку абітурієнтів є важливим кроком у створенні повноцінної та функціональної системи. Описані етапи демонструють комплексний підхід до розробки back-end компонентів та враховують важливі аспекти безпеки та захисту даних.

Продовження розвитку та удосконалення back-end частини системи дозволить розширити її можливості, поліпшити продуктивність та забезпечити високу якість обслуговування користувачів.

Висновки до розділу

У даному розділі було успішно реалізовано розробку програмного продукту для системи обліку абітурієнтів. Було створено зручний та інтуїтивно зрозумілий користувацький інтерфейс, реалізовано зберігання та обробку даних абітурієнтів у базі даних, а також реалізовано необхідні функції back-end частини системи. Під час розробки були використані сучасні технології, такі як Apache, MySQL, HTML5, CSS, Bootstrap 5, JS, jQuery, PHP 8.1 та SQL, що дозволило забезпечити ефективну

та надійну роботу системи.

Розробка front-end частини системи:

У цьому етапі була проведена розробка користувацького інтерфейсу системи. Використовуючи HTML5, CSS, Bootstrap 5, JS та jQuery, була створена зручна та інтуїтивно зрозуміла інтерактивна сторінка, яка дозволяє абітурієнтам легко взаємодіяти з системою.

Створення бази даних:

У цьому етапі була створена база даних для зберігання інформації про абітурієнтів. Використовуючи MySQL, були створені таблиці з необхідними полями, такими як id, pib, faculty, specialty, certificate, quotas, phone, email, password та role. Це дозволяє зручно зберігати та оновлювати дані про абітурієнтів у системі.

Розробка back-end частини системи:

У цьому етапі була проведена розробка back-end частини системи, яка відповідає за обробку запитів користувачів та взаємодію з базою даних. Використовуючи PHP 8.1 та SQL, були написані необхідні скрипти і запити для обробки даних абітурієнтів, авторизації та аутентифікації користувачів.

Результати роботи в цьому розділі послужать основою для подальших етапів розробки та вдосконалення інформаційної комп'ютерної системи для обліку абітурієнтів.

Висновки

Система має зручний інтерфейс, що дозволяє абітурієнтам з легкістю зареєструватися, заповнити свої особисті дані та переглядати їх. Крім того, вона надає можливість додавання нових документів зі змогою адміністратора переглядати та використовувати ці документи.

Для створення інформаційної комп'ютерної системи були використані сучасні технології та інструменти, такі як сервер Apache, база даних MySQL та стек технологій, що включає HTML5, CSS з використанням Bootstrap 5, JavaScript з бібліотекою jQuery, PHP 8.1 та мову запитів SQL. Цей стек технологій забезпечив ефективну розробку та функціональність системи.

Створена система дозволяє ефективно управляти обліком абітурієнтів, забезпечує зручний доступ до інформації та функцій для користувачів. Вона також забезпечує безпеку даних шляхом шифрування паролів та застосування відповідних заходів для запобігання несанкціонованому доступу.

Під час розробки системи було проведено аналіз існуючих систем університетів з точки зору абітурієнта. Цей аналіз дозволив виявити недоліки наявних систем та врахувати їх при розробці нової системи.

Усі ці аспекти підтверджують успішне створення інформаційної комп'ютерної системи для обліку абітурієнтів, яка відповідає потребам користувачів та допомагає ефективно виконувати облік та управління абітурієнтами.

Висновки роботи показують, що створена інформаційна комп'ютерна система є ефективним інструментом для обліку абітурієнтів та покращення процесу вступу до університету. Розширення системи у майбутньому може включати додаткові функціональності, інтеграцію з іншими системами та розширення аналітичних можливостей. Застосування такої системи допоможе університету підвищити ефективність та зручність обліку абітурієнтів та покращити їх досвід вступу до вищого навчального закладу.

Список використаних джерел

1. Moodle. [Електронний ресурс]. Режим доступу: <https://moodle.org/getinvolved>
2. Документація Кадри Плюс Україна. [Електронний ресурс]. Режим доступу: <https://andeesoft.com/ru/kpu/kpu.pdf/>
3. Купівля Кадри Плюс Україна. [Електронний ресурс]. Режим доступу: <https://andeesoft.com/ua/kpu/buy.html>
4. Kaushik, S. Web Application Development Methodologies: A Comparative Study. International Journal of Advanced Research in Computer Science, Volume 10, No. 4, 2019.
5. Gothapatha, B. Modern tools and current trends in web-development. Indonesian Journal of Electrical Engineering and Computer Science, Bhubaneswar, India, 2021.
6. Wikimedia Servers. [Електронний ресурс]. Режим доступу: https://upload.wikimedia.org/wikipedia/commons/thumb/8/87/Wikimedia_Servers.svg/570px-Wikimedia_Servers.svg.png
7. Apache HTTP Server. [Електронний ресурс]. Режим доступу: https://uk.wikipedia.org/wiki/Apache_HTTP_Server
8. Стек технологій. [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BA>
9. Сміт, Дж. А. (2018). Вступ до інформаційних систем: підтримка та трансформація бізнесу. Routledge.
10. Laudon, K. C., & Laudon, J. P. (2016). Інформаційні системи управління: Управління цифровою фірмою. Пірсон.
11. О'Брайен, Дж. А., Маракас, Г. М. (2017). Інформаційні системи управління. Освіта McGraw-Hill.
12. Коннолі, Т., і Бегг, К. (2014). Системи баз даних: практичний підхід до проектування, впровадження та управління. Пірсон.
13. Елмасрі, Р., і Нават, С. Б. (2016). Основи систем баз даних. Пірсон.
14. Дейт К. Дж. і Дарвен Г. (2014). Основа майбутніх систем баз даних: третій маніфест. Addison-Wesley Professional.
15. «MySQL. Навчальний посібник» - Юрій Філіппов, Видавництво "КУРС"

16.«Responsive Web Design with HTML5 and CSS3: Build responsive and future-proof websites to meet the demands of modern web users» by Ben Frain, Packt Publishing, 2019