**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА**

**ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ**

**Кафедра цифрової економіки та бізнес-аналітики**

**КУРСОВА РОБОТА**

**з навчальної дисципліни**

**“Проектування та адміністрування БД і СД”**

**на тему:**

**«Інформаційна система приймальної комісії»**

Галузь знань: 05 «Соціальні та поведінкові науки»

Спеціальність: 051 «Економіка»

Спеціалізація: «Інформаційні технології в бізнесі»

Освітній ступінь: бакалавр

**Науковий керівник:         Виконавець:**

к.ф.-м.н., доц. Депутат Б.Я.Курчак Х. М.        

           (прізвище, ім’я, по-батькові)                                 (прізвище, ім’я, по-батькові)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис)                  УФЕ-31с        група

                                                                                \_\_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис)

**“\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 р.                               “\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 2021 р.**

**Загальна кількість балів** \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, ПІП членів комісії)

**Львів 2021**

Зміст

[ВСТУП 3](#_Toc72746776)

[РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 5](#_Toc72746777)

[1.1. Постановка задачі 5](#_Toc72746778)

[1.2. Розробка моделі 6](#_Toc72746779)

[РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ 8](#_Toc72746780)

[2.1. Що таке реляційні бази даних? 8](#_Toc72746781)

[2.2. Робота з MySQL 11](#_Toc72746782)

[2.3. Розробка архітектури програмної системи 13](#_Toc72746783)

[2.4. Розробка структури бази даних 14](#_Toc72746784)

[2.5. Перелік таблиць бази даних 15](#_Toc72746785)

[2.6. Перелік полів таблиць бази даних 15](#_Toc72746786)

[2.7. Визначення типів даних таблиці бази даних 17](#_Toc72746787)

[2.8. Реалізація SQL-скрипту; 19](#_Toc72746788)

[РОЗДІЛ 3 « РОЗРОБКА ВЕБ-ДОДАТКУ» 28](#_Toc72746789)

[3.1. «Структура веб-сайту» 28](#_Toc72746790)

[3.2. Макет веб-сторінки 29](#_Toc72746791)

[3.3. Програмування користувацької частини 30](#_Toc72746792)

[3.4. Програмування клієнтської частини 33](#_Toc72746793)

[ВИСНОВКИ 38](#_Toc72746794)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 39](#_Toc72746795)

[ДОДАТКИ 40](#_Toc72746796)

ВСТУП

**Актуальність проблеми.** У вік інформаційних технологій все рухається до глобальної інформатизації. Інформатизація, що розвивається під впливом викликів сучасності,  сприяє різкому зростанню нового інформаційного виробництва, що, у свою чергу, обумовлює вдосконалення системи соціальних інформаційних комунікацій, систем структурування збереження  та підготовки до використання  інформаційних ресурсів, інструментів пошуку і орієнтації у великих інформаційних масивах. Невід’ємною складовою життя людини є інформація. З розвитком суспільства збільшується обсяг інформації, при цьому інформація перетворилась в один з найбільш важливих ресурсів, який розглядається поряд із трудовими, матеріальними та енергетичними ресурсами. Фактично, інформація стала чинником, що визначає ефективність будь-якої сфери діяльності людини. Розвиток технологій набрав масштабних темпів. Сидячи вдома сьогодні ми можемо купити книги чи одяг, замовити їжу чи навіть записатись на прийом до лікаря. Система роботи приймальної комісію також потребує залучення інформаційних технологій. Процес подачі заявки зарахування на факультет завжди буде актуальним, так як вища освіта є важливою і не кожен готовий сидіти у довгих чергах аби подати документи у ВНЗ. Недоліком є те, щоб вступити інколи абітурієнту доводиться навіть їхати з іншого міста і можливо він навіть й не дочекається своєї черги, також це займає багато часу. Тому розробка такого веб-сайту, який надасть абітурієнту можливість подати заявку на потрібний йому факультет, є актуальною.

**Мета та завдання.** Метою даної роботи є обґрунтування теоретичних основ та реалізація інформаційної робот приймальної комісії.

Для досягнення мети курсової роботи, необхідно вирішити такі

завдання:

1. коротко охарактеризувати інформаційну систему для приймальної комісії;
2. проаналізувати предметну область;
3. розробити архітектуру програмної системи та спроектувати структуру бази даних;
4. вибрати мову програмування та технологій для програмної реалізації описаного продукту;
5. розробити веб-сайт «Приймальна комісія» для подання заявок для вступу на факультет;

**Об’єкт дослідження**. Особливість розробки інформаційної системи для приймальної комісії за допомогою системи управління базами даних MYSQL.

**Предмет досліджень**. Застосування технологій створення веб-орієнтованого програмного забезпечення для розробки веб-сайту «Приймальна комісія».

**Використане програмне забезпечення**. В процесі розробки використовуватимуться наступні мови програмування

та технології:

Java - об'єктно-орієнтована мова програмування.

HTML- мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет.

CSS - спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.

Bootstrap - безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript.

MySQL- вільна система керування реляційними базами даних.

**Структура роботи**. Курсова робота складається з трьох розділів(«Аналіз предметної області», «Розробка бази даних» та «Розробка веб-додатку»), висновків, списку використаних джерел та додатків.

Загальний обсяг роботи − 47 сторінок

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

* 1. Постановка задачі

У сучасному світі інформаційні технології займають не абияке місце в житті суспільства. Сьогоднішній світ перейшов на новий етап життя, де головну роль виконує інформація, а також економіка, що будується на ній. Сучасний розвиток інформаційного суспільства безпосередньо пов'язаний з необхідністю збору, обробки і передачі величезних об'ємів інформації, перетворенням інформації у товар, як правило, значної вартості. Це стало причиною глобального переходу від індустріального суспільства до інформаційного. Поява всесвітньої мережі Інтернет спричинила масштабне зростання міжнародних спілкувань у різних сферах людського життя.

Головні ідеї сучасних інформаційних технологій базуються на концепції баз даних. Відповідно до цієї концепції, основою інформаційних технологій є дані, які повинні бути організовані в бази даних для ефективної роботи з інформацією.

Інформація є одним з найцінніших ресурсів суспільства поруч з традиційними матеріальними видами ресурсів, як нафта, метал, корисні копалини тощо, тому, процес переробки інформації, подібно до процесів переробки матеріальних ресурсів можна сприймати як технологію. Інформаційна технологія передбачає вміння грамотно працювати з інформацією і обчислювальною технікою. У свою чергу, дані - це інформація, представлена в певному вигляді, що дозволяє автоматизувати її збір, зберігання і подальшу обробку людиною або інформаційним засобом. Для комп'ютерних технологій дані - це інформація в дискретному, фіксованому вигляді, зручна для зберігання, обробки, а також для передачі по каналах зв'язку.

Інформаційні технологій дуже швидко перетворилися на життєво важливий стимул розвитку не тільки світової економіки, а й інших сфер людської діяльності. На сьогодні практично неможливо знайти сферу, в якій зараз не використовуються інформаційні технології.

Не можливо оцінити важливість застосування інформаційних технологій у сфері освіти та науковій сфері. Зараз важко уявити собі школу, в якій би не було комп'ютерного класу, існує багато електронних бібліотек, користуватися якими можна не виходячи з дому, що значно полегшує процес навчання і самоосвіти. І при цьому інформаційні технології сприяють розвитку наукових знань.

Онлайн приймальна комісія призначена для швидкої подачі заявок на факультет, що в свою чергу принесе користь як абітурієнту, так і університету. Фактично у кожному університеті є безліч різних факультетів, які до того ж можуть знаходить у різних кінцях міста. Подача заяви онлайн пришвидшить весь процес і забере купу паперової роботи. Проте, щоб подати заявку користувач повинен бути авторизований.

* 1. Розробка моделі

Система приймальної комісії має три елементи керування:

* Будь-який користувач;
* Авторизований користувач;
* Адміністратор;

Розглянемо структурна схему варіантів використання web-додатку(рис. 1.1).

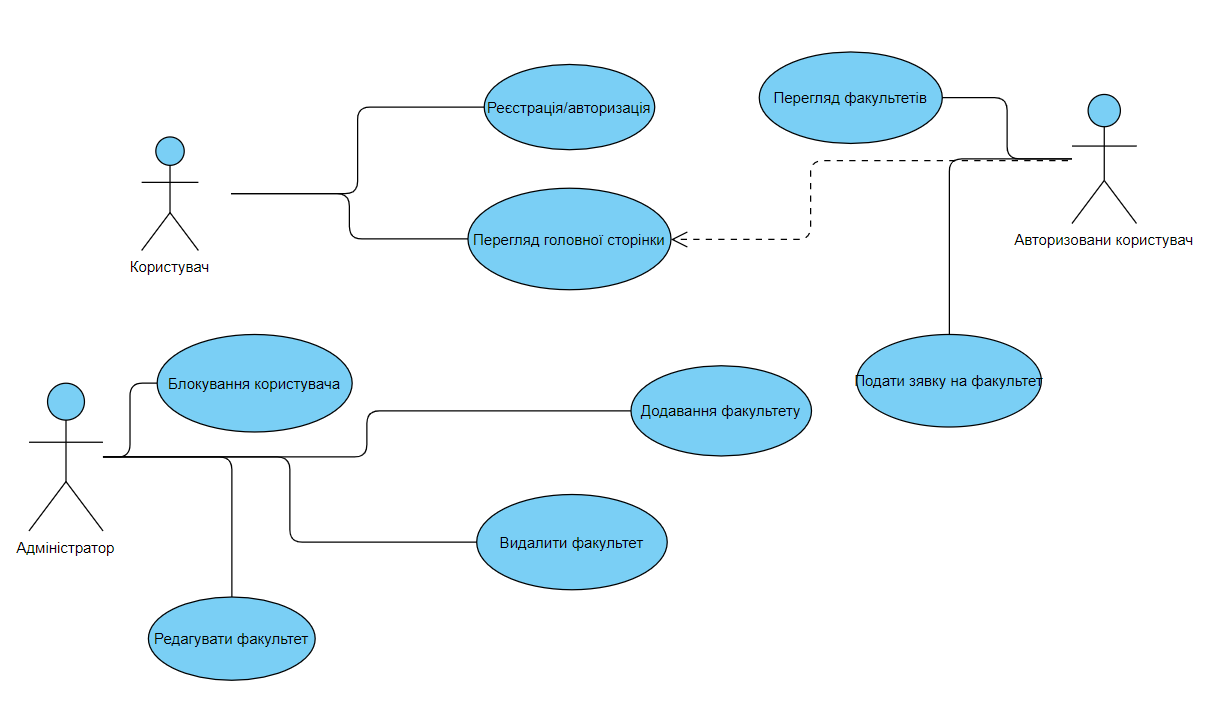


Рис. 1.1 Структурна схема варіантів використання web-додатку

Не зареєстрований користувач має змогу переглядати головну сторінку, щоб переглянути список можливих факультетів та подати заявку користувач повинен зареєструватись. Головний користувач сайту – це адміністратор. Він має змогу наповнювати БД системи новою інформацією, тобто даними про факультети, що з’явилися, їх опис тощо, та блокувати користувачів. (рис. 1.1).

РОЗДІЛ 2. РОЗРОБКА БАЗИ ДАНИХ

* 1. Що таке реляційні бази даних?

Поява комп'ютерної техніки в нашій сучасності ознаменувало інформаційний переворот у всіх сферах людської діяльності. Але для того, щоб вся інформація не стала непотрібним сміттям в глобальній мережі Інтернет, була винайдена система баз даних(СУБД), в якій матеріали сортуються, систематизуються, в результаті чого їх легко відшукати і уявити подальшій обробці.

База даних (БД) - спільно використовуваний набір логічно зв'язаних даних (і їх опис), призначений для задоволення інформаційних потреб організації.

СУБД (система управління базами даних) - програмне забезпечення, за допомогою якого користувачі можуть визначати, створювати і підтримувати базу даних, а також отримувати до неї контрольований доступ.

Системи управління базами даних існують вже багато років, багато хто з них зобов'язані своїм походженням системам з неструктурованими файлами на великих ЕОМ. Разом із загальноприйнятими сучасними технологіями в області систем управління базами даних починають з'являтися нові напрями, що обумовлено вимогами бізнесу, що росте, об'ємами корпоративних даних, що постійно збільшуються, і, звичайно ж, впливом технологій Internet.

Реляційна база даних — база даних, заснована на реляційній моделі даних. Така модель була розроблена в 1970-х роках доктором науки Едгаром Коддом. Вона являє собою логічно структуровану таблицю з полями, що описує дані, їх відносини між собою, операції, вироблені над ними, а головне - правила, які гарантують їх цілісність. Чому модель називається реляційної? В її основі лежать відносини (від лат. Relatio) між даними. Існує безліч визначень цього типу бази даних. Реляційні таблиці з інформацією набагато простіше систематизувати і надати обробці, ніж в мережевий або ієрархічної моделі. Як же це зробити? Досить знати особливості, структуру моделі і властивості реляційних таблиць. Для того щоб створити власну СУБД, слід скористатися одним з інструментів моделювання, продумати, з якою інформацією вам необхідно працювати, спроектувати таблиці і реляційні одно- і множинні зв'язку між даними, заповнити осередки сутностей і встановити первинний, зовнішні ключі.

Моделювання таблиць і проектування реляційних баз даних здійснюється за допомогою безкоштовних інструментів, таких як Workbench, PhpMyAdmin, Case Studio, dbForge Studio. Після детальної проектування слід зберегти графічно готову реляційну модель і перевести її в готовий SQL-код. На цьому етапі можна починати роботу з сортуванням даних, їх обробку і систематизацію.

Щоб відкрити вікно параметрів реляційної бази даних слід знати, з яких базових компонентів вона складається і для чого вони призначені.

Сутність. Таблиця реляційної бази даних може бути одна, а може бути цілий набір з таблиць, які характеризують описані об'єкти завдяки зберігаються в них даними. У них фіксовану кількість полів і змінне число записів. Таблиця реляційної моделі баз даних складається з рядків, атрибутів і макета.

Запис - змінне число рядків, що відображають дані, що характеризують описуваний об'єкт. Нумерація записів проводиться системою автоматично.

Атрибути - дані, що демонструють собою опис стовпців суті.

Поле. Являє собою стовпець суті. Їх кількість - фіксована величина, що встановлюється під час створення або зміни таблиці.

Тепер, знаючи складові елементи таблиці, можна переходити до властивостей реляційної моделі database:

* Суті реляційної БД двовимірні. Завдяки цій властивості з ними легко проробляти різні логічні і математичні операції.
* Порядок проходження значень атрибутів і записів в реляційної таблиці може бути довільним.
* Стовпець в межах однієї реляційної таблиці повинен мати свою індивідуальне назву.
* Всі дані в стовпці суті мають фіксовану довжину і однаковий тип.

Будь-який запис по суті вважається одним елементом даних.

* Складові компоненти рядків єдині в своєму роді. У реляційної сутності відсутні однакові рядки.

Існує 2 основних види зв'язків реляційних табличок:

* «Один-багато». Виникає при відповідно однією ключовою записи таблиці №1 декількох екземплярах другий сутності. Значок ключа на одному з кінців проведеної лінії говорить про те, що сутність знаходиться на стороні «один», другий кінець лінії часто відзначають символом нескінченності.
* Зв'язок «багато-багато» утворюється в разі виникнення між кількома рядками однієї сутності явного логічного взаємодії з низкою записів іншої таблиці.

Якщо між двома сутностями виникає конкатенація «один до одного», це означає, що ідентифікаційний ключ однієї таблиці присутній в інший сутності, тоді слід прибрати одну з таблиць, вона зайва. Але іноді виключно в цілях безпеки програмісти навмисно розділяють дві сутності. Тому гіпотетично зв'язок «один до одного» може існувати.

Базові правила нормалізації реляційної сутності

1. Значення назви поля для реляційної таблиці повинно бути унікальним, єдиним у своєму роді (перша нормальна форма - 1НФ).

2. Для таблиці, яка вже приведена до 1НФ, найменування будь-якого неідентіфікуючого поля має бути залежним від унікального ідентифікатора таблиці (2НФ).

3. Для всієї таблиці, що вже знаходиться в 2НФ, кожне неідентіфікуюче поле не може залежати від елемента іншого невпізнаного значення (3НФ суті).

* 1. Робота з MySQL

MySQL- така ж система, як Oracle, SQL Server. Ці продукти , як і будь – яка інша СУБД, дозволяють отримати доступ до даних, що зберігаються в їх базах даних, та маніпулювати цими даними, захищають дані від ушкодження та порушення їх цілосності, надають метадані, які необхідні для опису даних. Головною відмінністю між СУБД та РСУБД є те, що остання має справу лише з реляційними БД, підтримує зберігання даних у вигляді структур типа таблиць та відношень між таблицями

MySQL — вільна система керування реляційними базами даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. Зазначимо, що у вітчизняній літературі зустрічаються синомінічні абревіатури СУБД та СКБД. У даному підручнику перевага надається – СУБД.

Зараз 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. MySQLнадає багатий набір функціональних можливостей, які підтримують безпечне середовище для зберігання, обслуговування і отримання даних.

MySQL — характеризується великою швидкістю, стійкістю і простотою використання, була розроблена для підвищення швидкодії обробки великих баз даних.

Вихідні коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNІХ-системах, де є підтримка багатоканальності, що підвищує продуктивність системи в цілому.

Для некомерційного використання MySQL є безкоштовним.

Можливості сервера MySQL:

* простота у встановленні та використанні;
* підтримується необмежена кількість користувачів, що одночасно працюють із БД;
* кількість рядків у таблицях може досягати 50 млн.;
* висока швидкість виконання команд;
* наявність простої і ефективної системи безпеки.

Основні переваги MySQL. Серед основних переваг MySQL відзначають наступні:

* Масштабованість. MySQL може підтримувати роботу БД значних розмірів, що підтверджують її реалізації в Yahoo!, Google,HP, Associated Press. Згідно документації, що додається до MySQL, деякі БД, що використовуються компанією MySQL АВ (розробником MySQL), зберігають до 50 млн. записів.
* Переносність. MySQL працює на різних платформах, серед яких Unix, Linux, Windows, OS/2, Solaris, Mac OC. Окрім того, MySQL працює на різних платформах.
* Зв’язаність. MySQL має мережеву структуру. До MySQL можна одержувати доступ із будь-якої точки Internet кільком користувачам одночасно. MySQL має цілий ряд програмних інтерфейсів додатків (Application Programming Interface –API ), які дозволяють встановлювати з’єднання з MySQL із додатків, написаних на таких мовах як С, С++, Perl, PHP, Java, Python.
* Бeзпека. MySQL має систему контроля доступу до даних, забезпечує шифрування даних при передаванні.
* Швидкість функціонування.
* Зручність експлуатації. MySQL досить зручно встановлюється та реалізується, легко адмініструється.
* Відкритий код.
  1. Розробка архітектури програмної системи

Для розробки архітектури веб-сайту «Приймальна комісія» була взята клієнт-серверна архітектура додатку. Обрана архітектура найчастіше використовується в роботі з базами даних та мережі і забезпечує обмін даними між вказаними компонентами. Архітектура клієнт-сервер передбачає такі три основні компоненти:

* сервери, що обробляють отримані запити та видають відповідний результат;
* клієнти, що звертаються до серверів з запитами про дані;
* мережа, що забезпечує обмін даними між клієнтами та серверами.

Обробка та збереження даних відбувається на боці сервера, відображення даних і надсилання запитів на сервер виконується на боці клієнта. На рисунку 2.3.1 зображена трирівнева схема архітектури веб-додатку.

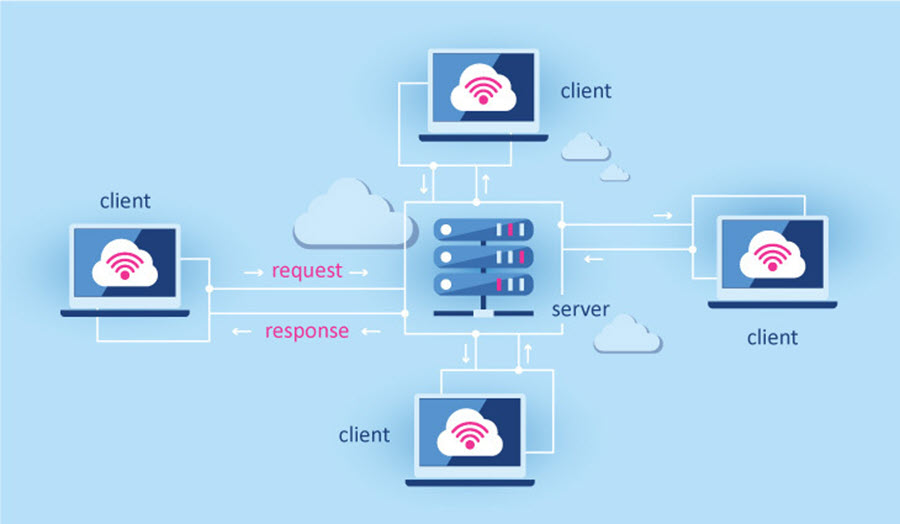


Рис.2.3.1 Трирівнева схема архітектури веб-додатку

Перший рівень – це клієнт, який відсилає запити на сервер та приймає результат обробки запитів. Клієнтом є браузер користувача.

Другий рівень – це бізнес-логіка додатку. Це логіка, за якою веб-сервер обробляє отримані від клієнта запити.

Третій рівень – це сама СУБД, яка отримує запити від сервера і повертає потрібні дані на сервер або зберігає їх.

* 1. Розробка структури бази даних

Для розробки своєї бази даних я обрала саме реляційну модель через ряд її переваг, а саме:

* Вона простота і доступна для розуміння користувачем. Єдиною використовуваною інформаційною конструкцією є «таблиця»;
* суворі правила проектування, які базуються на математичному апараті;
* повна незалежність даних. Зміни в прикладній програмі при зміні реляційної БД мінімальні;
* для організації запитів і написання прикладного ПЗ немає необхідності знати конкретну організацію БД у зовнішній пам'яті.

Діаграма концептуальної моделі даних представлена на рисунку 2.4.1

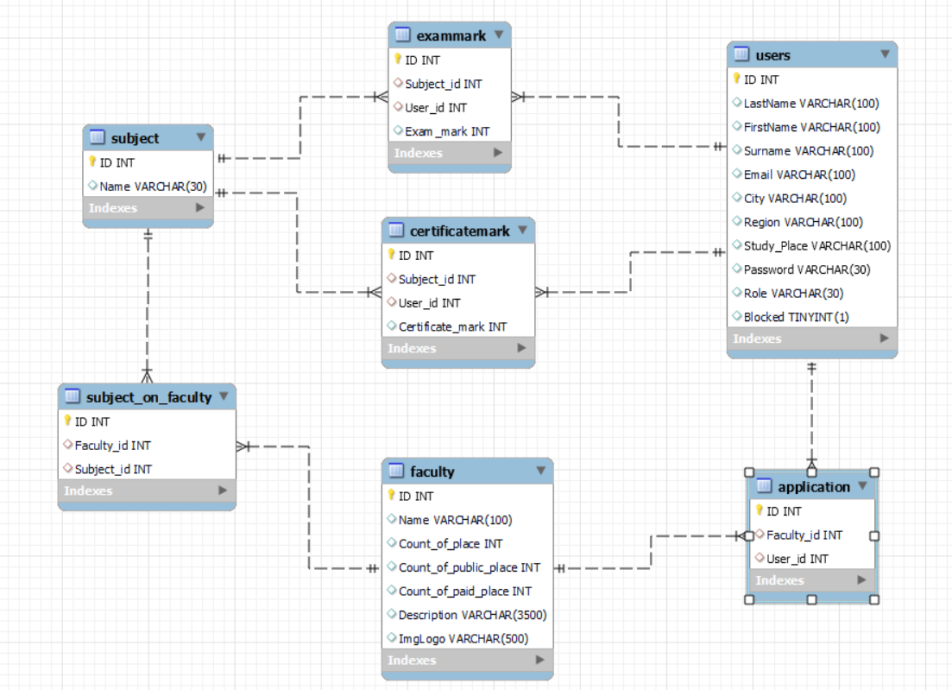


Рис.2.4.1 Концептуальна модель бази даних

* 1. Перелік таблиць бази даних

Таблиці бази даних для інформаційної системи приймальна комісія:

* user – інформація про абітурієнта, який зареєструвався;
* faculty- інформація про факультет
* subject- список предметів
* subject\_on\_faculty- предмети, що потрібні для вступу на факультет
* mark – таблиця збереження оцінок абітурієнта
* application- таблиця заявок на факультет
  1. Перелік полів таблиць бази даних

Поля, які ідентифікують властивості таблиці «user»:

* id- унікальне ід кожного користувача;
* LastName – прізвище користувача
* FirstName - ім’я користувача
* Surname – по батькові користувача
* Email –електронна адреса користувача,
* City – місто користувача
* Region – область користувача,
* Study\_Place- місце навчання користувача,
* Password - пароль,
* Role – роль користувача(юзер або адмін.)
* Blocked – заблокований/розблокований користувач

Поля, які ідентифікують властивості таблиці «faculty»:

* ID – унікальне ід факультету;
* Name – назва факультету ;
* Count\_of\_place – кількість місць на факультеті;
* Count\_of\_public\_place – кількість державних місць на факультеті;
* Count\_of\_paid\_place – кількість платних місць на факультеті;
* Description – опис факультету;
* ImgLogo – логотип.

Поля, які ідентифікують властивості таблиці «subject»:

* ID – унікальне ід кожного предмета;
* Name- назва предмета.

Поля, які ідентифікують властивості таблиці «subject\_on\_faculty»:

* Id- унікальне значення;
* Faculty\_id- ід факультету, на якому є предмет;
* Subject\_id- ід предмету, який є на факультеті.

Поля, які ідентифікують властивості таблиці «mark»:

* Id- унікальне значення;
* User\_id- ід користувача;
* Subject\_id- ід предмету;
* Exam\_mark- оцінка за екзамен з відповідного предмету;
* Certificate\_mark- оцінка за екзамен з відповідного предмету;

Поля, які ідентифікують властивості таблиці «application»:

* Id- унікальне значення;
* Faculty\_id- ід факультету;
* User\_id- ід користувача;
  1. Визначення типів даних таблиці бази даних

Перед створенням кожної таблиці бази даних необхідно визначити пити

даних кожного стовпця. В sql є безліч типів даних.

Основні типи даних в MYSQL:

* VARCHAR  – може зберігати не більше 255 символів(якщо кладете більше-перетворюється в TEXT).
* TIME –  час в форматі  ГГ: ХХ: СС.
* INT – діапазон від -2 147 483 648 до 2 147 483 647;

Типи даних в таблицях моєї бази даних:

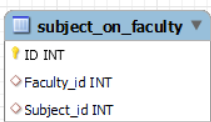


Рис.2.7.1 Таблиця «subject\_on\_faculty»

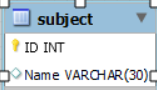


Рис.2.7.2 Таблиця «subject»

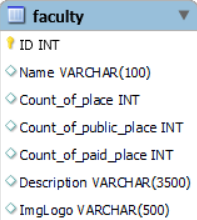


Рис.2.7.3 таблиця «faculty»



Рис.2.7.4 Таблиця «users»:

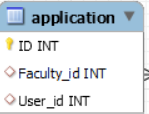


Рис.2.7.5 Таблиця «application»:

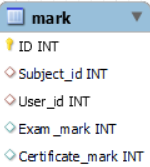


Рис.2.7.6 Таблиця «mark»:

* 1. Реалізація SQL-скрипту;

З допомогою запитів SQL можна отримувати практично будь-які необхідні дані з бази даних, тому вони зазвичай використовуються в якості джерела даних для звіту. Щоб реалізувати sql запит необхідно для початку під’єднатись до бази даних. Для java можна зробити наступним методом:

*Лістинг 2.8.1*

public class DBManager {

private final String CONTEXT\_LOOKUP;

private final String DATA\_SOURCE\_LOOKUP;

{

CONTEXT\_LOOKUP = "java:/comp/env";

DATA\_SOURCE\_LOOKUP = "jdbc/mysql";

}

private DataSource getDataSource() throws NamingException {

Context context = (Context) new InitialContext().lookup(CONTEXT\_LOOKUP);

return (DataSource) context.lookup(DATA\_SOURCE\_LOOKUP);

}

public Connection getConnection() throws NamingException, SQLException {

return getDataSource().getConnection();

}

Web-сайт реалізує такі запити до бази даних:

1. Запити типу «SELECT» :

* "select \* from users where email = ? and password = ?"
* "select \* from users where email = ?"
* "select subject.Name from subject\_on\_faculty join subject on Subject\_id = subject.ID join faculty on Faculty\_id = faculty.ID where Faculty\_id = ?"
* "select \* from faculty where id = ?"

Прикладом використання запиту SELECT у коді є:

*Лістинг 2.8.2*

public Faculty getFacultyById(int id) {  
 String sql ="select \* from faculty where id = ?";  
 ResultSet resultSet = null;  
 Faculty faculty = null;  
 try(Connection connection = dbManager.getConnection();  
 PreparedStatement statement = connection.prepareStatement(sql)) {  
 statement.setInt(1, id);  
 resultSet = statement.executeQuery();  
  
 if (resultSet.next()) {  
 faculty = new Faculty();  
 faculty.setId(resultSet.getInt("id"));  
 faculty.setName(resultSet.getString("name"));  
 faculty.setCount\_of\_places(resultSet.getInt("Count\_of\_place"));  
 faculty.setCount\_of\_paid\_places(resultSet.getInt("Count\_of\_paid\_place"));  
 faculty.setCount\_of\_public\_places(resultSet.getInt("Count\_of\_public\_place"));  
 }  
 } catch (SQLException | NamingException e) {  
 e.printStackTrace();  
 } finally {  
 assert resultSet != null;  
 close(resultSet);  
 }  
 return faculty;  
}

*Лістинг 2.8.3*

public List<Subject> getSubjectOnFacultyBYId(int id) {  
 String sql = "select subject.Name from subject\_on\_faculty join subject on Subject\_id = subject.ID join faculty on Faculty\_id = faculty.ID where Faculty\_id = ?";  
 ResultSet resultSet = null;  
 List<Subject> subjects = new ArrayList<>();  
  
 try (Connection connection = dbManager.getConnection();  
 PreparedStatement statement = connection.prepareStatement(sql)) {  
  
 statement.setInt(1, id);  
 resultSet = statement.executeQuery();  
  
 while (resultSet.next()) {  
 Subject subject = new Subject();  
  
 subject.setName(resultSet.getString("name"));  
  
 subjects.add(subject);  
  
 }  
 } catch (SQLException | NamingException e) {  
 e.printStackTrace();  
 }finally {  
 close(resultSet);  
 }  
 return subjects;  
 }  
}

*Лістинг 2.8.4*

public String checkUser(User user) {  
 String sql = "select \* from users where email = ? and password = ?";  
  
 ResultSet resultSet = null;  
  
 try (Connection connection = dbManager.getConnection();  
 PreparedStatement statement = connection.prepareStatement(sql)) {  
  
 statement.setString(1, user.getEmail());  
 statement.setString(2, user.getPassword());  
 statement.execute();  
  
 resultSet = statement.getResultSet();  
  
 if (resultSet.next()) {  
 return resultSet.getString("role");  
 }  
  
 } catch (SQLException | NamingException e) {  
 e.printStackTrace();  
 } finally {  
 close(resultSet);  
 }  
  
 return null;  
}

Запити типу «INSERT»:

* "insert into users values(default,?,?, ?, ?,?,?,?,?,default)"
* ,,,

Прикладом використання запиту INSERT у коді є:

*Лістинг 2.8.5*

public boolean addUser(Map<String, String[]> parametersMap) {  
 String sql = "insert into users values(default,?,?, ?, ?,?,?,?,?,default)";  
  
 try (Connection connection = dbManager.getConnection();  
 PreparedStatement statement = connection.prepareStatement(sql)) {  
  
 int k = 0;  
 statement.setString(++k, parametersMap.get(LASTNAME)[0]);  
 statement.setString(++k, parametersMap.get(FIRSTNAME)[0]);  
 statement.setString(++k, parametersMap.get(SURNAME)[0]);  
 statement.setString(++k, parametersMap.get(EMAIL)[0]);  
 statement.setString(++k, parametersMap.get(CITY)[0]);  
 statement.setString(++k, parametersMap.get(REGION)[0]);  
 statement.setString(++k, parametersMap.get(STUDY\_PLACE)[0]);  
 statement.setString(++k, parametersMap.get(PASSWORD)[0]);  
  
  
 statement.execute();  
 System.out.println(1);  
  
 } catch (SQLException | NamingException e) {  
 e.printStackTrace();  
 return false;  
 }  
  
 return true;  
}

Заповнення таблиця Faculty

insert into faculty value (default, "Факультет управління фінансами та бізнесу", 8, 3, 5, "Якщо ти хоч коли-небудь мріяв стати кваліфікованим фінансистом, чи менеджером, бухгалтером-ревізором, аудитором, працівником фіскальних органів і при цьому володіти комп’ютером, іноземною мовою та сучасними технологіями – то перед тобою відчиняються двері факультету управління фінансами та бізнесу. ", "https://upload.wikimedia.org/wikipedia/uk/thumb/7/74/Logo-lnu-financial.jpg/250px-Logo-lnu-financial.jpg" );

insert into faculty value (default, "Хімічний факультет", 8, 3, 5, "Нас оточують мільйони хімічних сполук, органічних і неорганічних, створених Природою і синтезованих Людиною. Завдання хімії – вивчати ці речовини та їхні перетворення, виявляти зв’язки між їхніми складом, структурою та властивостями.“Хімія наше життя – хімія наше майбутнє”.", "https://upload.wikimedia.org/wikipedia/commons/1/1b/Logo-lnu-faculty-chem.jpeg") ;

insert into faculty value (default, "Економічний факультет", 8, 3, 5,"Економіст – одна з найпрестижніших професій. Випускники нашого факультету є відомі в Україні та далеко за її межами вчені, політики, підприємці, педагоги. Нинішні студенти, засвоюючи знання з економіки, математики, іноземних мов та інформаційних технологій, творитимуть вітчизняну економічну та наукову еліту, якій буде під силу забезпечити гідне місце нашої держави у світовій економіці та протистояти складним викликам третього тисячоліття.", "https://admission.lnu.edu.ua/wp-content/uploads/2015/01/logo-faculty-economy.png");

insert into faculty value (default, "Факультет міжнародних відносин", 8, 3, 5,"На нашому факультеті ви отримаєте грунтовну підготовку з одного з напрямків міжнародних відносин (дипломатичної служби, міжнародного права, міжнародних економічних відносин, країнознавства, міжнародної інформації та міжнародного бізнесу), а також мовну підготовку принаймні з трьох іноземних мов. Вас чекає також інтенсивне та цікаве студентське життя: участь в студентських організаціях, можливість навчання закордоном, професійні практики, студентська наукова робота, спортивні секції.", "https://upload.wikimedia.org/wikipedia/uk/0/06/Logo-lnu-intrel.png");

insert into faculty value (default, "Факультет культури та мистецтв", 8, 3, 5,"Свій науковий, мистецько-науковий та науково-методичний доробок викладачі факультету публікують у міжнародних та українських наукових журналах, а також у монографіях, підручниках та навчальних посібниках. Щорічно на факультеті проводять наукові та студентські наукові конференції, де обговорюють сучасні проблеми відповідних галузей наук про культуру, мистецтво, бібліотечну та інформаційну справу.", "https://upload.wikimedia.org/wikipedia/commons/0/06/%D0%9A%D1%83%D0%BB%D1%8C%D1%82%D1%83%D1%80%D0%B0%D0%9B%D0%9C%D0%A3.jpg");

insert into faculty value (default, "Географічний факультет", 8, 3, 5, "Географічний факультет Львівського національного університету імені Івана Франка є провідним в Україні науковим і навчально-методичним центром підготовки спеціалістів високого рівня кваліфікації. Офіційно географічна освіта в Університеті започаткована у 1883 р. зі створення Інституту географії, який очолював відомий географ і геоботанік Антоній Реман. У 1945 р. на базі Інституту географії утворений географічний факультет.", "https://upload.wikimedia.org/wikipedia/uk/thumb/c/cd/LNU\_GEOGR.png/640px-LNU\_GEOGR.png");

insert into faculty value (default, "Юридичний факультет", 8, 3, 5,"Юридичний факультет Львівського національного університету імені Івана Франка – це не тільки місце підготовки нових кадрів – висококваліфікованих правників, а й установа, де формується еліта нації, а також твориться сучасна юридична наука. Поважаючи багатовікову історію та традиції, враховуючи виклики сьогодення, ми формуємо майбутнє!","https://upload.wikimedia.org/wikipedia/uk/thumb/e/e5/LNU\_JURID.png/640px-LNU\_JURID.png");

insert into faculty value (default, "Факультет прикладної математики та інформатики", 8, 3, 5, "Студенти факультету мають можливість отримати знання з усіх фундаментальних дисциплін, які складають основу комп’ютерних технологій, вив­чити основні мови програмування та ознайомитись із найсучаснішими IT-технологіями. Уже неодноразово команди студентів університету ставали призерами світових олімпіад з програмування.", "https://upload.wikimedia.org/wikipedia/uk/thumb/2/29/Logo-lnu-ami.jpeg/250px-Logo-lnu-ami.jpeg");

insert into faculty value (default, "Історичний факультет", 8, 3, 5, "Події, дати, особи становлять основу історичних знань, але не вичерпують їх. Кожен, хто стане студентом історичного факультету, навчиться характеризувати явища минулого в тих системах цінностей, у яких вони виникли, а тоді зіставляти їх із сучасними поглядами, моральними категоріями, визначати власне ставлення до подій. Іншими словами, навчиться розуміти історію, людей минулого, їхні турботи, потреби, радощі, надії, розчарування. А відтак навчиться розуміти себе й людей, котрі його оточують, будувати життєві плани і втілювати мрії в життя.", "https://upload.wikimedia.org/wikipedia/commons/8/81/%D0%86%D1%81%D1%82%D0%BE%D1%80%D0%B8%D1%87%D0%BD%D0%B8%D0%B9%D0%9B%D0%9D%D0%A32.gif");

insert into faculty value (default, "Біологічний факультет", 8, 3, 5,"Випускники біологічного факультету можуть працювати у науковій, екологічній, санітарно-епідеміологічній, сільськогосподарській сферах, харчовій та мікробіологічній промисловостях, рибництві, заповідній справі та природоохоронній діяльності, озелененні територій та парковому господарствах, в установах екологічної безпеки та надзвичайних ситуацій, у лабораторіях медико-діагностичного профілю і судово-медичної експертизи та закладах освіти всіх рівнів як в Україні, так і за кордоном","https://admission.lnu.edu.ua/wp-content/uploads/2015/01/logo-faculty-biology.png");

Заповнення таблиці Subject

insert into subject value (default,"Математика");  
insert into subject value (default,"Історія України");  
insert into subject value (default,"Українська мова та література");  
insert into subject value (default,"Хімія");  
insert into subject value (default,"Фізика");  
insert into subject value (default,"Біологія");  
insert into subject value (default,"Німецька мова");  
insert into subject value (default,"Англійська мова");

Заповнення таблиці Subject\_on\_faculty

insert into subject\_on\_faculty value (default, 1, 1);

insert into subject\_on\_faculty value (default, 1, 3);

insert into subject\_on\_faculty value (default, 1, 6);

insert into subject\_on\_faculty value (default, 2, 2);

insert into subject\_on\_faculty value (default, 2, 3);

insert into subject\_on\_faculty value (default, 2, 8);

insert into subject\_on\_faculty value (default, 3, 2);

insert into subject\_on\_faculty value (default, 3, 3);

insert into subject\_on\_faculty value (default, 4, 1);

insert into subject\_on\_faculty value (default, 4, 3);

insert into subject\_on\_faculty value (default, 4, 8);

insert into subject\_on\_faculty value (default, 5, 1);

insert into subject\_on\_faculty value (default, 5, 3);

insert into subject\_on\_faculty value (default, 5, 4);

insert into subject\_on\_faculty value (default, 7, 1);

insert into subject\_on\_faculty value (default, 7, 3);

insert into subject\_on\_faculty value (default, 7, 8);

insert into subject\_on\_faculty value (default, 8, 1);

insert into subject\_on\_faculty value (default, 8, 3);

insert into subject\_on\_faculty value (default, 8, 8);

insert into subject\_on\_faculty value (default, 9, 1);

insert into subject\_on\_faculty value (default, 9, 3);

insert into subject\_on\_faculty value (default, 9, 8);

РОЗДІЛ 3 « РОЗРОБКА ВЕБ-ДОДАТКУ»

* 1. «Структура веб-сайту»

В якості середовища для написання коду було обрано програмний продукт IntelliJ IDEA — використовується для редагування коду JavaScript, HTML, CSS, Java. За допомогою нього сайт розміщується на localhost.

Програмна реалізація веб-сервісу почалася з верстки шаблону сайту.

Верстка виконувалася мовою розмітки гіпертексту HTML використанням фреймворку Bootstrap. Для надання

розмітці стилів використовувалися каскадні таблиці стилів CSS.

Дана програма реалізована у вигляді web-сайту, який працює через будь-який доступний браузер.

Структура головної сторінки сайту приймальної комісії(рис.3.1.1.)

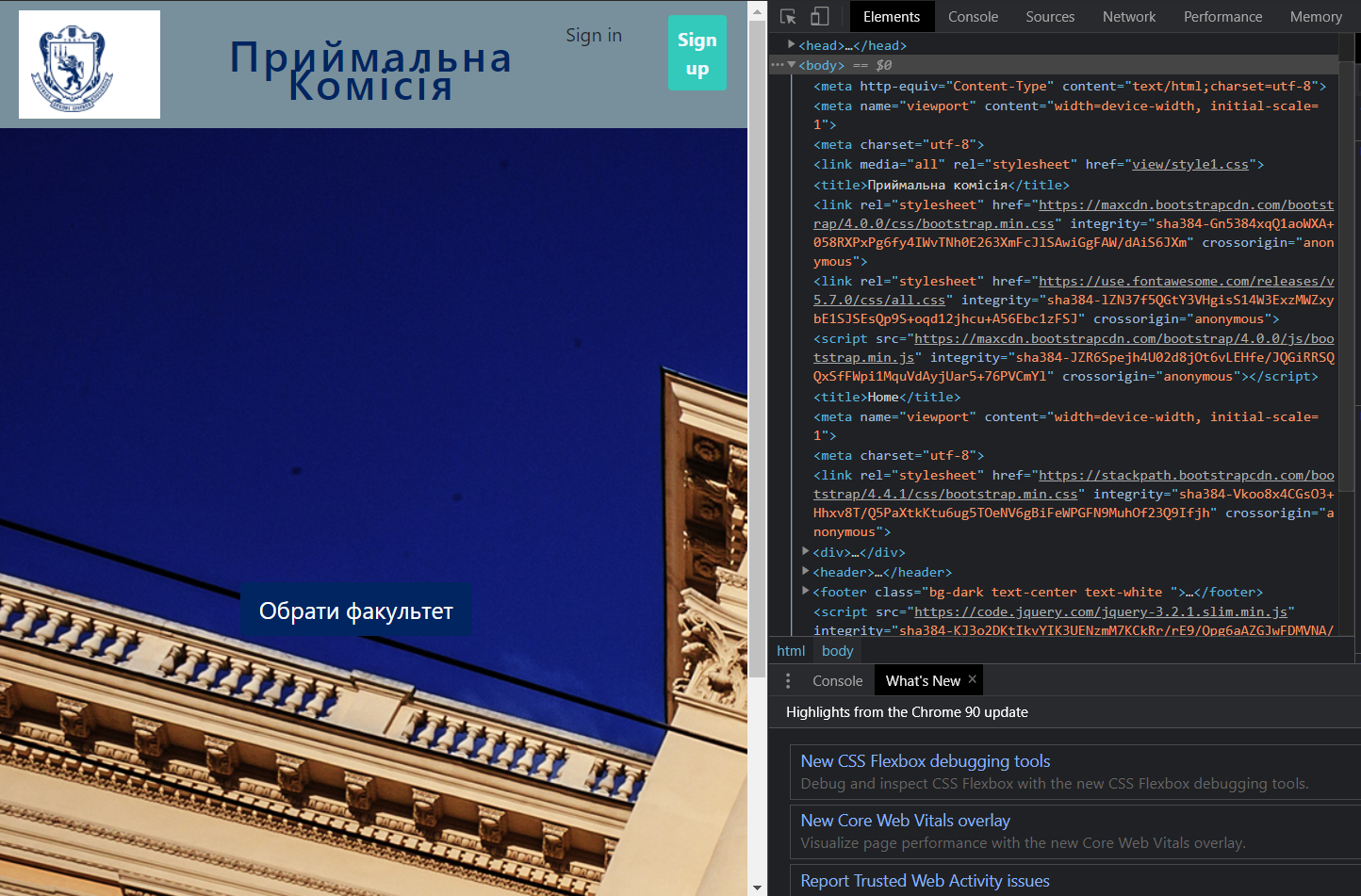


Рис.3.1.1. Головна сторінка сайту

* 1. Макет веб-сторінки

Клієнтська частина має такі головні сторінки:

* Sign in – сторінка, яка містить в собі форму авторизації на особистий профіль користувача(Додаток А);
* Sign up – сторінка, яка містить форму реєстрації, за допомогою якої користувач може зареєструватись в базу даних комісії(Додаток Б);
* Мій кабінет – містить інформацію про користувача(Додаток В);
* Головна сторінка зареєстрованого користувача містить список факультетів(рис.3.2.1);



Рис.3.2.1 Спиисок факультетів

* Інформація про факультет – містить повну інформацію про конкретний факультет(Додаток Г)
* Подання заявки на факультет – форма заявки (Додаток Д)

Адміністраторська сторона окрім вище згаданого на головній сторінці містить:

* Список всіх користувачів(рис.3.2.7);

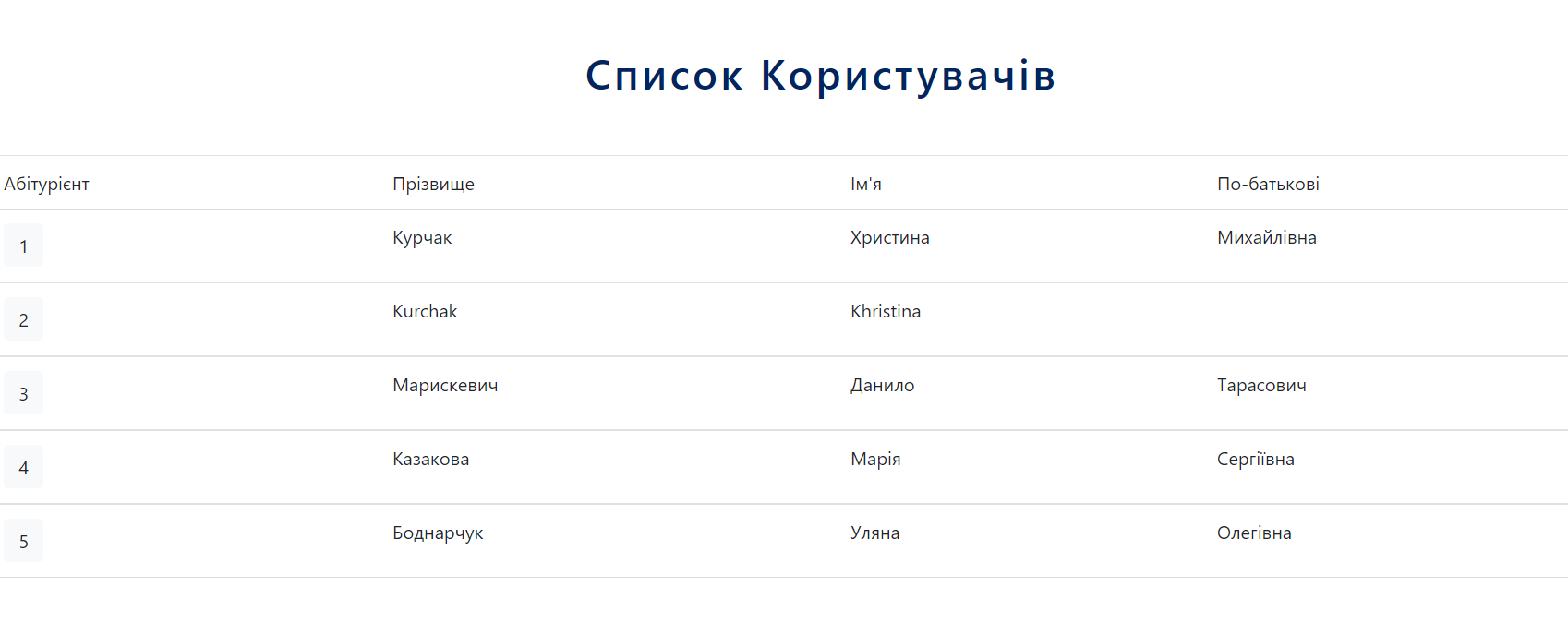


Рис.3.2.7 Список користувачів

* Інформацію про кожного конкретного користувача(Додаток Е)
* Форму редагування та видалення факультету(Додатки Ж і З)
  1. Програмування користувацької частини

В практичній частині серверна частина інформаційної системи(веб-сайту) для приймальної комісії створена на мові java.

Для прикладу розглянемо частину коду- сервлет для форми редагування/видалення факультету.

*Лістінг 3.3.1*

@WebServlet(name = "FacultyListServlet", value = "/facultyList")

public class FacultyListServlet extends HttpServlet {

private FacultyDao facultyDao;

@Override

public void init() throws ServletException {

facultyDao = new FacultyDao();

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String action = request.getParameter("action");

switch (action) {

case "delete":

deleteUser(request, response);

break;

case "update":

updateUser(request, response);

break;

default:

RequestDispatcher dispatcher = request.getRequestDispatcher("admin");

dispatcher.forward(request, response);

break;

}

}

private void updateUser(HttpServletRequest request, HttpServletResponse response) throws IOException {

String name = request.getParameter("name");

System.out.println(name);

int countOfPlace = Integer.parseInt(request.getParameter("count\_of\_places"));

System.out.println("co " + countOfPlace);

int countOfPublicPlace = Integer.parseInt(request.getParameter("count\_of\_public\_places"));

System.out.println("co " + countOfPublicPlace);

int countOfPaidPlace = Integer.parseInt(request.getParameter("count\_of\_paid\_places"));

System.out.println("co " + countOfPaidPlace);

String decription = request.getParameter("description");

System.out.println("co " + decription);

String logo = request.getParameter("logo");

int id = Integer.parseInt(request.getParameter("facultyID"));

System.out.println("id" + id);

Faculty updateFaculty = new Faculty(name, countOfPlace, countOfPublicPlace, countOfPaidPlace, decription, logo, id);

facultyDao.updateFaculty(updateFaculty);

response.sendRedirect("admin");

}

private void deleteUser(HttpServletRequest request, HttpServletResponse response) throws IOException {

int id = Integer.parseInt(request.getParameter("facultyID"));

facultyDao.deleteFacultyById(id);

response.sendRedirect("admin");

}

}

Вище наведена частина коду виконує відправлення даних з форми в базу даних відповідно до того факультету, який ми хочемо редагувати чи видалити. У разі успішного надсилання програма переходить на

основу сторінку сайту.

* 1. Програмування клієнтської частини

Програмування клієнтської частини є важливим етапом в створені веб-додатку, адже перше, що бачить користувач, коли попадає на веб-сторінку, це саме Front-end частину. Клієнтська частина повинна взаємодіяти з користувачем, передавати запити користувача серверу, отримувати результату запиту від серверної частини (програми-сервера) і представляти його в зручному та зрозумілому вигляді. В своїй я роботі я написало чимало коду для цієї частини, тому розглянемо лише приклад саме головної сторінки.

*Лістинг 3.4.1*

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ page contentType="text/html;charset=utf-8" %>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html;charset=utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta charset="utf-8">

<link media="all" rel="stylesheet" href="view/style1.css">

<title>Приймальна комісія</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.css"

integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"

integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"

crossorigin="anonymous"></script>

</head>

<body>

<%@include file="header.jsp"%>

<header>

<div class="section" id="home">

<div class="row">

<div class="center">

<c:choose>

<c:when test="${username == null}">

<a href="#faculty" class="button button2 btn-lg" data-toggle="modal">Обрати факультет</a>

</div>

<!--Grid column-->

</div>

<!-- Modal -->

<div class="modal fade" id="faculty" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle"

aria-hidden="true">

<div class="modal-dialog modal-dialog-centered" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLongTitle">Попередження</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

Щоб обрати факультет потрібно увіти в систему

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</c:when>

<c:otherwise>

<a href="facultyList#faculty" class="button button2 btn-lg" >Обрати факультет</a>

</div>

</c:otherwise>

</c:choose>

</header>

</body>

<footer class="bg-dark text-center text-white "> <!--'fixed-bottom'-->

<!-- Grid container -->

<div class="container p-2">

<!-- Section: Social media -->

<section class="mb-2">

<!-- Facebook -->

<a class="btn btn-outline-light btn-circle m-1" href="#" role="button">

<i class="fab fa-facebook-f"></i>

</a>

<!-- Twitter -->

<a class="btn btn-outline-light btn-circle m-1" href="#" role="button"

><i class="fab fa-twitter"></i

></a>

<!-- Google -->

<a class="btn btn-outline-light btn-circle m-1" href="#" role="button"

><i class="fab fa-google"></i

></a>

<!-- Instagram -->

<a class="btn btn-outline-light btn-circle m-1" href="#" role="button"

><i class="fab fa-instagram"></i

></a>

<!-- Linkedin -->

<a class="btn btn-outline-light btn-circle m-1" href="#" role="button"

><i class="fab fa-linkedin-in"></i

></a>

</section>

</div>

<!-- Grid container -->

<!-- Copyright -->

<div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2);">

© 2021 Copyright:

<a class="text-white" href="/">SelectionCommittee.com</a>

</div>

<!-- Copyright -->

</footer>

</html>

Потрібно зазначити, що програмування клієнтської частини є таким ж важливим, як і програмування серверної частини, адже одне без іншого гарно працювати не зможе.

ВИСНОВКИ

Під час виконання курсової роботи проаналізовано можливості використання HTML, CSS, JS, Java та роботу із базами даних. Для закріплення набутих знань розроблено базу даних приймальної комісії, а також веб-додаток, який дає змогу подати заявку на факультети оналайн.

На початковому етапі виконання було проаналізовано предметну область, поставлена мета і завдання у відповідності до змісту роботи та виданого завдання, а саме для інформаційної системи приймальної комісії.

Після цього у другому розділі була реалізовано база даних. А саме: детальний аналіз предметної області, ER-модель, декомпозиція та нормалізація відношень методом Сутність-зв'язок. Також створено універсальне відношення, визначені первинний і вторинний ключі. Створено 6 таблиць(«users», «faculty», «subject», «subject\_on\_faculty mark», «application»). Для опису використано деякі sql-скрпти та реалізація підключення бази даних.

В останньому розділі описано функціонал веб-додатку, а також висвітлено зовнішній вигляд сайту, його макет. Наведені приклади коду програмування користувацької та серверної частини для кращого розуміння роботи веб-додатку.

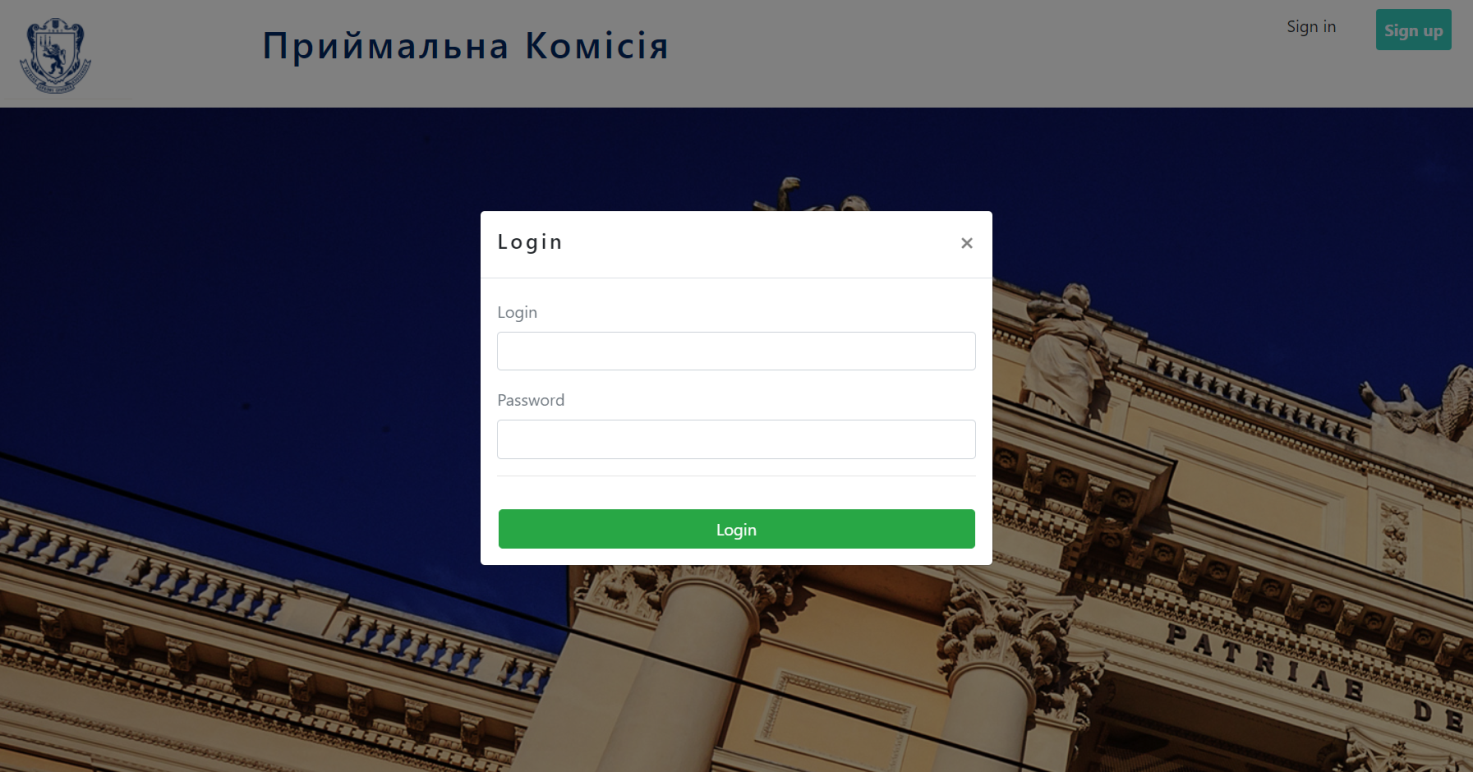
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документация по MySQL [Електронний ресурс]. – Режим доступу: <http://www.mysql.ru/docs/>
2. Сучасна інформаційна еволюція і особистість [Електронний ресурс].– Режим доступу: <http://www.nbuviap.gov.ua/index.php?option=com_content&view=article&id=1870:informatizatsiya-i-osobistist&catid=8&Itemid=350>
3. Bootstrap [Електронний ресурс]. – Режим доступу: <https://timeweb.com/ru/community/articles/plyusy-i-minusy-bootstrap-1>.
4. CSS styles[Електронний ресурс]. - Режим доступу: <http://htmlbook.ru/css>
5. Пасічник В. В. Організація баз даних та знань / В.В. Пасічник, В.А. Резніченко.– К.: Видавнича група BHV, 2017. – 384 с.
6. Вікіпедія. [Електронний ресурс]. - Режим доступу: <https://uk.wikipedia.org>
7. W3SCHOOLS ONLINE WEB TUTORIALS [Електронний ресурс]. – Режим доступу: <https://www.w3schools.com/>
8. ЧТО ТАКОЕ СУБД - RU-CENTER [Електронний ресурс]. – Режим доступу: <https://www.nic.ru/help/chto-takoe-subd_8580.html>

ДОДАТКИ

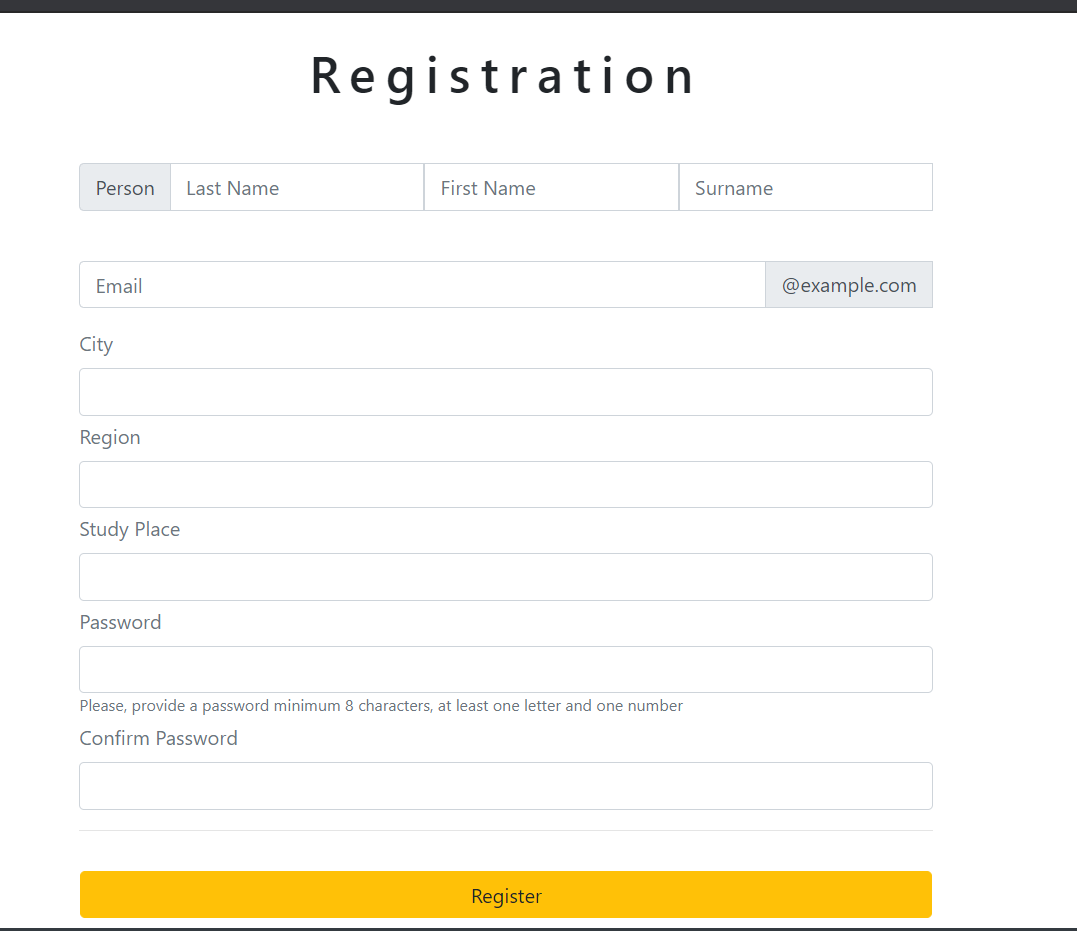
Додаток А

Авторизація



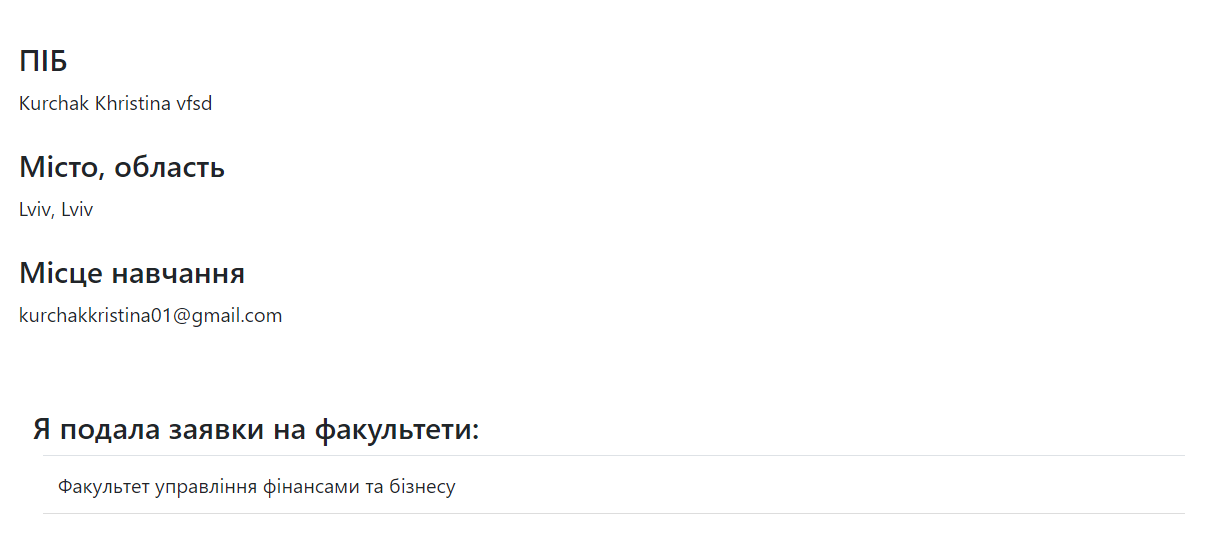
Додаток Б

Реєстрація



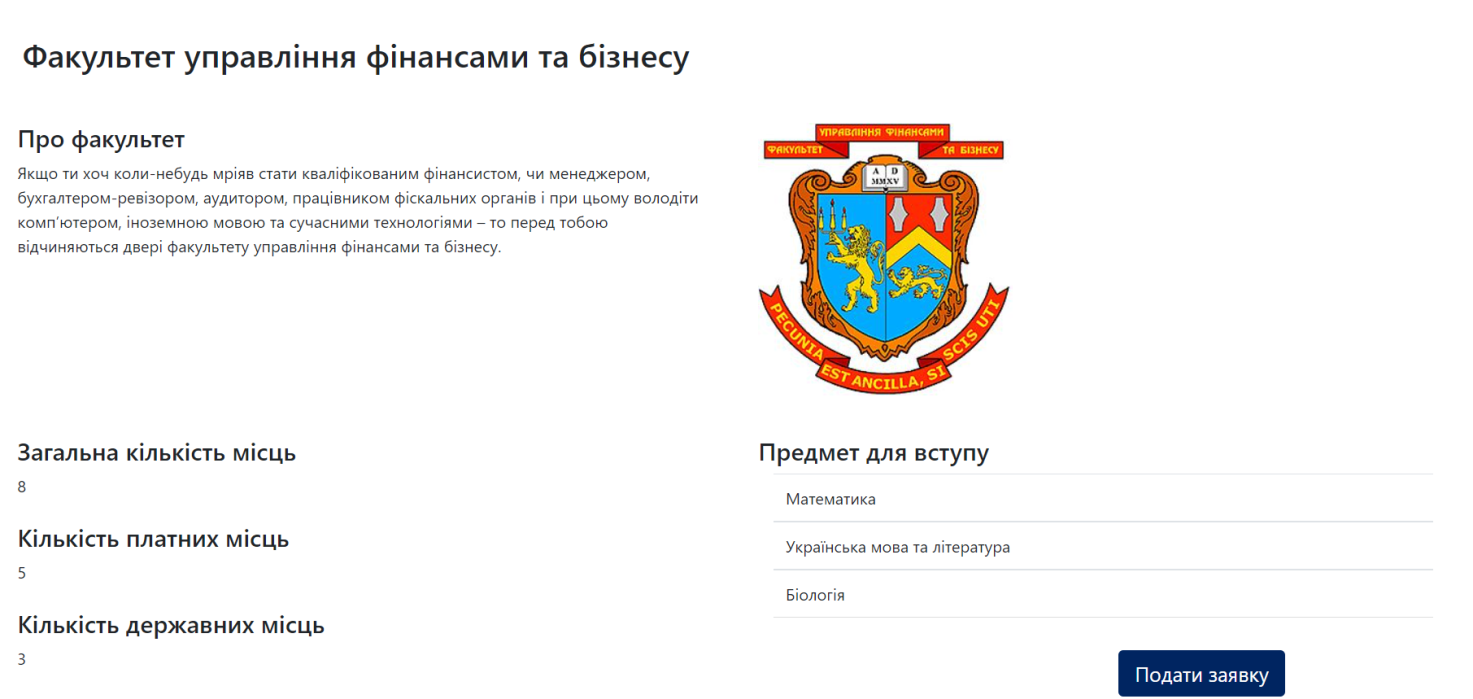
Додаток В

Мій кабінет



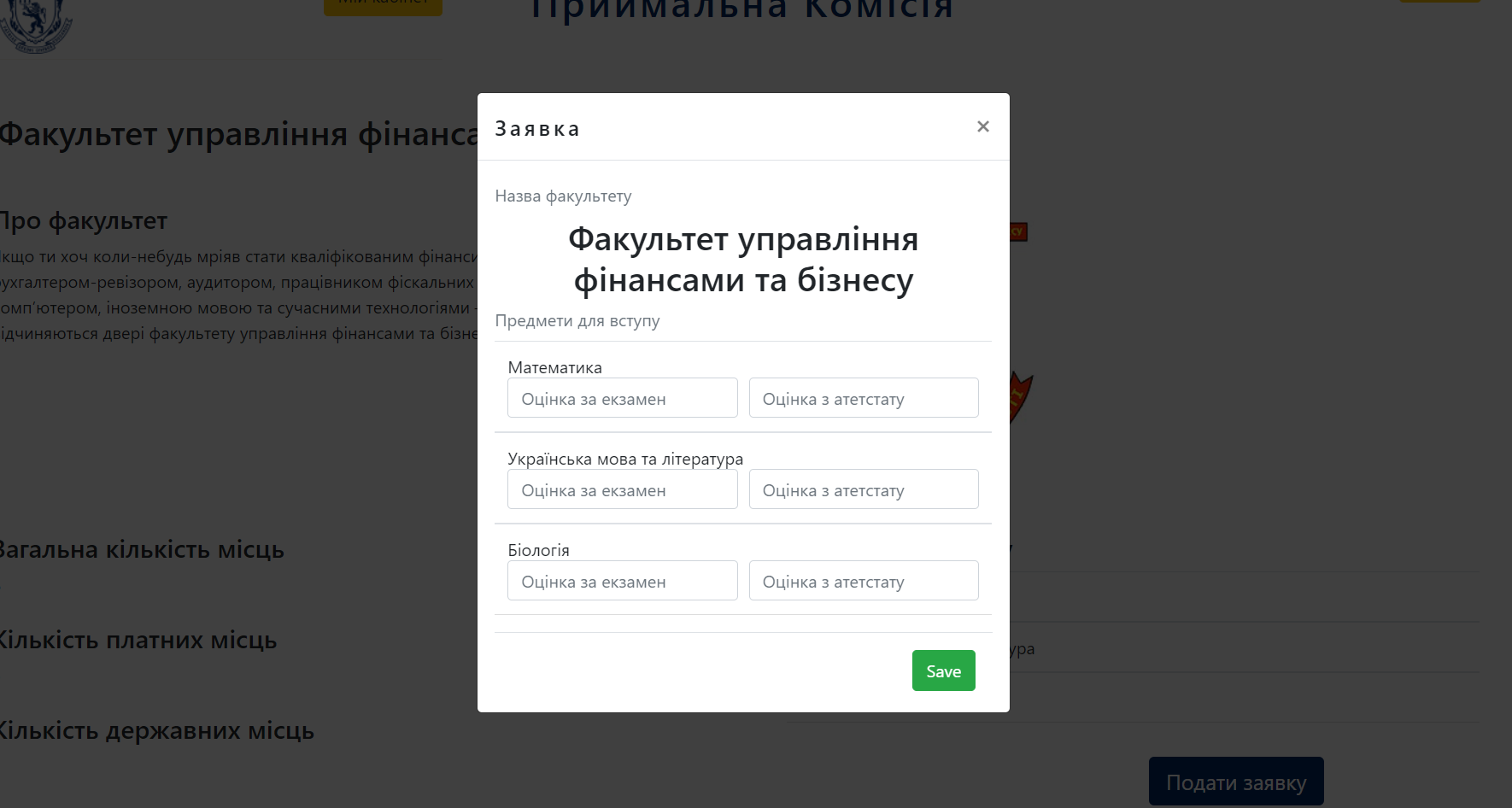
Додаток Г

Інформація про факультет(користувач)



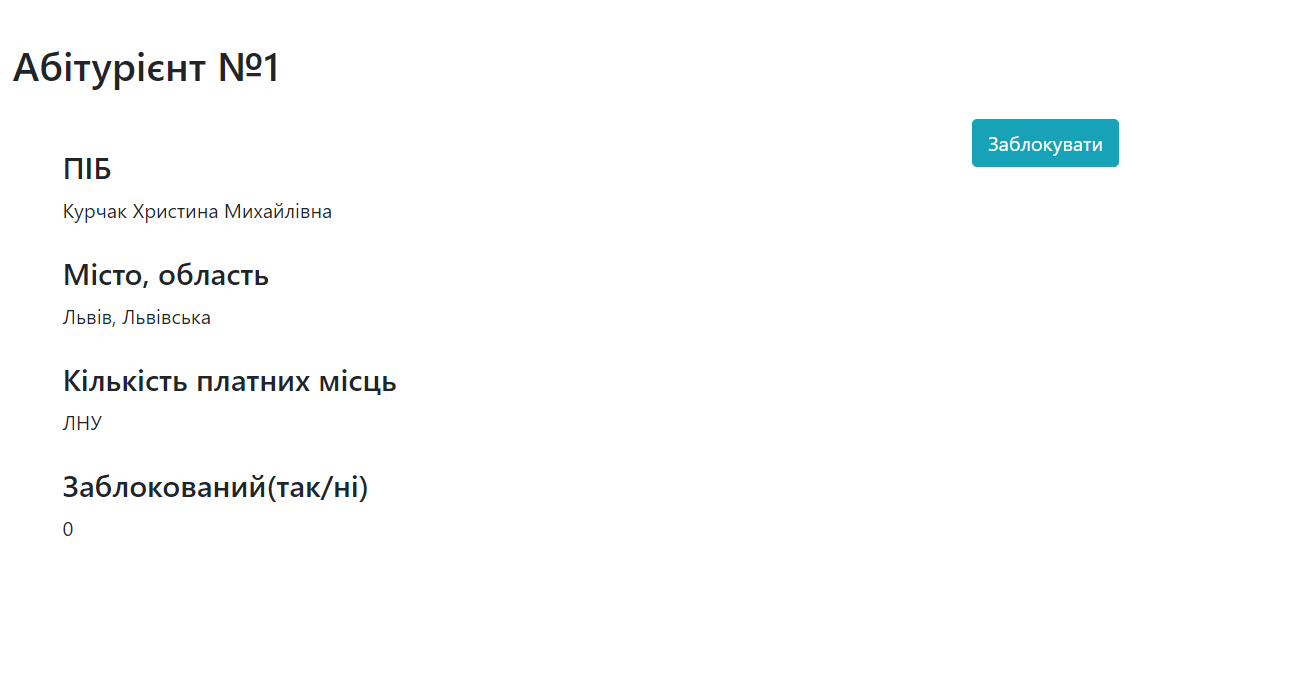
Додаток Д

Подання заявки на факультет



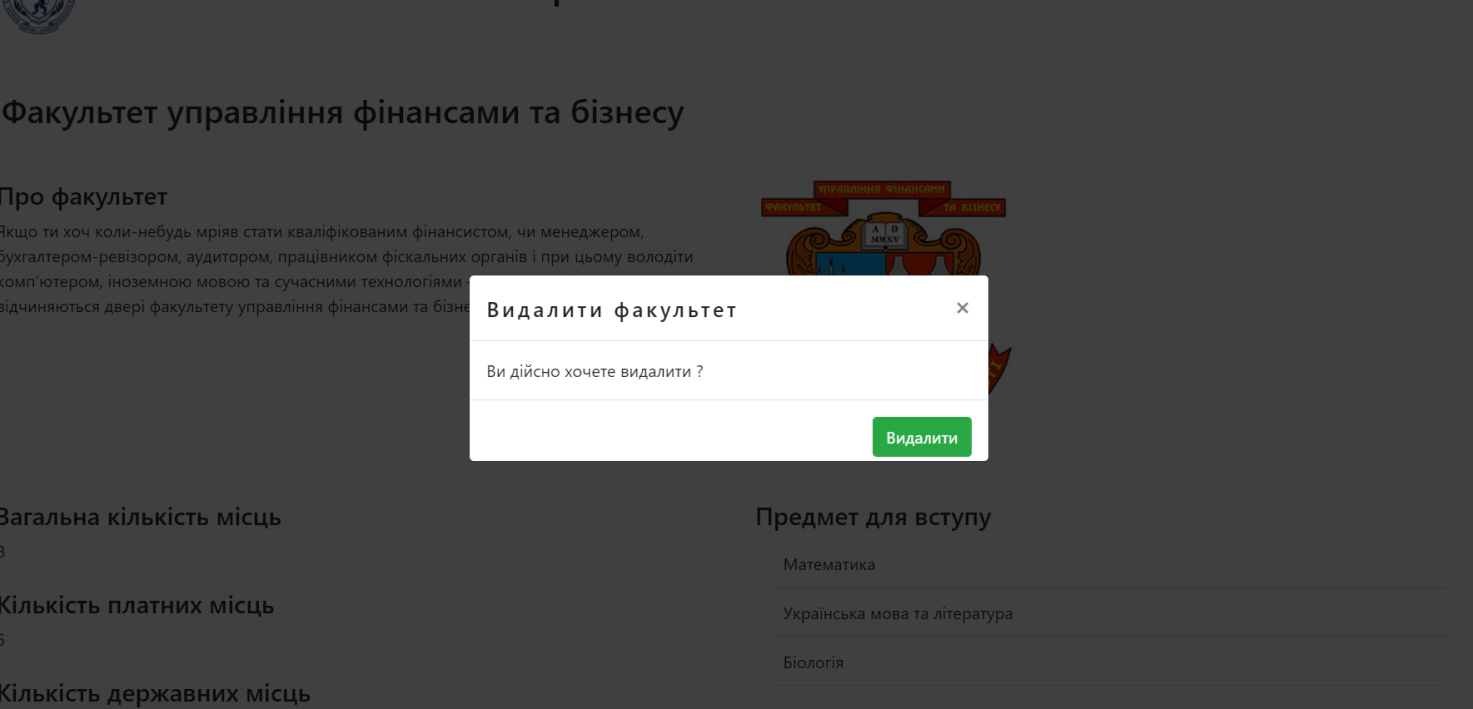
Додаток Е

Інформацію про користувача(адміністратор)



Додаток Ж

Видалення факультету(адміністратор)



Додаток З

Редагування факультету(адміністратор)

