

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Розробка програмного забезпечення мовою C# для  
обліку та аналізу даних рейтингу викладача»

на здобуття освітнього ступеня бакалавра  
зі спеціальності 121 Інженерія програмного забезпечення  
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень. Використання  
ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ Євген КУЗЬМЕНКО  
(підпис)

Виконав: здобувач вищої освіти групи ППЗ-51

\_\_\_\_\_ Євген КУЗЬМЕНКО

Керівник: \_\_\_\_\_ Віктор ГРЕБЕНЮК  
доктор філософії (PhD)

Рецензент: \_\_\_\_\_

**Київ 2024**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ Ірина ЗАМРІЙ

« \_\_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Кузьменко Євгену Євгеновичу

1. Тема кваліфікаційної роботи: «Розробка програмного забезпечення мовою С# для обліку та аналізу даних рейтингу викладача»

керівник кваліфікаційної роботи доктор філософії (PhD), доцент кафедри ІПЗ Віктор ГРЕБЕНЮК, затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. №36.

2. Строк подання кваліфікаційної роботи «28» травня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: Теоретичні та практичні навички розробки веб-додатків з використанням мови С# та фреймворку ASP.NET, розуміння принципів клієнт-серверної архітектури та досвід у тестуванні програмного забезпечення.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз предметної галузі.

2. Проектування програмного застосунку.

3. Розробка програмного застосунку.

4. Тестування програмного застосунку.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів
2. Ключові вимоги до web-застосунку
3. Програмні засоби реалізації
4. Діаграма прецедентів
5. Діаграма класів
6. Схема баз даних
7. Апробація результатів дослідження

6. Дата видачі завдання «28» лютого 2024 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02-06.03.2024	
2	Аналіз та дослідження існуючих аналогів	07.03-13.03.2024	
3	Вивчення матеріалів для аналізу існуючих рішень для обліку та аналізу рейтингу викладача	15.03-31.03.2024	
4	Аналіз переваг та недоліків існуючих програмних засобів	01.04-10.04.24	
5	Дослідження функціональних та нефункціональних вимог до застосунку	11.04-04.05.24	
6	Проектування інтерфейсу користувача	05.05-10.05.24	
7	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.24	
8	Розробка демонстраційних матеріалів	06.05-12.05.24	
9	Попередній захист роботи	13.05-31.05.2024	

Здобувач вищої освіти

\_\_\_\_\_ (підпис)

Євген КУЗЬМЕНКО

Керівник кваліфікаційної роботи

\_\_\_\_\_ (підпис)

Віктор ГРЕБЕНЮК





## РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 66 стор., 1 табл., 40 рис., 2 джерела.

*Мета роботи* – автоматизація процесу обліку та аналізу даних рейтингу викладачів.

*Об'єкт дослідження* – облік та аналіз даних рейтингу викладачів.

*Предмет дослідження* – програмне забезпечення для обліку та аналізу даних рейтингу викладачів.

*Короткий зміст роботи:* У роботі проведено огляд існуючих програмних застосунків для обліку та аналізу рейтингу викладача. Складена порівняльна таблиця існуючих програмних застосунків. Спроектвана архітектура бази даних, діаграми класів та прецедентів. В результаті розроблено застосунок для підтримки самостійної роботи здобувачів вищої освіти.

**КЛЮЧОВІ СЛОВА:** КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ДІАГРАМИ КЛАСІВ, ВЕБ ТЕХНОЛОГІЇ, БАЗИ ДАНИХ, АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.

## ЗМІСТ

<b>ВСТУП .....</b>	<b>10</b>
<b>1. АНАЛІЗ І ВИДІЛЕННЯ ПЕРЕВАГИ РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ.....</b>	<b>11</b>
1.1 Огляд існуючих програмних застосунків.....	11
1.1.1 Настільний додаток Microsoft Word .....	11
1.1.2 Настільний додаток Microsoft Excel .....	12
1.1.3 Порівняння переваг та недоліків існуючих додатків .....	13
<b>2. Проектування програмного застосунку .....</b>	<b>15</b>
2.1 Діаграма прецедентів.....	15
2.2 Діаграма класів.....	15
2.3 Схема структури баз даних.....	16
<b>3. РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ .....</b>	<b>19</b>
3.1 Вибір засобів програмної реалізації.....	19
3.1.1 Інтегроване середовище розробки Visual Studio .....	19
3.1.2 Мова програмування C#.....	19
3.1.3 ASP.NET .....	21
3.1.4 Entity Framework .....	22
3.1.5 MySQL .....	23
3.2 Використані архітектурні паттерни .....	25
3.2.1 Паттерн Модель - Вид – Контролер.....	25
3.2.2 Клієнт-серверна архітектура.....	26
<b>4. СЦЕНАРІЇ РОБОТИ РОЗРОБЛЕНОГО ЗАСТОСУНКУ .....</b>	<b>28</b>
4.1 Головна сторінка додатку .....	28
4.2 Реєстрація .....	28
4.3 Вхід.....	29
4.4 Список інститутів .....	29
4.5 Додати інститут.....	30
4.6 Редагувати інститут .....	30

4.7 Видалити інститут .....	31
4.8 Список кафедр.....	31
4.9 Додати кафедру .....	32
4.10 Редагувати кафедру .....	32
4.11 Видалити кафедру.....	33
4.12 Список посад .....	33
4.13 Додати посаду .....	34
4.14 Редагувати посаду.....	34
4.15 Видалити посаду .....	35
4.16 Список типів метрик.....	35
4.17 Додати тип метрики.....	36
4.18 Редагувати тип метрики .....	36
4.19 Видалити тип метрики .....	37
4.20 Список метрик.....	37
4.21 Додати метрику.....	38
4.22 Редагувати метрику .....	38
4.23 Видалити метрику.....	39
4.24 Рейтингова картка.....	39
4.25 Рейтинг викладачів університету.....	40
4.26 Рейтинг викладачів певної кафедри.....	40
4.27 Рейтинг викладачів кафедри.....	41
4.28 Рейтинг інститутів університету .....	41
4.29 Рейтинг кафедр університету .....	42
<b>ВИСНОВКИ.....</b>	<b>43</b>
<b>ПЕРЕЛІК ПОСИЛАНЬ .....</b>	<b>44</b>
<b>ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....</b>	<b>48</b>



## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

НПП - Науково-педагогічний працівник

ПК - Персональний комп'ютер

MVC - Архітектурний шаблон Модель – Вид - Контролер

## ВСТУП

У сучасному навчальному середовищі все частіше постає питання кар'єрного росту викладацького колективу але облік та аналіз результатів кожного викладача може забрати цінний час у який вони б могли навчати нове покоління студентів.

Процес обліку та аналізу даних рейтингу викладачів досі проводиться у електронних таблицях або текстових редакторах. У деяких університетах навіть використовуюся паперові носії. Така стратегія є морально застарілою.

Тому для зменшення навантаження на людей задіяних у цьому процесі був розроблений застосунок для автоматизації обліку та аналізу рейтингу викладачів. Він додає такі зручні інструменти як автоматичний розрахунок рейтингової картки викладача, аналіз рейтингу викладачів по всьому університету або окремій кафедрі, а також доступ до рейтингу усіх кафедр та інститутів.

Розробка велась за допомогою сучасної мови програмування C# та фреймворку ASP.NET тому він працює дуже швидко і не потребує серверного обладнання останніх поколінь. Він забезпечує простоту використання, швидкість роботи та захищеність так як доступний тільки з локальної мережі університету.

# 1. АНАЛІЗ І ВИДІЛЕННЯ ПЕРЕВАГИ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАСТОСУНКУ

## 1.1 Огляд існуючих програмних застосунків

### 1.1.1 Настільний додаток Microsoft Word

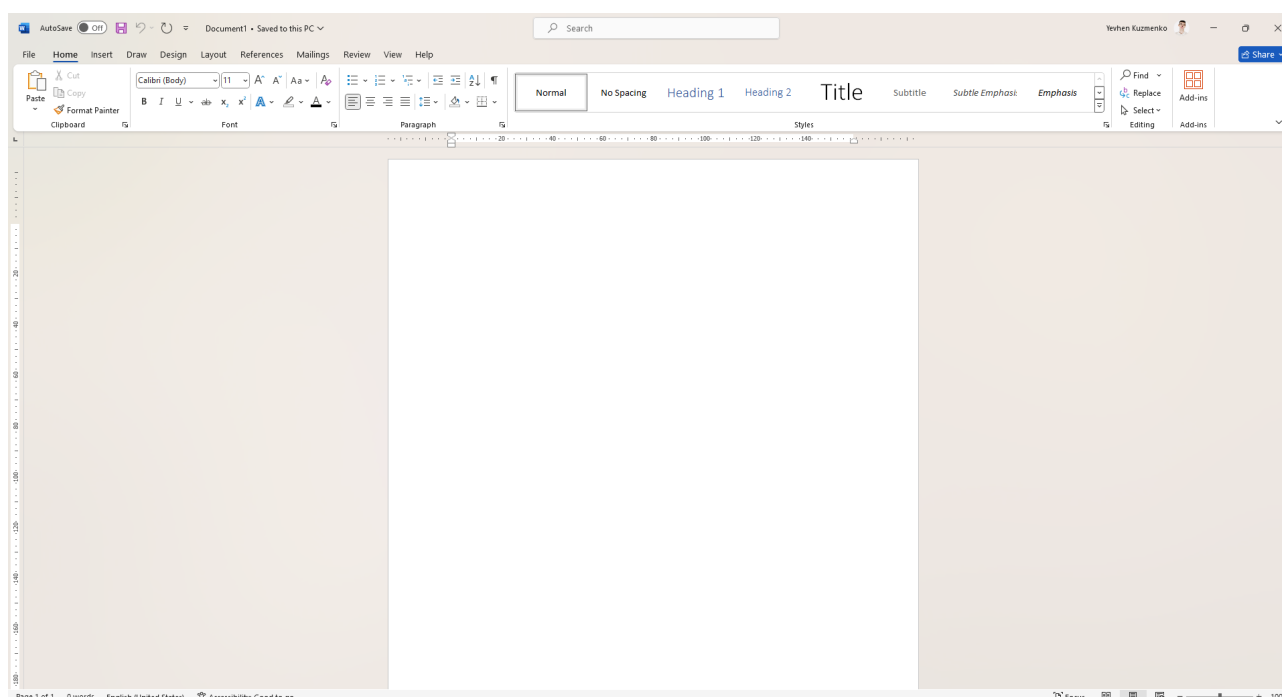


Рис. 1.1 – Інтерфейс додатку Microsoft Word

Microsoft Word є одним із найпопулярніших текстових редакторів у світі, широко використовуваним для створення, редагування та форматування текстових документів. Основними перевагами Word є прості та зрозумілі інструменти для створення та редагування тексту, можливість створення шаблонів для стандартних документів, вбудовані функції перевірки правопису та граматики, а також підтримка великої кількості форматів файлів для експорту та імпорту.

Переваги:

- Зручний інтерфейс.
- Підтримка платформ Windows, MacOS.

Недоліки:

- Немає можливості маніпулювати масивами даних.
- Дані зберігаються на ПК власника документу тому додавання, зміна, видалення даних вимагає пересилки документу іншій особі.
- Ціна додатку - 1899 грн на рік за 1 користувача,  $1899 * 50$  викладачів = 94950 грн на рік

### 1.1.2 Настільний додаток Microsoft Excel

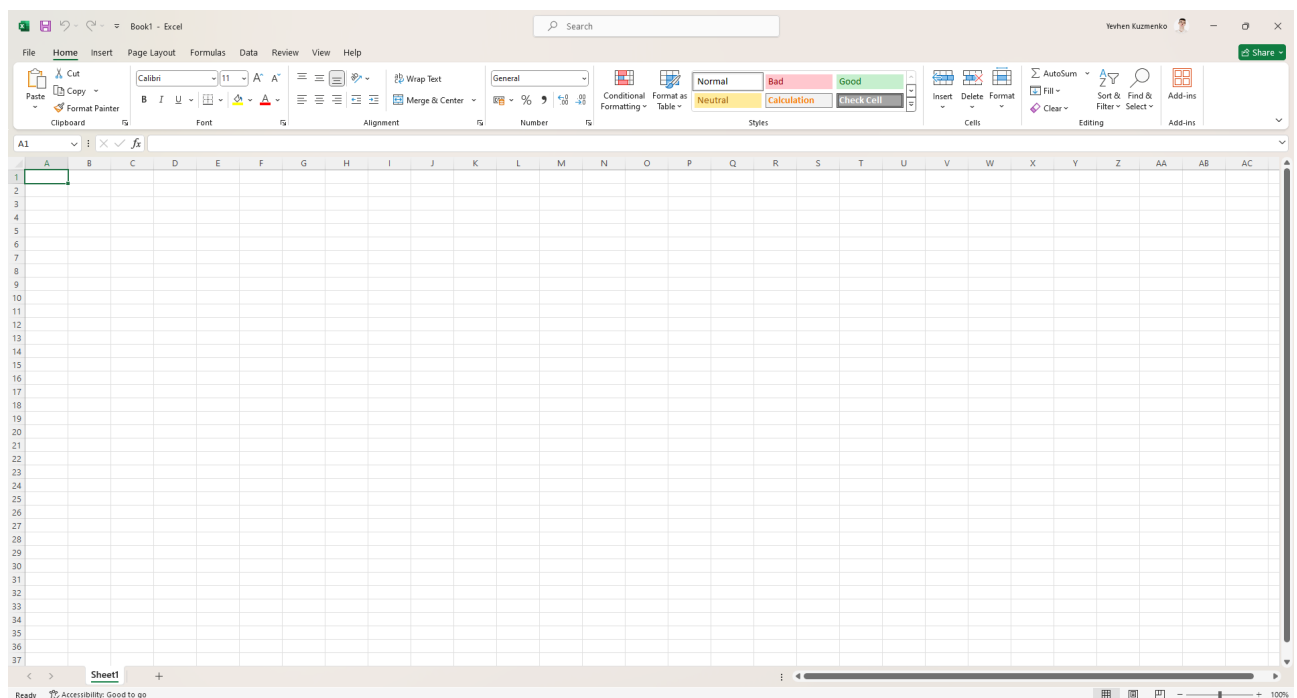


Рис. 1.2 – Інтерфейс додатку Microsoft Excel

Microsoft Excel – це потужний інструмент для роботи з електронними таблицями, який дозволяє здійснювати складні розрахунки, аналіз даних та

створення звітів. Основні переваги Excel включають можливість автоматизації розрахунків за допомогою формул та макросів, інструменти для створення графіків та діаграм для візуалізації даних, гнучкість у налаштуванні електронних таблиць під потреби користувача та широкі можливості для аналізу даних завдяки вбудованим функціям та інструментам.

Переваги:

- Зручний інтерфейс.
- Підтримка платформ Windows, MacOS.
- Можливість маніпулювати масивами даних за допомогою формул.

Недоліки:

- Дані зберігаються на ПК власника документу тому додавання, зміна, видалення даних вимагає пересилки документу іншій особі.
- Ціна додатку - 1899 грн на рік за 1 користувача,  $1899 * 50$  викладачів = 94950 грн на рік

### **1.1.3 Порівняння переваг та недоліків існуючих додатків**

Проаналізувавши переваги та недоліки попередніх додатків у контексті обліку та аналізу даних викладачів була складена порівняльна таблиця 1.1.

Таблиця 1.1

## Порівняльна таблиця існуючих додатків

Функціональна можливість застосунку	Microsoft Word	Microsoft Excel	Система рейтинг
Реєстрація та авторизація користувачів	-	-	+
Облік та аналіз рейтингових карток викладачів	-	Рейтинг розраховується за допомогою формул	Рейтинг формується автоматично
Аналіз та формування рейтингових списків інститутів, кафедр	-	-	Рейтинг формується автоматично
Аналіз та формування рейтингових списків викладачів по кафедрі, університету	-	-	Рейтинг формується автоматично
Стандартизована рейтингова картка викладача	-	+	Сторінка генерується автоматично та є однаковою для усіх викладачів
Підтримка платформ	Windows, MacOS	Windows, MacOS	Windows, MacOS, Linux, Android, iOS.
Ціна	Пакет із 2 додатків - 1899 грн на рік / 1 користувач		Безкоштовно / Не обмежено

## 2. Проектування програмного застосунку

### 2.1 Діаграма прецедентів

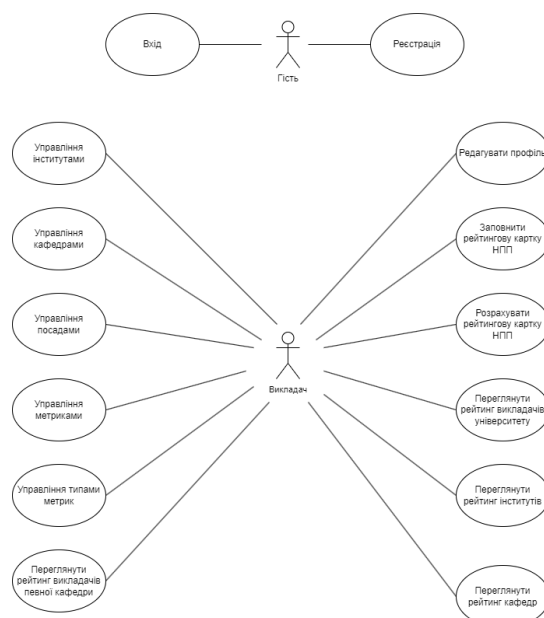


Рис. 2.1 – Діаграма прецедентів

Діаграма прецедентів демонструє основні функціональні можливості нашого веб-додатку для ведення обліку та аналізу даних рейтингу викладачів. Вона включає різноманітні дії, які можуть виконувати користувачі системи для ефективного управління інформацією про інститути, кафедри, посади та метрики, а також для автоматичного формування рейтингів. Таким чином, система надає комплексний інструмент для управління та аналізу академічної діяльності, сприяючи покращенню якості освіти та ефективності управлінських рішень в університеті.

### 2.2 Діаграма класів

У проекті була побудована діаграма класів тому, що вона є потужним інструментом для розуміння та проектування складних програмних систем та

допомагає розробникам краще керувати та аналізувати їх структуру. На рис.2.2 наведено діаграма класів.

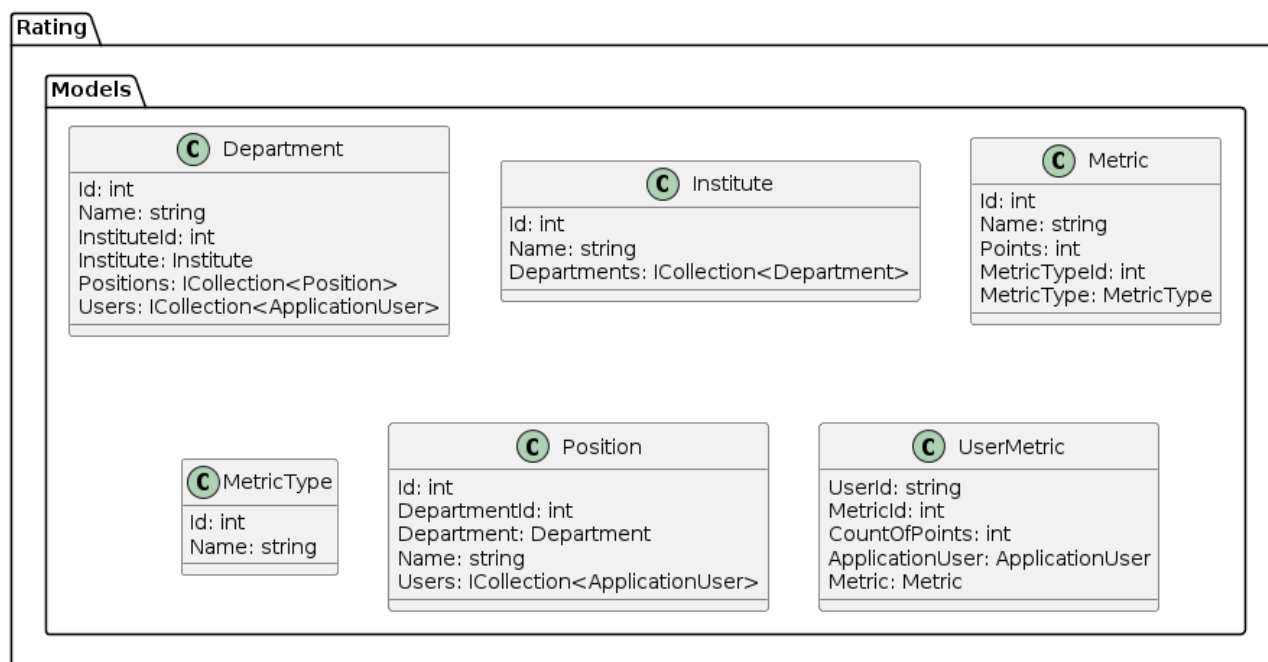


Рис. 2.2 – Діаграма класів

### 2.3 Схема структури баз даних

Схема баз даних визначає структуру та взаємозв'язки між таблицями, необхідними для зберігання та організації даних у системі. У нашому веб-додатку ми використовуємо базу даних для зберігання інформації про користувачів, кафедри, наукові інститути, посади, метрики та їхні взаємозв'язки.

На рис.2.3 представлено схема структури баз даних.





Рис. 2.3 – Схема структури баз даних

Опис призначення таблиць у базі даних:

#### Таблиця Institute

Ця таблиця містить інформацію про наукові інститути у складі університету. Кожен науковий інститут має унікальний ідентифікатор (Id) та назву (Name).

#### Таблиця Departments

Містить дані про кафедри університету. Кожна кафедра має унікальний ідентифікатор (Id), назву (Name) та ідентифікатор інституту, до якого вона належить (InstituteId), що визначає взаємозв'язок між кафедрами та науковими інститутами.

#### Таблиця Positions

Зберігає інформацію про посади, які можуть займати викладачі та співробітники університету. Кожна посада має унікальний ідентифікатор (Id), назву (Name) та ідентифікатор кафедри, до якої вона відноситься (DepartmentId).

### Таблиця MetricType

Тут зберігаються дані про типи метрик, що використовуються для обліку та аналізу рейтингу викладачів. Кожен тип метрик має свій унікальний ідентифікатор (Id) та назву типу метрики (Name).

### Таблиця Metric

Дана таблиця зберігає інформацію про існуючі метрики, що використовуються для обліку та аналізу рейтингу викладачів. Кожна метрика має унікальний ідентифікатор (Id), назву (Name), кількість балів (Points) та ідентифікатор типу метрики до якого вона належить (MetricTypeId).

### Таблиця UserMetrics

Встановлює зв'язок між користувачами та метриками. Вона містить ідентифікатор користувача (UserId), ідентифікатор метрики (MetricId) та кількість балів, яку отримав користувач за відповідну метрику (countOfPoints).

### Таблиця AspNetUsers

Ця таблиця містить дані про користувачів системи, які використовуються для аутентифікації та авторизації. Кожен користувач має унікальний ідентифікатор (Id), повне ім'я (FullName), науковий ступінь (AcademicDegree), вчене звання (AcademicTitle), посаду (PositionId) та інші особисті дані.

Загалом, структура нашої бази даних відповідає вимогам сучасних систем управління даними. Вона забезпечує надійне зберігання інформації, підтримує цілісність даних та дозволяє ефективно працювати з великими обсягами інформації. Крім того, наша структура є гнучкою та масштабованою, що дозволяє легко адаптувати її до змін у вимогах або бізнес-логіці, додаючи нові таблиці чи поля без значних зусиль.

Ця база даних є основою для побудови надійного, масштабованого та ефективного веб-додатку, що відповідає усім необхідним стандартам і забезпечує високу якість обслуговування для користувачів.

## 3. РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ

### 3.1 Вибір засобів програмної реалізації

#### 3.1.1 Інтегроване середовище розробки Visual Studio

Розглянемо Інтегроване середовище розробки Visual Studio. Visual Studio – це сучасне інтегроване середовище розробки (IDE), створене компанією Microsoft яке дозволяє створювати програмне забезпечення під різні платформи та має широку підтримку таких мов програмування як C#, C++, Visual Basic, JavaScript та інших. На рис. 3.1 представлено логотип Visual Studio.



Рис. 3.1 – Логотип Visual Studio

Одна з його найбільших переваг - це можливість встановлення на різних операційних системах: Windows, Linux та MacOS. Це не тільки демонструє відкритість Microsoft до різних спільнот розробників, а й дає можливість кожному розробнику працювати в зручному для нього середовищі.

Вбудовані інструменти включають дебагер, що допомагає виявляти й виправляти помилки у коді, інтеграцію з Git яка забезпечує зручний контроль версій, інструменти для проведення тестування та роботою з базами даних.

#### 3.1.2 Мова програмування C#

Розглянемо мову програмування C#. Мова C# (вимовляється як "сі-шарп") є найпопулярнішою мовою програмування на платформі .NET [1] та розроблена компанією Microsoft. Вона активно використовується в різних галузях ІТ, таких як

розробка веб-додатків, ігор, бізнес-додатків та ін. Завдяки своїй потужності, гнучкості та простоті синтаксису, С# здобула велику популярність серед розробників, як новачків, так і досвідчених професіоналів. Це забезпечує високий попит на розробників з навичками програмування на С#, що робить її привабливою для вивчення та використання. На рис.3.2 представлено логотип С#.



Рис. 3.2 – Логотип С#

Однією з ключових переваг С# є її здатність працювати на різних платформах. Завдяки платформі .NET, яка підтримує крос-платформну розробку, програми, написані на С#, можуть бути запуснені на Windows, macOS та Linux. Це досягається завдяки .NET Core, сучасному крос-платформному фреймворку з відкритим вихідним кодом. Таким чином, С# надає розробникам можливість створювати додатки, які можуть працювати на різних операційних системах без необхідності переписувати код для кожної з них.

С# також відома своєю можливістю створювати додатки для різних типів пристроїв, включаючи телефони, ПК та інші гаджети. Завдяки платформам Xamarin та .NET MAUI, розробники можуть створювати крос-платформні мобільні додатки для iOS та Android, використовуючи єдину кодову базу. Крім того, С# широко використовується для розробки настільних додатків для Windows, веб-додатків за допомогою ASP.NET та навіть ігор за допомогою ігрового рушія Unity.

Це робить С# універсальною мовою, здатною задовольнити потреби різних проектів та ринків.

Ще однією значною перевагою C# є розвинена екосистема та підтримка з боку спільноти. Мова має велику кількість бібліотек та фреймворків, які значно спрощують розробку різних типів додатків. Крім того, завдяки активній спільноті розробників, завжди можна знайти підтримку та відповіді на свої питання у форумах, блогах та спеціалізованих ресурсах. Microsoft регулярно випускає оновлення та нові версії C#, додаючи нові функції та вдосконалюючи існуючі можливості, що забезпечує актуальність та сучасність мови.

### 3.1.3 ASP.NET

Розглянемо ASP.NET. ASP.NET – це веб-фреймворк, розроблений компанією Microsoft, який дозволяє створювати динамічні веб-сайти, веб-додатки та веб-сервіси. Використовуючи платформу .NET, ASP.NET забезпечує потужний набір інструментів та бібліотек, що спрощують розробку складних веб-рішень. В основі ASP.NET лежить модель подій, яка дозволяє розробникам легко створювати інтерактивні та зручні для користувачів інтерфейси. Завдяки інтеграції з Visual Studio, ASP.NET пропонує розробникам інтуїтивно зрозумілий середовище для написання, налагодження та тестування коду. На рис.3.3 представлено логотип ASP.NET.



Рис. 3.3 – Логотип ASP.NET

Однією з основних переваг ASP.NET є його продуктивність. Завдяки компіляції коду на сервері та кешуванню результатів, веб-додатки, створені з використанням цього фреймворку, демонструють високу швидкість роботи та

ефективність. Крім того, ASP.NET підтримує асинхронні операції, що дозволяє обробляти велику кількість одночасних запитів без значного навантаження на сервер. Це робить ASP.NET ідеальним вибором для розробки високонавантажених веб-додатків.

Безпека є ще однією важливою характеристикою ASP.NET. Фреймворк включає в себе широкий спектр вбудованих засобів для забезпечення безпеки додатків, таких як автентифікація та авторизація користувачів, захист від SQL-ін'єкцій та міжсайтового скриптингу (XSS). Крім того, ASP.NET підтримує використання HTTPS для захисту передачі даних між клієнтом та сервером. Завдяки цим функціям розробники можуть створювати безпечні веб-додатки, які відповідають сучасним стандартам безпеки.

Ще однією важливою особливістю ASP.NET є його гнучкість та масштабованість. Фреймворк підтримує різноманітні архітектурні підходи, включаючи MVC (Model-View-Controller) та Web API, що дозволяє розробникам вибирати оптимальну структуру для своїх проєктів. Крім того, ASP.NET легко інтегрується з іншими технологіями та сервісами, що робить його універсальним інструментом для розробки веб-додатків різної складності та масштабу.

### **3.1.4 Entity Framework**

Розглянемо Entity Framework. Entity Framework (EF) – це об'єктно-реляційний маппер (ORM), розроблений Microsoft для платформи .NET. EF дозволяє розробникам працювати з базами даних, використовуючи об'єктно-орієнтовану парадигму програмування.

Замість написання SQL-запитів, розробники можуть маніпулювати даними, використовуючи об'єкти та їх властивості, що значно спрощує процес розробки та обслуговування додатків.

На рис. 3.4 представлено логотип Entity Framework.

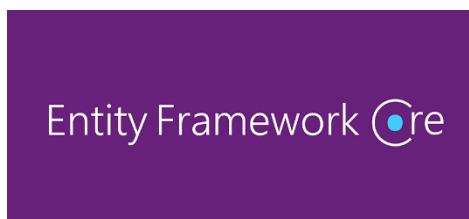


Рис. 3.4 – Логотип Entity Framework

Однією з основних переваг Entity Framework є його здатність автоматично створювати бази даних на основі класів моделі даних. Це дозволяє розробникам зосередитися на бізнес-логіці додатка, не витрачаючи час на написання складних SQL-запитів та налаштування бази даних. Крім того, EF підтримує міграції, що дозволяє легко керувати змінами в структурі бази даних протягом життєвого циклу проекту.

Entity Framework також забезпечує високу продуктивність та ефективність роботи з даними. Він оптимізує SQL-запити, що генеруються, зменшуючи кількість звернень до бази даних та мінімізуючи навантаження на сервер. Крім того, EF підтримує кешування та асинхронні операції, що дозволяє покращити продуктивність додатків та забезпечити швидкий доступ до даних навіть при високому навантаженні.

Безпека є ще однією важливою характеристикою Entity Framework. Він автоматично захищає додатки від таких поширених атак, як SQL-ін'єкції, забезпечуючи безпечне виконання запитів до бази даних. Крім того, EF підтримує рольову модель безпеки, що дозволяє контролювати доступ користувачів до даних на різних рівнях додатка. Завдяки цим функціям Entity Framework є надійним та безпечним інструментом для роботи з базами даних у додатках .NET.

### 3.1.5 MySQL

Розглянемо MySQL. MySQL - це відкрита система керування базами даних, що базується на мові структурованого запиту SQL (Structured Query Language). Розроблена компанією MySQL AB, яка була придбана корпорацією Sun

Microsoft у 2008 році, а потім увійшла в склад корпорації Oracle у 2010 році. Це одна з найпопулярніших реляційних систем керування базами даних (СКБД) у світі. На рис. 3.5 представлено логотип MySQL.



Рис. 3.5 – Логотип MySQL

MySQL використовується для зберігання, організації та управління великими обсягами структурованої інформації. Вона підтримує широкий спектр можливостей, включаючи транзакційні операції, індексацію даних, реплікацію та шифрування, що робить її популярним вибором для веб-розробників та організацій у всьому світі.

Однією з головних переваг MySQL є її відкритість та безкоштовність. Вона доступна для завантаження та використання безкоштовно під ліцензією GNU General Public License (GPL), що дозволяє користувачам не тільки ефективно використовувати систему, а й модифікувати її відповідно до власних потреб.

MySQL підтримує багатоплатформенність, тобто вона працює на багатьох операційних системах, включаючи Windows, Linux, macOS та інші. Це робить її універсальним рішенням для розробки програмного забезпечення, незалежно від того, на якій платформі вона буде використовуватися.

Ще однією важливою особливістю MySQL є її висока продуктивність та швидкодія. Вона може працювати з великими обсягами даних і виконувати складні запити швидко і ефективно. Це робить її ідеальним вибором для веб-сайтів та додатків з великою кількістю користувачів та великим обсягом трафіку.

Для розробників MySQL надає ряд інструментів та інтерфейсів для роботи з базами даних. Найпопулярнішим інтерфейсом є MySQL Workbench, що забезпечує графічний інтерфейс для створення, управління та оптимізації баз



даних MySQL. Крім того, для роботи з базами даних можна використовувати мови програмування, такі як PHP, Python, Java та інші, використовуючи різноманітні бібліотеки та модулі, що надають доступ до MySQL через різні API.

## 3.2 Використані архітектурні паттерни

### 3.2.1 Паттерн Модель - Вид – Контролер

Розглянемо використані архітектурні паттерни. Модель – Вид – Контролер – це архітектурний паттерн який ділить додаток на три групи компонентів: моделі, види та контролери [2]. На рис.3.6 наведено візуальне представлення роботи паттерну MVC.

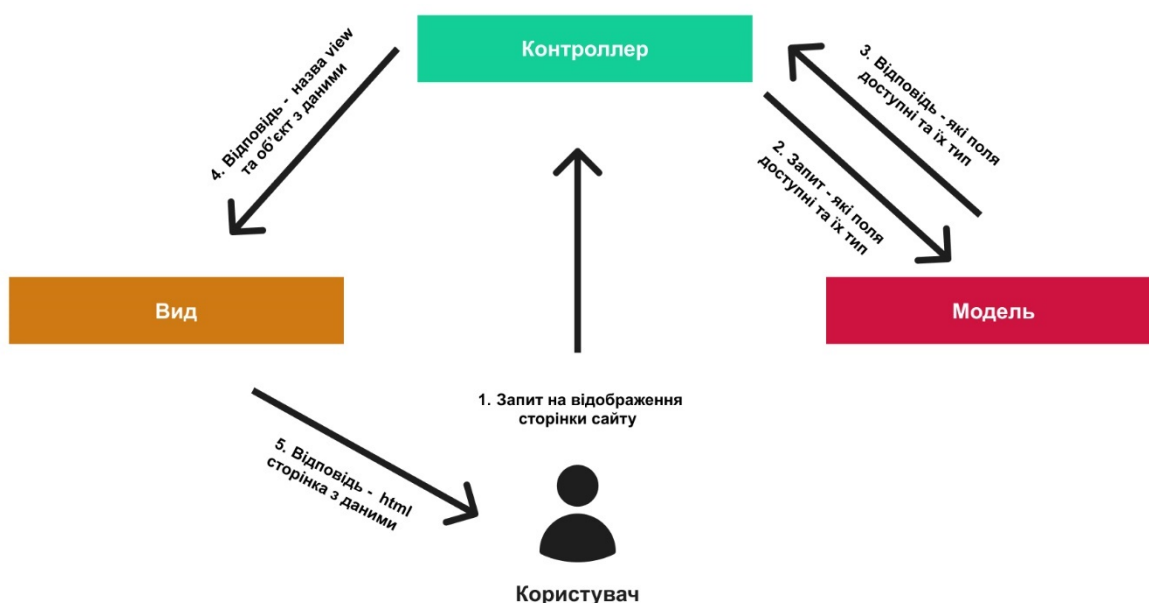


Рис. 3.6 – Візуальне представлення роботи паттерну MVC

Контролер – це клас який відповідає за обробку запитів від користувача. Він може взаємодіяти з моделями для отримання типів даних та створювати запити до бази даних яка поверне у відповідь потрібні дані. Після отримання даних від бази

даних він відправляє користувачу відповідь із зазначеною у кодї назвою виду та об'єкт з даними які необхідні для його відображення.

Модель – це клас який має внутрішні поля та методи для взаємодії з ними. Такі класи потрібні для обробки запитів від контролера. Вони допомагають зберегти назви полів та їх типів для подальшої синхронізації з базою даних.

Вид – це файл у якому міститься html розмітка та у який за допомогою шаблонізаторів можна вставляти дані які містяться у об'єкті отриманого від контролера.

### **3.2.2 Клієнт-серверна архітектура**

При виборі архітектури майбутнього програмного забезпечення було обрано тришарову клієнт-серверну архітектуру. Ця архітектура передбачає розділення системи на рівень клієнту, серверу та бази даних.

Шар клієнта - це частина програми або додатку, яка виконується на обладнанні кінцевого користувача. Зазвичай це веб-інтерфейс, мобільний додаток або десктопна програма. Принцип роботи полягає у тому, що клієнт робить запит на сервер, сервер його обробляє, а потім відправляє у відповідь клієнту сторінку з даними або помилкою.

Шар логіки - це частина програми або додатку, яка виконується на сервері. Після отримання сервером запиту від клієнта, він може звернутися до бази даних, провести обчислення або інші запрограмовані дії та відправити оброблений запит клієнту. Сервер має обмежену кількість ресурсів для обробки майбутніх запитів. При сильному навантаженні сервер ставить запити у чергу та поступово їх обробляє. Якщо кількість запитів у черзі перевищує доступні можливості для їх обробки, клієнт отримає помилку або відповідь з дуже великою затримкою.

Шар даних – це зазвичай реляційне сховище даних представлених у табличному вигляді та пов'язаних зв'язками між таблицями за допомогою ключів. Таке сховище може отримувати запити від серверної частини на додавання, редагування, видалення або фільтрацію даних з таблиць.

Основними перевагами тришарової архітектури є:

**Модульність:** Система розділена на три окремі модулі. Кожен модуль може супроводжуватися окремо від інших, що значно спрощує підтримку продукту.

**Масштабованість:** Окремі компоненти мають змогу регулювати кількість вживаних ними ресурсів відповідно до свого реального навантаження та можуть бути розміщені на різних серверах.

**Готовність до змін:** розділення на три окремі частини дає змогу вносити зміни в один модуль не впливаючи при цьому на інші.

Таким чином, трирівнева клієнт-серверна архітектура є ефективним підходом до розробки програмного забезпечення, що дозволяє досягти балансу між функціональністю, масштабованістю, безпекою та стабільністю системи.

## 4. СЦЕНАРІЇ РОБОТИ РОЗРОБЛЕНОГО ЗАСТОСУНКУ

### 4.1 Головна сторінка додатку

На головній сторінці додатку зображений логотип університету, а також присутнє верхнє навігаційнє меню з якого можна потрапити на сторінки Реєстрація та Авторизація. На рис.4.1 представлена головна сторінка додатку.

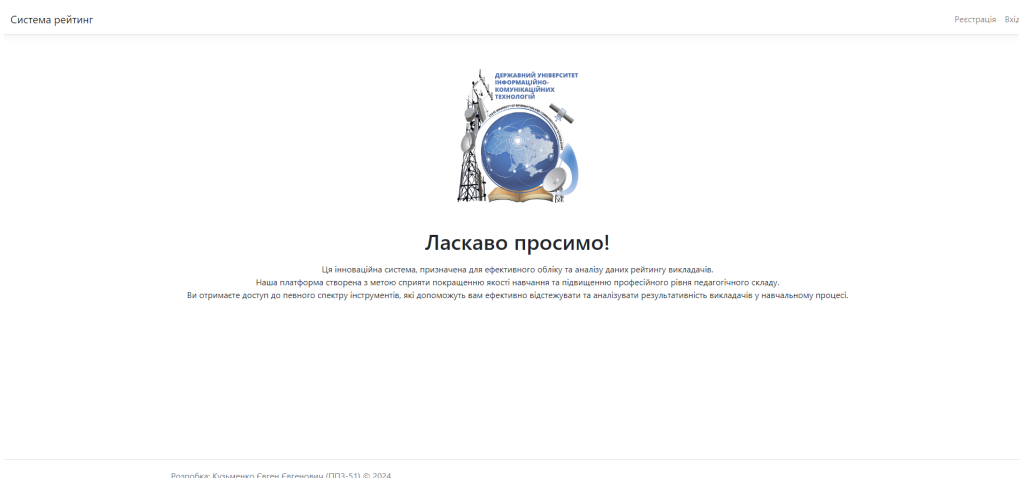


Рис. 4.1 – Головна сторінка додатку

### 4.2 Реєстрація

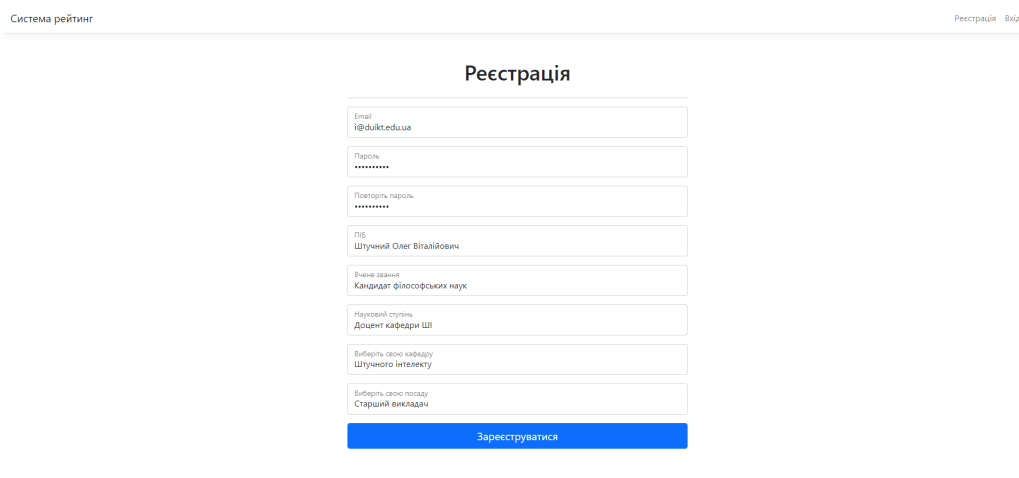


Рис. 4.2 – Сторінка реєстрації

На сторінці реєстрації присутня форма яка запитує інформацію про email, пароль, ПІБ, вчене звання, науковий ступінь, кафедру та поточну посаду.

### 4.3 Вхід

Рис. 4.3 – Сторінка входу

Форма входу запитує інформацію про email та пароль які були вказані у процесі реєстрації. Також сторінка містить посилання на реєстрацію та відновлення пароля.

### 4.4 Список інститутів

Рис. 4.4 – Сторінка список інститутів

Викладач може переглядати список інститутів які були внесені у систему. Кожен інститут має посилання на сторінки редагування та видалення.

## 4.5 Додати інститут

Система рейтинг | Институти | Кафедри | Посади | Тили метрик | Метрики | Рейтингова картка | Рейтингові списки | Викладач Іван Петрович | Вийти

### Додати інститут

Назва інституту

[Додати](#)

[Повернутись до списку інститутів](#)

Розробка: Кузьменко Євген Євгенович (ПІІЗ-51) © 2024

Рис. 4.5 – Сторінка додати інститут

Викладач може додавати нові інститути. Форма додавання нового інституту містить назву інституту.

## 4.6 Редагувати інститут

Система рейтинг | Институти | Кафедри | Посади | Тили метрик | Метрики | Рейтингова картка | Рейтингові списки | Викладач Іван Петрович | Вийти

### Редагувати інститут

Назва інституту  
Навчально-науковий інститут Інформаційних технол.

[Редагувати](#)

[Повернутись до списку інститутів](#)

Розробка: Кузьменко Євген Євгенович (ПІІЗ-51) © 2024

Рис. 4.6 – Сторінка редагувати інститут

Викладач може редагувати вибраний інститут. Форма редагування має містити назву інституту. Редагування інституту відбувається через його ідентифікатор у системі.

## 4.7 Видалити інститут

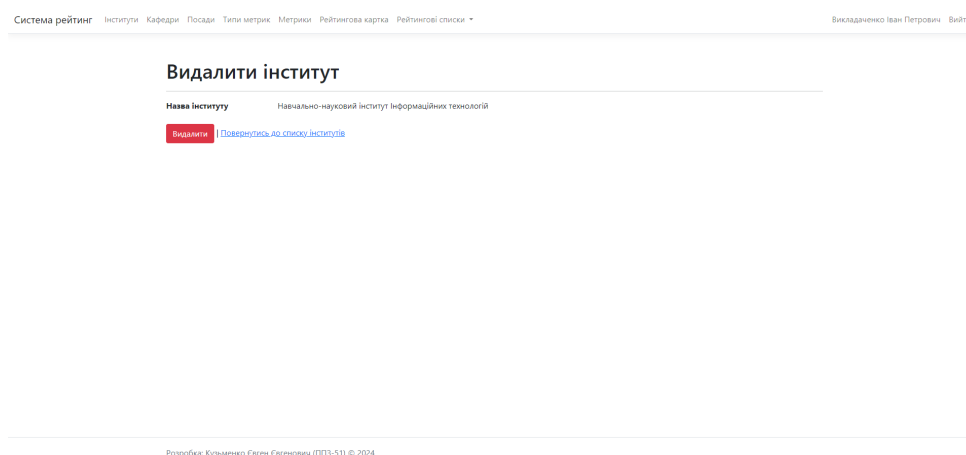


Рис. 4.7 – Сторінка видалити інститут

Викладач може видалити вибраний інститут. Видалення інституту відбувається через його ідентифікатор у системі.

## 4.8 Список кафедр

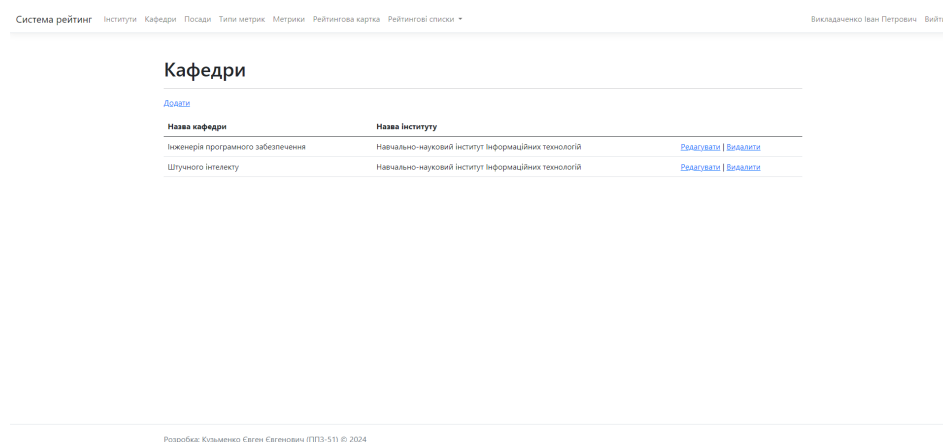


Рис. 4.8 – Сторінка список кафедр

Викладач може переглядати список кафедр які були внесені у систему. Кожна кафедра має посилання на сторінки редагування та видалення.

## 4.9 Додати кафедру

Система рейтинг Інститути Кафедри Посади Типи метрик Метрики Рейтингова картка Рейтингові списки \* Викладач Іван Петрович Вийти

### Додати кафедру

Назва кафедри

Назва інституту  
Навчально-науковий інститут інформаційних тех

[Додати](#)

[Повернутись до списку кафедр](#)

Розробка: Курьменко Євген Євгенович (ПІІТ-51) © 2024

Рис. 4.9 – Сторінка додати кафедру

Викладач може додавати нові кафедри. Форма додавання нової кафедри містить назву кафедри та назву інституту до якого вона належить.

## 4.10 Редагувати кафедру

Система рейтинг Інститути Кафедри Посади Типи метрик Метрики Рейтингова картка Рейтингові списки \* Викладач Іван Петрович Вийти

### Редагувати кафедру

Назва кафедри  
Інженерія програмного забезпечення

Назва інституту  
Навчально-науковий інститут інформаційних тех

[Редагувати](#)

[Повернутись до списку кафедр](#)

Розробка: Курьменко Євген Євгенович (ПІІТ-51) © 2024

Рис. 4.10 – Сторінка редагувати кафедру



Викладач може редагувати вибрану кафедру. Форма редагування містить назву кафедри та назву інституту до якого вона належить. Редагування кафедри відбувається через її ідентифікатор у системі.

#### 4.11 Видалити кафедру

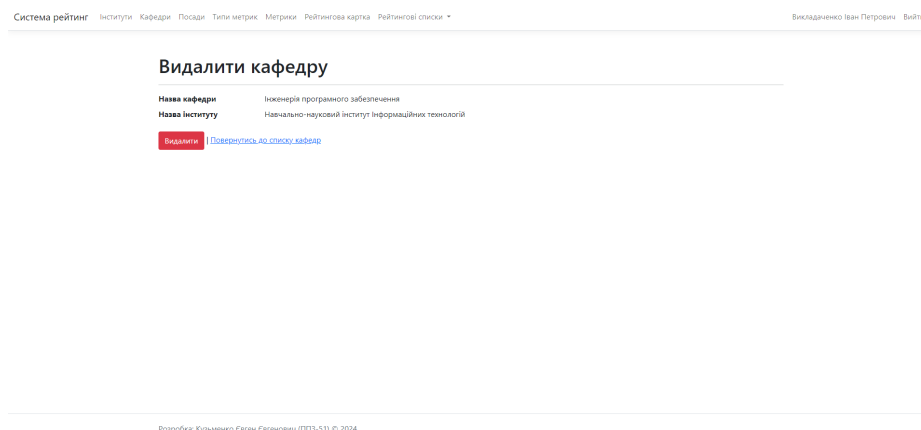


Рис. 4.11 – Сторінка видалити кафедру

Викладач може видалити вибрану кафедру. Видалення кафедри відбувається через її ідентифікатор у системі.

#### 4.12 Список посад

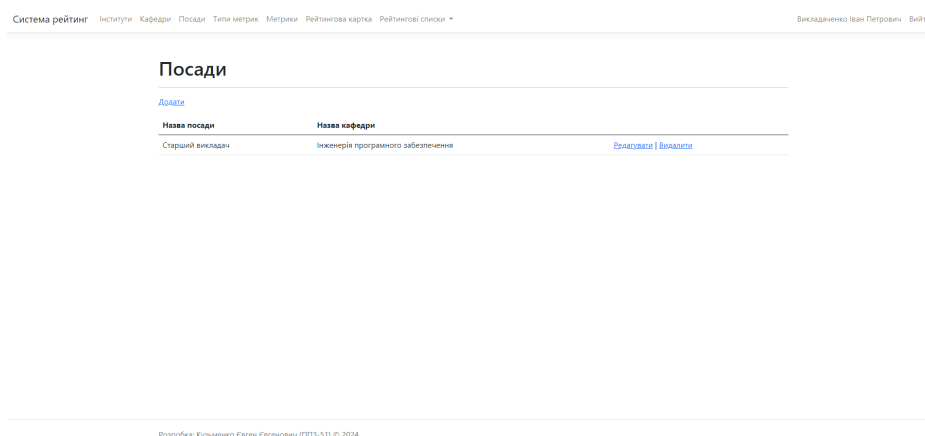


Рис. 4.12 – Сторінка список посад

Викладач може переглядати список посад які були внесені у систему. Кожна посада має посилання на сторінки редагування та видалення.

### 4.13 Додати посаду

The screenshot shows a web interface for adding a new position. At the top, there is a navigation bar with the following items: Система рейтинг, Інститути, Кафедри, Посади, Типи метрик, Метрики, Рейтингова картка, Рейтингові списки. On the right side of the navigation bar, it says: Викладач Іван Петрович, Вийти. The main heading of the form is 'Додати посаду'. Below the heading, there is a form with two input fields: 'Назва посади' (Position Name) and 'Назва кафедри' (Department Name) with a dropdown menu showing 'Інженерія програмного забезпечення'. There is a blue 'Створити' (Create) button and a link 'Повернутись до списку посад' (Return to position list).

Рис. 4.13 – Сторінка додати посаду

Викладач може додавати нові посади. Форма додавання нової посади містить назву посади та назву кафедри до якої вона належить.

### 4.14 Редагувати посаду

The screenshot shows a web interface for editing an existing position. At the top, there is a navigation bar with the following items: Система рейтинг, Інститути, Кафедри, Посади, Типи метрик, Метрики, Рейтингова картка, Рейтингові списки. On the right side of the navigation bar, it says: Викладач Іван Петрович, Вийти. The main heading of the form is 'Редагувати посаду'. Below the heading, there is a form with two input fields: 'Назва посади' (Position Name) and 'Старий викладач' (Old Lecturer). Below these fields, there is a dropdown menu for 'Назва кафедри' (Department Name) showing 'Інженерія програмного забезпечення'. There is a blue 'Редагувати' (Edit) button and a link 'Повернутись до списку посад' (Return to position list).

Рис. 4.14 – Сторінка редагувати посаду

Викладач може редагувати вибрану посаду. Форма редагування містить назву посади та назву кафедри до якої вона належить.

## 4.15 Видалити посаду

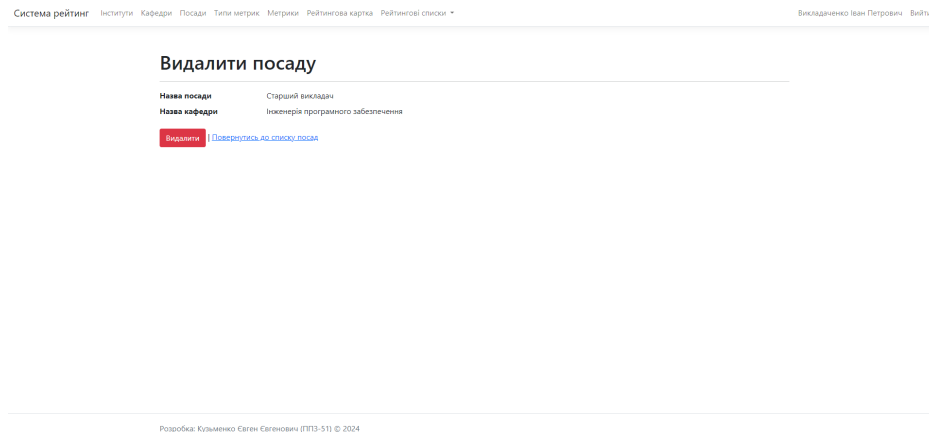


Рис. 4.15 – Сторінка видалити посаду

Викладач може видалити вибрану посаду. Видалення посади відбувається через її ідентифікатор у системі.

## 4.16 Список типів метрик

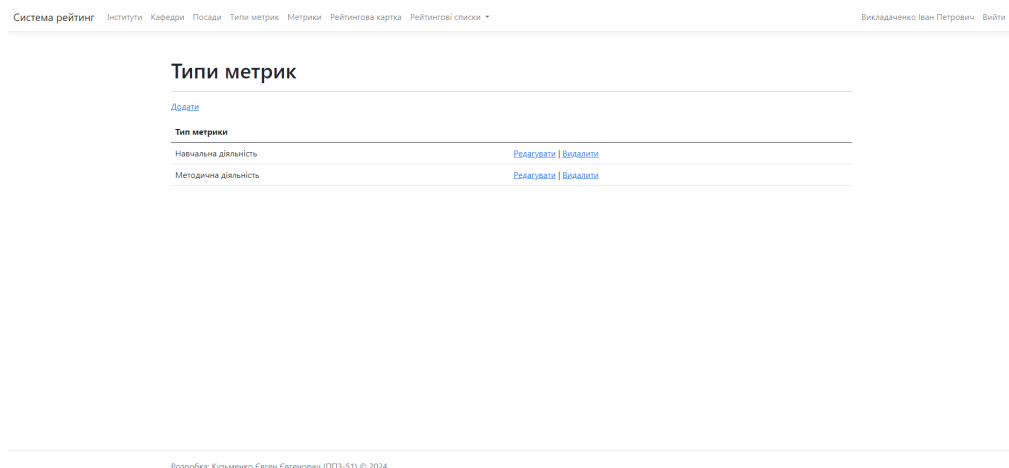


Рис. 4.16 – Сторінка список типів метрик

Викладач може переглядати список типів метрик які були внесені у систему. Кожен тип метрик має посилання на сторінки редагування та видалення.

## 4.17 Додати тип метрики

Система рейтинг Інститути Кафедри Посади Типи метрик Метрики Рейтингова картка Рейтингові списки Вихід

Викладач Іван Петрович

### Додати тип метрики

Тип метрики

[Додати](#)

[Повернутись до списку типів метрик](#)

Розробка: Кузьменко Євген Євгенович (ПТЗ-51) © 2024

Рис. 4.17 – Сторінка додати тип метрики

Викладач може додавати нові типи метрик. Форма додавання нового типу метрик містить назву типу метрик.

## 4.18 Редагувати тип метрики

Система рейтинг Інститути Кафедри Посади Типи метрик Метрики Рейтингова картка Рейтингові списки Вихід

Викладач Іван Петрович

### Редагувати тип метрики

Тип метрики  
Навчальна діяльність

[Редагувати](#)

[Повернутись до списку типів метрик](#)

Розробка: Кузьменко Євген Євгенович (ПТЗ-51) © 2024

Рис. 4.18 – Сторінка редагувати тип метрики

Викладач може редагувати вибраний тип метрик. Форма редагування містить назву типу метрик. Редагування типу метрик відбувається через її ідентифікатор у системі.

## 4.19 Видалити тип метрики

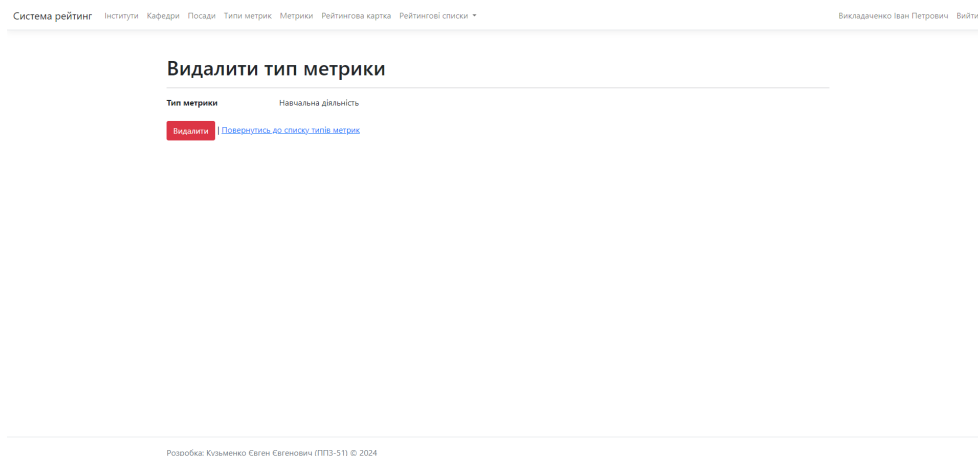


Рис. 4.19 – Сторінка видалити тип метрики

Викладач може видалити вибраний тип метрик. Видалення типу метрик відбувається через її ідентифікатор у системі.

## 4.20 Список метрик

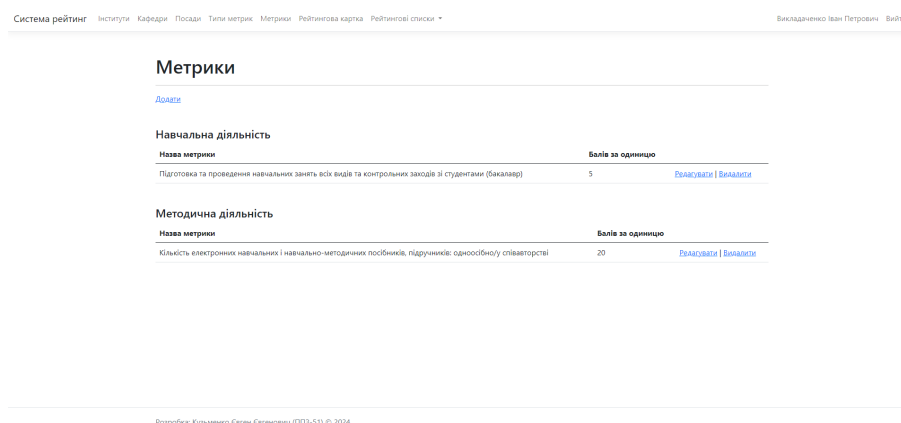


Рис. 4.20 – Сторінка список метрик

Викладач може переглядати список метрик які були внесені у систему. Кожна метрика має посилання на сторінки редагування та видалення.

## 4.21 Додати метрику

The screenshot shows a web form titled "Додати метрику" (Add Metric). At the top, there is a navigation bar with the text "Система рейтингів" and several menu items: "Інструменти", "Кабінет", "Посади", "Типи метрик", "Метрики", "Рейтингова картка", and "Рейтингові статистика". On the right side of the navigation bar, it says "Викладач: Іван Петрович" and "Вийти".

The form itself has the following fields:

- A text input field for "Назва метрики" (Metric Name).
- A text input field for "Бали за одиницю" (Points per unit).
- A dropdown menu for "Тип метрики" (Metric Type) with "Рейтингова діяльність" (Rating Activity) selected.
- A blue "Додати" (Add) button.
- A link below the button: "Сторінка додавання метрик" (Metric addition page).

At the bottom of the page, there is a footer: "Розробка: Кривинська Світлана (PFD-51) © 2024".

Рис. 4.21 – Сторінка додати метрику

Викладач може додавати нові метрики. Форма додавання нової метрики містить назву метрики, кількість балів за одиницю метрики та назву типу метрик до якої вона належить.

## 4.22 Редагувати метрику

The screenshot shows a web form titled "Редагувати метрику" (Edit Metric). At the top, there is a navigation bar with the text "Система рейтингів" and several menu items: "Інструменти", "Кабінет", "Посади", "Типи метрик", "Метрики", "Рейтингова картка", and "Рейтингові статистика". On the right side of the navigation bar, it says "Викладач: Іван Петрович" and "Вийти".

The form itself has the following fields:

- A text input field for "Назва метрики" (Metric Name) with a warning message: "Підприємство на проведення навчання зазвичай використовує цей код".
- A dropdown menu for "Тип метрики" (Metric Type) with "Рейтингова діяльність" (Rating Activity) selected.
- A text input field for "Бали за одиницю" (Points per unit) with the value "0" entered.
- A blue "Зберегти" (Save) button.
- A link below the button: "Сторінка додавання метрик" (Metric addition page).

At the bottom of the page, there is a footer: "Розробка: Кривинська Світлана (PFD-51) © 2024".

Рис. 4.22 – Сторінка редагувати метрику

Викладач може редагувати вибрану метрику. Форма редагування містить назву метрики, кількість балів за одиницю метрики та назву типу метрик до якого вона належить. Редагування метрики відбувається через її ідентифікатор у системі.

## 4.23 Видалити метрику

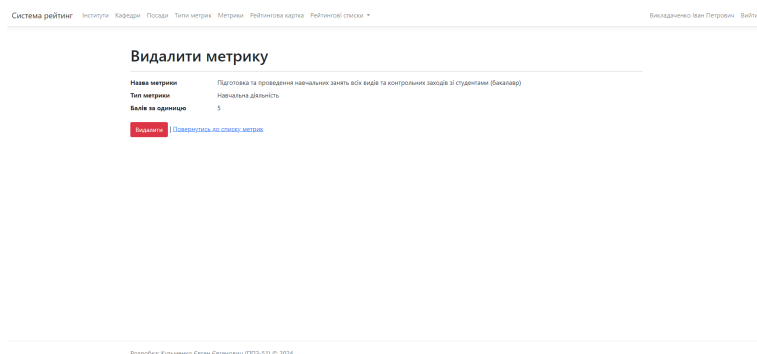


Рис. 4.23 – Сторінка видалити метрику

Викладач може видалити вибрану метрику. Видалення метрики відбувається через її ідентифікатор у системі.

## 4.24 Рейтингова картка

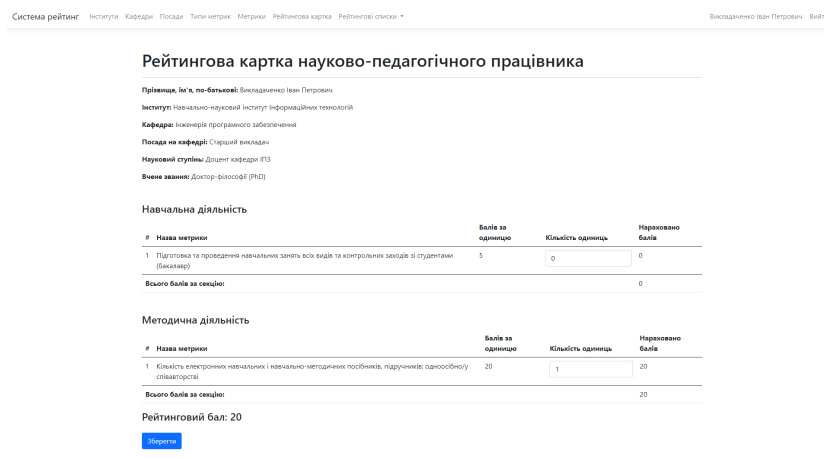


Рис. 4.24 – Сторінка рейтингова картка

Викладач може переглядати та заповнювати особисту картку рейтингу НПП. Картка містить ПІБ, назву інституту, назву кафедри, посаду на кафедрі,

науковий ступінь, академічне звання, типи метрик, метрики, кількість балів за одиницю метрики, кількість одиниць метрики, кількість нарахованих балів, кількість балів за тип метрики, рейтинговий бал викладача.

#### 4.25 Рейтинг викладачів університету

Система рейтинг Інститути Кафедри Посади Типи метрик Метрики Рейтингова картка Рейтингові списки Вихід

Викладач Іван Петрович Вийти

### Рейтинг викладачів університету

#	ПІБ	Кафедра	Посада на кафедрі	Науковий ступінь	Вчене звання	Рейтинг
1	Штуней Олександр Віталійович	Штучного інтелекту	Старший викладач	Кандидат філософських наук	Доцент кафедри ШІ	125
2	Викладач Іван Петрович	Інженерів програмного забезпечення	Старший викладач	Доктор філософії (PhD)	Доцент кафедри ІТЗ	20

Розробка: Кузьменко Євген Євгенович (ІТЗ-51) © 2024

Рис. 4.25 – Сторінка рейтинг викладачів університету

Викладач може переглядати рейтинг викладачів університету. Рейтинг представлений у вигляді таблиці яка є ранжованою за кількістю рейтингу та містити номер місця у рейтингу, ПІБ, назву кафедри, посаду на кафедрі, науковий ступінь, вчене звання та рейтинг викладачів.

#### 4.26 Рейтинг викладачів певної кафедри

Система рейтинг Інститути Кафедри Посади Типи метрик Метрики Рейтингова картка Рейтингові списки Вихід

Викладач Іван Петрович Вийти

### Рейтинг викладачів певної кафедри

Виберіть кафедру

[Інженерів програмного забезпечення](#) [Штучного інтелекту](#)

Розробка: Кузьменко Євген Євгенович (ІТЗ-51) © 2024

Рис. 4.26 – Сторінка рейтинг викладачів певної кафедри



Викладач може переглядати рейтинг викладачів певної кафедри. На сторінці є посилання на сторінки з рейтингом кожної кафедри.

## 4.27 Рейтинг викладачів кафедри

Система рейтинг | Інститути | Кафедри | Посади | Типи метрик | Метрики | Рейтингова карта | Рейтингові списки

Викладач Іван Петрович | Вийти

### Рейтинг викладачів кафедри Інженерія програмного забезпечення

#	ПІБ	Посада на кафедрі	Науковий ступінь	Вчене звання	Рейтинг
1	Викладач Іван Петрович	Старший викладач	Доктор-філософ (PhD)	Доцент кафедри ПЗ	20

Розробка: Кузьменко Євген Євгенович (ПІІЗ-51) © 2024

Рис. 4.27 – Сторінка рейтинг викладачів кафедри

Викладач може переглядати рейтинг викладачів певної кафедри. Рейтинг представлений у вигляді таблиці яка є ранжованою за кількістю рейтингу та містить номер місця у рейтингу, ПІБ, посаду на кафедрі, науковий ступінь, вчене звання та рейтинг викладачів.

## 4.28 Рейтинг інститутів університету

Система рейтинг | Інститути | Кафедри | Посади | Типи метрик | Метрики | Рейтингова карта | Рейтингові списки

Викладач Іван Петрович | Вийти

### Рейтинг інститутів університету

#	Назва інституту	Сукупний рейтинг
1	Національно-науковий інститут інформаційних технологій	145

Розробка: Кузьменко Євген Євгенович (ПІІЗ-51) © 2024

Рис. 4.28 – Сторінка рейтинг інститутів університету

Викладач може переглядати рейтинг інститутів університету. Рейтинг представлений у вигляді таблиці яка є ранжованою за кількістю рейтингу та містить номер місця у рейтингу, назву інституту та сумарний рейтинг.

## 4.29 Рейтинг кафедр університету

Система рейтинг Інститути Кафедри Посади Типи метрик Метрики Рейтингова картка Рейтингові списки \* Викладченко Іван Петрович Вийти

---

### Рейтинг кафедр університету

---

#	Назва кафедри	Сумарний рейтинг
1	Штучного інтелекту	125
2	Інженерів програмного забезпечення	20

---

Розробка: Кузьменко Сігєн Євгенівич (ПІ3-51) © 2024

Рис. 4.29 – Сторінка рейтинг кафедр університету

Викладач може переглядати рейтинг кафедр університету. Рейтинг представлений у вигляді таблиці яка є ранжованою за кількістю рейтингу та містить номер місця у рейтингу, назву кафедри та сумарний рейтинг.

## ВИСНОВКИ

В процесі виконання кваліфікаційної роботи, було досягнуто наступних результатів:

1. Проаналізовано та складено таблицю переваг та недоліків існуючих засобів ведення обліку та аналізу даних рейтингу викладачів.
2. Розроблено функціональні та нефункціональні вимоги для створення програмного забезпечення яке автоматизує процес обліку та аналізу даних рейтингу викладачів.
3. Розроблено інтерфейс користувача з урахуванням брендового стилю університету.
4. Спроектвані діаграми класів, прецедентів та схема таблиць баз даних.
5. Розроблено програмне забезпечення яке автоматизує підрахунок індивідуального рейтингу НПП по університету, окремій кафедрі, а також сумарний рейтинг по кафедрам та інститутам.
6. Проведено ручне тестування застосунку
7. Робота пройшла апробацію:

Кузьменко Є. Є. Розробка програмного забезпечення для обліку та аналізу даних. Науково-практична конференція «Перспективи та проблематика інтелектуальних систем», 31 травня 2024 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.:ДУІКТ, 2024. С.45.

Кузьменко Є. Є. Гребенюк В. В., Розробка Web-застосунку для моніторингу аналітичних рейтингів науково-педагогічних працівників. IV Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті». Збірник тез. – К.: ДУІКТ, 2024. С.65.

На основі отриманих даних та досліджень, розроблено програмний застосунок для обліку та оцінки даних рейтингу викладача, який враховує всі проаналізовані потреби та можливості.

## ПЕРЕЛІК ПОСИЛАНЬ

1. <https://dotnet.microsoft.com/en-us/languages/csharp>
2. <https://visualstudio.microsoft.com/vs/>
3. Т.К. В. Machine learning algorithms for social media analysis: A survey [Електронний ресурс] / В. Т.К., R. Chandra Sekhara, В. Annushree // ScienceDirect. – 2021. – Режим доступу до ресурсу: <https://www.sciencedirect.com/science/article/abs/pii/S1574013721000356>.
4. Документація «AMQP (Advanced Message Queuing Protocol)» [Електронний ресурс] - Режим доступу: <https://www.amqp.org/> 18.10.2023
5. Документація «RabbitMQ»: [Електронний ресурс] - Режим доступу: <https://www.rabbitmq.com/> 18.10.2023
6. Article «Why Google Stores Billions of Lines of Code in a Single Repository»: - Rachel Potvin and Josh Levenberg [Електронний ресурс] - Режим доступу: <https://dl.acm.org/doi/pdf/10.1145/2854146> 18.10.2023
7. "Впровадження архітектур мікросервісів" Боріс Шоль, Лео Штудер, Майк Амундсен
8. "Сервісно-Орієнтована Архітектура: Концепції, Технології та Дизайн" Томас Ерл
9. Клієнт-серверна архітектура [Електронний ресурс] // QATestLab. – 2020. – Режим доступу до ресурсу: <https://training.qatestlab.com/blog/technical-articles/client-server-architecture/>.
10. Агрегація в MongoDB: зменшення сукупного трубопроводу та карти [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: <https://uk.myservername.com/aggregation-mongodb>.
11. The top programming languages [Електронний ресурс] // GitHub. – 2022. – Режим доступу до ресурсу: <https://octoverse.github.com/2022/top-programming-languages>.

12. "Шаблони проектування сервісів: Фундаментальні рішення для SOAP/WSDL та RESTful веб-сервісів" Роберт Деньно
13. Рейтинг мов програмування 2023 [Електронний ресурс] // Редакція DOU. – 2023. – Режим доступу до ресурсу: <https://dou.ua/lenta/articles/language-rating-2023/>.
14. "Data Structures and Algorithms in Python", Michael T. Goodrich and Roberto Tamassia, 4th edition, Wiley, 2022.
15. "OSPF: The Basics", Juniper Networks, 2023.
16. Tennis Sense: A platform for extracting semantic information from multi-camera tennis data [Електронний ресурс]. Режим доступу: <https://ieeexplore.ieee.org/document/5201152/authors#authors>
17. Shvachych G., Moroz B., Ivaschenko O. , Sushko I., Pobochii I. The implementation features of the aggregation mode of network interface in the multiprocessors computing system. Комп'ютерне моделювання та оптимізація складних систем : Матеріали IV міжнар. науково-практ. конф., м. Дніпро, 1 – 2 листоп. 2018 р. Дніпро. 2018. С. 200 – 202.
18. Ільїн О.Ю., Катков Ю.І., Вергун Д.С., Шашлов А.В. Особливості розгортання мікросервісних додатків за допомогою системи керування контейнерами.
19. The Computer and the Brain (The Silliman Memorial Lectures Series) 3rd Edition by von Neumann, John (Author), Ray Kurzweil (Foreword), Yale University Press; 3 edition, 2012, 136 pages, ISBN-10: 0300181116, ISBN-13: 978-0300181111.
20. Computer Architecture: A Quantitative Approach 5th Edition by John L. Hennessy (Author), David A. Patterson (Author), Morgan Kaufmann; 5 edition (September 30, 2011), 856 pages, ISBN-10: 012383872X, ISBN-13: 978-8178672663.
21. Microservices vs. Service-Oriented Architecture. by Mark Richards. Publisher: O'Reilly Media, Inc. Release Date: April 2016. ISBN: 9781491975657.
22. Ivan Zmerzlyi Microservice architecture – [Електронний ресурс] – 2019 – Режим доступу: <https://medium.com/@IvanZmerzlyi/microservice-architecture-f8a382291ff4>. Дата доступу: жовтень 2023.

23. Xiao Ma Microservice Architecture at Medium – [Електронний ресурс] – 2019 – Режим доступу: <https://medium.engineering/microservice-architecture-at-medium-9c33805eb74f>. Дата доступу: жовтень 2023.
24. Netflix MSA Platform – MeltingCon – [Електронний ресурс] – 2019 – <https://meltingcon.github.io/2018/assets/files/%EC%A0%95%EC%9C%A4%EC%A7%84.pdf>. Дата доступу: жовтень 2023.
25. Docker Cookbook/ by Sébastien Goasguen. Copyright © 2016 Sébastien Goasguen. All rights reserved. Printed in the United States of America. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
26. Чи завжди потрібні Docker, мікросервіси та реактивне програмування? – [Електронний ресурс] – 2019 – Режим доступу: <https://www.dataart.com.ua/news/chi-zavzhdi-potribni-dockermikroservisi-ta-reaktivne-programuvannya>. Дата доступу: жовтень 2023.
27. Kubernetes: Up and Running, 2nd Edition \ Dive into the Future of Infrastructure. By Brendan Burns, Kelsey Hightower, Joe Beda – Publisher: O'Reilly Media, Release Date: October 2023. Pages: 278.
28. Cloud Native DevOps with Kubernetes – by Justin Domingus, John Arundel. Publisher: O'Reilly Media, Inc. Release Date: March 2019. ISBN: 9781492040750.
29. Official Kubernetes documentation <https://kubernetes.io/docs/concepts/overview/what-iskubernetes>. Дата доступу: жовтень 2023.
30. Ivaschenko V., Shvachych G., Ivaschenko O., Busygin, V. Improving the efficiency of multiprocessors system through in-line interface network aggregation. Системні технології. Дніпро. 2018. No 2(115). P. 84 – 93. Книга "Мікросервіси. Патерни розробки та рефакторинг" (Microservices: Design Patterns and Refactoring): авторство - Г. Макклелланд, М. Джонсон, С. Хуттон.
31. Cloud-assisted body area networks: State-of-the-art and future challenges – [Електронний ресурс]. Режим доступу: <https://www.researchgate.net/>

publication/271918886\_Cloud-assisted-body-area\_networks\_State-of-the-art\_and\_future\_challenges.

32. The Fundamentals of Data Warehouse + Data Lake = Lake House. –  
Режим доступа:<https://towardsdatascience.com/the-fundamentals-of-data-warehouse-data-lake-lake>

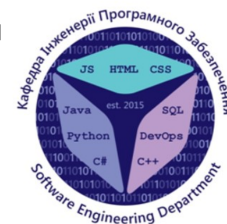
## ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО -КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО -НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### Розробка програмного забезпечення мовою C# для обліку та аналізу даних рейтингу викладача

Виконав студент 5 курсу  
Групи ППЗ-51  
Кузьменко Євген Євгенович  
Керівник роботи

Доктор Філософії ( PhD ), доцент кафедри ІПЗ Гребенюк Віктор Вікторович  
Київ – 2024

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** - автоматизація процесу обліку та аналізу даних рейтингу викладачів.
- **Об'єкт дослідження** – облік та аналіз даних рейтингу викладачів.
- **Предмет дослідження** – програмне забезпечення для обліку та аналізу даних рейтингу викладачів.



## ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Проаналізувати існуючі переваги та недоліки засобів ведення обліку та аналізу даних рейтингу викладачів.
2. Розробити функціональні та нефункціональні вимоги для автоматизації процесу обліку та аналізу даних рейтингу викладачів.
3. Розробити інтерфейс користувача з урахуванням брендового стилю університету.
4. Спроекувати діаграми класів, прецедентів та схему таблиць баз даних.
5. Розробити програмне забезпечення яке автоматизує підрахунок індивідуального рейтингу НПП по університету, окремій кафедрі, а також сумарний рейтинг по кафедрам та інститутам.
6. Провести ручне тестування готового застосунку.

3

## АНАЛІЗ АНАЛОГІВ

Функціональна можливість застосунку	Microsoft Word	Microsoft Excel	Система рейтинг
Реєстрація та авторизація користувачів	-	-	+
Облік та аналіз рейтингових карток викладачів	-	Рейтинг розраховується за допомогою формул	Рейтинг формується автоматично
Аналіз та формування рейтингових списків інститутів, кафедр	-	-	Рейтинг формується автоматично
Аналіз та формування рейтингових списків викладачів по кафедрі, університету	-	-	Рейтинг формується автоматично
Стандартизована рейтингова картка викладача	-	+	Сторінка формується автоматично та є однаковою для усіх викладачів
Підтримка платформ	Windows, MacOS	Windows, MacOS	Windows, MacOS, Linux, Android, iOS.
Ціна	Пакет із 2 додатків - 1899 грн на рік / 1 користувач		Безкоштовно / Не обмежено

4

## КЛЮЧОВІ ВИМОГИ ДО WEB-ЗАСТОСУНКУ

### Функціональні:

1. Викладач може авторизуватися у системі за e-mail та паролем.
2. Викладач може переглядати та заповнювати особисту картку рейтингу НПП.
3. Викладач може переглядати рейтинг викладачів.
4. Викладач може переглядати рейтинг викладачів певної кафедри.
5. Викладач може переглядати рейтинг інститутів університету.
6. Викладач може переглядати рейтинг кафедр університету.

### Нефункціональні

1. Система повинна зберігати паролі користувачів у зашифрованому вигляді.
2. Система повинна правильно відображатися у сучасних веббраузерах (Google Chrome, Mozilla Firefox)
3. Система повинна підтримувати українську мову.

5

## ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



**C#**  
Об'єктно-орієнтована  
мова програмування  
від Microsoft



**.NET**  
Платформа для  
запуску C# додатків на  
різних платформах



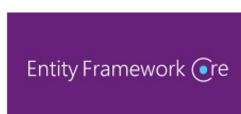
**ASP.NET**  
Веб-фреймворк  
для створення  
веб-додатків



**ASP.NET Identity**  
Система  
аутентифікації та  
авторизації



**MySQL**  
Реляційна база  
даних



**Entity framework**  
ORM-фреймворк  
для .NET



**HTML**  
Мова  
розмітки для  
створення  
веб-сторінок



**CSS**  
Мова стилів  
для  
оформлення  
веб-сторінок



**JavaScript**  
Мова  
програмування  
для додавання  
інтерактивності до  
веб-сторінок.

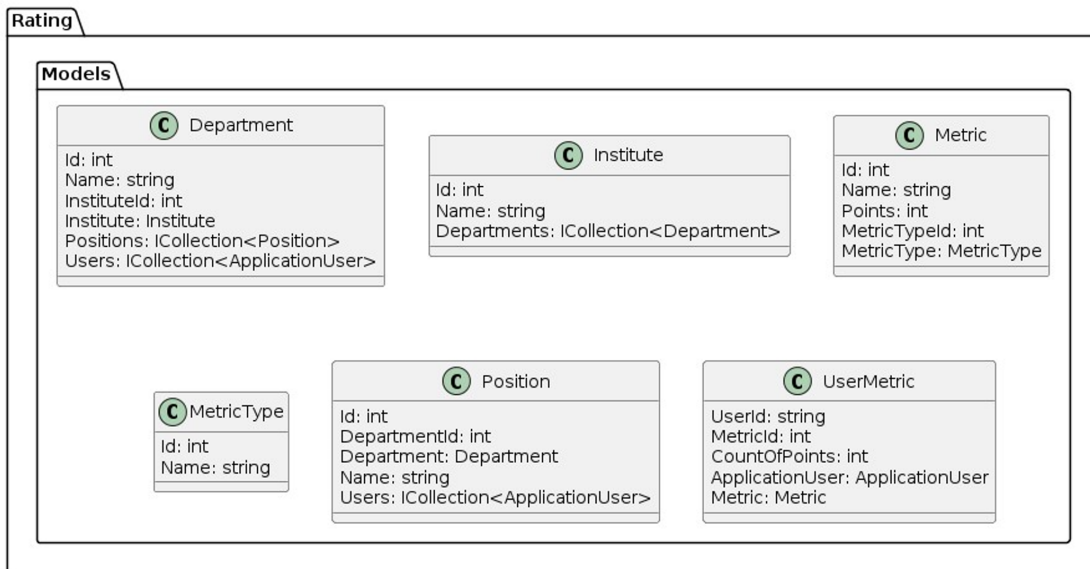
6

## ДІАГРАМА ПРЕЦЕДЕНТІВ



7

## ДІАГРАМА КЛАСІВ



8



## ЕКРАННІ ФОРМИ

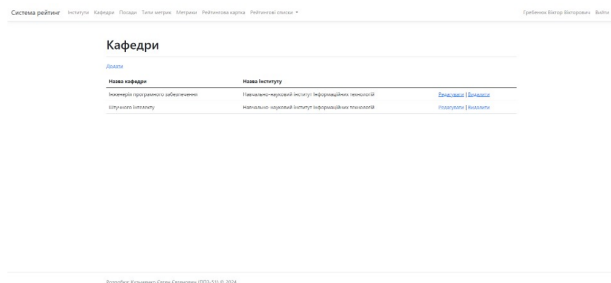


Рис. 3 Екран управління кафедрами

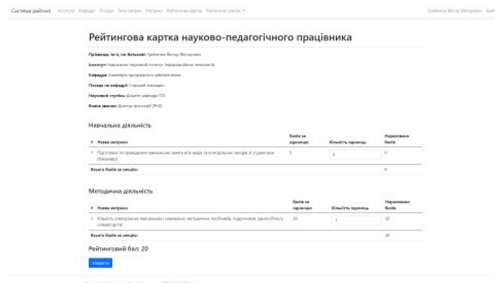


Рис. 4 – Екран заповнення рейтингової картки НПП

11

## ЕКРАННІ ФОРМИ

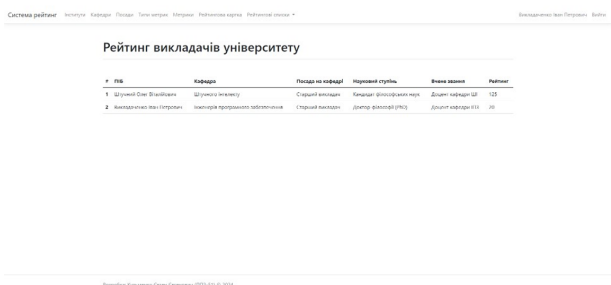


Рис. 5 Екран рейтингу викладачів університету

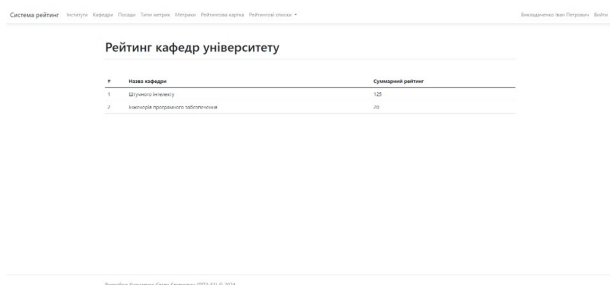


Рис. 6 – Екран рейтингу кафедр університету

12

## АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- 1. Кузьменко Є. Є., Розробка програмного забезпечення для обліку та аналізу даних. Науково-практична конференція «Перспективи та проблематика інтелектуальних систем», 31 травня 2024 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2024. С.45-46.
- 2. Кузьменко Є. Є., Гребенюк В. В. Розробка Web-застосунку для моніторингу аналітичних рейтингів науково-педагогічних працівників. IV Всеукраїнська науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті», 24 квітня 2024 р., Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.:ДУІКТ, 2024. С.34-35.

12

## ВИСНОВКИ

1. Проаналізовано та складено таблицю переваг та недоліків існуючих засобів ведення обліку та аналізу даних рейтингу викладачів.
2. Розроблено функціональні та нефункціональні вимоги для створення програмного забезпечення яке автоматизує процес обліку та аналізу даних рейтингу викладачів.
3. Розроблено інтерфейс користувача з урахуванням брендового стилю університету.
4. Спроектвані діаграми класів, прецедентів та схема таблиць баз даних.
5. Розроблено програмне забезпечення яке автоматизує підрахунок індивідуального рейтингу НПП по університету, окремій кафедрі, а також сумарний рейтинг по кафедрам та інститутам.
6. Проведено ручне тестування застосунку методами функціонального та димового тестування.

14