

РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет математики та інформатики

Кафедра цифрових технологій та методики навчання інформатики

«До захисту допущено»

Завідувач кафедри

_____ Ігор ВОЙТОВИЧ

«_____» _____ 2024р.

протокол № _____

ЯЛОВЕНКО ЛЮБОМИР

КВАЛІФІКАЦІЙНА РОБОТА

за освітнім ступенем «магістр»

на тему:

**РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЕКАНАТУ ДЛЯ
ЕФЕКТИВНОГО УПРАВЛІННЯ ОСВІТНІМ ПРОЦЕСОМ**

Виконав:

здобувач II курсу

групи М-ЦТ-2

спеціальності 015 «Професійна
освіта (Цифрові технології)»

Любомир ЯЛОВЕНКО

Науковий керівник:

кандидат педагогічних наук

Тетяна ШРОЛЬ

АНОТАЦІЯ

Яловенко Л.В. Розробка інформаційної системи деканату для ефективного управління освітнім процесом. Кваліфікаційна робота на здобуття освітнього ступеня магістра за спеціальністю 015 «Професійна освіта (Цифрові технології)». Рівненський державний гуманітарний університет, Рівне, 2024. 101 с.

Кваліфікаційна робота присвячена особливостям розробки інформаційної системи деканату, спрямованої на підвищення ефективності та автоматизації управління освітнім процесом у закладах вищої освіти.

У ході роботи здійснено теоретичний аналіз існуючих інформаційних систем управління освітнім процесом. Визначено переваги та недоліки таких систем, досліджено технології реалізації та особливості проектування інформаційних систем управління освітнім процесом. Визначено функціональні вимоги для інформаційної системи деканату відповідно до потреб користувачів.

У процесі розробки інформаційної системи деканату проведено порівняння вебфреймворків Django та Spring Boot, визначено переваги Django для реалізації проєкту. Обґрунтовано вибір стеку технологій для розробки інформаційної системи, побудованої за принципом архітектури MVT (Model-View-Template), який включає HTML, CSS, Bootstrap, JavaScript, AJAX та HTMX для фронтенду, Django для бекенду, SQLite як базу даних на етапі розробки з подальшим переходом на PostgreSQL, а також інструменти управління розробкою Git і GitHub.

У результаті кваліфікаційної роботи розроблено прототип інформаційної системи деканату. Безпосередньо в системі реалізовано функціональні модулі, які виконують функціональні вимоги таких користувачів як студенти, викладачі, менеджери та адміністратори. Окрім того, матеріали роботи можуть бути використані для навчання майбутніх ІТ-фахівців розробці інформаційних систем.

Ключові слова: інформаційна система, деканат, управління освітнім процесом, фреймворк Django, автоматизація.

ABSTRACT

Yalovenko L.V. Development of the dean's information system for effective management of the educational process. Qualifying work of the master on a specialty 015.39 Vocational Education (Digital Technologies). Rivne State University for the Humanities, Rivne, 2024. 101 p.

The qualification work is devoted to the peculiarities of developing an information system of the dean's office aimed at improving the efficiency and automation of educational process management in higher education institutions.

In the course of the work, a theoretical analysis of existing information systems for managing the educational process was carried out. The advantages and disadvantages of such systems are identified, the implementation technologies and design features of information systems for managing the educational process are studied. The functional requirements for the dean's information system are determined in accordance with the needs of users.

In the process of developing the dean's information system, a comparison of the Django and Spring Boot web frameworks is carried out, the advantages of Django for project implementation are determined. The choice of a technology stack for the development of an information system built on the principle of MVT (Model-View-Template) architecture, which includes HTML, CSS, Bootstrap, JavaScript, AJAX and HTMX for the frontend, Django for the backend, SQLite as a database at the development stage with further transition to PostgreSQL, as well as Git and GitHub development management tools, is substantiated.

As a result of the qualification work, a prototype of the dean's information system was developed. Functional modules that fulfil the functional requirements of such users as students, teachers, managers and administrators are implemented in the system. In addition, the materials of this work can be used to teach future IT specialists how to develop information systems.

Keywords: information system, dean's office, educational process management, Django framework, automation.

ЗМІСТ

ЗМІСТ	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	6
РОЗДІЛ I. ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ УПРАВЛІННЯ ОСВІТНІМ ПРОЦЕСОМ	8
1.1. Поняття інформаційної системи та особливостей управління освітнім процесом	8
1.2. Аналіз існуючих інформаційних систем для управління освітнім процесом.....	13
1.3. Організація та основні функції деканату	19
1.4. Сучасні технології розробки інформаційних систем для управління освітнім процесом	31
РОЗДІЛ II. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ ДЕКАНАТУ	37
2.1. Визначення та характеристика етапів проєктування ІС.....	37
2.2. Визначення функціональних вимог до ІС.....	39
2.3. Проєктування структури бази даних ІС	43
2.4. Обґрунтування вибору стеку технологій для розробки ІС	56
РОЗДІЛ III. РОЗРОБКА ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ОСВІТНІМ ПРОЦЕСОМ НА РІВНІ ДЕКАНАТУ	61
3.1. Особливості програмної реалізації архітектури та функціональних модулів системи	61
3.2. Організація тестування та налагодження ІС	67
3.3. Рекомендації щодо використання та впровадження ІС	70
ВИСНОВКИ	77
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТКИ	86

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІС	–	інформаційна система
ЗВО	–	заклад вищої освіти
MVC	–	Model View Controller
MVT	–	Model View Template
ІСУОП	–	Інформаційна система управління освітнім процесом

ВСТУП

В умовах цифрової трансформації освітніх процесів, інформаційні системи стають ключовими інструментами для підвищення ефективності управління вищими навчальними закладами. Університети та відповідно факультети стикаються з великою кількістю адміністративних задач, що включають розподіл навчального навантаження, контроль успішності студентів, автоматизацію документообігу та планування освітнього процесу. Використання інформаційних систем дозволяє оптимізувати ці процеси, підвищити оперативність прийняття управлінських рішень та знизити ризик людських помилок. Незважаючи на наявність існуючих рішень, багато деканатів потребують адаптації систем під специфічні вимоги закладу, що обумовлює актуальність дослідження. Необхідно створювати інноваційні рішення для ефективного управління освітнім процесом, інтеграції з іншими інформаційними ресурсами та підтримки аналізу великої кількості даних, що робить тему дослідження надзвичайно важливою як у науковому, так і в прикладному аспекті.

Мета дослідження: розглянути сучасні технології проєктування і розробки інформаційних систем та розробити на їх основі інформаційну систему деканату, що забезпечить ефективне управління освітнім процесом шляхом автоматизації ключових адміністративних процедур і планування освітнього процесу.

Відповідно до мети сформульовано ряд завдань:

- дослідити поняття та зміст інформаційної системи, зокрема систем управління освітнім процесом;
- проаналізувати існуючі інформаційні системи для управління освітнім процесом та виявити їх особливості, переваги і недоліки;
- дослідити технології реалізації та особливості проєктування інформаційних систем управління освітнім процесом;
- визначити функціональні вимоги для інформаційної системи деканату відповідно до потреб користувачів;
- розробити архітектуру та функціональні модулі системи;
- реалізувати прототип системи та перевірити його працездатність в умовах тестування;

– оцінити ефективність запропонованої системи та можливості її впровадження у закладах вищої освіти.

Об’єкт дослідження – управління освітнім процесом в закладах вищої освіти.

Предмет дослідження – технології розробки інформаційної системи, що забезпечує ефективне управління освітнім процесом на рівні деканату.

Для досягнення мети та завдань дослідження використані такі методи: *теоретичні* (аналіз, синтез, порівняння, узагальнення даних наукових та методичних джерел з розробки інформаційних систем та управління освітніми процесами); *емпіричні* (проєктування та тестування прототипу системи, аналіз отриманих результатів, оцінка ефективності впровадження).

Практичне значення результатів дослідження полягає в можливості впровадження розробленої інформаційної системи у закладах вищої освіти для підвищення ефективності управління освітнім процесом. Так, запропонована система сприятиме автоматизації рутинних задач на рівні деканату, що дозволить адміністрації закладу зосередитися на стратегічному плануванні та покращенні якості освіти. Крім того, матеріали роботи можуть бути використанні в підготовці майбутніх ІТ-фахівців для навчання програмуванню та розробки інформаційних систем засобами вебфреймворку Django мовою Python.

Основні теоретичні результати дослідження були представлені на III Всеукраїнській науково-практичній конференції «Підготовка педагогів до професійної діяльності в умовах змішаного навчання» (м. Рівне, 28-29 травня 2024р.) та XVII Всеукраїнської науково-практичної конференції «Інформаційні технології в професійній діяльності» (м. Рівне, 5 листопада 2024р.). В результаті опубліковано 2 тез конференцій (Яловенко, 2024а, с. 89-93), (Яловенко, 2024б, с. 114-117) та отримано відповідні сертифікати (додаток А).

Кваліфікаційна робота складається із переліку умовних позначень, вступу, трьох розділів, висновків, списку використаних джерел (51 найменування) та 4 додатків. Загальний обсяг роботи – 101 сторінка, основного тексту – 85 сторінок. Робота містить 16 таблиць та 27 рисунків.

РОЗДІЛ I

ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБКИ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ УПРАВЛІННЯ ОСВІТНІМ ПРОЦЕСОМ

1.1. Поняття інформаційної системи та особливостей управління освітнім процесом

В сучасному світі інформаційні системи стали невід'ємною частиною будь-якої сфери діяльності, особливо в галузі освіти, де вони слугують інструментом для підвищення ефективності управління освітнім процесом. Освітня сфера, яка завжди характеризувалася складністю своїх завдань і багаторівневістю структур, потребує нових підходів до організації, аналізу та оптимізації навчальних процесів. Саме тут на допомогу приходять інформаційні системи, що дозволяють інтегрувати всі аспекти освітньої діяльності, від управління ресурсами до моніторингу академічної успішності студентів.

Дослідженням інформаційних систем та впровадження їх в управління освітнім процесом закладів вищої освіти займаються Н. Кисіль, Л. Пархуць, Я. Лазор, С. Гордійчук, В. Ніколашин, Т. Куценко, Ю. Триус, І. Ощипок. Незважаючи на значний прогрес у розвитку таких систем, у наукових дослідженнях рідко зустрічаються огляди їх функціональних можливостей, особливо в контексті практичного застосування у вищих навчальних закладах.

Розглянувши численні наукові дослідження та публікації, можна дійти висновку, що єдиного універсального визначення поняття «інформаційна система» на сьогодні не існує. Це зумовлено багатогранністю самого явища, різноманітністю підходів до його вивчення, а також контекстуальною залежністю використання цього терміна в різних наукових дисциплінах та галузях практики. У роботі (Кисіль, Гаталяк, & Горбаль, 2004, с. 243) інформаційну систему, з технічної точки зору, визначають набором взаємозалежних компонентів, які виконують функції збирання, обробки, збереження і розподілу інформації для підтримки процесу прийняття рішень і управління організацією. У дослідженні (Лазор, 2017, с. 83) автор підходить до висновку, що «інформаційна система – це організований

комплекс організаційно-технічних заходів (сукупність підприємств, підрозділів і фахівців), а також безпосередньо інформаційних технологій і інформаційних ресурсів, призначених для функціонування інформаційних процесів, зокрема створення, поширення, використання, систематизації, збереження і знищення інформації». Автор роботи (Куценко & Попова, 2016, с. 99) вказує на те, що місією інформаційної системи є процес виробництва інформації за допомогою якої проводиться ефективне управління всіма ресурсами організації а також створення технічного та інформаційного середовища для управління її діяльністю. Звернувшись до нормативно-правових актів України визначенню терміну «інформаційна система» надаються такі роз'яснення:

- 1) «інформаційна система – це автоматизована система, комп'ютерна мережа або система зв'язку» (Про Положення про технічний захист інформації в Україні, 2008, пункт 5 частини 1 статті 2);
- 2) «інформаційна система – це автоматизована система, комп'ютерна мережа або система зв'язку» (Про затвердження Положення про дозвільний порядок проведення робіт з технічного захисту інформації для власних потреб, 2024, пункт 2 частини 1 статті 2);
- 3) «інформаційна (автоматизована) система – це організаційно-технічна система, в якій реалізується технологія обробки інформації з використанням технічних і програмних засобів» (Про захист інформації в інформаційно-телекомунікаційних системах, 2024, пункт 8 частини 1 статті 1).

У своєму дослідженні Н. Кисіль, З. Гаталяк і Н. Горбаль запропонували власну вдосконалену класифікацію інформаційних систем (рис.1.1) (Кисіль, Гаталяк, & Горбаль, 2004., с. 244).

За рівнем або сферою діяльності
• Державні • Територіальні (регіональні) • Міжгалузові • Об'єднання організацій • Галузові або відомчі
За ступенем інтеграції функцій
• Багаторівневі ІС зінтеграцією за рівнями управління • Багаторівневі ІС зінтеграцією за рівнями планування та ін.
За рівнем автоматизації
• Автоматизованого збирання та обробки інформації • Інформаційно-пошукові • Інформаційно-довідкові • Інформаційно-керівні (управлінські) • Документальні • Фактографічні • Інтелектуальні системи
За сферою застосування
• Системи для наукових досліджень • Системи автоматизованого проєктування • Системи організаційного управління • Системи управління організаційно-технічними процесами • Системи управління виробничими процесами • Системи управління технологічними процесами • Навчальні
За ступенем централізації обробки інформації
• Централізовані ІС • Децентралізовані ІС • ІС колективного використання
За функціональним призначенням
• Культурологічні • Владні • Науково-технічні • Соціальні • Фінансово-економічні • Міжнародні організації
За часом обробки інформації
• Реального часу • Квасіреального часу • Нереального (ірреального) часу

Рис. 1.1. Класифікація інформаційних систем

Отже, узагальнюючи різні підходи до визначення поняття «інформаційна система», можна зазначити, що вона є комплексним явищем, яке інтегрує технічні, організаційні, та інформаційні компоненти. Водночас, її роль у сфері освіти виходить за межі суто технічного забезпечення: інформаційні системи стають інструментом стратегічного управління, підвищення ефективності освітнього процесу, моніторингу його результативності та забезпечення прозорості діяльності закладів вищої освіти.

У своїй роботі С. Гордійчук (2016, с. 93) виокремлює основні переваги використання інформаційних систем управління освітнім процесом. Тоді як, в роботі (Триус, Кожем'якін, Аширова, & Заспа, 2020, с. 28) визначаються проблеми, які інформаційна система дозволяє вирішити, зокрема зниження складності управлінських рішень, обробку великих обсягів інформації, забезпечення оперативності прийняття рішень, координацію дій та збереження накопичених знань і досвіду. Крім того, автор виділяє основні характеристики інформаційної системи для ефективної розробки та впровадженні її в управління освітнім процесом і розглядає основні етапи створення інформаційної системи для підтримки освітньої діяльності, такі як: моделювання, проектування, розробка (рис. 1.2) (Триус та ін., 2020, с. 29-34).



Рис. 1.2. Основні етапи створення інформаційної системи для підтримки освітньої діяльності.

У роботі (Ощипок, & Петришин, 2023, с. 287) автор підсумовує, що інформаційна система ЗВО має інтегруватися із сучасними web-технологіями для підвищення прозорості та якості освітнього процесу, забезпечення актуальності даних і зменшення витрат ресурсів на підтримку інформаційної інфраструктури.

Отже, інформаційні системи стали ключовим інструментом у сучасному управлінні освітнім процесом, дозволяючи вирішувати численні завдання, які раніше потребували значних зусиль і ресурсів. Їх інтеграція в освітню сферу сприяє підвищенню ефективності управлінських рішень, оптимізації процесів моніторингу та аналізу даних, а також забезпечує прозорість і доступність інформації для всіх учасників освітнього процесу.

Подальший розвиток інформаційних систем у сфері освіти відкриває широкі перспективи для вдосконалення освітніх підходів, адаптації до сучасних викликів і потреб суспільства, а також забезпечення конкурентоспроможності закладів вищої освіти. Успішна реалізація цих завдань потребує тісної співпраці між науковцями, розробниками програмного забезпечення та освітніми установами, що дозволить максимально ефективно використовувати потенціал інформаційних технологій для розвитку освіти.

1.2. Аналіз існуючих інформаційних систем для управління освітнім процесом

Інформаційні системи (ІС) займають ключову роль у сучасному суспільстві, оскільки дозволяють зберігати, обробляти та передавати великі обсяги даних, необхідні для прийняття рішень та ефективної роботи організацій і підприємств. Ці системи є основою цифрової інфраструктури, що дозволяє забезпечувати оперативний доступ до інформації, об'єднуючи людей, процеси та технології в єдиний інформаційний простір.

Зокрема, під інформаційною системою розуміють «взаємозв'язану сукупність концепцій, методів, технологій, технічних і програмних засобів, використовуваних для автоматизації процесів збору, реєстрації, обробки, зберігання та надання інформації користувачу з метою досягнення поставлених цілей» (Юрчук, 2015, с. 54).

Інформаційні системи виконують широкий спектр функцій, які сприяють підтримці діяльності організацій і забезпечують ефективне управління інформаційними потоками. Завдяки різноманітним функціям ІС можна автоматизувати рутинні процеси, покращити якість прийняття рішень та підвищити продуктивність роботи. Розглянемо основні функції, що виконують інформаційні системи:

- «обробка інформації з усіх напрямків, що цікавлять систему управління;
- формування всієї необхідної для управління зовнішньої та внутрішньої інформації;

- забезпечення неперервного накопичення, систематизації, збереження і відновлення всієї необхідної інформації;
- передача необхідної інформації в задані пункти;
- реалізація швидкісного доступу, пошуку і видачі необхідної інформації;
- здійснення інформативно-керівного впливу на об'єкт управління і його ланки в разі відхилення їхніх параметрів функціонування від заданих значень;
- забезпечення оптимальних розрахунків в міру зміни мети, критеріїв та умов функціонування об'єктів управління;
- перевірка та ідентифікація документів;
- інформаційна безпека;
- визначення основних тенденцій, закономірностей та показників розвитку об'єкта управління;
- визначення основних показників техніко-економічного рівня виробництва і господарської діяльності;
- формування всіх обліково-звітних, планово-організаційних, конструкторсько-технологічних та інших форм документів» (Сазонець, 2014, с. 43-44).

Для ефективного управління інформаційними ресурсами існують різні підходи до класифікації інформаційних систем, що дозволяє обирати оптимальні рішення відповідно до конкретних умов та потреб організації. Зокрема, інформаційні системи можна класифікувати:

«1) за характером організації пошуку виокремлюють системи:

- інформаційно-пошукові;
- інформаційно-управлінські;
- інформаційно-довідкові.

2) за реалізацією режиму поширення та пошуку інформації розрізняють:

- системи з режимом вибіркового поширення та пошуку інформації;
- системи з ретроспективним режимом поширення і пошуку;
- інтегральні системи.

3) за характером функціональності інформаційні системи поділяють на:

- монофункціональні;
- багатофункціональні.

4) за масштабістю реалізації:

- глобальні;
- регіональні;
- локальні.

5) за рівнем безпеки:

- інформаційні системи з найвищим рівнем безпеки;
- з високим рівнем безпеки;
- з низьким рівнем безпеки» (Сазонець, 2014, с. 45-47).

6) за архітектурою:

- «настільні (Desktop), або локальні ІС, в яких всі компоненти (бази даних (БД), системи управління базами даних (СКБД), клієнтські програми) знаходяться на одному комп'ютері;
- розподілені (Distributed) ІС, в яких компоненти розподілені по декількох комп'ютерів.

7) за ступенем автоматизації:

- автоматизовані інформаційні системи, в яких автоматизація може бути неповною (тобто потрібне постійне втручання персоналу);
- автоматичні: інформаційні системи, в яких автоматизація є повною, тобто втручання персоналу не потрібно або потрібно лише епізодично.

8) за характером обробки, використання даних:

- інформаційно-пошукові системи виробляють введення, систематизацію, зберігання, видачу інформації з запиту користувача без складних перетворень даних;
- інформаційно-вирішальні системи здійснюють усі операції переробки інформації за певним алгоритмом.

9) за охопленням завдань:

- персональна ІС призначена для вирішення певного кола завдань однієї людини;

- групова ІС орієнтована на колективне використання інформації членами робочої групи або підрозділу;
- корпоративна ІС в ідеалі охоплює всі інформаційні процеси цілого підприємства, досягаючи їх повної узгодженості, беззбитковості і прозорості. Такі системи інколи називають системами комплексної автоматизації підприємства» (Дядюн, 2015, с.34-35).

В цілому інформаційні системи підрозділяються на:

- ERP (Enterprise Resource Planning) – це система для планування та управління ресурсами підприємства, яка інтегрує різні бізнес-процеси в єдину платформу, включаючи фінанси, постачання, виробництво та людські ресурси для покращення ефективності та прийняття рішень.

- CRM (Customer Relationship Management) – модель взаємодії, яка ставить клієнта в центр філософії бізнесу, фокусуючись на ефективному управлінні маркетингом, продажами та обслуговуванням клієнтів для підвищення лояльності та оптимізації взаємодії з ними.

- ECM (Enterprise Content Management) – стратегічна інфраструктура та технічна архітектура для підтримки єдиного життєвого циклу неструктурованої інформації (контенту), що охоплює всі типи та формати даних, від їх створення до зберігання та архівування.

- CPM (Corporate Performance Management) – концепція управління ефективністю підприємства, що включає стратегічне та фінансове управління, орієнтуючись на досягнення бізнес-цілей і поліпшення результативності організації.

- HRM (Human Resource Management) – сфера діяльності, яка забезпечує організацію кваліфікованими кадрами, займається оптимізацією використання персоналу та підтримкою розвитку людських ресурсів для досягнення цілей підприємства.

- EAM (Enterprise Asset Management) – інформаційна система для автоматизації процесів технічного обслуговування, ремонту та післяпродажного

обслуговування обладнання, забезпечуючи його ефективне використання та тривалу експлуатацію.

- EDMS (Electronic Document Management System) – система для управління електронними документами на підприємстві, включаючи їх створення, зберігання, пошук, обробку та архівування, забезпечуючи ефективний документообіг.

- Workflow (Business Process Management, BPM) – система для управління бізнес-процесами та документообігом, що охоплює всю організацію та включає автоматизацію процесів документообігу, від простих завдань до складних маршрутів і версій документів.

- Collaboration – система, яка підтримує електронну взаємодію людей, що не є формалізованою, як у випадку з workflow, і не є архівом, як у EDMS, але забезпечує зручний обмін інформацією та спільну роботу в реальному часі (Дядюн, 2015, с.35-36).

Окрім вище описаних систем варто виділити інформаційну систему управління освітнім процесом. ІСУОП – це сукупність цифрових інструментів та технологій, що використовуються для організації, контролю та оптимізації освітнього процесу. Вона надає можливості для автоматизації завдань, підвищення ефективності навчання, покращення управління та забезпечення комунікації між усіма учасниками освітнього середовища. Такі системи дозволяють централізовано керувати навчальними планами, розкладом занять, оцінюванням знань, обліком відвідуваності та іншими аспектами освітнього процесу.

В табл.1.1 представлено основні функції інформаційної системи управління освітнім процесом.

Таблиця 1.1

Основні функції інформаційної системи управління освітнім процесом

№	Функція	Опис
1	2	3
1	Управління навчальними планами та розкладом	Можливість створення, зміни та адміністрування розкладу занять, курсів, навчальних програм та матеріалів для полегшення організації освітнього процесу, покращення структури навчального плану і його доступності.
2	Контроль відвідуваності та успішності	Система надає інструменти для обліку відвідуваності занять і успішності студентів. Це дозволяє відслідковувати академічні показники кожного студента, надаючи адміністрації та викладачам необхідні дані для моніторингу та аналізу результатів.
3	Адміністрування навчальних матеріалів	Викладачі можуть завантажувати, зберігати та розповсюджувати навчальні матеріали через систему, надаючи студентам легкий доступ до всіх ресурсів, необхідних для навчання, будь то лекції, книги, презентації чи відео.
4	Автоматизація процесу оцінювання	ІСУОП спрощує процеси оцінювання завдяки автоматичній перевірці тестів, генерації звітів та наданню оцінок. Студенти можуть бачити свої результати в реальному часі, що сприяє своєчасному зворотному зв'язку.
5	Комунікація та підтримка спільноти	Більшість систем управління освітнім процесом надають можливість обміну повідомленнями між студентами, викладачами та адміністрацією. Це можуть бути чати, форуми або інтеграція з месенджерами, що забезпечує активну комунікацію та підтримку спільноти.
6	Аналітика та звітність	ІСУОП збирає та аналізує дані щодо успішності студентів, активності викладачів та загального стану освітнього процесу. Це дозволяє генерувати звіти та аналітику, необхідну для прийняття управлінських рішень і покращення навчального середовища.

Переваги використання ІСУОП:

– ефективність управління та комунікації спрощує взаємодію між адміністрацією, викладачами та студентами;

- прозорість процесу навчання дозволяє всім учасникам відстежувати свій прогрес та отримувати миттєвий зворотний зв'язок;
- автоматизація рутинних процесів економить час і ресурси, оптимізуючи адміністрування освітнього процесу;
- аналітика та оптимізація освітнього процесу надає дані для вдосконалення підходів у навчанні та підвищення результативності.

У сучасних закладах вищої освіти України активно впроваджуються автоматизовані системи управління, які допомагають підвищити ефективність організації освітнього процесу, забезпечити якісний облік студентів, викладачів, а також оптимізувати адміністративну діяльність. Нижче наведені основні системи, розроблені для різних вишів країни:

- «АСУ «Університет» – розробник ТОВ «UNITEX+» [<http://www.unitex.com.ua/products/commercial-software/automated-system-for-higher-education-institution/>];
- Пакет комп'ютерних систем ПП «Політек-софт» [<http://www.politek-soft.kiev.ua/index.php?do=products>];
- АСУ «ВНЗ» – розробник Науково-дослідний інститут прикладних інформаційних технологій [<http://ndipit.com.ua>, <https://vuz.osvita.net/ua/>];
- ІАС «Університет» – розробник Херсонський державний університет [<http://www.kspu.edu>];
- Електронна система управління ЗВО «Сократ» – розробник Вінницький національний аграрний університет [<http://socrates.vsau.org/index.php/ua/>];
- ІАС управління університетом – розробник Черкаський національний університет імені Богдана Хмельницького» (Триус та ін, 2020, с. 28-29).

1.3. Організація та основні функції деканату

Деканат є одним із ключових структурних підрозділів закладу вищої освіти, що виконує функції організації, управління та координації освітнього процесу. Сучасні інформаційні системи для автоматизації роботи деканату дозволяють

значно підвищити ефективність виконання цих завдань. Усі функції та модулі, які реалізуються в таких системах, спрямовані на створення зручного інструментарію для роботи з великим обсягом даних, забезпечення оперативного доступу до інформації, формування звітності та інтеграції з іншими модулями навчального закладу.

Інформаційна система деканату, зазвичай, включає модулі для роботи з навчальними планами, обліком студентів та викладачів, ведення звітності та документообігу. Основним завданням такої системи є створення централізованої бази даних, що містить інформацію про студентів, їх академічні успіхи, розклад занять, сесійні результати та інші аспекти освітнього процесу. Завдяки цьому деканат може швидко отримувати актуальні дані для прийняття рішень.

Однією з ключових функцій деканату є ведення бази даних студентів, що включає їхні персональні дані, навчальні досягнення, академічні довідки, а також інформацію про вибрані дисципліни. Ця інформація є основою для формування індивідуальних навчальних планів кожного студента. Система дозволяє не лише автоматизувати цей процес, але й забезпечити його гнучкість, що особливо важливо в умовах сучасних освітніх програм, які пропонують велику кількість дисциплін на вибір. Крім того, система деканату забезпечує облік навчальних контрактів, що є актуальним для студентів, які навчаються на платній основі.

Важливим аспектом діяльності деканату є контроль за виконанням освітнього процесу. У цьому контексті система дозволяє формувати розклад занять, сесій, перездач та автоматично оновлювати його в разі змін. Завдяки інтеграції з вебпорталом університету, ці дані стають доступними як для студентів, так і для викладачів у режимі реального часу. Особливу роль відіграє модуль ведення розкладу, який дозволяє уникати конфліктів між заняттями та максимально оптимізувати використання ресурсів навчального закладу.

Документообіг деканату також автоматизується завдяки інформаційним системам. Це включає формування різних видів довідок, звітів, академічних документів, таких як додатки до дипломів, та експорт даних до систем створення студентських квитків. Автоматизація цих процесів мінімізує можливість помилок,

пов'язаних із людським фактором, і значно скорочує час, необхідний для виконання рутинних завдань.

Інтеграція інформаційної системи деканату з іншими модулями університету є одним із ключових аспектів її функціонування. Зокрема, вона забезпечує обмін даними з системою управління кадрами для формування навчального навантаження викладачів, модулем наукових рад для підготовки звітності, а також модулем відділу кадрів для оновлення інформації про штат викладачів. Крім того, система передбачає можливість синхронізації з Єдиною державною електронною базою з питань освіти (ЄДЕБО), що забезпечує актуальність даних про студентів і їхні документи про освіту.

Таким чином, автоматизована інформаційна система деканату виконує роль центрального вузла координації освітнього процесу, забезпечуючи ефективне управління всіма аспектами освітньої діяльності. Вона дозволяє інтегрувати рутинні операції в єдиний інформаційний простір, забезпечити актуальність і точність даних, а також сприяти покращенню взаємодії між усіма учасниками освітнього процесу. Розвиток таких систем відкриває можливості для впровадження інноваційних підходів до організації навчання, які відповідають сучасним викликам у сфері освіти.

Для більш детального аналізу функціональних можливостей автоматизованих систем управління закладами вищої освіти доцільно порівняти три найпоширеніші програмні рішення: АСУ «Університет», розроблену ТОВ «UNITEX+», пакет комп'ютерних систем ППІ «Політек-софт» та АСУ «ВНЗ» від Науково-дослідного інституту прикладних інформаційних технологій. Кожна з цих систем має власні особливості в організації роботи модулів, що впливають на їхню ефективність і зручність у використанні. Аналіз та оцінка ключових функцій модулів цих систем щодо їхньої придатності для різних закладів освіти наведено в табл. 1.2.

Таблиця 1.2

Порівняння особливостей організації роботи модулів АСУ

№	АСУ	Модуль	Опис
1	2	3	4
1	АСУ «Університет»	Модуль «Структура ВНЗ»	Модуль «Структура ВНЗ» є основою для функціонування всієї АСУ «Університет». Він створює модель організаційної структури закладу, включаючи інформацію про кадровий склад і керівництво. Модуль визначає термінологію, якість оформлення екранних форм і звітів, що генеруються системою. Його довідник організований у деревоподібну структуру з оптимальним поділом на розділи, що забезпечує зручність навігації та використання («Автоматизована система управління ПП «Юнітех +»», б. д.).
		Модуль «Навчальна частина»	Модуль «Навчальна частина» автоматизує облік навчальних дисциплін, розподіл їх між кафедрами, а також формує навчальні плани для різних спеціальностей та форм навчання. Він обчислює навантаження викладачів і студентів, враховуючи академічні потоки, групи та ставки. Модуль генерує звіти про навчальне навантаження, а також автоматично конвертує їх у формати для подальшого використання, забезпечуючи ефективний документообіг («Автоматизована система управління ПП «Юнітех +»», б. д.).
		Модуль «Кафедра»	Модуль «Кафедра» автоматизує облік та управління діяльністю кафедри, включаючи ведення особових справ співробітників та кадрове забезпечення освітнього процесу. Він дозволяє розподіляти навчальне навантаження серед викладачів на основі розрахунків навчальної частини ЗВО, а також генерувати звіти про навантаження і ставки викладацького складу. Модуль підтримує складання розкладів, формування індивідуальних планів роботи співробітників та кафедри, а також тематичних планів з дисциплін. Завдяки автоматизації конвертації звітної інформації у формати *.rtf, *.xls, *.pdf, *.htm, забезпечується зручний документообіг («Автоматизована система управління ПП «Юнітех +»», б. д.).

1	2	3	4
		Модуль «Навчальний розклад»	Модуль «Навчальний розклад» автоматизує створення та облік навчальних розкладів у ЗВО. Він забезпечує паспортизацію аудиторного фонду та навчальних лабораторій, закріплених за підрозділами, а також формує графік освітнього процесу на основі даних про курс, ОКР та форму навчання студентів. Модуль автоматично формує розклад аудиторних занять, використовуючи дані про навчальне навантаження кафедр та викладачів. Він також підтримує конвертацію та експорт розкладів у формати *.rtf, *.xls, *.pdf, *.htm для подальшого використання за межами АСУ, а також візуалізацію розкладів на сайті ЗВО з можливістю фільтрації інформації («Автоматизована система управління ПП «Юнітех +»», б. д.).
		Модуль «Абітурієнт»	Модуль «Абітурієнт» автоматизує роботу Приймальної комісії ЗВО, включаючи реєстрацію абітурієнтів, введення їхніх даних, генерування необхідної документації та проведення конкурсного відбору. Всі дані про абітурієнта заносяться до бази, включаючи особисту інформацію, пільги, рівень освіти, вступні випробування тощо. Програма автоматично генерує заяви, договори, довідки та інші документи. Для проведення конкурсу формуються списки кандидатів на основі результатів випробувань, сертифікатів та пільг. Модуль також публікує результати на сайті ЗВО , формуючи аналітичні та звітні дані («Автоматизована система управління ПП «Юнітех +»», б. д.).
2		Модуль «Навчальний план»	Модуль «Навчальний план» для вищих навчальних закладів автоматизує створення, редагування та обрахунок навчальних планів. Вона дозволяє отримувати звітні документи та

1	2	3	4
	ПП «Політек-софт»		формувати вакансії для педнавантаження відповідно до норм Наказу Міністерства освіти і науки України №450. Система визначає години для консультацій, індивідуальних занять, заліків, екзаменів та інших видів навчального навантаження для різних форм навчання (денна, вечірня, заочна) («Пакет програм «Деканат» для вищих навчальних закладів III-IV рівнів акредитації», б. д.).
		Модуль «Навчальний процес (Університет)»	Модуль «Навчальний процес (Університет)» призначений для управління навчальною діяльністю у вищому навчальному закладі. Він відображає структуру освітнього процесу, обчислює навантаження підрозділів і викладачів, автоматизує його розподіл та аналіз. Модуль забезпечує підрахунок годин роботи викладачів за тиждень, семестр і рік, а також формує документи, такі як звіти про педагогічне навантаження, бюджети годин дисциплін, штатні списки кафедр тощо. Це полегшує управління навчальним процесом і сприяє оптимізації його організації («Пакет програм «Деканат» для вищих навчальних закладів III-IV рівнів акредитації», б. д.).
		Модуль «П-Кафедра-Web»	Модуль «ПС-Кафедра-Web» призначена для кафедр вищих навчальних закладів і працює через Web-інтерфейс, що усуває необхідність в її встановленні на місцях. Вона забезпечує фіксацію та аналіз виконання навчального навантаження кафедри протягом року, полегшуючи процес моніторингу та управління навчальним процесом без додаткових технічних налаштувань («Пакет програм «Деканат» для вищих навчальних закладів III-IV рівнів акредитації», б. д.).
		Модуль «Розклад»	Модуль «Розклад» призначена для формування розкладу занять з деталізацією для кожного тижня семестру на основі плану педнавантаження та даних про аудиторний фонд. Вона дозволяє

1	2	3	4
			друкувати розклад для навчальних груп, заповнення аудиторій та зайнятості викладачів. З версії від 20.05.2009 програма має функцію автоматичного складання розкладу, яка наразі працює в тестовому режимі («Пакет програм «Деканат» для вищих навчальних закладів III-IV рівнів акредитації», б. д.).
		Модуль «ПС-Студент-Web»	Модуль «ПС-Студент-Web» використовується деканатами ЗВО для обліку студентів та контролю їх успішності. Вона побудована на Web-інтерфейсі, що виключає необхідність установки на комп'ютери. Модуль фіксує події руху студентів і контролює успішність за традиційною та кредитно-модульною системами. Забезпечує отримання звітів по успішності студентів, семестрових та річних відомостей, зведених відомостей для диплому та інших звітів («Пакет програм «Деканат» для вищих навчальних закладів III-IV рівнів акредитації», б. д.).
		Модуль «ПС-Додаток до диплому-Web»	Модуль «ПС-Додаток до диплому-Web» призначена для друку додатків до дипломів нової форми, згідно з Наказом Міністерства освіти і науки України від 07.04.2000. Вона автоматично отримує необхідну інформацію для друку з бази даних, сформовану програмою «ПС-Студент-Web». Програма працює як об'єкт (Active X), який завантажується в інтернет-браузер через мережу з сервера («Пакет програм «Деканат» для вищих навчальних закладів III-IV рівнів акредитації», б. д.).
		Модуль «ПС-Академ. довідка-Web»	Модуль «ПС-Академ. довідка-Web» призначена для друку академічних довідок. Вона автоматично отримує необхідну інформацію з бази даних, сформовану програмою «ПС-Студент-Web». Програма побудована як об'єкт (Active X), який автоматично завантажується в

1	2	3	4
			інтернет-браузер через мережу з сервера («Пакет програм «Деканат» для вищих навчальних закладів III-IV рівнів акредитації», б. д.).
3	АСУ «ВНЗ»	Модуль «Студенти»	Модуль «Студенти» забезпечує комплексне збереження та управління інформацією про студентів у межах багатофункціональної анкети. У ній відображаються персональні дані, статус і деталі навчання, інформація про зарахування, випуск, студентський квиток, навчальну групу та місце проживання в гуртожитку. Дані про попередню освіту включають тип і назву закладу, документ про освіту, оцінки, а також вивчені іноземні мови. Модуль автоматично синхронізує зміни в анкеті з навчальною карткою студента, забезпечуючи актуальність інформації. Додатково враховуються пільги, участь у громадській діяльності, дані про успішність і результати довузівської підготовки. Система дозволяє ефективно обробляти всі аспекти інформації про студентів, забезпечуючи зручність і точність роботи («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «Контракти студентів»	Модуль «Контракти студентів» забезпечує ефективне управління контрактами та договорами студентів. Він дозволяє зберігати, відстежувати та оновлювати інформацію про фінансові угоди, такі як умови оплати, терміни, суми та інші важливі деталі контрактів. Цей модуль забезпечує прозорість процесу управління фінансовими зобов'язаннями студентів, спрощує ведення документації та надає можливість оперативно оновлювати дані («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «Сесії»	Модуль «Сесії» забезпечує автоматизацію процесів, пов'язаних з підготовкою екзаменаційних відомостей та веденням журналу екзаменаційних відомостей. Він включає функції генерації та

1	2	3	4
			додавання екзаменаційних відомостей, оформлення відомостей для перездачі, а також актуалізацію відомостей. Модуль також відповідає за занесення оцінок до бази даних: це включає занесення оцінок у відомість та проставлення оцінок для групи в базу даних, забезпечуючи правильне відображення оцінок напроти прізвищ студентів. Крім того, модуль дозволяє перевести студентів на наступний курс або здійснити відрахування та випуск студентів і груп («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «Навчальні плани»	Модуль «Навчальні плани» дозволяє створювати, редагувати та відстежувати навчальні плани для студентів, забезпечуючи доступ до розкладів, навчальних матеріалів і завдань. Він включає функції для внесення дисциплін, атестацій та практик, а також дозволяє редагувати додаткові параметри для практик. Модуль створює графік навчального процесу для кожного плану і забезпечує можливість друку навчального плану з усіма необхідними даними, що спрощує офлайн-використання та архівування інформації («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «Відділ кадрів»	Модуль «Відділ кадрів» забезпечує ефективне управління персоналом навчального закладу. Він дозволяє вести облік працівників, зберігати та оновлювати інформацію про їхню зайнятість, кваліфікацію, заробітну плату та інші кадрові дані. У програмі є форма «Анкета співробітника», яка включає закладки для заповнення особистої інформації, даних про посади, відпустки, стаж, освіту, родичів, військовий обов'язок та особливі відмітки, такі як нагороди та пільги. Закладка «Різне» дозволяє фіксувати додаткові дані, включаючи підвищення кваліфікації та наукову роботу. Також є можливість вести історію співробітника, що містить записи про зміни у його анкетних даних («АС «деканат» – АСУ «ВНЗ»», б. д.).

1	2	3	4
		Модуль «Освітньо-професійна програма підготовки (ОПП)»	Модуль «Освітньо-професійна програма підготовки (ОПП)» забезпечує визначення стандартів навчання для різних освітньо-кваліфікаційних рівнів (ОКР) та галузей знань. Стандарти можуть змінюватись щорічно, визначаючи кількість годин (кредитів) для кожної дисципліни, співвідношення між обов'язковими та вибірковими дисциплінами, а також їх розподіл між різними етапами навчання. Модуль також регулює кваліфікації, що можуть бути присвоєні студентам після завершення курсу за відповідним стандартом («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «Дисципліни вільного вибору студентів (ДВВС)»	Модуль «Дисципліни вільного вибору студентів (ДВВС)» призначений для управління дисциплінами, що пропонуються студентам як варіативна частина навчальних програм. Дисципліни групуються за наборами, типами та видами, що дозволяє реалізувати різні моделі роботи з варіативною складовою. Дані з цього модуля використовуються при створенні навчальних планів. Процес запису на дисципліни включає вибір студентами предметів, їх аналіз методистом, а також підтвердження або відмову від вибору дисциплін. Після цього методист додає підтверджені дисципліни до індивідуальних навчальних планів студентів через АС «Деканат» («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «ІНП студента (Індивідуальний навчальний план)»	Модуль «ІНП студента (Індивідуальний навчальний план)» забезпечує формування індивідуального плану для студента, який включає всі дисципліни, що студент має скласти, як разом із навчальною групою, так і індивідуально. ІНП може бути сформовано через «Журнал екзаменаційних відомостей» або шляхом прив'язки плану безпосередньо до студента («АС «деканат» – АСУ «ВНЗ»», б. д.).

1	2	3	4
		Модуль «Академічні довідки студента»	Модуль «Академічні довідки студента» забезпечує внесення інформації про академічні довідки, що надаються студентами при переведенні з інших ЗВО або факультетів («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «Індивідуальні відомості»	Модуль «Індивідуальні відомості» формує відомості для окремих дисциплін студента за потребою. Типи відомостей включають: підвищення оцінки, академічну різницю, індивідуальний графік, атестаційний лист екстерна та позапланову дисципліну («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «Створення замовлення додатків до дипломів»	Модуль «Створення замовлення додатків до дипломів» дозволяє замовляти друк додатків до дипломів, включаючи європейські додатки (DIPLOMA SUPPLEMENT). Після виконання операцій створюється файл у форматі XML («АС «деканат» – АСУ «ВНЗ»», б. д.).
		Модуль «Створення резервної копії»	Модуль «Створення резервної копії» забезпечує збереження важливих даних шляхом ручного створення резервних копій системи. Це дозволяє зберегти налаштування та інформацію для їх відновлення у разі збою чи втрати даних («АС «деканат» – АСУ «ВНЗ»», б. д.).

Підсумувавши можна сказати, що усі представлені системи мають свої переваги, однак кожна з них орієнтована на різні аспекти управління та автоматизації процесів.

АСУ «Університет», розроблена компанією ТОВ «UNITEX+», вирізняється широкою функціональністю в організації та управлінні навчальним процесом. Система містить модулі, що автоматизують процеси управління навчальним планом, розкладом, діяльністю кафедр, а також абітурієнтами. Крім того, система забезпечує зручну організацію даних та документообіг, що полегшує адміністративну роботу та зменшує ймовірність помилок. Система має розвинену структуру для формування звітності та збереження важливих даних у зручних для користувача форматах.

Пакет комп'ютерних систем ПП «Політек-софт» відрізняється інтеграцією з вебінтерфейсами для кафедр та студентів, що забезпечує зручність доступу та роботи без необхідності встановлення програмного забезпечення на локальні комп'ютери. Водночас система забезпечує автоматизацію процесів розподілу навчального навантаження, формування розкладів та аналізу успішності студентів, що дозволяє ефективно контролювати освітній процес. Важливою особливістю є можливість формування різноманітних звітів, що спрощує процес аналізу та планування діяльності ЗВО.

АСУ «ВНЗ» від Науково-дослідного інституту прикладних інформаційних технологій, у свою чергу, орієнтована на комплексне управління інформацією про студентів, викладачів і навчальні програми. Модулі, що стосуються створення та управління навчальними планами, відділом кадрів, а також обліком сесій та академічної успішності, дозволяють реалізувати повний цикл управлінських процесів в навчальному закладі. Додатково система забезпечує зручну роботу з академічними довідками та замовленнями додатків до дипломів, що може бути важливим для великих установ з великою кількістю студентів.

Таким чином, вибір найкращої автоматизованої системи залежить від конкретних потреб закладу вищої освіти. Якщо головною метою є забезпечення широкої автоматизації освітнього процесу та управління навчальними планами,

то АСУ «Університет» є оптимальним вибором. У випадку, коли важливими є зручність інтеграції з вебресурсами та спрощення моніторингу навчального навантаження, система ПП «Політек-софт» надає вагомі переваги. Для установ, які потребують глибокої аналітики студентських даних та керування кадровими процесами, найкращим варіантом є АСУ «ВНЗ», оскільки вона має розвинену функціональність в обліку та управлінні даними про студентів і викладачів.

1.4. Сучасні технології розробки інформаційних систем для управління освітнім процесом

У сучасному світі побудова інформаційних систем вимагає використання ефективних, продуктивних і гнучких засобів, особливо в управлінні навчальними процесами. Ці системи стають все складнішими, а вимоги до них стають все більш вимогливими. Жорстка конкуренція на ринку змушує розробників працювати швидше, швидко адаптуватися до змін і надавати користувачам безпечні, надійні та масштабовані рішення. Тому важливо вибрати правильний інструмент розробки, який не тільки відповідає вашим технічним потребам, але й дозволяє оптимізувати час і ресурси, необхідні для реалізації вашого проєкту. Існує багато інструментів для розробки вебзастосунків, але виділяються два фреймворки: Django та Spring Boot.

Вебфреймворк Django – «це безкоштовна платформа з відкритим кодом для розробки мовою Python, розгорнута на вебсервері, що може допомогти розробникам швидко створити вебінтерфейс, який є багатофункціональним, безпечним і масштабованим» (Яловенко & Шроль, 2024а, с. 91). Django слідує архітектурному патерну модель–вид–контролер (MVC), де модель представляє структуру даних, вид відповідає за інтерфейс користувача, а контролер керує взаємодією між моделлю та видом. Вебфреймворк використовує простоту й читабельність мови Python, а також багату бібліотеку, яка сприяє розвитку екосистеми фреймворку та полегшує інтеграцію з різними інструментами і технологіями. Django використовує архітектурний патерн «Модель–Вид–

Шаблон» (MVT), який є варіацією моделі MVC. Він забезпечує розділення відповідальностей і модульність дизайну додатків.

Модель представляє структуру даних і відповідає за взаємодію з базою даних через ORM (Object Relational Mapping). Вид займається обробкою запитів користувачів і взаємодією з моделлю, а шаблон визначає, як дані відображаються, використовуючи вбудований механізм створення динамічного HTML. Django має потужну систему ORM, яка дозволяє працювати з базами даних через об'єкти Python, абстрагуючи SQL-запити, і підтримує такі бази даних, як PostgreSQL, MySQL, SQLite та Oracle. Завдяки активній спільноті та доступній документації Django легко вивчати навіть новачкам. Він слідує принципу «з коробки», надаючи вбудовані інструменти, такі як ORM, адмін-інтерфейс та механізми аутентифікації. Це дозволяє пришвидшити розробку, але водночас обмежує гнучкість. Проста структура та дизайн сприяють швидкому засвоєнню цього фреймворку, навіть якщо у користувача немає значного досвіду в програмуванні. Django пропонує інструменти для юніт-, інтеграційного та асинхронного тестування, а також вбудовані механізми безпеки, що захищають від XSS, CSRF та інших загроз.

Розпочати розробку вебзастосунку за допомогою фреймворку Django набагато ефективніше, ніж починати з нуля, оскільки вам потрібно окремо розробляти бекенд, API, JavaScript і карту сайту. Django дозволяє веброзробникам зосередитися на розробці унікальної функціональності програми з більшою гнучкістю, ніж використання основних інструментів веброзробки (Mantosh, 2023).

Django став основою для всесвітньо відомих проєктів, таких як Instagram, YouTube, NASA, Spotify, DropBox і Pinterest.

В табл. 1.3 представлено переваги та недоліки вебфреймворку Django.

Таблиця 1.3

Переваги та недоліки вебфреймворку Django

№	Переваги	Недоліки
1	Універсальна структура. Вебфреймворк підходить для розробки додатків різного рівня складності, таких як соціальні мережі, сайти новин і системи керування контентом. Крім того, він підтримує інтеграцію з вмістом у будь-якому форматі (HTML, XML, JSON тощо) і сумісний із будь-яким фреймворком JavaScript.	Монолітна структура платформи. Це може бути перевагою чи недоліком, залежно від потреб користувача. Фреймворк має визначений набір файлів і змінних, а його файлову структуру легко зрозуміти та вивчити, але це не дозволяє вам реалізувати свою власну.
2	Надання базових функцій безпеки: керування обліковими записами користувачів, контроль транзакцій, захист від підроблених міжсайтових запитів, клікджекінгу тощо.	Більше підходить для масштабних проєктів. Усі функції містять велику кількість коду, який потребує часу на обробку сервером і може спричинити проблеми з низькопродуктивним обладнанням.
3	Масштабованість. Цей фреймворк дотримується шаблонів і принципів дизайну, які сприяють повторному використанню коду та легкому обслуговуванню. Основний принцип «ніколи не повторювати» гарантує уникнення дублювання. Завдяки компонентній архітектурі Django кожен рівень незалежний від інших, що дозволяє розширювати вашу програму на будь-якому етапі розробки.	
4	Інтегрована система інтернаціоналізації. Можливість розробляти багатомовні сайти без встановлення додаткових плагінів.	

Spring Boot – «це вебфреймворк на основі Java з відкритим кодом, який побудований на основі популярного Spring Framework і використовує його функції та можливості» (Яловенко & Шроль, 2024а, с. 91). Spring Boot

використовує основу з фреймворку Spring та дотримується принципу «налаштування за умовчанням». Це означає, що він надає розумні стандартні налаштування, спрощуючи роботу та зменшуючи потребу в складному налаштуванні й повторюваному коді. Spring Boot підтримує стандартну архітектуру MVC (Model-View-Controller), яка популярна в розробці вебзастосунків. У цій архітектурі модель описує структуру даних і бізнес-логіку програми, вид займається представленням даних, отриманих від моделі, а контролер керує потоком даних між моделлю і видом.

Фреймворк також підтримує мікросервісну архітектуру, що сприяє модульності та розподіленому підходу. Використання Java, сувора типізація та висока продуктивність роблять Spring Boot придатним для корпоративних додатків. Він спрощує доступ до баз даних завдяки проєкту Spring Data, підтримуючи як реляційні бази даних (наприклад, MySQL, PostgreSQL), так і NoSQL (наприклад, MongoDB). Spring Boot інтегрується з JUnit для тестування, забезпечуючи можливість юніт-, інтеграційного та кінцевого тестування. Його модульна архітектура дозволяє кастомізувати конфігурації, що робить його ідеальним вибором для проєктів із різними вимогами, зокрема мікросервісів.

Окрім цього, Spring Boot пропонує зручний підхід до розробки корпоративних застосунків, надаючи конфігурацію за замовчуванням і усуває необхідність використовувати шаблонний код. Фреймворк орієнтований на зменшення складності налаштування та розгортання Spring-застосунків, впроваджуючи принцип «конвенція важливіша за конфігурацію» («Django vs spring boot 2023 – know the differences – cyber sucess», 2023).

Spring Boot широко застосовується компаніями різного масштабу – від стартапів до світових гігантів, таких як Netflix, Spotify і Amazon.

На рис. 1.3 визначено переваги та недоліки вебфреймворку Spring Boot (Керри, 2023).



Рис. 1.3. Переваги та недоліки вебфреймворку Spring Boot

В табл. 1.4 представлено порівняльну характеристику вебфреймворків Django і Spring Boot.

Таблиця 1.4

Порівняльна характеристика вебфреймворків

№ п/п	Параметри	Django	Spring Boot
1	2	3	4
1.	Мова	Python	Java
2.	Екосистема	Багата бібліотеками, активна спільнота	Всеосяжна екосистема Spring
3.	Архітектура	Модель–Вид–Шаблон (MVT)	Модель–Вид–Контролер (MVC), мікросервіси
4.	Підтримка баз даних	ORM-система, підтримка багатьох БД	Spring Data, широка сумісність з базами даних

1	2	3	4
5.	Спільнота та документація	Жвава спільнота, зручна документація для початківців	Розширена спільнота Spring, детальна документація
6.	Гнучкість і кастомізація	Вбудовані інструменти, менше можливостей для налаштування	Модульний, кастомізований, легкий підхід
7.	Крива навчання	Відносно плавна, Django Admin для спрощення	Крутіша, але з докладною документацією
8.	Підтримка тестування	Потужна система тестування	Інтеграція з JUnit, розширена підтримка тестування
9.	Особливості безпеки	Вбудовані механізми безпеки	Надійні механізми безпеки, модульний підхід

Тому вибір між Django і Spring Boot залежить від конкретних потреб вашого проєкту. Якщо метою є створити працездатну вебпрограму з мінімальною конфігурацією, Django стане вдалим вибором. Якщо для важливі масштабованість, інтеграція з різними системами та гнучкість архітектури для адаптації до мінливих вимог, то краще вибрати Spring Boot. У будь-якому випадку, обидва фреймворки надають потужні інструменти для розробки сучасних інформаційних систем, особливо для управління освітніми процесами.

РОЗДІЛ II

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ ДЕКАНАТУ

2.1. Визначення та характеристика етапів проєктування ІС

Під час розробки інформаційної системи для організації роботи деканату було проведено комплексний процес проєктування, який охоплює низку ключових етапів, спрямованих на створення ефективного інструменту для управління освітнім процесом. Кожен етап відіграє важливу роль у формуванні цілісного та функціонального рішення, що відповідає сучасним вимогам і потребам користувачів. Від аналізу предметної області до тестування готової системи всі дії були спрямовані на досягнення високої якості кінцевого продукту та забезпечення його зручності у використанні.

Першим етапом – це дослідження предметної області. Його метою є глибокий аналіз наявних рішень у сфері інформаційних систем, які використовуються для управління навчальним процесом у вищих навчальних закладах. Цей етап передбачав ознайомлення з функціональними можливостями вже існуючих систем, оцінку їхніх переваг та недоліків, а також визначення потреб і очікувань користувачів. Особливу увагу було приділено тому, які функції повинна виконувати інформаційна система, щоб бути зручною, функціональною та відповідати сучасним вимогам. У результаті цього етапу було сформовано чітке уявлення про основні функціональні модулі майбутньої системи, зокрема управління даними про студентів, планування освітнього процесу та облік успішності.

Другий етап проєктування полягає у виборі технологій, які використовуватимуться під час розробки. Цей етап є одним із найважливіших, оскільки вибір відповідних технологій впливає на ефективність роботи системи, її масштабованість, стабільність та безпеку. Ураховуючи вимоги проєкту, було проведено оцінку різних технологічних рішень, після чого обрано ті, які найкраще відповідали поставленим завданням. Основний акцент було зроблено на

використанні сучасних фреймворків і мов програмування, що забезпечують швидку розробку, легку підтримку і можливість майбутнього розширення системи.

На третьому етапі здійснюється проєктування інтерфейсу користувача. Цей процес базується на принципах мінімалізму та зручності використання, що особливо важливо для систем, які мають щоденну взаємодію з великою кількістю користувачів. У ході цього етапу були розроблені макети, що демонструють, як виглядатимуть основні екрани та елементи управління. Мінімалістичний дизайн інтерфейсу дозволяє користувачам швидко орієнтуватися в системі, зменшує візуальне навантаження та сприяє підвищенню загальної продуктивності роботи.

Наступним *четвертим етапом* є frontend розробка, що зосереджується на створенні клієнтської частини системи. У ході роботи було реалізовано візуальні компоненти, розроблені під час проєктування інтерфейсу, та забезпечено їхню інтерактивність. Для цього використовували сучасні технології, зокрема HTML, CSS, JavaScript і фреймворк Bootstrap, що дозволило створити адаптивний дизайн, здатний коректно відображатися на різних пристроях. Значну увагу було приділено забезпеченню високої продуктивності роботи інтерфейсу, а також його стабільної роботи у різних браузерах, що зробило систему зручною та надійною у використанні.

П'ятий етап присвячено розробці back-end частини системи, яка забезпечує обробку даних, функціонування бізнес-логіки та взаємодію з базами даних. Для реалізації серверної частини використовується фреймворк Django, що дозволяє створити надійну та масштабовану архітектуру. Було впроваджено модулі для авторизації, управління користувачами, персоналізованих розкладів та інтеграції з календарними сервісами. Система також забезпечує можливість внесення даних про дисципліни, управління розкладами та генерування звітів, що відповідає потребам усіх категорій користувачів, від студентів до адміністраторів. Усі елементи були спроектовані з урахуванням гнучкості, безпеки та можливості подальшого вдосконалення.

Останній етап передбачає тестування розробленої системи. Цей процес включає перевірку її функціональності, надійності та безпеки. На цьому етапі виявляються та усуваються можливі помилки, а також здійснюється оптимізація роботи окремих модулів. Тестування розробленої ІС проводилася в кілька етапів: від базового функціонального до стресового та навантажувального, щоб упевнитися в здатності системи працювати стабільно навіть у складних умовах.

Кожен із описаних етапів є невід'ємною частиною процесу проектування інформаційної системи деканату. Послідовність цих етапів дозволяє забезпечити можливість створення якісного, функціонального та зручного програмного продукту, що відповідає сучасним вимогам та сприяти ефективному управлінню освітнім процесом у навчальному закладі.

2.2. Визначення функціональних вимог до ІС

Основними завданнями інформаційного забезпечення трансформації освіти є підтримка рівного доступу до якісної освіти, забезпечення справедливості у розподілі ресурсів, а також підвищення ефективності та якості освітніх послуг шляхом впровадження систем моніторингу та аналізу даних. Таку підтримку забезпечують сучасні інформаційні системи управління освітою (Educational Management Information Systems, EMIS). Основні функції EMIS включають збір, зберігання, інтеграцію та обробку освітніх даних і статистики з використанням сучасного програмного та апаратного забезпечення. Зокрема, під освітньою управлінською інформаційною системою розуміють « не тільки набір формалізованих та інтегрованих операційних процесів, процедур та організаційно-правових заходів, але й повноцінна інституційна культура, яка забезпечує суспільство актуальними та достовірними даними про стан розвитку освіти» (Лондар та ін., 2020, с. 5).

Інформаційна система управління освітнім процесом є важливим інструментом для ефективної координації та оптимізації діяльності закладів освіти. Вона забезпечує автоматизацію багатьох рутинних процесів, що стосуються планування занять, моніторингу результатів навчання, а також

управління розкладом і робочими планами. Система допомагає налагодити безперебійну комунікацію між студентами, викладачами, адміністрацією та іншими учасниками освітнього процесу, підвищуючи якість навчання та управління.

Для того, щоб ефективно організувати освітній процес, необхідно насамперед визначити його цілі та завдання, окреслити основні проблеми, які необхідно вирішити, визначити учасників процесу, уточнити їхні ролі та методи взаємодії. Це перший крок у системному аналізі системи. На другому етапі необхідно розробити можливі варіанти вирішення основної проблеми та запропонувати шляхи вдосконалення системи (Львов, б. д., с. 3)

Також важливим етапом при розробці такої системи є чітке визначення її функціональних вимог, які повинні відповідати потребам різних категорій користувачів і сприяти досягненню основних завдань освітньої установи.

Ключовими принципами побудови такої системи є зручність використання, доступність та актуальність інформації. Інтуїтивно зрозумілий інтерфейс повинен забезпечувати легкий доступ до необхідних даних, що дозволить користувачам швидко знаходити потрібну інформацію. Важливо, щоб система підтримувала доступ з різних пристроїв, включаючи комп'ютери, смартфони та планшети, і працювала на різних операційних системах, таких як Windows, macOS, Android та iOS. Це сприятиме тому, щоб користувачі могли отримувати інформацію та взаємодіяти з системою без прив'язки до одного типу пристрою або програмного забезпечення.

Однією з ключових вимог є актуальність даних, що зобов'язує систему автоматично оновлювати розклад та іншу інформацію в режимі реального часу. Це забезпечує своєчасне інформування студентів та викладачів про будь-які зміни у графіку, що запобігає плутанині та допомагає ефективно планувати свій час. Система також має забезпечувати повноту та деталізацію інформації. Наприклад, розклад повинен містити не лише час проведення занять, а й деталі про вид заняття (лекція, семінар, лабораторна робота), назву дисципліни, аудиторію та викладача. Додатковою перевагою є можливість інтерактивного доступу до

додаткових даних про курси та викладачів, що дозволяє студентам глибше ознайомитися з програмою навчання.

Функціональні можливості системи залежать від ролі кожного користувача:

- студента;
- викладача;
- методиста;
- менеджера деканату;
- адміністратора.

Кожна з цих категорій має власні потреби та пріоритети, що впливають на інструменти й можливості, які надає система.



Рис. 2.1. Діаграма варіантів використання інформаційної системи

Для студентів особливо важливо мати доступ до персоналізованого розкладу, який можна фільтрувати за різними параметрами, такими як факультет, курс, група або викладач. Інтеграція з календарними сервісами дозволяє студентам ефективніше планувати свій час, а також вчасно отримувати сповіщення про зміни або нагадування про важливі події. Викладачі, зі свого боку,

потребують не лише доступу до розкладу, а й можливості перегляду інформації про свої дисципліни та кількість відпрацьованих годин. Доступ до електронних журналів дозволяє викладачам вести облік відвідуваності та успішності студентів безпосередньо в системі, що підвищує прозорість та зменшує паперову роботу.

Методисти освітніх установ мають інші потреби. Їх робота пов'язана з формуванням навчальних планів і внесенням даних про дисципліни, кількість годин, кредити та форми контролю. Інформаційна система повинна надати їм інструменти для внесення та редагування цієї інформації, а також для генерації звітів, які допоможуть аналізувати ефективність освітнього процесу. Можливість аналізу даних є важливим компонентом для оптимізації освітньої діяльності, що дозволяє приймати обґрунтовані рішення щодо змін у програмах або розкладах.

Менеджери деканату виконують важливі функції з управління освітнім процесом і потребують доступу до інструментів для формування груп студентів, створення розкладів, управління обліковими записами співробітників і розробки графіків освітнього процесу. Інформаційна система має допомогти їм забезпечувати координацію між різними підрозділами та уникати конфліктів у розкладі. Можливість редагування та оновлення розкладу в режимі реального часу є критичною для ефективного управління процесом навчання.

Адміністратори системи володіють найширшими можливостями, оскільки їх завдання включають управління обліковими записами всіх користувачів, моніторинг роботи системи та забезпечення безпеки даних. Вони мають можливість додавати, видаляти та блокувати користувачів, а також встановлювати рівні доступу для різних категорій співробітників. Система повинна забезпечувати багаторівневу авторизацію, щоб захистити персональні дані користувачів і запобігти несанкціонованому доступу. Важливим компонентом є також резервне копіювання інформації, що гарантує відновлення даних у разі технічних збоїв.

Сучасні інформаційні системи можуть включати додаткові можливості для покращення взаємодії між користувачами. Наприклад, інтеграція чату або функцій для обміну повідомленнями дозволяє студентам і викладачам

спілкуватися безпосередньо в межах платформи. Сповіщення про важливі події або зміни в розкладі також сприяють оперативному інформуванню та запобіганню непорозумінь.

Забезпечення безпеки та конфіденційності даних є однією з пріоритетних задач. Для цього важливо використовувати сучасні протоколи шифрування та системи авторизації, такі як двофакторна автентифікація. Це гарантує, що лише уповноважені особи матимуть доступ до конфіденційної інформації. Крім того, система повинна забезпечувати зручне резервне копіювання, що дозволить швидко відновити дані у разі виникнення технічних проблем або кібератак.

Таким чином, визначення функціональних вимог до інформаційної системи управління освітнім процесом є комплексним завданням, що охоплює багато аспектів. Важливо врахувати потреби кожної категорії користувачів, забезпечити зручність використання, інтеграцію з іншими сервісами та підтримку різних пристроїв. Окрім цього, система повинна мати високий рівень безпеки та підтримувати оперативне оновлення даних. Реалізація таких вимог сприятиме підвищенню ефективності управління освітнім процесом, покращенню взаємодії між учасниками та забезпеченню безперервного доступу до актуальної інформації для всіх користувачів системи.

2.3. Проєктування структури бази даних ІС

Для створення інформаційної системи потрібно спроектувати базу даних, яка б відповідала попередньо визначеним вимогам, оскільки від її правильної організації залежить ефективність обробки даних, масштабованість та надійність системи. У контексті розробки інформаційної системи деканату для ефективного управління освітнім процесом, важливо забезпечити структуру, яка відображатиме взаємозв'язки між студентами, викладачами, дисциплінами, розкладом занять та іншими елементами освітнього процесу. В проєктуванні потрібно описати підходи до моделювання бази даних, визначити основні сутності і зв'язки між ними, а також розробити модель, що відповідає специфіці завдань системи та забезпечує зручність роботи з даними.

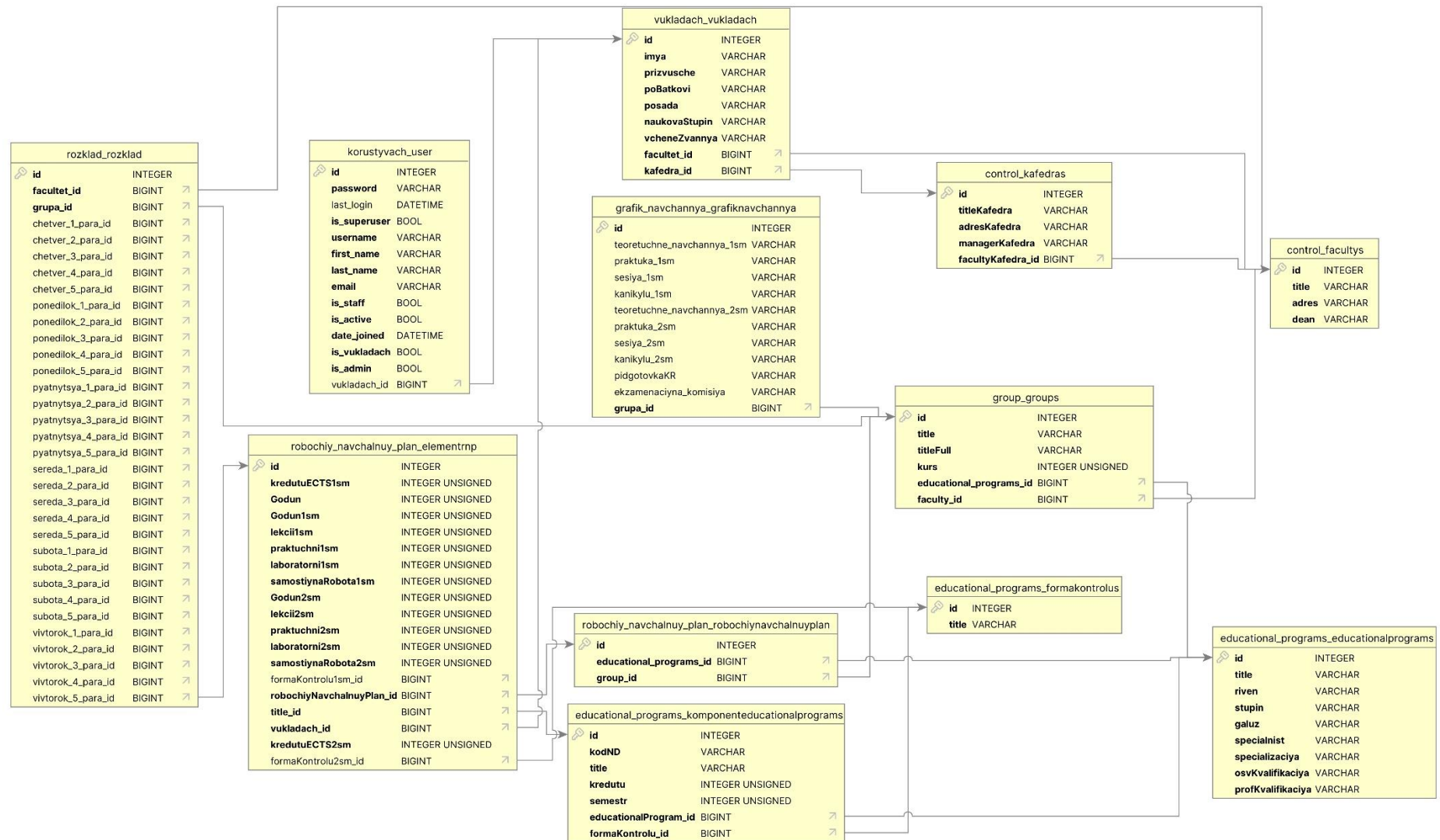


Рис. 2.2. Схема структури бази даних інформаційної системи

База даних передбачає певні сутності:

- факультет;
- кафедра;
- група;
- викладач;
- освітня програма;
- компонент освітньої програми;
- робочий навчальний план;
- елемент робочого навчального плану;
- користувач;
- графік навчання;
- розклад;
- формат контролю.

Сутність «Факультет» передбачає збереження номера ідентифікатора, назву, адресу та ППІ декана факультету. Пов'язана зв'язком один до багатьох із сутностями «Група», «Розклад», «Викладач» та «Кафедра».

Таблиця 2.1

Поля сутності «Факультет»

Назва поля	Опис	Тип даних
Id	Номер ідентифікатора (первинний ключ)	Integer
Title	Назва факультету	varchar(64)
Adres	Адреса факультету	varchar(64)
Dean	Декан факультету (ППІ)	varchar(64)

Сутність «Кафедра» передбачає збереження номера ідентифікатора, назву, адресу, ППІ завідувача кафедри та факультет до якого відноситься. Пов'язана зв'язком один до багатьох із сутністю «Викладач».

Таблиця 2.2

Поля сутності «Кафедра»

Назва поля	Опис	Тип даних
Id	Номер ідентифікатора (первинний ключ)	Integer
titleKafedra	Назва кафедри	varchar(64)
adresKafedra	Адреса кафедри	varchar(64)
managerKafedra	Завідувач кафедри	varchar(64)
FacultyKafedra_ id	Номер ідентифікатора факультету (зовнішній ключ)	Integer

Сутність «Група» передбачає збереження номера ідентифікатора, коротку назву групи, повну назву групи, курс на якому навчається група, ідентифікатор викладача який є куратором групи, ідентифікатор факультету до якого прив'язана група та ідентифікатор освітньої програми за якою навчається група. Пов'язана зв'язком один до багатьох із сутностями «Робочий навчальний план», «Розклад» та «Графік навчання».

Таблиця 2.3

Поля сутності «Група»

Назва поля	Опис	Тип даних
1	2	3
Id	Номер ідентифікатора (первинний ключ)	integer
Title	Коротка назва групи	varchar(64)
titleFull	Повна назва групи	varchar(64)
Kurs	Курс	integer

1	2	3
faculty	Номер ідентифікатора факультету (зовнішній ключ)	integer
educational_programs	Номер ідентифікатора освітньої програми (зовнішній ключ)	integer

Сутність «Викладач» передбачає збереження номера ідентифікатора, прізвище, ім'я, по батькові викладача, займана посада, наукова ступінь, вчене звання, кафедра та факультет до яких відноситься викладач. Пов'язана зв'язком один до багатьох із сутностями «Користувач» та «Елемент робочого навчального плану».

Таблиця 2.4

Поля сутності «Викладач»

Назва поля	Опис	Тип даних
id	Номер ідентифікатора (первинний ключ)	integer
prizvusche	Прізвище викладача	varchar(64)
imya	Ім'я викладача	varchar(64)
poBatkovi	По батькові викладача	varchar(64)
posada	Посада викладача	varchar(64)
naukovaStupin	Наукова ступінь викладача	varchar(64)
vcheneZvannya	Вчене звання викладача	varchar(64)
facultet	Номер ідентифікатора факультету (зовнішній ключ)	integer
kafedra	Номер ідентифікатора кафедри (зовнішній ключ)	integer

Сутність «Освітня програма» передбачає збереження номера ідентифікатора, назва освітньої програми, рівень, ступінь, галузь, спеціальність, спеціалізацію, освітню кваліфікацію та професійну кваліфікацію. Пов'язана зв'язком один до багатьох із сутностями «Робочий навчальний план», «Компонент освітньої програми» та «Група».

Таблиця 2.5

Поля сутності «Освітня програма»

Назва поля	Опис	Тип даних
id	Номер ідентифікатора (первинний ключ)	integer
title	Назва освітньої програми	varchar(64)
riven	Рівень	varchar(64)
stupin	Ступінь	varchar(64)
galuz	Галузь	varchar(64)
specialnist	Спеціальність	varchar(64)
specializaciya	Спеціалізація	varchar(64)
osvKvalifikaciya	Освітня кваліфікація	varchar(128)
profKvalifikaciya	Професійна кваліфікація	varchar(128)

Сутність «Компонент освітньої програми» передбачає збереження номера ідентифікатора, код навчальної дисципліни, назву, кількість кредитів, семестр в якому навчається, освітню програму до якої відноситься та формат контролю. Пов'язана зв'язком один до багатьох із сутністю «Елемент робочого навчального плану»

Таблиця 2.6

Поля сутності «Компонент освітньої програми»

Назва поля	Опис	Тип даних
id	Номер ідентифікатора (первинний ключ)	integer
kodND	Код навчальної дисципліни	varchar(64)
title	Назва навчальної дисципліни	varchar(64)
kredutu	Кількість кредитів	integer
formaKontrolu	Формат контролю (зовнішній ключ)	integer
semestr	Семестр в якому вивчається дисципліна	integer
educationalProgram	Освітня програма (зовнішній ключ)	integer

Сутність «Робочий навчальний план» передбачає збереження номера ідентифікатора, освітню програму та групу до яких відноситься. Пов'язана зв'язком один до багатьох із сутністю «Елемент робочого навчального плану».

Таблиця 2.7

Поля сутності «Робочий навчальний план»

Назва поля	Опис	Тип даних
id	Номер ідентифікатора (первинний ключ)	integer
educational_programs	Освітня програма (зовнішній ключ)	integer
group	Група (зовнішній ключ)	integer

Сутність «Елемент робочого навчального плану» передбачає збереження номера ідентифікатора, загальну кількість годин дисципліни, кількість годин, кредитів, годин для лекцій, годин для практичних, годин для лабораторних, годин для самостійного вивчення, формат контролю на перший семестр та аналогічно для другого семестру. Також робочий навчальний план, компонент освітньої програми та викладач до яких відноситься. Пов'язана зв'язком один до багатьох із сутністю «Розклад».

Таблиця 2.8

Поля сутності «Елемент робочого навчального плану»

Назва поля	Опис	Тип даних
1	2	3
id	Номер ідентифікатора (первинний ключ)	integer
title	Компонент освітньої програми (зовнішній ключ)	integer
vukladach	Викладач (зовнішній ключ)	integer
Godun	Загальна кількість годин	integer
kredutuECTS1sm	Кількість кредитів на перший семестр	integer
Godun1sm	Кількість годин на перший семестр	integer
lekcii1sm	Кількість годин для лекцій на перший семестр	integer
praktuchni1sm	Кількість годин для практичних на перший семестр	integer
laboratorni1sm	Кількість годин для лабораторних на перший семестр	integer
samostiynaRobota1sm	Кількість годин для самостійного вивчення на перший семестр	integer

1	2	3
formaKontrolu1sm	Формат контролю в першому семестрі	integer
kredutuECTS2sm	Кількість кредитів на другому семестр	integer
Godun2sm	Кількість годин на другий семестр	integer
lekcii2sm	Кількість годин для лекцій на другий семестр	integer
praktuchni2sm	Кількість годин для практичних на другий семестр	integer
laboratorni2sm	Кількість годин для лабораторних на другий семестр	integer
samostiynaRobota2sm	Кількість годин для самостійного вивчення на другий семестр	integer
formaKontrolu2sm	Формат контролю в другому семестрі	integer
robochiyNavchalnyPlan	Робочий навчальний план (зовнішній ключ)	integer

Сутність «Користувач» є розширенням стандартної сутності фреймворку Django хоч і не передбачає використання усіх полів батьківської сутності. Зберігається номер ідентифікатора, пароль користувача, позначка чи є супер-користувачем, логін, позначка чи є викладачем, позначка чи є адміністратором, викладач до якого відноситься.

Таблиця 2.9

Поля сутності «Користувач»

Назва поля	Опис	Тип даних
id	Номер ідентифікатора (первинний ключ)	integer
password	Пароль користувача	varchar(64)
is_superuser	Позначка чи є супер-користувачем	bool
username	Логін користувача	varchar(64)
is_vukladach	Позначка чи є викладачем	bool
is_admin	Позначка чи є адміністратором	bool
vukladach_id	Викладач (зовнішній ключ)	integer

Сутність «Графік навчання» передбачає збереження номера ідентифікатора, ідентифікатор групи, дати теоретичного навчання в першому семестрі, дати практики в першому семестрі, дати зимової сесії, дати зимових канікул, дати теоретичного навчання в другому семестрі, дати практики в другому семестрі, дати літньої сесії, дати літніх канікул, дати підготовки до кваліфікаційних робіт, дати захисту кваліфікаційних робіт.

Таблиця 2.10

Поля сутності «Графік навчання»

Назва поля	Опис	Тип даних
1	2	3
id	Номер ідентифікатора (первинний ключ)	integer
grupa	Номер ідентифікатора групи (зовнішній ключ)	integer

1	2	3
teoretuchne_navchannya_1sm	Дати теоретичного навчання у першому семестрі	varchar(64)
praktuka_1sm	Дати практики у першому семестрі	varchar(64)
sesiya_1sm	Дати зимової сесії	varchar(64)
kanikylu_1sm	Дати зимових канікули	varchar(64)
teoretuchne_navchannya_2sm	Дати теоретичного навчання у другому семестрі	varchar(64)
praktuka_2sm	Дати практики у другому семестрі	varchar(64)
sesiya_2sm	Дати літньої сесії	varchar(64)
kanikylu_2sm	Дати літніх канікули	varchar(64)
pidgotovkaKR	Дати підготовки до кваліфікаційних роботи	varchar(64)
ekzamenaciyna_komisiya	Дати захисту кваліфікаційних робіт	varchar(64)

Сутність «Розклад» передбачає збереження номера ідентифікатора, ідентифікатор факультету, ідентифікатор групи та кожної із п'яти пар на день.

Таблиця 2.11

Поля сутності «Розклад»

Назва поля	Опис	Тип даних
1	2	3
id	Номер ідентифікатора (первинний ключ)	integer
grupa	Номер ідентифікатора групи (зовнішній ключ)	integer

1	2	3
facultet	Номер ідентифікатора факультету (зовнішній ключ)	integer
ponedilok_1_para	Понеділок 1 пара (зовнішній ключ)	integer
ponedilok_2_para	Понеділок 2 пара (зовнішній ключ)	integer
ponedilok_3_para	Понеділок 3 пара (зовнішній ключ)	integer
ponedilok_4_para	Понеділок 4 пара (зовнішній ключ)	integer
vivtorok_1_para	Вівторок 1 пара (зовнішній ключ)	integer
vivtorok_2_para	Вівторок 2 пара (зовнішній ключ)	integer
vivtorok_3_para	Вівторок 3 пара (зовнішній ключ)	integer
vivtorok_4_para	Вівторок 4 пара (зовнішній ключ)	integer
vivtorok_5_para	Вівторок 5 пара (зовнішній ключ)	integer
sereda_1_para	Середа 1 пара (зовнішній ключ)	integer
sereda_2_para	Середа 2 пара (зовнішній ключ)	integer
sereda_3_para	Середа 3 пара (зовнішній ключ)	integer
sereda_4_para	Середа 4 пара (зовнішній ключ)	integer
sereda_5_para	Середа 5 пара (зовнішній ключ)	integer
chetver_1_para	Четвер 1 пара (зовнішній ключ)	integer
chetver_2_para	Четвер 2 пара (зовнішній ключ)	integer
chetver_3_para	Четвер 3 пара (зовнішній ключ)	integer
chetver_4_para	Четвер 4 пара (зовнішній ключ)	integer

1	2	3
chetver_5_para	Четвер 5 пара (зовнішній ключ)	integer
pyatnytsya_1_para	П'ятниця 1 пара (зовнішній ключ)	integer
pyatnytsya_2_para	П'ятниця 2 пара (зовнішній ключ)	integer
pyatnytsya_3_para	П'ятниця 3 пара (зовнішній ключ)	integer
pyatnytsya_4_para	П'ятниця 4 пара (зовнішній ключ)	integer
pyatnytsya_5_para	П'ятниця 5 пара (зовнішній ключ)	integer
subota_1_para	Субота 1 пара (зовнішній ключ)	integer
subota_2_para	Субота 2 пара (зовнішній ключ)	integer
subota_3_para	Субота 3 пара (зовнішній ключ)	integer
subota_4_para	Субота 4 пара (зовнішній ключ)	integer
subota_5_para	Субота 5 пара (зовнішній ключ)	integer

2.4. Обґрунтування вибору стеку технологій для розробки ІС

Обґрунтування вибору стеку технологій для розробки інформаційної системи деканату є ключовим етапом розробки програмного продукту, адже саме від обраних технологій залежить функціональність, зручність використання, масштабованість та надійність майбутньої системи. Було обрано один з популярних, на даний час, стек технологій:

1) Фронтенд (технології користувацького інтерфейсу):

- HTML
- CSS
- Bootstrap
- JavaScript (JS)
- AJAX
- HTMX

2) Бекенд (серверна частина і логіка додатка):

- Django

3) База даних (сховище даних):

- SQLite (на етапі розробки)
- PostgreSQL (у перспективі)

4) Інструменти управління розробкою:

- Git
- GitHub

Розглянемо докладно кожен елемент стеку технологій та обґрунтуємо його застосування в контексті завдань, поставлених перед інформаційною системою.

HTML (HyperText Markup Language) – «це мова гіпертекстової розмітки, яка використовується для опису структури вмісту, представленого на вебсторінці» (Shaikh, 2022). Є основою будь-якого вебзастосунку, оскільки забезпечує структурування вмісту вебсторінок. Він дозволяє розміщувати текст, зображення, форми та інші елементи, необхідні для інтерактивного інтерфейсу системи деканату. У рамках даного проєкту HTML використовується для

створення базової розмітки сторінок, що забезпечує користувачам інтуїтивно зрозумілий доступ до функціоналу системи.

CSS (Cascading Style Sheets) є потужним інструментом для визначення стилю та макета вебсторінок, що забезпечує їх естетичну привабливість, зручність використання та доступність. Основною функцією CSS є стандартизація візуальних характеристик вебсайту, таких як шрифти, кольори, розташування елементів та інші аспекти дизайну, що спрощує підтримку та оновлення інтерфейсу. CSS також надає можливість створювати адаптивний дизайн, який дозволяє вебсторінкам автоматично підлаштовуватися під різні розміри екранів, що є критично важливим у сучасному середовищі багатоплатформного доступу (Pranav, 2024). Це сприяє забезпеченню позитивного досвіду користувачів незалежно від пристрою, який вони використовують. Завдяки цьому CSS є невід’ємною частиною розробки сучасних вебзастосунків, що відповідають сучасним вимогам до дизайну та функціональності.

Bootstrap є одним із найпоширеніших CSS-фреймворків, що значно оптимізує процес розробки вебінтерфейсів. Завдяки наявності широкого набору готових шаблонів, стилів і компонентів, Bootstrap дозволяє розробникам ефективно створювати функціональні та естетично привабливі інтерфейси, суттєво скорочуючи час на розробку з нуля.

Орієнтованість Bootstrap на мобільні пристрої забезпечує адаптивність дизайну, що дозволяє вебзастосункам безперешкодно адаптуватися до різних розмірів екрана та типів пристроїв. Це робить Bootstrap особливо цінним у сучасних умовах, коли користувачі взаємодіють із вебсайтами через широкий спектр пристроїв. Використання цього фреймворка сприяє створенню професійного та зручного дизайну з мінімальними затратами ресурсів і часу (Community, 2024).

JavaScript — є ключовою мовою програмування для реалізації інтерактивності на стороні клієнта. У рамках проєкту його використання дозволяє створювати динамічні елементи інтерфейсу, такі як валідація форм у реальному часі, динамічне оновлення даних та інтерактивна взаємодія користувача з

системою (Suresh, 2023). JavaScript забезпечує ефективність і зручність використання, що є важливими вимогами до інформаційної системи деканату.

AJAX (Asynchronous JavaScript and XML) – використовується для асинхронного обміну даними між клієнтом і сервером. Завдяки AJAX система може оновлювати дані на сторінках без перезавантаження, що дозволяє здійснювати запити та отримувати дані із сервера після завантаження сторінки, а також надсилати інформацію на сервер у фоновому режимі. Це значно підвищує швидкість роботи та зручність для користувача, забезпечуючи безперервну роботу інтерфейсу без порушення інших процесів («GeeksforGeeks», 2022). Для системи деканату це означає миттєве оновлення інформації про студентів, розклади або інші важливі дані.

HTMX є сучасною технологією, що забезпечує простий і ефективний підхід до створення інтерактивних вебзастосунків без зайвої складності. На відміну від традиційних фронтенд-фреймворків, HTMX дозволяє використовувати декларативний HTML-синтаксис для виконання динамічних завдань, що зменшує залежність від великих JavaScript-бібліотек і спрощує підтримку коду. Технологія підтримує інтеграцію з існуючою серверною логікою, дозволяючи зберігати її недоторканою, а також забезпечує швидке оновлення сторінок без повного перезавантаження завдяки вбудованій підтримці AJAX (Tusharupadhyay, 2024). У рамках проєкту HTMX дозволяє значно спростити реалізацію інтерактивних функцій, зберігаючи при цьому високий рівень продуктивності.

Django обрано як основний фреймворк для розробки серверної частини системи завдяки його універсальності, масштабованості та широкому набору вбудованих інструментів. Django забезпечує швидку розробку завдяки модульній архітектурі, можливостям адміністрування та автоматизованій роботі з базами даних. Цей фреймворк також надає потужний механізм автентифікації та авторизації, що критично важливо для захисту даних у системі деканату.

Django забезпечує оптимальну архітектуру для розробки завдяки слабкій зв'язності компонентів, що працюють незалежно з мінімальними залежностями, сприяючи модульності та легкості підтримки. Його динамічні можливості, як-от

інспекція Python, дозволяють уникати надлишкового коду, зберігаючи лаконічність додатків і прискорюючи процес розробки. Принцип DRY (Don't Repeat Yourself) забезпечує уніфікованість даних, мінімізуючи дублювання та підтримуючи структурованість коду, тоді як акцент на явності замість неявності, відповідно до PEP 20, сприяє простоті розуміння навіть для новачків. Послідовність у підходах до написання коду та користувацького досвіду робить Django надійним вибором для різноманітних проєктів і галузей, включно з масштабними платформами, такими як Instagram, Disqus і Spotify (Mantosh, 2023).

SQLite обрано як базу даних на етапі розробки завдяки її легкості, швидкості та простоті використання. Вона ідеально підходить для прототипування і тестування системи. «SQLite – це база даних на основі sql із унікальною особливістю: уся база даних міститься в одному файлі. Здебільшого з цієї причини багато людей історично розглядали його як просту базу даних для простих випадків використання» (Dodds, 2023). У подальшому, з огляду на потреби у масштабуванні та роботі з великими обсягами даних, планується перехід на PostgreSQL. Привабливість PostgreSQL вирізняється не лише своїми технічними характеристиками, але й високою зручністю та універсальністю, що робить його оптимальним вибором для широкого спектра вебпроєктів, від невеликих до масштабних корпоративних систем. Його здатність ефективно обробляти складні запити та великі обсяги даних дозволяє розробникам зосереджуватися на інноваційних аспектах, знижуючи необхідність у глибокому управлінні базами даних. Підтримка принципів відкритого коду забезпечує безперервний розвиток PostgreSQL через активний внесок глобальної спільноти, що сприяє його адаптивності та забезпечує надійні функції безпеки, гарантуючи актуальність і потужність у сучасних вебрішеннях (Smith, 2024).

Git і GitHub використовуються для управління версіями коду та командної роботи над проєктом. Git дозволяє розробникам ефективно контролювати зміни в коді, а GitHub забезпечує платформу для спільної роботи, зберігання проєкту та розгортання додатка. Ці інструменти є стандартом у сучасній розробці програмного забезпечення і значно підвищують якість роботи команди.

Вибраний стек технологій дозволяє забезпечити високу продуктивність, надійність, зручність використання та адаптивність інформаційної системи деканату. Кожен компонент стеку виконує специфічну роль, сприяючи досягненню поставлених цілей і створенню ефективного інструменту для управління освітнім процесом.

Кожна технологія у стеку була обрана з урахуванням вимог проєкту та перспектив масштабування. Таке поєднання забезпечує стабільну роботу системи, її інтерактивність і відповідність сучасним стандартам безпеки та функціональності.

Завдяки збалансованому вибору технологій система відповідає потребам користувачів і надає інструменти для ефективного управління освітнім процесом.

РОЗДІЛ III

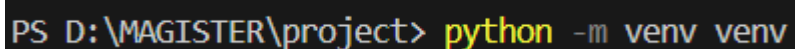
РОЗРОБКА ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ОСВІТНІМ ПРОЦЕСОМ НА РІВНІ ДЕКАНАТУ

3.1. Особливості програмної реалізації архітектури та функціональних модулів системи

Згідно рекомендацій зазначених в роботах (Подоба, 2016), (Areeba, 2024), (Biswas, 2023), (Khosrowabadi, 2022), (Getachew, 2024), («Django Software Foundation», б. д.), («GeeksforGeeks», 2023), (Melé, 2022), (Vincent, 2023), («W3Schools», б. д.), (Zapke-Gründemann, 2015) проведено розробку інформаційної системи.

Для початку роботи з вебфреймворком Django необхідно створити віртуальне середовище. Віртуальне середовище являє собою ізолюваний простір у системі, призначений для встановлення бібліотек Python окремо від загальної системної конфігурації. Це забезпечує запобігання можливим конфліктам між різними версіями бібліотек.

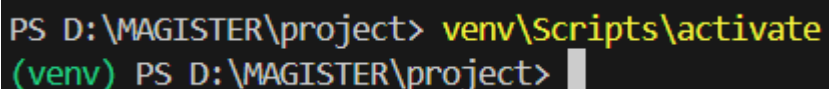
Процес створення віртуального середовища здійснюється в заздалегідь підготовленій окремій директорії за допомогою терміналу, шляхом виконання відповідної команди (рис. 3.1).



```
PS D:\MAGISTER\project> python -m venv venv
```

Рис. 3.1. Команда створення віртуального середовища

Після створення віртуального середовища його потрібно активувати командою «activate» (рис. 3.2).



```
PS D:\MAGISTER\project> venv\Scripts\activate  
(venv) PS D:\MAGISTER\project> █
```

Рис. 3.2. Команда активації віртуального середовища

Активувавши віртуальне середовище потрібно встановити Django, вказавши актуальну версію, за допомогою системи керування пакетами PIP (рис. 3.3).

```
(venv) PS D:\MAGISTER\project> pip install django==4.2.7
```

Рис. 3.3. Команда встановлення Django

Після встановлення Django потрібно створити проєкт вказавши його ім'я (рис. 3.4).

```
(venv) PS D:\MAGISTER\project> django-admin startproject decanat
```

Рис. 3.4. Команда створення проєкту

В результаті буде створено каталог «decanat» з файлами (рис. 3.5):

- «__init__.py» – зазвичай порожній, сигналізує інтерпритатору, що каталог є модулем;
- «asgi.py» – відповідає за конфігурацію ASGI (Asynchronous Server Gateway Interface) для проєкту;
- «settings.py» – основний файл конфігурації проєкту;
- «urls.py» – файл маршрутизації URL;
- «wsgi.py» – відповідає за конфігурацію WSGI (Web Server Gateway Interface) для проєкту;
- «manage.py» – головний файл, з якого відбувається запуск проєкту.

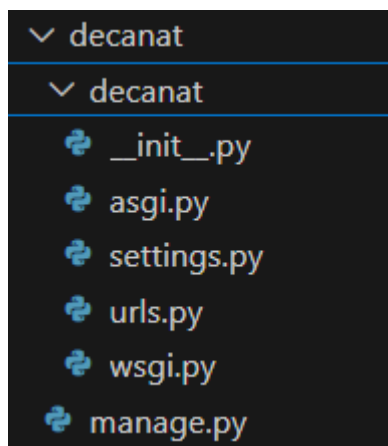


Рис. 3.5. Структура каталогу проєкта

Django проєкт включає в себе набір модулів, які взаємодіють між собою. Кожен модуль відповідає за обробку даних однієї із сутностей бази даних. Для створення нового модуля використовується команда «startapp» (рис. 3.6).

```
PS D:\MAGISTER\project\decanat> python manage.py startapp main
```

Рис. 3.6. Команда створення модуля

Після створення модуля його необхідно зареєструвати (рис. 3.7) у файлі «settings.py», щоб забезпечити інформування інтерпретатора про його існування та інтеграцію в проєкт.

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'main',
    'control',
    'educational_programs',
    'group',
    'robochiy_navchalnuy_plan',
    'vukladach',
]
```

Рис. 3.7. Перелік зареєстрованих модулів

Кожен з модулів має певну структуру файлів (рис. 3.8):

- «__pycache__» – каталог, що містить скомпільовані у байт-код файли Python для оптимізації виконання;
- «migrations» – каталог, у якому зберігаються файли міграцій;
- «templates» – каталог для зберігання HTML-шаблонів, які використовуються для створення інтерфейсу користувача;
- «__init__.py» – файл, який позначає каталог як модуль Python;
- «admin.py» – файл для налаштування адміністративного інтерфейсу Django;
- «apps.py» – файл конфігурації модуля;
- «forms.py» – файл для визначення форм Django;
- «models.py» – файл для визначення моделей даних;
- «tests.py» – файл для написання тестів, що перевіряють функціональність застосунку;
- «urls.py» – файл маршрутизації, у якому визначаються шляхи (URL) для цього модуля та їх відповідні обробники (views);
- «views.py» – файл для створення представлень (views), які обробляють запити користувачів і повертають відповіді, наприклад, HTML-сторінки чи JSON-дані.

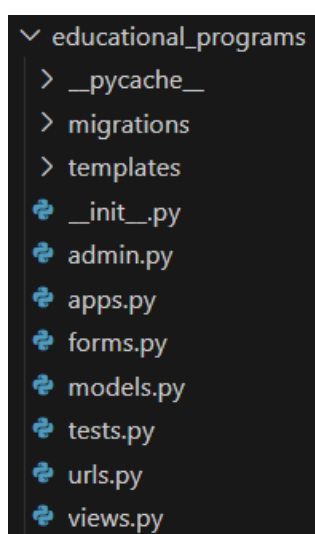


Рис. 3.8. Структура файлів модуля

Оскільки Django використовує архітектурний підхід MVT (Model-View-Template), доцільно проаналізувати структуру одного з модулів проєкту відповідно до принципів цієї архітектури.

До Model належить файл `models.py`, у якому визначаються сутності бази даних за допомогою класів (рис. 3.9). У цьому файлі для кожної сутності задаються поля, що відображають відповідну структуру таблиць у базі даних. Крім того, у моделі можна реалізовувати логіку, яка стосується роботи з даними, наприклад, кастомні методи для обробки полів.

Для того щоб створити таблицю в базі даних на основі класу з файлу `models.py` потрібно створити міграції за допомогою команди «makemigrations» і виконати їх за допомогою команди «migrate».

```
class EducationalPrograms(models.Model):
    title=models.CharField('Назва',max_length=150)
    riven=models.CharField('Рівень',max_length=75)
    stupin=models.CharField('Ступінь',max_length=75)
    galuz=models.CharField('Галузь знань',max_length=75)
    specialist=models.CharField('Спеціальність',max_length=75)
    specializaciya=models.CharField('Спеціалізація',max_length=75)
    osvKvalifikaciya=models.CharField('Освітня кваліфікація',max_length=150)
    profKvalifikaciya=models.CharField('Професійна кваліфікація',max_length=150)

    def __str__(self):
        return self.title
```

Рис. 3.9. Клас сутності «Освітня програма»

Додатково до Model можна віднести файл `forms.py`, де створюються класи (рис. 3.10) форм для зручної взаємодії користувача з даними. Форми дозволяють не лише генерувати HTML-елементи для введення інформації, але й перевіряти коректність введених даних згідно визначених правил валідації.

```

class EducationalProgramsForm(ModelForm):
    class Meta:
        model=EducationalPrograms
        fields='__all__'

class KomponentEducationalProgramsForm(ModelForm):
    class Meta:
        model=KomponentEducationalPrograms
        fields='__all__'
        widgets = {
            'kreditu': forms.NumberInput(
                attrs={
                    'class': 'form-control'
                }
            ),
            'semestr': forms.NumberInput(
                attrs={
                    'class': 'form-control'
                }
            )
        }

```

Рис. 3.10. Класи форм сутності «Освітня програма»

У компоненті View основним є файл views.py (Додаток Б), який містить логіку обробки запитів і формування відповідей. У цьому файлі визначаються функції або класи, які відповідають за обробку запитів користувачів, отримання необхідних даних із бази через моделі та передачу цих даних до шаблонів для подальшого відображення. Файл urls.py (рис. 3.11) також належить до View, оскільки в ньому описуються маршрути, що забезпечують зв'язок між URL-адресами та відповідними представленнями. Він визначає, які функції або класи викликаються при доступі до конкретних шляхів у вебзастосунку.

```

urlpatterns = [
    path('', EducationalProgramsList.as_view(), name='list_educational_programs'),
    path('create/', EducationalProgramsCreate.as_view(), name='create_educational_programs'),
    path('update/<int:pk>/', EducationalProgramsUpdate.as_view(), name='update_educational_programs'),
    path('delete-komponent-educational-programs/<int:pk>/',
         delete_komponent_educational_programs, name='delete_komponent_educational_programs'),
]

```

Рис. 3.11. URL-адреси модуля

Шаблони (Templates) у Django відповідають за відображення даних, що отримуються через представлення. Прикладом є файл index.html (Додаток Б), який

є HTML-шаблоном, що використовується для рендерингу сторінок (рис. 3.11). Шаблони можуть включати динамічний вміст за допомогою вбудованого шаблонного двигуна Django, який підтримує спеціальні теги та фільтри для маніпуляції даними. У цьому файлі визначаються елементи інтерфейсу користувача, такі як таблиці, форми або списки, що дозволяють представити дані у зручній для сприйняття формі.

Спеціальність * Тест2

Спеціалізація * Тест2

Освітня кваліфікація * Тест2

Професійна кваліфікація * Тест2

Додати компонент

Код н/д *	Назва *	Кредити *	Форма контролю *	Семестр *	Delete?	Custom Delete btn
		1	-----	1	☐	

Add More

Submit

Рис. 3.12. Відображення шаблону модуля

Таким чином, архітектура MVT забезпечує чіткий розподіл логіки роботи з даними, їх обробки та відображення, що сприяє структурованій і гнучкій розробці вебзастосунків.

3.2. Організація тестування та налагодження ІС

Тестування вебзастосунків є ключовим етапом у процесі розробки програмного забезпечення, що забезпечує їхню стабільність, функціональність і відповідність очікуванням користувачів. Основна мета тестування полягає в тому,

щоб виявити потенційні дефекти, перевірити відповідність системи вимогам і гарантувати безперебійність її роботи в різних умовах. Процес тестування охоплює різні аспекти, такі як функціональність, продуктивність, безпека, сумісність і зручність використання. Вебзастосунки працюють у динамічному середовищі, що вимагає врахування різних конфігурацій браузерів, операційних систем і пристроїв. Таким чином, тестування стає багатогранним процесом, який включає як ручні, так і автоматизовані методи перевірки. Ефективно організоване тестування дозволяє не лише підвищити якість продукту, але й зменшити витрати на його підтримку в майбутньому.

Під час тестування інформаційної системи була проведена перевірка на коректність її роботи згідно рекомендацій зазначених в роботах (Admin, 2022), (Novak, 2016), («QATestLab», 2023), (Багрій & Петровський, 2022), (Цюцюра, Криворучко, Жирова, & Котенко, 2019), (Mohan, 2022), (Бубенко, Герасимов, Карпенко, & Морозов 2023). В результаті були здійснені такі перевірки:

1. Виконано перевірку валідації полів на відсутність заповнених даних, у процесі якої було перевірено коректність функціонування механізму відображення повідомлень про помилку у випадку, якщо відповідне поле залишено незаповненим (рис. 3.12). За результатами тестування встановлено, що всі поля коректно виконують функцію перевірки, забезпечуючи відображення відповідного повідомлення про помилку у разі порушення вимоги заповнення.

Додати групу

Назва * ЦТ-2

Повна назва * Заповніть це поле

Курс * 2

Educational programs * Професійна освіта(Цифрові технології)

Faculty * Факультет математики та інформатики

Submit

Рис. 3.13. Валідація поля відсутність даних

2. Проведено перевірку відповідності даних, внесених у форму, до даних, які були записані в базу даних (рис. 3.13, рис.3.14). За результатами перевірки встановлено, що всі дані коректно передаються та зберігаються, що свідчить про правильність роботи механізму передачі інформації між інтерфейсом користувача та базою даних.

Рис. 3.14. Форма із заповненими даними

id	imya	prizvusche	poBatkovi	posada	naukovaStupin	vcheneZvannya	facultet_id	kafedra_id
Фі...	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
2	Василь	Боярський	Станіславович	Старший викладач	Професор	доктор педагогічних наук	1	1

Рис. 3.15. Відображення даних з форми в базі даних

3. Проведено перевірку на кросбраузерність інформаційної системи в браузерах Google Chrome, Mozilla Firefox та Opera. В результаті тестування було встановлено, що всі функціональні компоненти системи коректно відображаються та працюють у зазначених браузерах, що свідчить про їх сумісність із сучасними стандартами вебтехнологій.

4. Проведено перевірку на адаптивність інформаційної системи для мобільних пристроїв (рис. 3.15). У результаті тестування встановлено, що

більшість елементів сторінок відображаються некоректно, що свідчить про недостатню адаптацію інтерфейсу під різні розширення екранів. Для забезпечення належного користувацького досвіду необхідно внести додаткові зміни до стилів та структури сторінок, орієнтуючись на вимоги до адаптивного дизайну для різних типів мобільних пристроїв.

Размеры: Samsung Galaxy S8+ 360 x 740 75% Без ограничения

Додати викладача

Ім'я * Василь

Прізвище * Боярський

По батькові * Станіславович

Посада * Старший викладач

Науковий ступінь * Професор

Вчене звання * доктор педагогічних наук

Факультет *

Рис. 3.16. Відображення сторінки на мобільному пристрої

3.3. Рекомендації щодо використання та впровадження ІС

Важливим аспектом розробки інформаційних систем є розроблення розгорнутих та доступних до розуміння рекомендацій щодо використання інформаційної системи для трьох категорій користувачів: студента, викладача та адміністратора. Інформаційна система забезпечує ефективне управління навчальним процесом та доступ до відповідної інформації. Перед початком роботи кожному користувачеві необхідно перейти за вебадресою

«<https://yalovenko.pythonanywhere.com/>», де розміщена система, використовуючи стандартний веббраузер.

Студенту надається можливість переглядати розклад занять, графік навчання та робочий навчальний план для обраної групи. Після переходу на вебсторінку системи студент бачить інтерфейс із відповідними меню та кнопками (рис.3.16). Для перегляду розкладу необхідно виконати наступні дії:

1. У випадаючому меню «Факультет» обрати потрібний факультет.
2. У випадаючому меню «Група» вибрати свою групу.
3. Натиснути відповідну кнопку

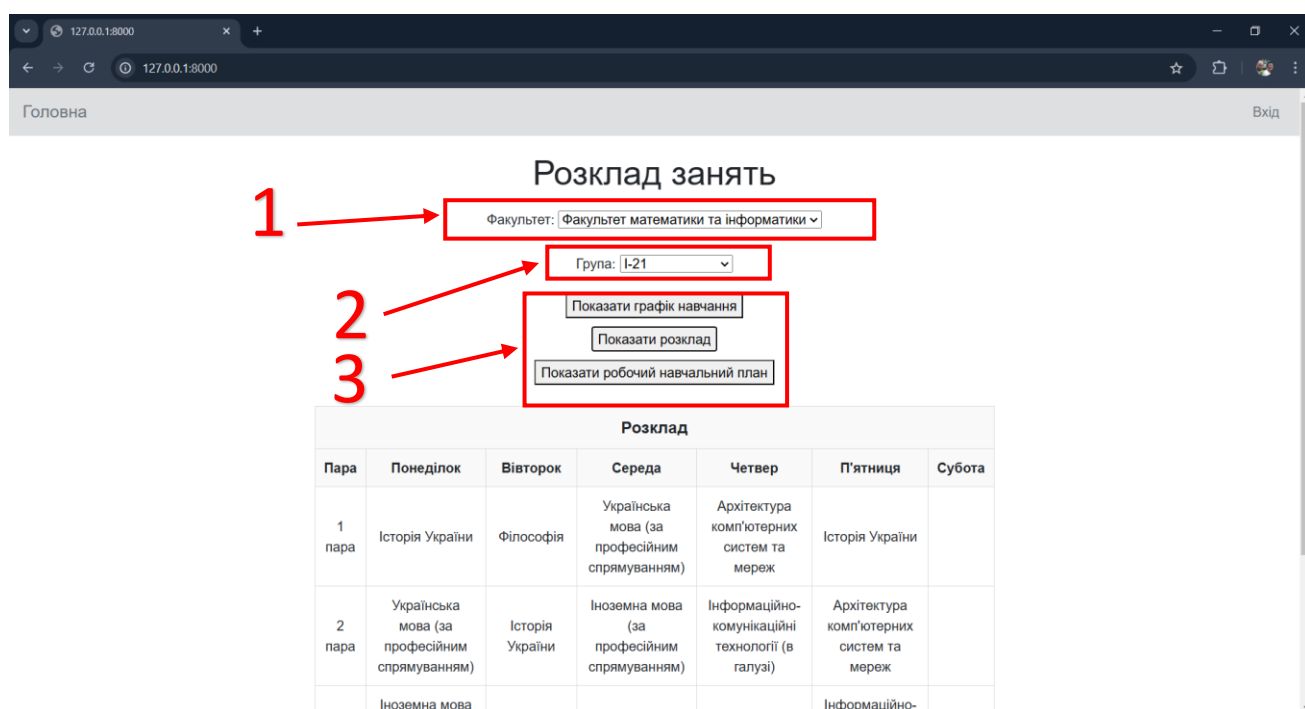


Рис. 3.17. Рекомендації дій для відображення розкладу, графіку навчання та робочого навчального плану

Після виконання зазначених дій на екрані з'явиться розклад занять на поточний тиждень, графік навчання та робочий навчальний план. Ця функціональність дозволяє студентам оперативно отримувати актуальну інформацію про навчальний процес.

Викладач має розширений доступ до функцій інформаційної системи, що включає можливості студента та доступ до персонального кабінету. Для того, щоб увійти до особистого кабінету, викладачу необхідно (рис. 3.17):

1. Натиснути кнопку «Вхід» у верхньому правому куті вебсторінки.
2. На сторінці авторизації ввести свій логін і пароль.
3. Натиснути кнопку «Увійти».

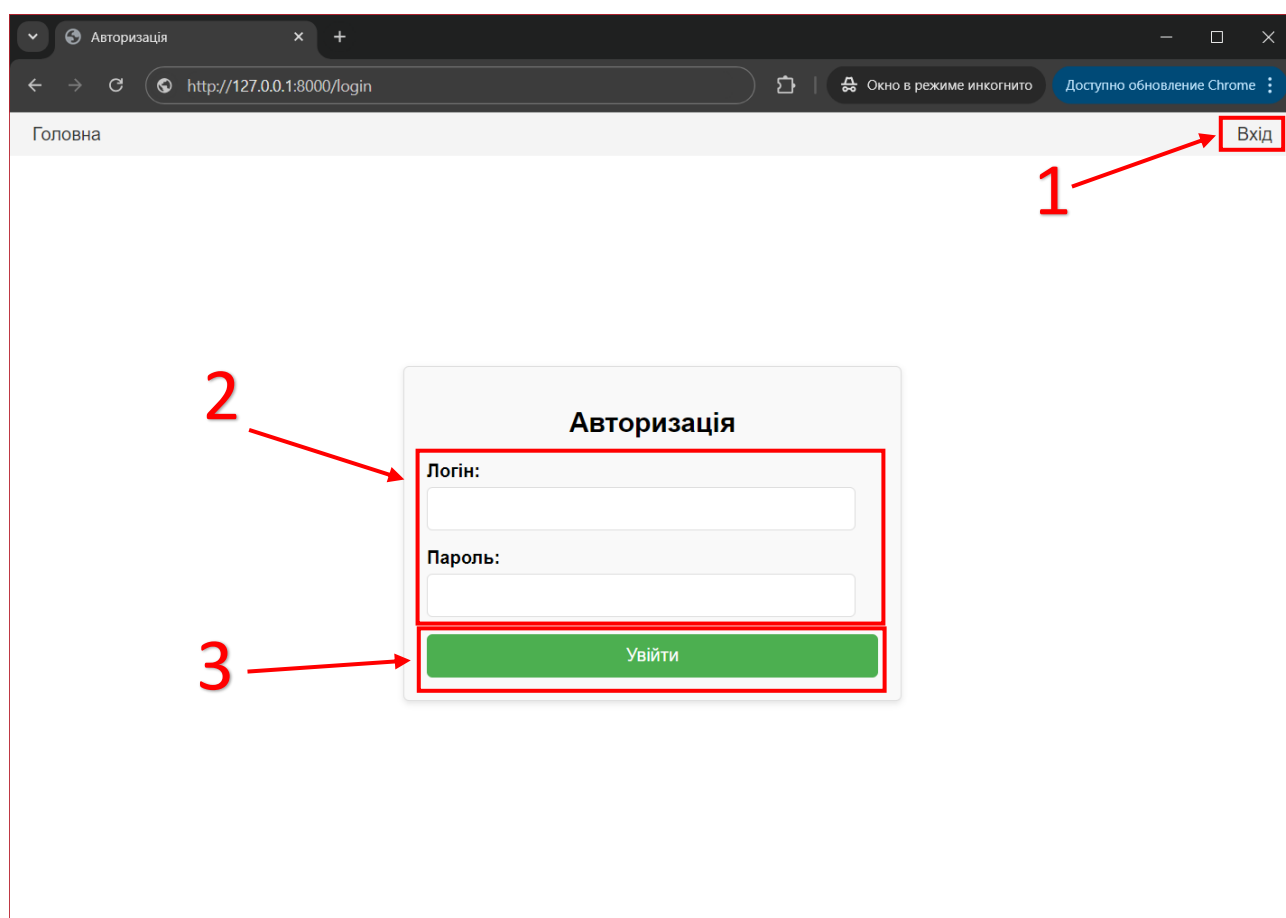


Рис. 3.18. Рекомендації дій для входу в інформаційну систему

Після авторизації викладач отримує доступ до особистого кабінету (рис. 3.18), де розміщена наступна інформація:

- персональні дані викладача;
- перелік дисциплін, які викладач закріплений викладати, із зазначенням кількості годин;
- загальна кількість годин, передбачена для викладача;

- особистий розклад, що включає інформацію про заняття з певних дисциплін у відповідних групах.

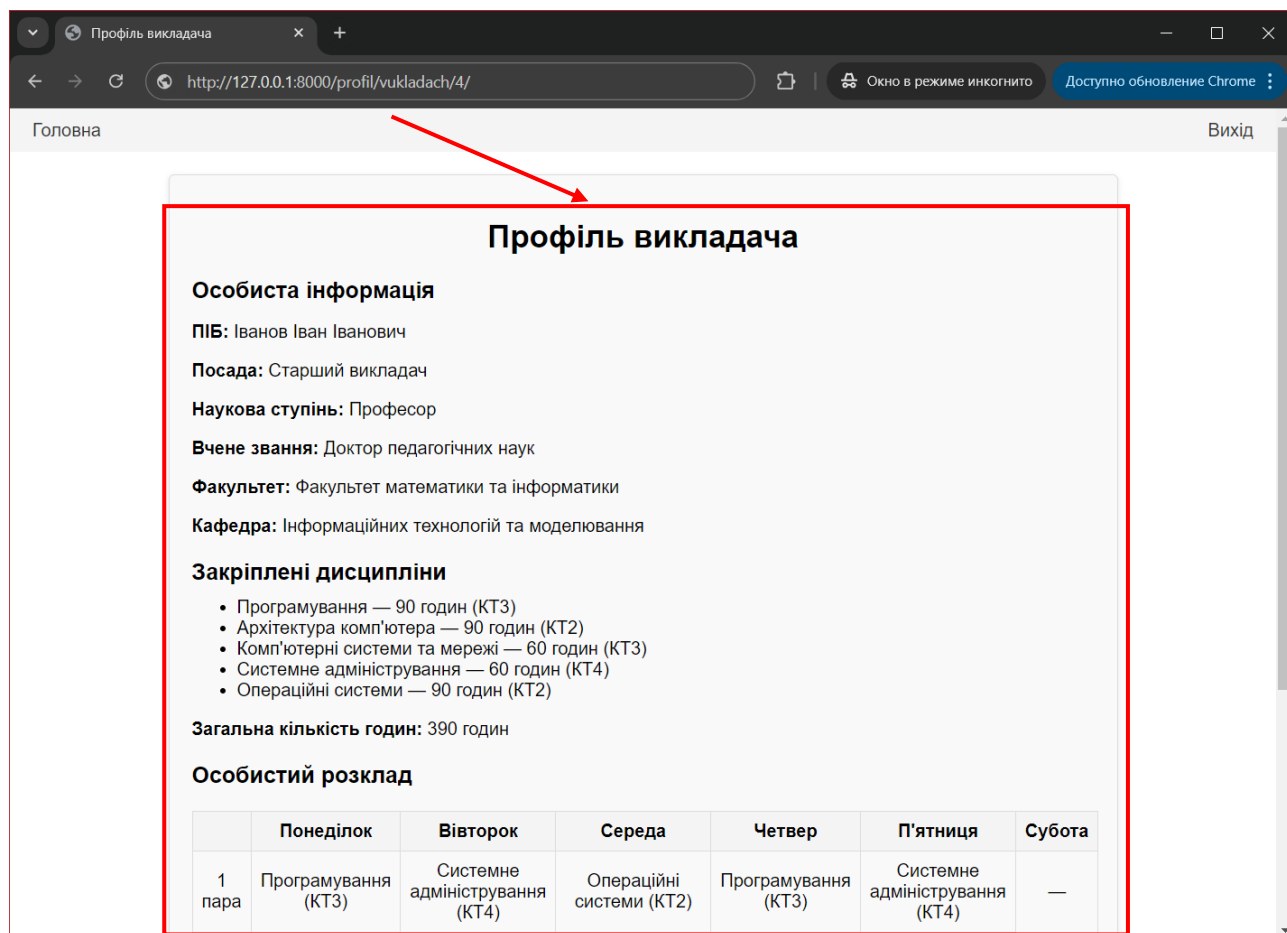


Рис. 3.19. Відображення інформації профілю викладача

Даний функціонал спрямований на оптимізацію роботи викладача, забезпечуючи швидкий доступ до необхідної інформації.

Адміністратор володіє найширшим спектром функціональних можливостей, включаючи створення, редагування та видалення різних категорій даних. Для отримання доступу до цих функцій, адміністратор повинен пройти авторизацію, аналогічно до викладача:

1. Натиснути кнопку «Вхід» у верхньому правому куті вебсторінки.
2. На сторінці авторизації ввести логін і пароль.
3. Натиснути кнопку «Увійти».

Після успішної авторизації адміністратор отримує доступ до таких категорій (рис. 3.19):

- факультети;
- кафедри;
- освітньо-професійні програми;
- групи;
- викладачі;
- робочі навчальні плани;
- розклади;
- графіки навчання;
- користувачі.

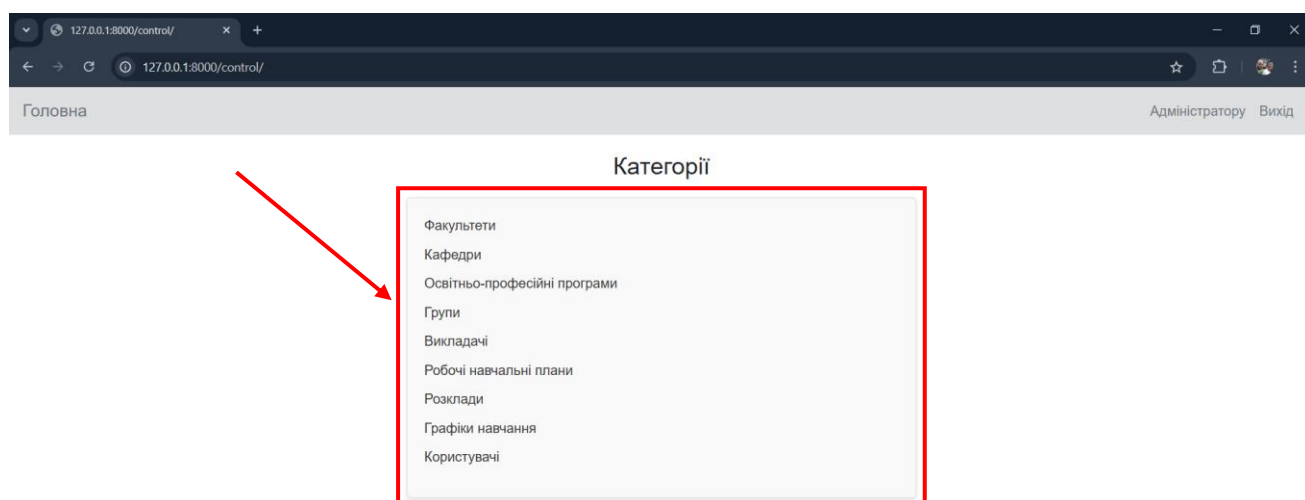


Рис. 3.20. Панель вибору категорій адміністратором

Для роботи з будь-якою категорією адміністратор може вибрати її зі списку. На сторінці відображатиметься повний перелік об'єктів відповідної категорії (рис. 3.20). Функціональні можливості включають:

- видалення об'єкта шляхом натискання кнопки «Видалити» навпроти нього;

- створення нового об'єкта за допомогою кнопки «Створити», що відкриває сторінку введення даних;
- редагування існуючого об'єкта через кнопку «Редагувати», яка перенаправляє на сторінку редагування даних.

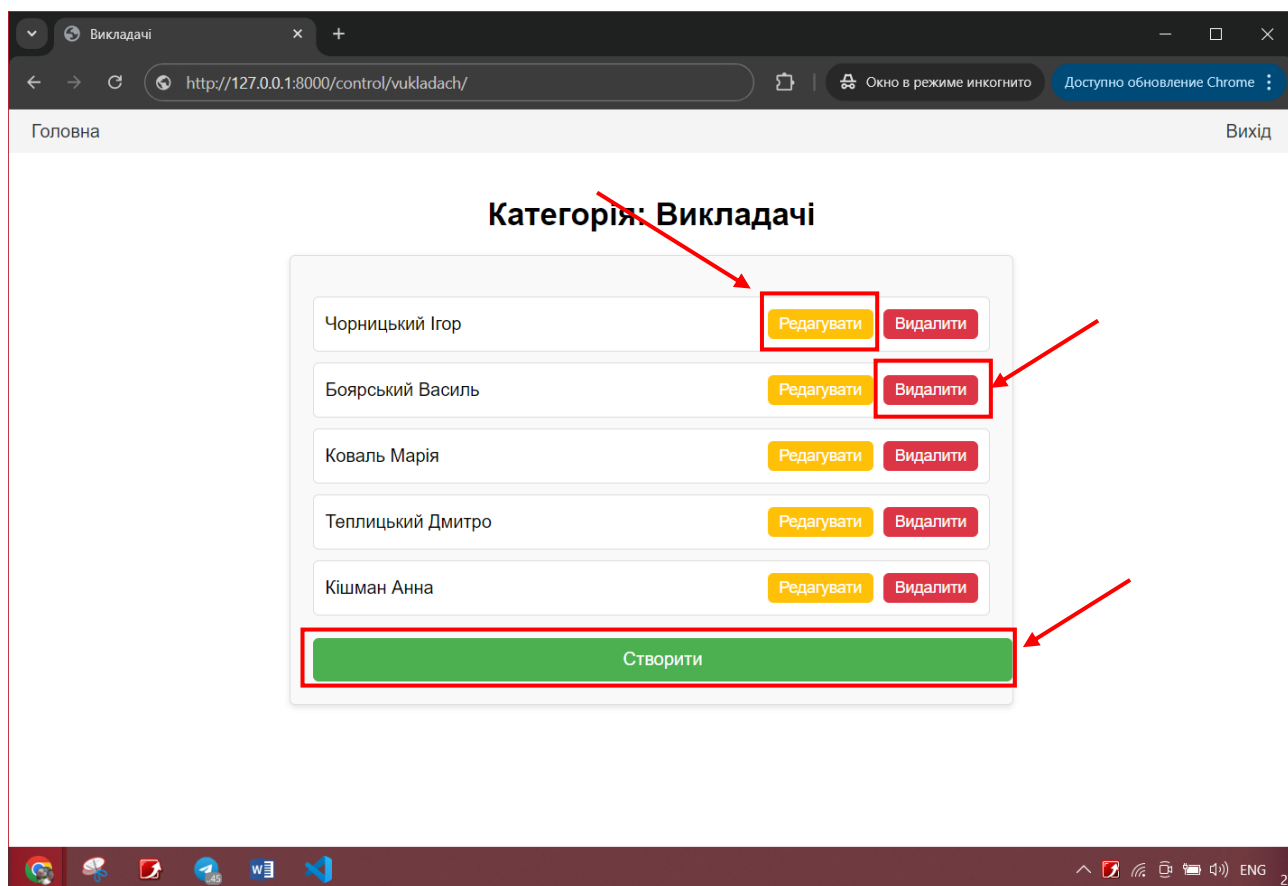


Рис. 3.21. Відображення можливих функцій над об'єктами із списку категорії «Викладачі»

Сторінки створення та редагування мають аналогічну структуру: користувач вводить дані у відповідні текстові поля або вибирає варіанти зі списків (рис. 3.21). Для збереження змін необхідно натиснути кнопку «Зберегти».

127.0.0.1:8000/control/vuklada... x +

127.0.0.1:8000/control/vukladach/update/4/ ☆ | Окно в режиме инкогнито | Доступно обновление Chrome

Прізвище * Теплицький

По батькові * Богданович

Посада * Старший викладач

Наукова ступінь * Професор

Вчене звання * доктор педагогічних наук

Facultet * Факультет математики та інформатики

Kafedra * Кафедра інформаційних технологій та моделювання

Зберегти

Рис. 3.22. Панель редагування об'єкта «Викладач»

Після завершення роботи адміністратор може розлогуватися (вийти із системи), натиснувши кнопку «Вийти» у верхньому правому куті вебсторінки.

Кожен користувач має дотримуватися правил коректного використання системи, забезпечувати конфіденційність даних для авторизації та своєчасно оновлювати інформацію (за наявності таких прав). Інформаційна система створена для забезпечення зручності та ефективності освітнього процесу, тому її використання потребує уважності та відповідальності.

ВИСНОВКИ

У результаті проведеного дослідження було проаналізовано існуючі інформаційні системи для управління освітнім процесом. Визначено, що ці системи значно покращують ефективність організації освітнього процесу завдяки автоматизації рутинних завдань, зниженню часу на обробку даних і підвищенню якості управлінських рішень. Переваги таких систем включають можливість централізованого управління навчальними планами, автоматизацію оцінювання та контроль успішності студентів, а також поліпшену комунікацію між усіма учасниками освітнього процесу. Проте існують і недоліки, зокрема обмеження у функціональності деяких систем, недостатня інтеграція з іншими програмними продуктами, а також питання безпеки даних. Аналіз функціональних можливостей ІСУОП показав, що ці системи потребують подальшого вдосконалення для досягнення високої прозорості, доступності інформації та зменшення ресурсних витрат на підтримку інфраструктури.

У процесі розробки інформаційної системи деканату було досліджено різноманітні технології реалізації та особливості проєктування інформаційних систем управління освітнім процесом. Зокрема, виявлено, що вибір технологічних рішень має визначальний вплив на ефективність, масштабованість та безпеку системи. Було розглянуто два основні вебфреймворки – Django та Spring Boot, які є потужними інструментами для створення сучасних інформаційних систем. Вибір між ними залежить від специфічних вимог проєкту, зокрема, від необхідної швидкості розробки, масштабованості та гнучкості архітектури. Важливим етапом проєктування стало також створення інтерфейсу користувача, орієнтованого на зручність і мінімалізм, що сприяє підвищенню ефективності взаємодії з системою.

Визначено функціональні вимоги для інформаційної системи деканату відповідно до потреб користувачів. Система повинна забезпечувати зручний доступ до актуальної та детальної інформації, підтримувати різні пристрої та гарантувати безпеку даних. Студенти мають мати доступ до персоналізованого

розкладу, викладачі – до обліку відвідуваності та успішності, методисти – до редагування навчальних планів, а менеджери деканату – до оновлення розкладів у реальному часі.

Розробка архітектури та функціональних модулів системи здійснена на основі принципу архітектури MVT (Model-View-Template), що забезпечує ефективне розділення логіки представлення, обробки даних та взаємодії з користувачем. Функціональні модулі системи забезпечують розподілене та ефективне виконання ключових операцій, необхідних для автоматизації управління освітнім процесом.

Розробка прототипу інформаційної системи деканату для ефективного управління освітнім процесом успішно реалізована з використанням вебфреймворка Django. Для забезпечення коректної роботи системи була здійснена серія тестувань, що включали перевірку валідації даних, їх коректної передачі в базу даних, сумісності з різними браузерами та адаптивності для мобільних пристроїв. В результаті тестування було встановлено, що система демонструє високий рівень функціональності, хоча й потребує вдосконалення адаптивності для забезпечення належного користувацького досвіду на мобільних пристроях.

Аналіз результатів тестування інформаційної системи для ефективного управління освітнім процесом на рівні деканату свідчить про високий потенціал цієї системи, оскільки вона охоплює ключові аспекти автоматизації адміністративних процесів, що сприяє підвищенню ефективності управлінської діяльності в закладі вищої освіти. Використання такої системи дозволяє значно скоротити час на обробку адміністративних запитів, зменшити ймовірність помилок при введенні даних і забезпечити зручний доступ до інформації для студентів і викладачів. Проте, на практиці система потребує подальшого вдосконалення та додаткових інвестицій, що дозволить поліпшити її функціональні можливості та інтеграцію з іншими інформаційними системами навчального закладу.

Розроблена інформаційна система, а також результати кваліфікаційної роботи можуть бути інтегровані в робочі процеси деканату факультету математики та інформатики. Крім того, впровадження системи дозволить провести її тестування в реальних умовах з метою виявлення недоліків і подальшого їх усунення. Реалізація цієї системи також створює можливості для запуску процесів самореклами з метою початку та підтримки комерційної діяльності. Окрім того, матеріали роботи можуть бути використані для навчання майбутніх ІТ-фахівців розробці інформаційних систем.

У перспективі доцільно розглянути можливість додавання нових функцій, таких як мобільні додатки для студентів та викладачів, а також розширення можливостей для аналітики та прийняття рішень на основі отриманих даних. Удосконалена версія цієї інформаційної системи має великий потенціал для впровадження в закладах вищої освіти, що дозволить не тільки оптимізувати адміністративні процеси, а й підвищити загальний рівень якості освітнього процесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Автоматизована система управління ПП «Юнітех +». Отримано з <https://www.unitex.com.ua/products/commercial-software/automated-system-for-higher-education-institution/>
2. АС «деканат» - АСУ «ВНЗ». АСУ «ВНЗ». Отримано з <https://vuz.osvita.net/as-dekanat/>
3. Багрій, Р., & Петровський, С. (2022). Особливості сучасного тестування вебдодатків. *Вісник Хмельницького національного університету*, 70-74. Отримано з <http://surl.li/lbmezx>
4. Бубенко, М., Герасимов, В., Карпенко, В., & Морозов, С. (2023). Аналіз методів тестування вебдодатків. *Регіональний міжвузівський збірник наукових праць Системні технології*, 82-90. Отримано з https://nmetau.edu.ua/download/st/st_4_147_2023_full.pdf
5. Гордійчук, С. (2016). *Застосування інтегрованої інформаційної системи в управління освітнім процесом у медичному коледжі*. Отримано з <https://pi.ioid.gov.ua/images/pdf/2016%202/14.pdf>
6. Дядюн, С. (2015). *Прикладні інформаційні технології в охороні праці та техногенній безпеці*. 401 Authorization Required. Отримано з http://univer.nuczu.edu.ua/tmp_metod/2486/kurs_lekcij.pdf
7. Керри, Е. (2023). *Переваги та недоліки використання Spring Boot. Функції для рядків у Java*. JavaRush. Отримано з <https://javarush.com/ua/groups/posts/uk.3380.kava-breyk-75-perevagi-ta-nedolki-vikoristannja-spring-boot-funkc-dlja-rjadkv-u-java>
8. Кисіль, Н. М., Гаталяк, З. П., & Горбаль, Н. І. (2004). Класифікація інформаційних систем. Лісове господарство, лісова, паперова і деревообробна промисловість. № 29. 242-249. Отримано з https://moodle.znu.edu.ua/pluginfile.php/674975/mod_folder/content/0/242_Ky sil_LG_29.pdf?forcedownload=1

9. Куценко, Т., & Попова, О. (2016). *Значення і особливості застосування сучасних інформаційних систем в управлінні підприємством*. Отримано з https://www.researchgate.net/publication/311938046_ZNACENNA_I_OSOBLIVOSTI_ZASTOSUVANNA_SUCASNIH_INFORMACIJNIH_SISTEM_V_U_PRAVLINNI_PIDPRIEMSTVOM
10. Лазор, Я. (2017). *Поняття та види інформаційних систем*. Academic Journals and Conferences|. Отримано з <https://science.lpnu.ua/sites/default/files/journal-paper/2017/jun/4950/vnulpurn201683714.pdf>
11. Лондар, С., Литвинчук, А., Терещенко, Г., Кир'янов, А., Іриневич, Ю., Гапон, В.,... Криштоф, С. (2020). *Розвиток інформаційних систем управління освітою як інструмент реалізації державної освітньої політики*. Отримано з https://iea.gov.ua/wp-content/uploads/2020/07/Rozvitok-IS-2020-monografiyaFINAL_sajt.pdf
12. Львов, М.С., Співаковський, О.В., & Щедролосьєв, Д.С. (2005). Інформаційна система управління вищим навчальним закладом як платформа реалізації управління академічним процесом. *Вісник Харківського національного університету. Серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління.»* Отримано з <https://www.kspu.edu/FileDownload.ashx/97.pdf?id=fa9fe084-29ad-430a-a9c5-19f0d5e80360>
13. Ніколашин, В., & Мушеник, І. (2020). *Поняття та види інформаційних систем*. Отримано з <http://188.190.43.194:7980/jspui/bitstream/123456789/8062/1/TK-2020-305-308.pdf>
14. Ощипок, І., & Петришин, Н. (2023). *Інформаційні технології в нових умовах освітнього процесу і управління ним*. Отримано з https://www.researchgate.net/publication/373311522_INFORMACIJNI_TEHN

OLOGII_V_NOVIN_UMOVAN_OSVITNOGO_PROCESU_I_UPRAVLINN A_NIM

15. *Пакет програм «Деканат» для вищих навчальних закладів III-IV рівнів акредитації.* Політек-СОФТ. Отримано з <http://www.politek-soft.kiev.ua/index.php?do=products&product=deanery34>
16. Пархуць, Л., & Ясинська, С. (2014). *Інформаційні системи в освіті: Автоматизовані навчальні системи.* Отримано з <https://core.ac.uk/download/pdf/147037367.pdf>
17. Подоба, В. (2016). *Веброзробка з python та django для початківців.*
18. Про затвердження Положення про дозвільний порядок проведення робіт з технічного захисту інформації для власних потреб, Наказ Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України № 9 (2024). (Україна). Отримано з <https://zakon.rada.gov.ua/laws/show/z0245-02#Text>
19. Про захист інформації в інформаційно-телекомунікаційних системах, Закон України № 80/94-ВР (2024). (Україна). Отримано з <https://zakon.rada.gov.ua/laws/show/80/94-вр#Text>
20. Про Положення про технічний захист інформації в Україні, Указ Президента України № 1229/99 (2008). (Україна). Отримано з <https://zakon.rada.gov.ua/laws/show/1229/99#Text>
21. Сазонець, О. (2014). *Інформаційні системи і технології в управлінні зовнішньоекономічною діяльністю.* Отримано з <http://surl.li/dtigkl>
22. Триус, Ю., Кожем'якін, О. Аширова, А., & Заспа, Г. (2020). *Інформаційно-аналітична система підтримки освітньої діяльності структурних підрозділів закладів вищої освіти.* Отримано з https://www.researchgate.net/publication/350112480_INFORMATIONO-ANALITICNA_SISTEMA_PIDTRIMKI_OSVITNOI_DIALNOSTI_STRUKTURNIH_PIDROZDILIV_ZAKLADIV_VISOI_OSVITI
23. Цюцюра, М., Криворучко, О., Жирова, Т., & Котенко, Н. (2019). *Сучасні технології тестування і захисту вебсторінок. Управління розвитком*

- складних систем*, 100-105. Отримано з <https://urss.knuba.edu.ua/files/zbirnyk-39/17.pdf>
- 24.Юрчук, Н. (2015). *Інформаційні системи в управлінні діяльністю підприємства*. Журнал Агросвіт - наукове фахове видання з питань економіки. Отримано з http://www.agrosvit.info/pdf/19_2015/12.pdf
 - 25.Яловенко, Л.В., & Шроль, Т.С. (2024а). Аналіз технологій для розробки інформаційної системи менеджменту освітнього процесу. *Підготовка педагогів до професійної діяльності в умовах змішаного навчання : матеріали III Всеукраїнської науково-практичної конференції / Рівне : РВВ РДГУ*, 89–93.
 - 26.Яловенко, Л.В., & Шроль, Т.С. (2024b). Визначення функціональних вимог до інформаційної системи управління освітнім процесом. *Інформаційні технології в професійній діяльності: матеріали XVII Всеукраїнської науково-практичної конференції / Рівне : РВВ РДГУ*, 114–117.
 - 27.Admin. (2022). Layout testing. basic requirements. Online courses from QATestLab | Main page. Retrieved from Отримано з <https://en.training.qatestlab.com/blog/technical-articles/layout-testing-basic-requirements/>
 - 28.Areeba, S. (2024). Django inline formset factory with examples. Medium. Retrieved from <https://awstip.com/django-inline-formset-factory-with-examples-27576b915b5a>
 - 29.Biswas, N. (2023). Beginner app with Django. Medium. Retrieved from <https://awstip.com/beginner-app-with-django-27823c78e3a4>
 - 30.Community, J. (2024). *Bootstrap technology in web development: A comprehensive overview*. Medium. Retrieved from <https://medium.com/@imjobreadyit/bootstrap-technology-in-web-development-a-comprehensive-overview-f2f34e46ca18>
 - 31.Django Software Foundation. *Writing your first Django app, part 1 | Django documentation*. Django Project. Retrieved from <https://docs.djangoproject.com/en/5.1/intro/tutorial01/>

32. *Django vs spring boot 2023 - know the differences - cyber sucess.* (2023). Cyber Sucess. Retrieved from <https://www.cybersuccess.biz/django-vs-spring-boot/>
33. Dodds, K. (2023). *Why you should probably be using SQLite.* Epic Web Dev. Retrieved from <https://www.epicweb.dev/why-you-should-probably-be-using-sqlite>
34. GeeksforGeeks. (2022). *What are AJAX applications in web development ? - GeeksforGeeks.* Retrieved from <https://www.geeksforgeeks.org/what-are-ajax-applications-in-web-development/>
35. GeeksforGeeks. (2023). *How to create a Django project? - GeeksforGeeks.* Retrieved from <https://www.geeksforgeeks.org/how-to-create-a-django-project/>
36. Getachew, S. (2024). *What is jinja templating in django? A comprehensive guide for beginners and experts.* Medium. Retrieved from <https://python.plainenglish.io/what-is-jinja-templating-in-django-a-comprehensive-guide-for-beginners-and-experts-caf2906859a5>
37. IBM. *What is django? | IBM.* IBM - United States. Retrieved from <https://www.ibm.com/topics/django>
38. Khosrowabadi, J. (2022). *Django tutorial on how to create A booking system for A health clinic.* Medium. Retrieved from <https://blog.devgenius.io/django-tutorial-on-how-to-create-a-booking-system-for-a-health-clinic-9b1920fc2b78>
39. Mantosh, M. (2023). *What is the django web framework? | the pycharm blog.* The JetBrains Blog. Retrieved from <https://blog.jetbrains.com/pycharm/2023/11/what-is-the-django-web-framework/>
40. Melé, A. (2022). *Django 4 By Example - Fourth Edition: Build powerful and reliable Python web applications from scratch.*
41. Mohan, G. (2022). *Full stack testing: A practical guide for delivering high quality software.* O'Reilly Media, Incorporated.
42. Novak, K. (2016). *How to test the adaptive/responsive design using a browser.* OTAKOYI – UI/UX, Web, Branding. Retrieved from

- <https://otakoyi.ua/en/blog/how-to-test-the-adaptive-responsive-design-using-a-browser>
43. Pranav, A. (2024). *Unleashing the importance of CSS in web development*. Antino | Product Development Company. Retrieved from <https://www.antino.com/blog/importance-css-web-development>
 44. QATestLab. (2023). *Навчальний посібник «SOFTWARE TESTING FOR UNIVERSITIES»*.
 45. Shaikh, A. (2022). *How to select the right technology stack for web application development*. Software Development Company-Peerbits. Retrieved from <https://www.peerbits.com/blog/choose-right-technology-stack-for-web-application-development.html>
 46. Smith, J. (2024). *PostgreSQL for website developers | simplify backups*. Effortless Data Backup and Recovery for Modern Databases. Retrieved from <https://upback.cloud/blog/postgresql-web-development>
 47. Suresh, D. (2023). *Introduction to javascript and its role in web development*. Medium. Retrieved from <https://medium.com/@dhanushkumarsuresh/introduction-to-javascript-and-its-role-in-web-development-8d920aa7b553>
 48. Tusharupadhyay. (2024). *Building responsive and interactive applications with HTMX*. Medium. Retrieved from <https://medium.com/@tusharupadhyay691/building-responsive-and-interactive-applications-with-htmx-e37aabfacf20>
 49. Vincent, W. (2023). *Django for beginners: Build websites with python and django*.
 50. W3Schools. *Django tutorial*. W3Schools Online Web Tutorials. Retrieved from <https://www.w3schools.com/django/index.php>
 51. Zapke-Gründemann, M. (2015). *Django-marcador tutorial – django-marcador v1.5.0 tutorial*. django-marcador - Basic Django Tutorial. Retrieved from <http://django-marcador.keimlink.de/en/>

ДОДАТКИ

Додаток А



Рис. А.1. Сертифікат конференції



Рис. А.2. Сертифікат конференції



Рис. Б.1. QR-код з посиланням на репозиторій інформаційної системи деканату

Посилання на репозиторій інформаційної системи деканату:

<https://github.com/Kok1ysh/dean-sManager.git>



Рис. В.1. QR-код з посиланням на інформаційну систему деканату

Посилання на інформаційну систему деканату:

<https://yalovenko.pythonanywhere.com/>

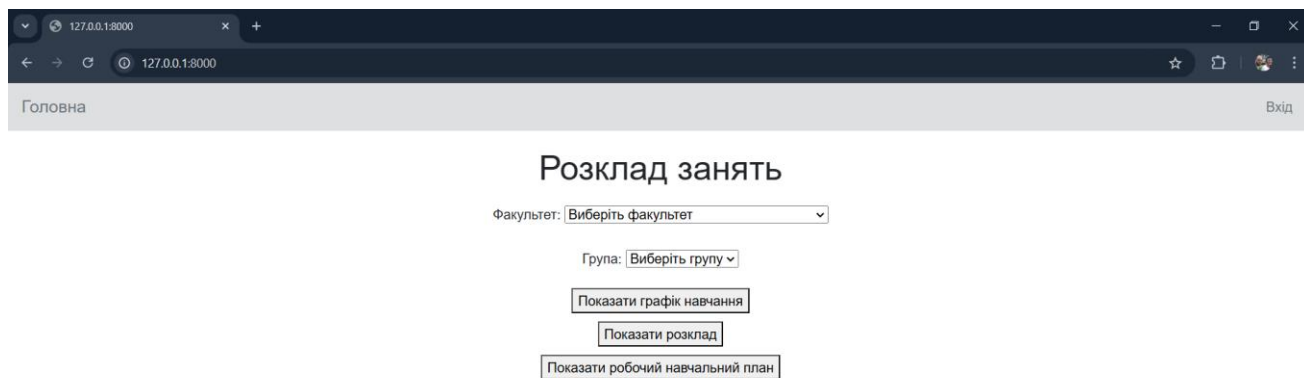


Рис. Г.1. Головна сторінка інформаційної системи деканату

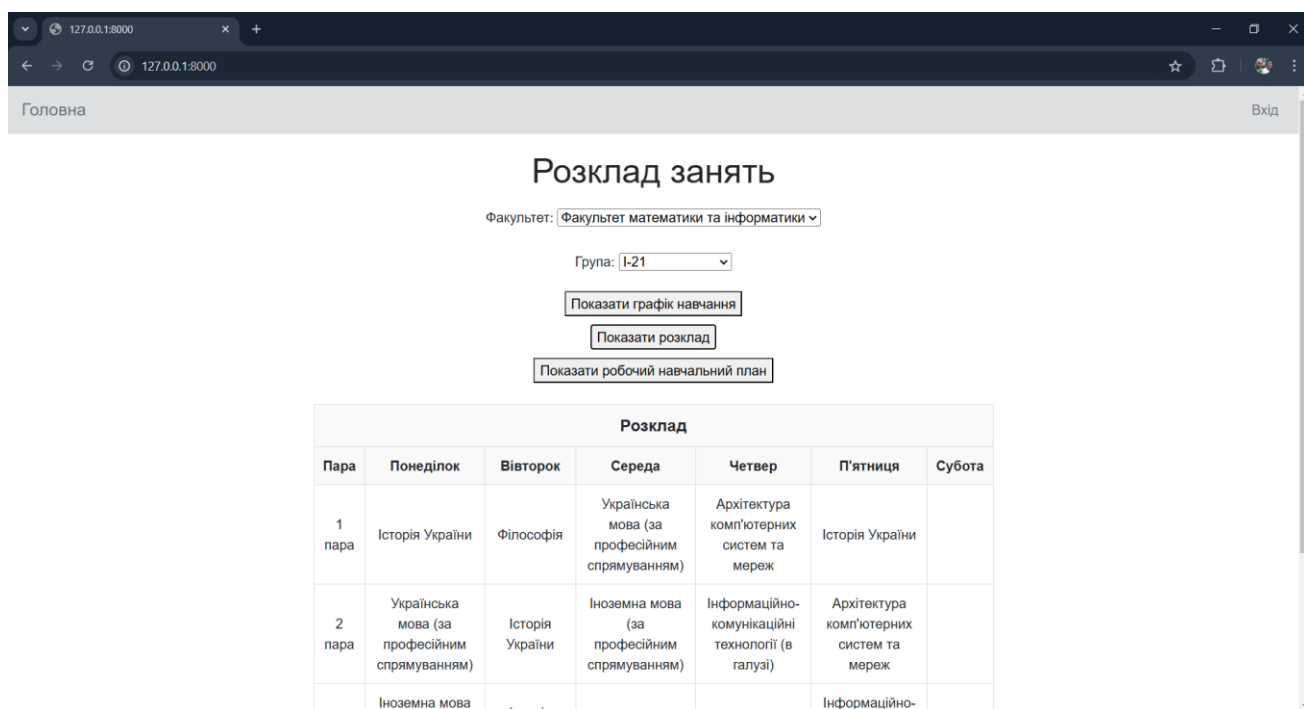
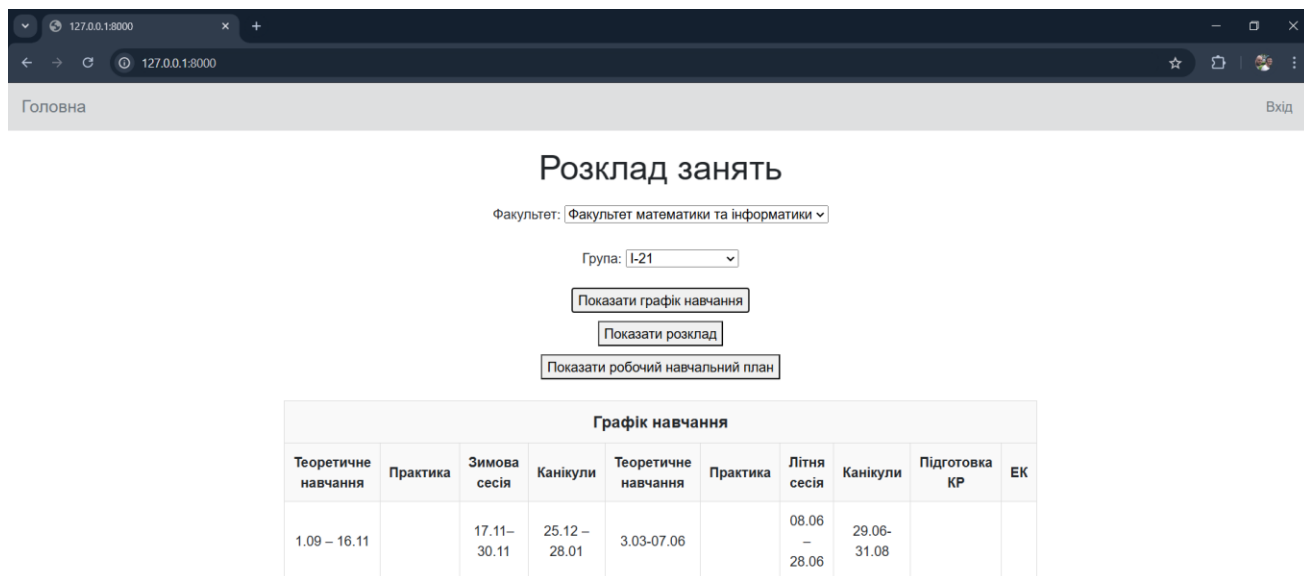


Рис. Г.2. Відображення розкладу групи I-21 факультету математики та інформатики



Розклад занять

Факультет:

Група:

Графік навчання									
Теоретичне навчання	Практика	Зимова сесія	Канікули	Теоретичне навчання	Практика	Літня сесія	Канікули	Підготовка КР	ЕК
1.09 – 16.11		17.11 – 30.11	25.12 – 28.01	3.03-07.06		08.06 – 28.06	29.06-31.08		

Рис. Г.3. Відображення графіку навчання групи I-21 факультету математики та інформатики

127.0.0.1:8000

Головна

Вхід

Розклад занять

Факультет:

Факультет математики та інформатики

Група:

I-21

Показати графік навчання

Показати розклад

Показати робочий навчальний план

Робочий навчальний план

Предмет	Викладач	Години	Кредити ECTS (1см)	Години (1см)	Лекції (1см)	Практичні (1см)	Лабораторні (1см)	Самостійна робота (1см)	Форма контролю (1см)	Кредити ECTS (2см)	Години (2см)	Лекції (2см)	Практичні (2см)	Лабораторні (2см)	Самостійна робота (2см)	Форма контролю (2см)
Історія України	Боярський Василь	90	3	90	30	30	0	30	Екзамен	0	0	0	0	0	0	
Історія																

Рис. Г.4. Відображення робочого навчального плану групи I-21 факультету математики та інформатики

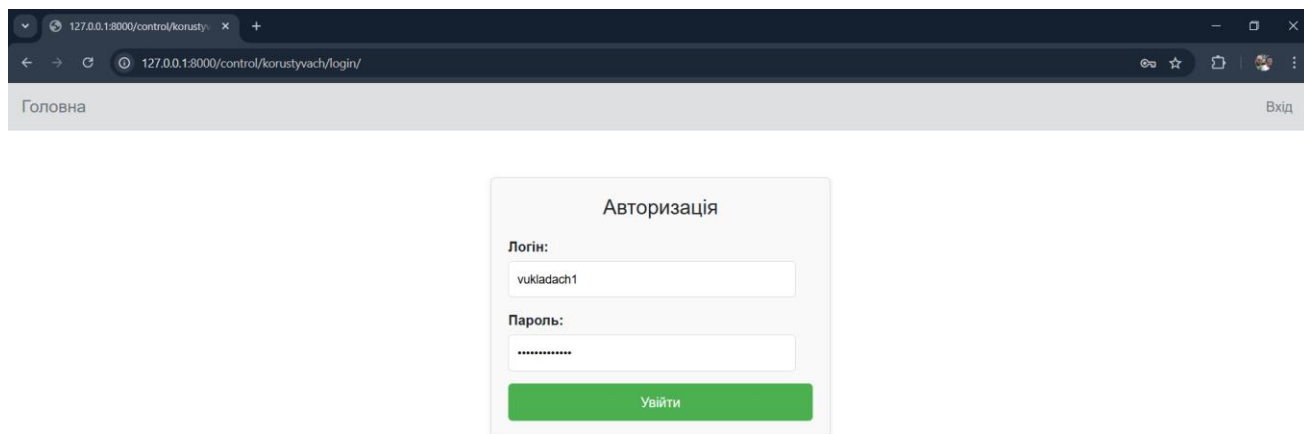


Рис. Г.5. Сторінка авторизації в інформаційній системі деканату

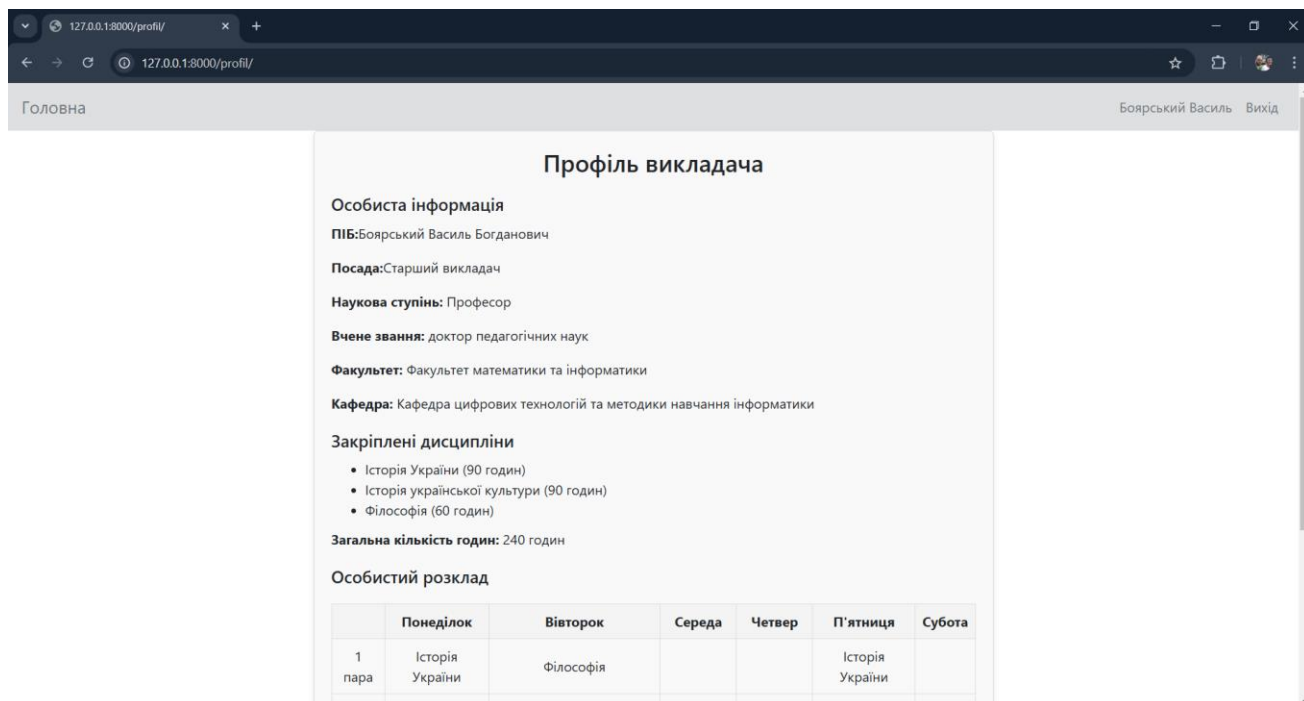


Рис. Г.6. Сторінка профілю викладача

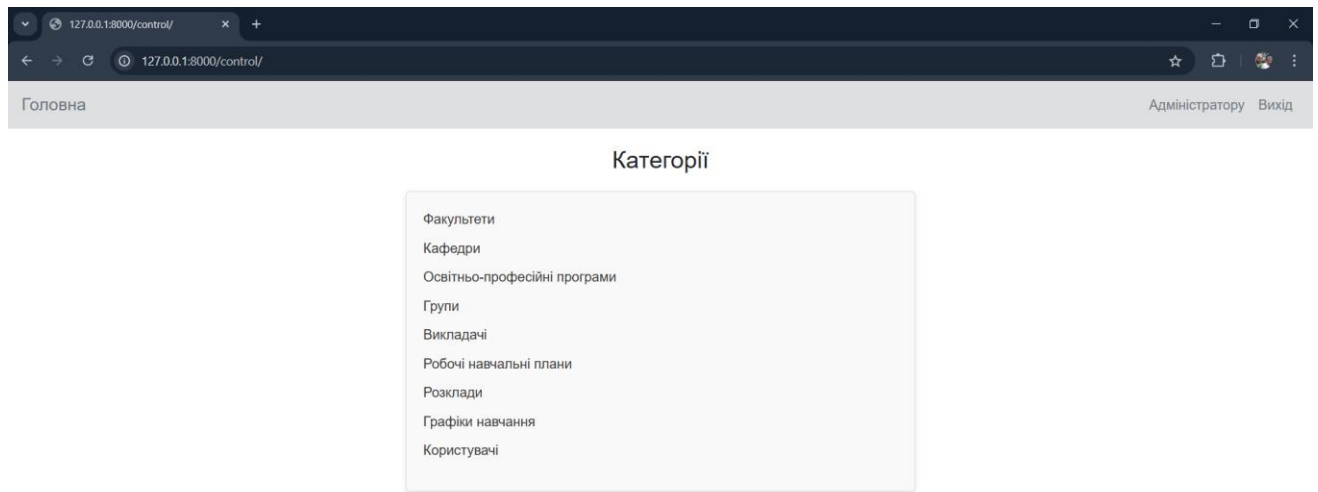


Рис. Г.7. Сторінка категорій управління адміністратора

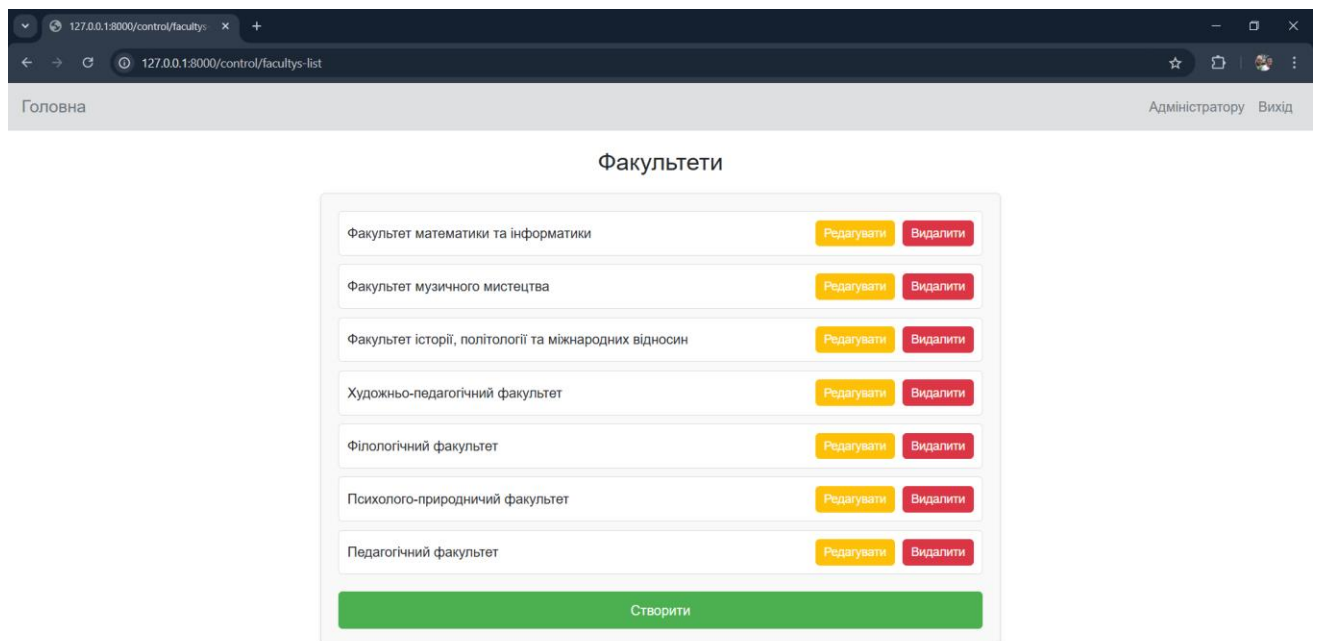


Рис. Г.8. Сторінка категорії «Факультети»

Додати факультет

Назва * Факультет математики тз

Адреса факультету * 33000, м. Рівне, вул. Плас

Декан факультету * Максимцев Юрій Роман

Submit

Рис. Г.9. Сторінка форми створення та редагування категорії «Факультети»

Кафедри

Кафедра цифрових технологій та методики навчання інформатики	Редагувати	Видалити
Кафедра інформаційних технологій та моделювання	Редагувати	Видалити
Кафедра вищої математики	Редагувати	Видалити
Кафедра математики з методикою викладання	Редагувати	Видалити
Кафедра історії України	Редагувати	Видалити
Кафедра всесвітньої історії	Редагувати	Видалити
Кафедра політичних наук	Редагувати	Видалити

Створити

Рис. Г.10. Сторінка категорії «Кафедри»

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/control/update-kafedras/1/'. The page has a header with 'Головна' on the left and 'Адміністратору Вихід' on the right. The main content area is titled 'Додати кафедру' and contains a form with the following fields:

- Назва * Кафедра цифрових техн
- Адреса кафедри * Рівне
- Завідувач кафедри * Войтович Ігор Станіслав
- FacultyKafedra * Факультет математики та інформатики

A 'Submit' button is located at the bottom of the form.

Рис. Г.11. Сторінка форми створення та редагування категорії «Кафедри»

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/control/educational-programs-create-or-update/'. The page has a header with 'Головна' on the left and 'Адміністратору Вихід' on the right. The main content area is titled 'Освітньо-професійні програми' and contains a table with three rows of programs:

Середня освіта (Інформатика)	Редагувати	Видалити
Професійна освіта (Цифрові технології)	Редагувати	Видалити
Комп'ютерні науки	Редагувати	Видалити

Below the table is a large green button labeled 'Створити'.

Рис. Г.12. Сторінка категорії «Освітньо-професійні програми»

Спеціальність * 014 Середня освіта (Інфс)

Спеціалізація * Середня освіта (Інформа

Освітня кваліфікація * Бакалавр освіти за спеці

Професійна кваліфікація * Вчитель інформатики

Додати компонент

Код н/д *	Назва *	Кредити *	Форма контролю *	Семестр *	Delete?	Custom Delete btn
OK1	Історія України	4	Екзамен	1	☐	Delete
OK2	Історія української культ	3	Залік	2	☐	Delete
OK3	Українська мова (за про	3	Екзамен	1	☐	Delete
OK4	Філософія	3	Екзамен	4	☐	Delete

Рис. Г.13. Сторінка форми створення та редагування категорії «Освітньо-професійні програми»

Головна

Адміністратору Вихід

Групи

I-21	Редагувати	Видалити
ЦТ-31	Редагувати	Видалити
КН-41	Редагувати	Видалити

Створити

Рис. Г.14. Сторінка категорії «Групи»

Головна Адміністратору Вихід

Додати групу

Назва *

Повна назва *

Kurs *

Educational programs *

Faculty *

Submit

Рис. Г.15. Сторінка форми створення та редагування категорії «Групи»

Головна Адміністратору Вихід

Викладачі

Боярський Василь	Редагувати	Видалити
Коваль Ігор	Редагувати	Видалити
Кішман Марія	Редагувати	Видалити
Коваль Анна	Редагувати	Видалити
Мудрик Дмитро	Редагувати	Видалити

Створити

Рис. Г.16. Сторінка категорії «Викладачі»

Ім'я * Василь

Прізвище * Боярський

По батькові * Богданович

Посада * Старший викладач

Наукова ступінь * Професор

Вчене звання * доктор педагогічних наук

Facultet * Факультет математики та інформатики

Kafedra * Кафедра цифрових технологій та методики навчання інформатики

Зберегти

Рис. Г.17. Сторінка форми створення та редагування категорії «Викладачі»

Головна Адміністратору Вихід

Робочі навчальні плани

014 Середня освіта (Інформатика) І-21	Редагувати	Видалити
015 Професійна освіта(за спеціалізаціями) ЦТ-31	Редагувати	Видалити
122 Комп'ютерні науки КН-41	Редагувати	Видалити

Створити

Рис. Г.18. Сторінка категорії «Робочі навчальні плани»

Головна Адміністратору Вихід

Додати робочий навчальний план

Educational programs * Середня освіта (Інформатика)

Group * I-21

Додати предмет

Предмет*	Викладач*	Години*	Кредити ECTS(1cm)*
Історія України	Боярський Василь	90	3
Історія української культури	Боярський Василь	90	3
Філософія	Боярський Василь	60	2
Українська мова (за професійним спрямуванням)	Кішман Марія	90	3

Рис. Г.19. Сторінка форми створення та редагування категорії «Робочі навчальні плани»

Головна Адміністратору Вихід

Розклади

I-21	Редагувати	Видалити
ЦТ-31	Редагувати	Видалити
КН-41	Редагувати	Видалити

Створити

Рис. Г.20. Сторінка категорії «Розклади»

Головна Адміністратору Вихід

Факультет

Факультет математики та інформатики

Група

I-21

	Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота
1 пара	Історія Украї...	Філософія	Українська м...	Архітектура ...	Історія Украї...	-----
2 пара	Українська м...	Історія Украї...	Іноземна мо...	Інформаційн...	Архітектура ...	-----
3 пара	Іноземна мо...	Історія украї...	Філософія	Філософія	Інформаційн...	-----
4 пара	-----	-----	-----	-----	-----	-----
5 пара	-----	-----	-----	-----	-----	-----

Зберегти

Рис. Г.21. Сторінка форми створення та редагування категорії «Робочі навчальні плани»

Головна Адміністратору Вихід

Графіки навчання груп

I-21

Редагувати

Видалити

ЦТ-31

Редагувати

Видалити

КН-41

Редагувати

Видалити

Створити

Рис. Г.22. Сторінка категорії «Графіки навчання груп»

Головна Адміністратору Вихід

Додати графік навчання

Група I-21

Теоретичне навчання(1 семестр) 1.09 – 16.11

Практика(1 семестр)

Зимова сесія 17.11– 30.11

Канікули(1 семестр) 25.12 –28.01

Теоретичне навчання(2 семестр) 3.03-07.06

Практика(2 семестр)

Рис. Г.23. Сторінка форми створення та редагування категорії «Графіки навчання груп»

Головна Адміністратору Вихід

Користувачі

admin	Редагувати	Видалити
yalovenko	Редагувати	Видалити
boyarskuy_vasul	Редагувати	Видалити
Test	Редагувати	Видалити
wasdwasd	Редагувати	Видалити
wasdwasd2	Редагувати	Видалити
vukladach	Редагувати	Видалити
admin123	Редагувати	Видалити

Створити

Рис. Г.24. Сторінка категорії «Користувачі»

Додати користувача

Ім'я

Прізвище

Логін *

Пароль *

Є адміном ☐

Є викладачем ☒

Викладач

Зберегти

Рис. Г.25. Сторінка форми створення та редагування категорії «Користувачі»