

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота
за освітнім ступенем «бакалавр»
на тему:
**«Інформаційна система підтримки електронного документообігу закладу
вищої освіти. Супроводження індивідуального навчального плану здобувача
вищої освіти»**

Виконав:

здобувач IV курсу
групи ІПЗ–41
спеціальності 121 «Інженерія
програмного забезпечення»
Герман Тарас Романович

Науковий керівник:

к.т.н., доцент Сяський В. А.

Рівне – 2025

ЗМІСТ

СПИСОК ТЕРМІНІВ ТА УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП.....	5
РОЗДІЛ 1 ОСНОВНІ ПОЛОЖЕННЯ ТА ПОНЯТТЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ У ЗАКЛАДАХ ВИЩОЇ ОСВІТИ	7
1.1 Понятійний апарат та нормативно – правова база забезпечення електронного документообігу в закладах вищої освіти.....	7
1.2 Інформаційна система підтримки електронного документообігу	10
1.3 Забезпечення інформаційної підтримки реалізації індивідуальної освітньої траєкторії здобувачів вищої освіти у закладах вищої освіти	12
РОЗДІЛ 2 МЕТОДОЛОГІЧНІ ОСНОВИ ТА ВИМОГИ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	14
2.1 Життєвий цикл програмного забезпечення	14
2.2 Методологія розробки.....	15
2.3 Функціональні вимоги	16
2.4 Нефункціональні вимоги	17
РОЗДІЛ 3 КОНЦЕПТУАЛЬНА МОДЕЛЬ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ ІНДІВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ	18
3.1 Постановка та формалізація задачі.....	18
3.2 Вихідні дані інформаційної системи	20
3.3 Основні бази даних інформаційної системи.....	21
3.4 Місце інформаційної підсистеми в структурі інтегрованої платформи забезпечення взаємодії з іншими підсистемами	22
РОЗДІЛ 4 АРГУМЕНТАЦІЯ ТА ОБГРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТІВ РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	23
4.1 Мова програмування Python, декларативна мова програмування SQL, формат даних JSON, протокол HTTP	23
4.2 Середовище розробки	25
4.3 Вибір бібліотек.....	26
4.4 Використання прикладного програмного інтерфейсу	28

4.5	Docker.....	29
4.6	Мікросервісна архітектура	30
РОЗДІЛ 5 ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ СЕПРОВОДЖЕННЯ ІНДИВІДУАЛЬНОГО НАВЧАЛЬНОГО ПЛАНУ		33
5.1	Розроблення баз даних	33
5.2	Модуль обслуговування індивідуального навчального плану здобувача освіти	36
5.3	Прикладний програмний інтерфейс	39
5.4	Інтеграція з іншими мікросервісами інформаційної системи.....	40
ВИСНОВКИ.....		44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		46

СПИСОК ТЕРМІНІВ ТА УМОВНИХ ПОЗНАЧЕНЬ

ЄДЕБО – єдина державна електронна база з питань освіти.

ЗВО – заклад вищої освіти.

API – прикладний програмний інтерфейс.

ВСТУП

Актуальність дослідження. В умовах швидкої диджеталізації заклади вищої освіти можуть стикатися із проблемою ефективності управлінських і освітніх процесів, зокрема через впровадження інформаційних систем. Однією із актуальних проблем сьогодення є автоматизація супроводження індивідуального навчального плану здобувача вищої освіти, яка відіграє ключову роль у гнучкості навчального процесу, забезпеченні права вибору напряму навчання та в подальшому формування компетентності здобувача.

Індивідуальний навчальний план є ключовим інструментом, що забезпечує гнучкість навчального процесу, дозволяючи студенту обирати дисципліни відповідно до власних інтересів, професійних цілей та компетентнісної моделі, затвердженої закладом освіти.

Мета дослідження полягає у розробці та впровадженні інформаційної системи підтримки електронного документообігу, орієнтованої на генерацію та подальше обслуговування індивідуального навчального плану здобувача вищої освіти, для вивільнення людського ресурсу, автоматизації деяких управлінських функцій та забезпеченні гнучкого підходу до навчання.

Завдання дослідження.

1. Дослідити рівень автоматизації процесів електронного документообігу у закладах вищої освіти.
2. Дізнатися про особливості ведення та супроводження індивідуальних навчальних планів, узагальнити вимоги до підсистеми.
3. Розробити архітектуру інформаційної системи, розробити прототип інформаційної підсистеми системи та реалізувати основні функціональні модулі.
4. Здійснити експериментальну інтеграції з іншими сервісами системи та перевірити ефективність роботи.
5. Проаналізувати результати та запропонувати подальшу траєкторію розвитку підсистеми.

Об'єктом дослідження є процес електронного документообігу в закладах вищої освіти, його організація та супроводження індивідуальних навчальних планів, що забезпечать зберігання, обробку, доступ та оновлення інформації.

Предметом дослідження є принципи, методи та засоби розробки інформаційної системи, що забезпечують функціонування системи індивідуальних навчальних планів. Особлива увага приділяється структурі навчальних даних, процесів оновлення даних у сховищах та інтеграції з іншими підсистемами.

Методи дослідження. Теоретичні методи: аналіз, узагальнення, систематизація інформації з наукової літератури, моделювання функціональних та структурних компонентів. Емпіричні методи: спостереження за процесами документообігу у закладах вищої освіти, аналіз функціональних вимог до системи, впровадження системи у заклад вищої освіти.

Практичне значення дослідження полягає в можливості використання розробленої системи у закладах вищої освіти для автоматизації формування індивідуальних навчальних планів. Система дає змогу вивільнити людський ресурс, підвищити ефективність управління освітнім процесом, зменшити обсяг рутинної роботи, покладеної на плечі адміністрації, а також покращити якість навчання та взаємодію здобувача із закладом вищої освіти.

Апробація та впровадження результатів дослідження.

IV Всеукраїнська науково-практична конференція «ПІДГОТОВКА ПЕДАГОГІВ ДО ПРОФЕСІЙНОЇ ДІЯЛЬНОСТІ В УМОВАХ ЗМІШАНОГО НАВЧАННЯ» (м. Рівне, 14 травня 2025)

Структура роботи.

Структура та обсяг роботи обумовлені метою і завданнями проведеного дослідження. Вона складається із вступу, п'яти розділів, висновків, списку використаних джерел та додатків. Обсяг курсової роботи становить 42 сторінки. Список використаних джерел складає 20 найменувань.

РОЗДІЛ 1

ОСНОВНІ ПОЛОЖЕННЯ ТА ПОНЯТТЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ У ЗАКЛАДАХ ВИЩОЇ ОСВІТИ

1.1 Понятійний апарат та нормативно – правова база забезпечення електронного документообігу в закладах вищої освіти

У процесі цифровізації управлінських та освітніх функцій ЗВО актуальності набирає правомірність, цілісність та прозорість таких процесів, для їх забезпечення варто мати чітко визначену термінологію, та знати нормативно-правову базу.

Електронні документи в Україні регламентуються законами, постановами Кабінету міністрів та іншими нормативно-правовими актами, які визначають їх правовий статус, порядок збереження, використання, обміну та передачі до архіву.

Згідно із Законом України «Про електронні документи та електронний документообіг» [1] поняття електронний документ визначається наступним чином:

«Електронний документ – документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов’язкові реквізити документа.

Склад та порядок розміщення обов’язкових реквізитів електронних документів визначається законодавством.

Електронний документ може бути створений, переданий, збережений і перетворений електронними засобами у візуальну форму.

Візуальною формою подання електронного документа є відображення даних, які він містить, електронними засобами або на папері у формі, придатній для приймання його змісту людиною».

Існування електронного документу, що мають юридичну силу, неможливе без кваліфікованого електронного підпису, вимоги використання такого підпису визначені у Законі України «Про електронний цифровий підпис» [2].

Якщо розглядати використання електронних документів з точки зору Цивільного кодексу України, то слід зазначити, що відповідно до статей 205, 207,

ЦКУ [3] електрона форма правочину є допустимою, а електронний документ з кваліфікованим електронним підписом прирівнюється до паперового варіанту.

Крім того, важливу роль у регулюванні електронного документообігу відіграють і інші законодавчі акти. Зокрема Закон України «Про інформацію» [4] встановлює правові засади інформаційних відносин, класифікує види інформації, визначає права та обов'язки суб'єктів, а також гарантує захист інформації, у тому числі в електронній формі.

У свою чергу, Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» [5] визначає основні принципи, вимоги та методи забезпечення безпеки даних, що мають місце в інформаційних системах. Він встановлює правові механізми захисту електронних документів від неакціонованого доступу, що є критичним для ЗВО при впровадженні системи електронного документообігу, які оперують особистими даними.

Також важливо згадати і про уніфікацію та стандартизацію документів, які визначені у положенні ДСТУ 4163:2020 «Уніфікована система організаційно-розпорядчої документації. Вимоги до оформлення документів» [6], що застосовується як для паперових, так і для електронних документів. Цей стандарт регламентує оформлення реквізитів, шрифтів, полів, розміщення елементів документа, що спрощує інтеграцію електронних систем обробки документів.

Крім цього, при організації електронного документообігу у ЗВО слід враховувати положення Закону України «Про Національний архівний фонд та архівні установи» [7], який встановлює вимоги щодо зберігання електронних документів, термінів їх збереження, передачі до архіву та знищення. Цей документ передбачає можливість ведення архівів у цифровій формі, за умови дотримання вимог безпеки та цілісності електронних даних.

Особливої уваги потребує впровадження систем електронного документообігу у контексті законодавства про захист персональних даних. Закон України «Про захист персональних даних» [8] регламентує обробку, зберігання та використання інформації, яка ідентифікує особу. У закладах вищої освіти це стосується, зокрема, персональних даних здобувачів освіти, викладачів,

адміністративного персоналу. Порушення вимог цього закону може призвести до юридичної відповідальності, тому системи електронного документообігу мають підтримувати інструменти контролю доступу, логування дій, шифрування тощо.

Сучасні електронні системи документообігу передбачають інтеграцію з державними інформаційними системами, такими як ЄДЕБО. Відповідно до Постанови Кабінету Міністрів України № 752 [9] від 13 липня 2011 року, кожен заклад освіти зобов'язаний забезпечити своєчасне і повне внесення даних до ЄДЕБО. Це ще раз підкреслює важливість електронного та автоматизованого документообігу.

Однією з переваг електронного документообігу в ЗВО є можливість автоматизації рутинних процесів, таких як погодження заяв, оформлення наказів, реєстрація академічних відпусток, розклад консультацій тощо. Це знижує адміністративне навантаження на працівників та підвищує ефективність внутрішніх процесів управління. Наприклад, студент, перебуваючи в іншому місті, може подати заяву на академічну відпустку або обрати вибірккову дисципліну без необхідності фізичної присутності в деканаті.

Важливим є також аспект цифрової грамотності персоналу. Ефективне функціонування електронного документообігу можливе лише за умови, що всі учасники процесу – адміністрація, викладачі володіють базовими навичками користування інформаційними системами. З цією метою у ЗВО проводяться внутрішні навчання, інструктажі та впроваджуються тренінги по інформаційній безпеці.

Ще одним важливим аспектом є прозорість. Електронний документообіг унеможливорює несанкціоновані зміни документів, підробки чи втрату даних. Кожна дія в системі фіксується: хто, коли і що саме виконував. Такий рівень контролю сприяє довірі як серед внутрішніх користувачів (студентів, викладачів), так і в зовнішньому середовищі (державні органи, роботодавці, міжнародні партнери).

Електронний документообіг дає змогу оптимізувати, автоматизувати процеси обробки, створення цих документів, зменшити кількість помилок та спростити

пошук інформації. Така організаційна структура може передбачати захист даних, доступність та можливість інтегрування з іншими системами. Варто згадати і те, що електронний документообіг дозволяє закладу освіти повноцінно функціонувати в умовах війни, тобто дистанційному форматі. Але для якісного впровадження електронного документообігу варто пройти всі етапи впровадження (рис. 1.1).

Етапи впровадження електронного документообігу	
1 етап	Аналіз потреб
2 етап	Вибір платформи
3 етап	Навчання персоналу
4 етап	Пілотний запуск
5 етап	Повне впровадження
6 етап	Підтримка та модернізація системи

Рисунок 1.1 Етапи впровадження електронно документообігу

1.2 Інформаційна система підтримки електронного документообігу

Інформаційна система – автоматизована система, комп'ютерна мережа або система зв'язку [15].

Інформаційна система – організаційно–технічна система обробки інформації за допомогою технічних і програмних засобів [14].

Інформаційна система – система, призначена для одержання, обробки, зберігання, відображення та/або реєстрації даних про технічний стан конструкцій, систем, елементів, їх властивості та/або функціонування[13].

Інформаційна система в умовах швидкої глобалізації є невід'ємною складовою функціонування будь-якої організації. Основними завданнями інформаційної системи є обробка даних, автоматизація управлінських процесів, зберігання інформації, а також інтеграція з іншими системами та підсистемами. Вони можуть охоплювати різні напрями в закладах вищої освіти: управління навчальним процесом, фінансами, кадрами, документообігом тощо.

Найдавнішими і найпоширенішими інформаційними системами вважають бібліотеки. В бібліотеках збирають книжки, зберігають їх, створюють каталоги різного призначення, для полегшення пошуку потрібної книги.

Структурою інформаційної системи є: технічні засоби обробки даних, програмне забезпечення і персонал, який обслуговує цю інформаційну систему. Чотири складові частини утворюють внутрішню інформаційну основу.

Крім вищезазначених елементів, важливими характеристиками будь-якої сучасної інформаційної системи є її інтегрованість, масштабованість, адаптивність і безпека. Інтегрованість дозволяє поєднувати інформаційну систему з іншими програмними компонентами або державними інформаційними ресурсами. Масштабованість забезпечує можливість розширення системи у разі збільшення обсягів інформації або кількості користувачів. Адаптивність проявляється в здатності системи швидко змінюватися у відповідь на нові вимоги навчального процесу або зміни в підході до ведення електронних документів. Безпека – ключовий фактор, що гарантує захист персональних даних студентів, викладачів та внутрішньої документації ЗВО.

У контексті закладів вищої освіти, інформаційна система повинна відповідати специфіці освітнього процесу, враховувати навчальні плани, типи документів, особливості вибіркового дисциплін, систему оцінювання, а також мати підтримку ролей користувачів: студентів, викладачів, науково-педагогічних працівників, працівників деканатів, гарантів освітніх програм. Саме завдяки гнучкій побудові інформаційної системи можливим є супровід таких елементів, як індивідуальний навчальний план здобувача освіти.

Зазвичай, система працює в межах клієнт-серверної архітектури або на базі вебтехнологій, що дозволяє забезпечити віддалений доступ до її ресурсів, що важливо, адже багато студентів та викладачів перебувають за межами країни. Важливою особливістю також є наявність інтерфейсів для експорту/імпорту інформації, формування статистичних і звітних даних, а також надання можливості аналітичного аналізу освітніх показників.

До допоміжних підсистем інформаційної системи можуть належати:

- модулі планування та моніторингу навчального навантаження;
- підсистеми електронної взаємодії (наприклад, сповіщення про зміни в індивідуальному навчальному плані);
- механізми верифікації та електронного підпису документів;
- архів збережених версій документів.

1.3 Забезпечення інформаційної підтримки реалізації індивідуальної освітньої траєкторії здобувачів вищої освіти у закладах вищої освіти

Індивідуалізація освітнього процесу в системі вищої освіти в Україні є однією з важливих цілей для забезпечення якісної професійної підготовки майбутніх фахівців різних галузей та підвищення їхньої конкурентоспроможності на ринку праці. Ідея та механізми формування індивідуальної освітньої траєкторії (ІОТ) здобувачів вищої освіти закладені в Законі України «Про освіту» [16], Законі України «Про вищу освіту» [17], Законі України «Про внесення змін до деяких законів України щодо розвитку індивідуальних освітніх траєкторій та вдосконалення освітнього процесу» [18]. Відповідно до положень цих нормативних документів ІОТ студента закладу вищої освіти передбачає створення освітніх можливостей для втілення персоналізованої стратегії досягнення освітніх цілей, що базується на самостійному виборі здобувачем провайдерів освітніх послуг, освітніх програм, способів реалізації та часових характеристик процесу навчання, а також окремих освітніх компонентів.

Основні характеристики індивідуального навчального плану

Індивідуальний навчальний план встановлює:

Послідовність вивчення обов'язкових і вибіркових дисциплін, передбачених освітньою програмою.

Форму засвоєння матеріалу, яка може включати аудиторну роботу, самостійне навчання, практику чи стажування.

Темп навчання, що дозволяє здобувачеві проходити окремі курси за прискореним графіком або з урахуванням академічних відпусток.

Процес розробки та затвердження

Формування індивідуального навчального плану відбувається на основі стандарту вищої освіти від Міністерства освіти і науки України та у тісній співпраці між здобувачем і закладом вищої освіти, зокрема його адміністрацією та викладацьким складом. Зазвичай, перед початком навчального року або семестру здобувач обирає дисципліни з переліку вибіркових компонентів, який пропонується університетом. На основі цих виборів план затверджується і стає обов'язковим для виконання.

Вплив на подальше працевлаштування

Додаток до диплома, сформований на основі індивідуального навчального плану, є важливим документом для майбутнього працевлаштування здобувача. Він дозволяє потенційним роботодавцям ознайомитися з переліком вивчених дисциплін, їхнім змістом і оцінками, що підтверджує відповідність кваліфікації випускника вимогам конкретної посади.

РОЗДІЛ 2

МЕТОДОЛОГІНІ ОСНОВИ ТА ВИМОГИ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Інженерія програмного забезпечення є фундаментальною дисципліною, яка охоплює процеси аналізу, проєктування, реалізації, тестування та підтримки програмних систем. Застосування її принципів дозволяє створювати якісне, ефективне та надійне програмне забезпечення, яке відповідає як технічним, так і бізнес-вимогам. У рамках розробки інформаційної системи підтримки електронного документообігу в закладі вищої освіти було враховано ключові аспекти інженерії програмного забезпечення: життєвий цикл програмного забезпечення, вибір відповідної методології розробки, а також чітке формулювання функціональних і нефункціональних вимог.

2.1 Життєвий цикл програмного забезпечення

Життєвий цикл програмного забезпечення – це послідовність етапів, які проходить програмне забезпечення від моменту формулювання вимог до завершення його експлуатації. У межах цього проєкту було орієнтовано на **інкрементну модель життєвого циклу програмного забезпечення**, яка передбачає створення системи поетапно, з поступовим нарощенням функціональності. Такий підхід дозволив частково впроваджувати й тестувати модулі системи ще до завершення розробки повного функціоналу, що дало змогу оперативно враховувати зауваження користувачів і адаптувати архітектуру під реальні потреби.

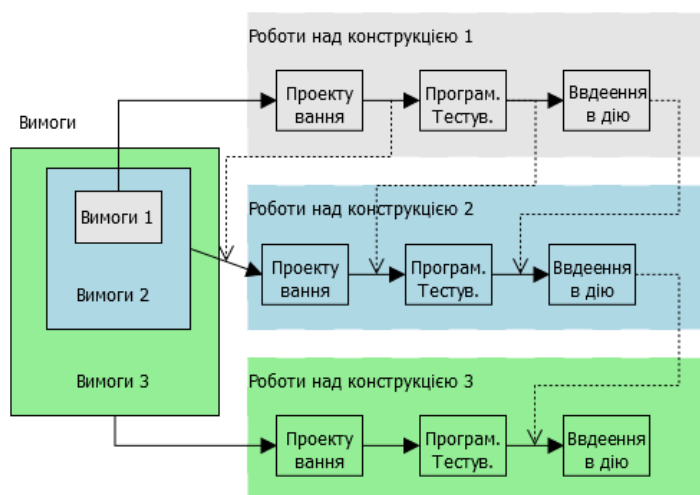


Рисунок 2.1 Ітераційна модель життєвого циклу програмного забезпечення

Класичні етапи життєвого циклу програмного забезпечення, дотриманих у межах цього проекту:

1. **Аналіз вимог** – вивчення потреб користувачів і особливостей внутрішніх процесів навчального закладу.
2. **Проектування** – розробка архітектури системи, визначення структури баз даних, API та взаємодії модулів.
3. **Реалізація** – програмування окремих компонентів системи із застосуванням Python, SQL, JSON, REST API.
4. **Тестування** – перевірка функціонування модулів, виявлення й виправлення помилок.
5. **Розгортання** – налаштування оточення, використання Docker для контейнеризації компонентів.
6. **Підтримка** – оновлення, виправлення помилок та адаптація системи до нових умов експлуатації.

2.2 Методологія розробки

Для організації розробки було обрано методологію **Agile (гнучка розробка)** у поєднанні з практиками **Scrum**, що дозволило забезпечити адаптивність до змін,

часті ітерації, швидке впровадження нових функцій та отримання зворотного зв'язку. Ітеративна природа цієї методології дозволила ефективно реалізовувати окремі функціональні блоки, тестувати їх у ізоляції та інтегрувати в загальну систему поступово. Це особливо важливо для складних систем, таких як електронний документообіг, де вимоги можуть уточнюватися протягом усього циклу розробки.

Застосування Scrum полягало у плануванні коротких спринтів, визначенні завдань, формуванні беклогу продукту та виконанні регулярних рев'ю. Завдяки цьому вдалось досягти високої прозорості процесу розробки, знизити ризики затримок і втрати якості коду.

2.3 Функціональні вимоги

Функціональні вимоги – це вимоги, що визначають конкретні функції, які має реалізовувати система. У рамках даного проєкту до основних функціональних вимог належать:

- Автентифікація користувачів.
- Формування, збереження та редагування індивідуального навчального плану.
- Генерація звітних документів (у форматі .docx) .
- Взаємодія з базою даних для зберігання структурованої інформації.
- Надання REST API для інтеграції з іншими інформаційними системами закладу.
- Реалізація розмежування прав доступу відповідно до ролей (викладач, адміністратор).
- Функціональні вимоги формувалися на основі аналізу ділових процесів вищого навчального закладу та з урахуванням досвіду потенційних користувачів системи.

2.4 Нефункціональні вимоги

Нефункціональні вимоги визначають якісні характеристики програмного забезпечення, що не пов'язані з конкретними діями користувача, але мають суттєвий вплив на ефективність експлуатації системи. У цьому проєкті було сформульовано такі основні нефункціональні вимоги:

- **Надійність** – система повинна функціонувати стабільно протягом тривалого часу без збоїв.
- **Масштабованість** – можливість обробки зростаючого навантаження без зниження продуктивності.
- **Модульність** – реалізація функціоналу у вигляді окремих модулів або сервісів.
- **Безпека** – захист персональних даних користувачів, захист від несанкціонованого доступу.
- **Портативність** – можливість розгортання на різних середовищах за допомогою Docker.
- **Продуктивність** – забезпечення швидкого відгуку системи на дії користувача.

Урахування цих вимог дозволило досягти високої якості програмного забезпечення, що відповідає як потребам користувачів, так і загальним критеріям надійності, продуктивності та зручності.

РОЗДІЛ 3

КОНЦЕПТУАЛЬНА МОДЕЛЬ ІНФОРМАЦІЙНОЇ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ ІНДИВІДУАЛЬНОЇ ОСВІТНЬОЇ ТРАЄКТОРІЇ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

3.1 Постановка та формалізація задачі

У сучасному цифровому суспільстві питання автоматизації навчального процесу стає надзвичайно важливим для закладів вищої освіти. Одним із ключових елементів академічного управління є ведення індивідуального навчального плану здобувача вищої освіти. Індивідуальний навчальний план є документом, що визначає освітню траєкторію студента, дозволяючи йому самостійно обирати дисципліни відповідно до своїх інтересів, потреб і компетентнісної моделі підготовки. У зв'язку з реформами у сфері освіти, зокрема запровадженням елементів автономії закладів вищої освіти та орієнтації на студентоцентрований підхід, зростає потреба в автоматизованих інформаційних системах, здатних забезпечити ефективне управління цим процесом.

Наразі в багатьох закладах вищої освіти процес формування та супроводу індивідуального навчального плану здебільшого виконується вручну або з використанням фрагментарних електронних засобів, що призводить до ряду проблем: втрати даних, дублювання інформації, труднощів у комунікації між підрозділами, а також неможливості централізованого контролю й аналітики. Тому створення сучасної веборієнтованої інформаційної системи підтримки ІНП є актуальним напрямом дослідження та впровадження.

Метою роботи є розробка інформаційної системи підтримки електронного документообігу в закладі вищої освіти, яка дозволяє формувати, редагувати, зберігати та супроводжувати індивідуальні навчальні плани студентів у цифровому середовищі.

Для досягнення поставленої мети необхідно виконати наступні завдання:

1. Проаналізувати нормативно–правову базу, що регламентує формування індивідуального навчального плану.
2. Дослідити існуючі інформаційні системи, які частково реалізують подібний функціонал.
3. Сформулювати функціональні та нефункціональні вимоги до розроблюваної системи.
4. Обґрунтувати вибір мов програмування, бібліотек, форматів даних, архітектурних підходів.
5. Спроекувати архітектуру системи на основі мікросервісного підходу.
6. Реалізувати окремі модулі, що забезпечують взаємодію з користувачем, базою даних та зовнішніми сервісами.
7. Забезпечити захист персональних даних та контроль доступу на основі ролей.
8. протестувати функціональність програмного забезпечення та провести його оцінювання за визначеними критеріями.

Актуальність теми дослідження зумовлена потребою забезпечення гнучкості освітнього процесу. Університети, які впроваджують цифрові рішення для управління навчальними траєкторіями, демонструють вищий рівень підготовки до освітніх реформ, а також до запитів сучасних здобувачів освіти. Розробка системи, яка б дозволяла ефективно керувати ІНП у режимі онлайн, не лише сприяє підвищенню ефективності внутрішніх процесів, а й покращує загальну якість освітньої послуги.

Об'єктом дослідження є процес управління індивідуальним навчальним планом здобувача вищої освіти. Предметом дослідження є методи та засоби розробки інформаційної системи, що забезпечує цифровий супровід ІНП.

Практична значущість роботи полягає у створенні програмного продукту, який може бути використаний у вищих навчальних закладах для автоматизації роботи з індивідуальними планами, зниження адміністративного навантаження, уникнення помилок та забезпечення прозорості академічного процесу.

3.2 Вихідні дані інформаційної системи

Вихідні дані інформаційної системи мають різні форми подання та є результатом опрацювання вхідної інформації, дій користувачів та алгоритмів логіки системи.

До основних типів вихідних даних належать:

- **Сформований індивідуальний навчальний план здобувача освіти** – документ, який містить обов'язкові та вибіркові дисципліни, розподіл по семестрах, обсяг кредитів ECTS, форми контролю (екзамен, залік), а також додаткову інформацію, визначену освітньою програмою.

ДРУГИЙ КУРС								
Календарний термін навчання з 9 лютого 2026 року до 28 червня 2026 року								
№ з/п	Назви навчальних дисциплін	ЧЕТВЕРТИЙ СЕМЕСТР						
		Тривалість семестру 17 тижнів						
		Кількість кредитів	Кількість годин					Форма семестрового контролю (екзамен, залік)
			аудиторних					
			загальна	лекцій	практичних	лабораторних	самостійна робота і контрольні заходи	
Обов'язкові навчальні дисципліни								
1	Історія української культури	3	90	20	10	0	60	з
2	Бази даних та інформаційні системи	5	150	26	0	24	100	і
3	Комп'ютерні мережі	5	150	26	0	24	100	і
4	Чисельні методи	4	120	20	0	20	80	і
5	Теорія програмування	4	120	20	0	20	80	і
6	Навчальна (обчислювальна) практика	3	90				90	з
Всього		24						
Вибіркові навчальні дисципліни								
7	Програмування на базі технології .net	3	90	16	0	14	60	з
8	Веб-програмування	3	90	16	0	14	60	з
Всього за семестр		30						х

доц.Максимцев Ю.Р.
(підписав та ініціали)

(підпис декана)

доц.Максимцев Ю.Р.
(прізвище та ініціали)

Рисунок 3.1 Сформована сторінка індивідуального навчального плану

- **Дані для викладачів** – перелік студентів, які обрали певні дисципліни, інформація.
- **Експорт у стандартні формати** – сформовані документи можуть бути збережені у форматах PDF, DOCX, XLSX, CSV. Це дозволяє легко обмінюватись

інформацією між структурними підрозділами або надавати її зовнішнім користувачам (наприклад, при акредитації чи перевірках).

– **API-вихід** – для інтеграції з іншими системами закладу інформаційна система надає структуровані вихідні дані через REST API.

3.3 Основні бази даних інформаційної системи

Основу функціонування інформаційної системи супроводу індивідуального навчального плану становлять взаємопов'язані бази даних, що інтегруються між собою через єдину цифрову платформу. До складу цих баз входить спеціальність, курс навчання, групу, академічний статус (активний, у відпустці, відрхований тощо), а також повну історію їхньої освітньої траєкторії, яка включає всі вибрані дисципліни, успішність, а також участь у практиках, стажуваннях і додаткових освітніх заходах.

База навчальних дисциплін містить повний перелік усіх освітніх компонентів, як обов'язкових, так і вибіркових, із зазначенням кількості кредитів ECTS, форм контролю (іспит, залік, курсовий проєкт, практична робота), семестрової структури, анотацій, вимог до попередніх курсів та компетентностей, які набуває здобувач після їхнього успішного опанування. Ця база є ключовою для формування індивідуальних навчальних планів, забезпечуючи гнучкість та адаптивність освітнього процесу.

Центральною складовою системи є база індивідуальних навчальних планів, яка зберігає інформацію про вибрані здобувачем дисципліни, дати їхнього вибору, а також повну історію змін, що дозволяє відстежувати еволюцію освітньої траєкторії та аналізувати динаміку навчального навантаження.

Також система використовує базу викладачів, що містить детальну інформацію про викладацький склад – персональні дані, наукові ступені і звання, спеціалізацію, закріплення за навчальними курсами, а також інформацію про

навантаження та участь у навчально-методичній роботі. Ця база забезпечує ефективне планування ресурсів і контроль за якістю викладання.

Крім того, інформаційна система передбачає збереження адміністративних документів, таких як навчальні плани освітніх програм, внутрішні положення, нормативні акти та інструкції, що регламентують організацію навчального процесу. Всі ці дані постійно оновлюються і підтримуються в актуальному стані для забезпечення відповідності сучасним освітнім вимогам.

3.4 Місце інформаційної підсистеми в структурі інтегрованої платформи забезпечення взаємодії з іншими підсистемами

Запропоноване програмне рішення може бути адаптоване для різних навчальних закладів, враховуючи їх специфіку, й забезпечити цифрову трансформацію документообігу в межах освітнього процесу.

Результатом роботи є вебзастосунок, що забезпечує:

1. Створення і збереження індивідуального навчального плану студента.
2. Зручний інтерфейс для викладача, адміністратора й здобувача освіти.
3. Можливість динамічного вибору дисциплін, формування звітності, перевірки актуальності навантаження.
4. API для інтеграції з іншими електронними системами закладу.
5. Масштабовану та безпечну серверну частину з підтримкою роботи в різних середовищах.

Запропоноване програмне рішення може бути адаптоване для різних навчальних закладів, враховуючи їх специфіку, й забезпечити цифрову трансформацію документообігу в межах освітнього процесу.

РОЗДІЛ 4

АРГУМЕНТАЦІЯ ТА ОБГРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТІВ РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Мова програмування Python, декларативна мова програмування SQL, формат даних JSON, протокол HTTP

З аналізу специфікації системи зрозуміло, що система буде складатися із декількох мікросервісів. Саме для нашого мікресервісу було обрано мову Python адже це нескладна мова програмування високого рівня з простим та зрозумілим для кожного синтаксисом, широко використовувана для вирішення простих задач, наприклад написання скриптів для рутинної роботи. Python підтримує модульність та повторне використання коду, що сприяє швидкому написанню його, а також містить в собі багато парадигм розробки програмного забезпечення включаючи у себе процедурну, об'єктно – орієнтовану, функціональні та аспектно – орієнтовану мову програмування. У контексті цієї інформаційної системи мова Python обрана для реалізації серверної логіки, зокрема – обробки HTTP-запитів, авторизації користувачів, роботи з базою даних, генерації звітів та іншого.

Для зберігання даних, які потрібні для формування індивідуального плану навчання, використовується декларативна мова програмування SQL, призначена для створення, модифікації та запиту до реляційних баз даних. Особливістю SQL є те, що користувач описує, який результат бажає отримати, не вказуючи покроковий алгоритм його досягнення. У межах даної системи SQL використовується для створення структури бази даних, реалізації запитів до таблиць, збереження інформації про студентів, дисципліни, семестри, академічні результати, а також логування змін і запитів. Платформою для реалізації SQL у системі виступає PostgreSQL – об'єктно-реляційна система керування базами даних з відкритим вихідним кодом.

Обмін даними між клієнтом і сервером, а також між окремими модулями системи реалізовано у форматі JSON. JSON – це легкий текстовий формат обміну даними, що підтримує вкладені структури (словники, масиви), є зручним для читання як людиною, так і машиною, та має широку підтримку практично всіма мовами програмування. Формат JSON обрано, як стандарт для серіалізації об’єктів, передачі інформації через API-запити, а також для збереження тимчасових або проміжних налаштувань системи. Наприклад, при запиті на отримання індивідуального навчального плану клієнтська частина надсилає HTTP-запит і отримує відповідь у форматі JSON, що дозволяє зручно обробити інформацію у браузері або мобільному застосунку.

```
{
  "look": "1",
  "at": 2,
  "some": "asdfjkl;",
  "JSON": [
    {
      "data": 1234,
      "arranged": "(and how!)"
    },
    "as an example"
  ]
}
```

Рисунок 4.1 Приклад JSON

Протокол HTTP – це протокол прикладного рівня, який є основою обміну даними в мережі інтернет. Він визначає правила, за якими користувач надсилає запити до сервера, на що сервер відправляє відповідь. HTTP не має станів це значить, що кожен запит є незалежним і не зберігає інформацію про попередні запити. Така архітектура спрощує масштабування та обробку великої кількості одночасних підключень за рахунок принципу «запит-відповідь». HTTP використовується в більшості інформаційних систем, зокрема у веб-додатках, RESTful API, мобільних застосунках, клієнт-серверних архітектурах, а також у мікросервісних системах. У контексті даної кваліфікаційної роботи протокол HTTP виступає базовим інструментом взаємодії між клієнтом, та серверною частиною, що забезпечують супровід індивідуального навчального плану здобувача освіти.

Наприклад, для отримання переліку дисциплін студент надсилає запит GET до відповідного ресурсу, а для оновлення обраного курсу – PUT із новими даними. Для завантаження довідок або звітів система генерує відповідні DOCX – файли, доступні за HTTP – запитом з відповідним рівнем доступу.

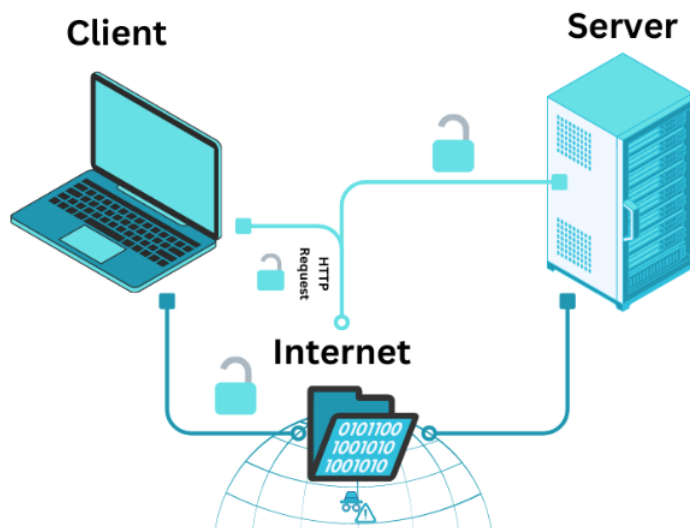


Рисунок 4.2 Приклад запиту та відповіді на GET запит

4.2 Середовище розробки

Для розробки програмного забезпечення було обрано сучасне середовище розробки Visual Studio Code, яке поєднує у собі високу функціональність, гнучкість налаштування, розширюваність за рахунок плагінів та зручність користування. VS Code є безкоштовним редактором коду з відкритим вихідним кодом, підтримуваним компанією Microsoft. Завдяки широкому спектру розширень, інтеграції з системами контролю версій, терміналом, дебагером, а також підтримці великої кількості мов програмування, середовище VS Code має популярність у сфері веброзробки, розробки API, системної інтеграції та автоматизації процесів.

У процесі реалізації інформаційної системи супроводу індивідуального навчального плану здобувача вищої освіти VS Code використовувалося для створення серверного коду на мові Python, написання SQL-запитів, конфігурування Docker-контейнерів, роботи з файлами у форматі JSON, а також тестування API. Однією з ключових переваг середовища є підсвічування синтаксису з помилками,

автодоповнення/авто виправлення коду та миттєвого відображення синтаксичних помилок, що дозволяє значно пришвидшити процес розробки та зменшити кількість помилок у коді.

Для забезпечення ефективної роботи з базою даних PostgreSQL у VS Code було встановлено відповідне розширення – PostgreSQL. Завдяки йому з'явилася можливість напряду з редактора підключатися до бази даних, виконувати запити, переглядати таблиці, перевіряти результати запитів у реальному часі та здійснювати базове адміністрування бази даних. Це дало змогу не лише пришвидшити розробку SQL-структур, а й зручно налагоджувати взаємодію між кодом та базою даних.

Для швидкого розгортання на машинах інших користувачів чи сервері, управління залежностями, а також моделювання мікросервісної архітектури було використано інструмент Docker, підтримку якого також реалізовано у VS Code за допомогою відповідного плагіна. Цей плагін дозволяє напряду в середовищі розробки створювати, запускати, зупиняти та налагоджувати контейнери, а також переглядати їх журнали, структуру та конфігурації. Застосування Docker дозволило стандартизувати середовище виконання, уникнути проблем сумісності залежностей та забезпечити переносимість програмного забезпечення між різними машинами та етапами розробки.

4.3 Вибір бібліотек

Для реалізації інформаційної системи було обрано низку бібліотек і модулів, які забезпечують зручну та ефективну роботу з базами даних, реалізацію вебсерверу, обробку HTTP – запитів, фільтрування та стандартизації даних, а також роботи з файлами та системними ресурсами. Використання перевірених бібліотек з відкритим вихідним кодом дозволяє суттєво скоротити час розробки, зменшити кількість помилок та підвищити стабільність програмного забезпечення.

Бібліотека `psycopg2` використовується для інтеграції з базою даних PostgreSQL. Це одна з найпопулярніших бібліотек для Python, яка дозволяє виконувати SQL – запити, керувати транзакціями, обробляти їх та здійснювати повноцінну взаємодію з реляційною базою даних. Вона забезпечує низький рівень доступу до БД і дозволяє реалізувати як прості, так і складні операції – від вибірок до конфігураційних змін схеми.

Бібліотека `json` – стандартний модуль Python – застосовується для серіалізації та десеріалізації даних у форматі JSON. Це необхідно для обміну інформацією між серверною частиною та клієнтами, а також для зберігання проміжних або кінцевих даних у легкому текстовому форматі, зручному як для машинної обробки, так і для ручного перегляду.

Фреймворк Flask є основою вебсерверної частини системи. Він дозволяє легко реалізовувати маршрути, обробку запитів, роботу з API, шаблонами та статичними файлами. Простота й гнучкість Flask робить його ідеальним вибором для невеликих і середніх проєктів, які не потребують громіздкої архітектури. У межах даної системи він забезпечує отримання HTTP – запитів, обробку даних, генерацію відповідей, а також організацію структури сервісу.

Бібліотека `requests` використовується для виконання HTTP – запитів з боку серверної частини до зовнішніх сервісів. Вона дозволяє реалізувати інтеграцію з API сторонніх систем або отримувати інформацію з хмарних сервісів, що може бути необхідним для підключення до єдиних реєстрів або освітніх платформ.

Модуль `os` застосовується для взаємодії з операційною системою, зокрема – для роботи з файлами, шляхами, середовищем змінних, а також для забезпечення гнучкої конфігурації системи залежно від середовища її запуску.

Додатково, у межах реалізації системи використовуються модулі внутрішньої структури – зокрема `get_xlsx`, який відповідає за формування звітів у форматі Excel на основі індивідуального навчального плану, та `db`, що містить функції для взаємодії з базою даних, інкапсуляцію SQL-запитів і обробку результатів.

4.4 Використання прикладного програмного інтерфейсу

У сучасному програмному забезпеченні прикладні програмні інтерфейси є ключовим інструментом для забезпечення взаємодії між різними програмними компонентами. У контексті розробки інформаційної системи супроводу індивідуального навчального плану API було використано для створення чіткої логіки обміну даними між клієнтською частиною та серверною частиною системи. Такий підхід дозволяє відокремити інтерфейс користувача від внутрішньої логіки обробки запитів і роботи з базою даних, що значно спрощує підтримку та масштабування програмного продукту.

API у проєкті реалізує архітектурний підхід REST, що базується на використанні стандартних методів HTTP – протоколу: GET – для отримання даних, POST – для створення, PUT – для оновлення, DELETE – для видалення. Така архітектура є логічною, гнучкою та зрозумілою як для розробників, так і для зовнішніх систем, які в майбутньому можуть взаємодіяти з інформаційною системою через API.

Обмін даними між клієнтською частиною та сервером здійснюється у форматі JSON, що забезпечує зручність передачі структурованих даних та сумісність з іншими мовами програмування й платформами. Усі відповіді API мають уніфіковану структуру, що дозволяє клієнтській частині легко обробляти отримані результати. Наприклад, при зверненні до ресурсу навчального плану у відповідь повертається структурований об'єкт, що містить список освітніх компонент, кредити, форми контролю та семестровий розподіл.

У межах реалізованої системи прикладний програмний інтерфейс виконує функції не лише передачі даних, а й їхньої первинної обробки. Перед виконанням запитів проводиться валідація вхідних параметрів, перевірка прав доступу користувача, обробка винятків та формування коректної відповіді. Такий підхід дозволяє забезпечити надійність системи, уникнути помилок та зберегти цілісність даних.

API забезпечує можливість інтеграції інформаційної системи з іншими сервісами, наприклад, для обміну даними з централізованими базами, аналітичними платформами чи внутрішніми порталами закладу вищої освіти. У разі розширення функціональності системи або створення мобільного застосунку, існуючий API може бути повторно використаний без необхідності зміни серверної частини, що також сприяє повторному використанню коду та зниженню витрат на подальший розвиток.

4.5 Docker

У процесі розробки інформаційної системи було прийнято рішення про використання технології контейнеризації, зокрема платформи Docker. Docker – це програмне середовище з відкритим вихідним кодом, яке дозволяє розгортати, масштабувати та запускати застосунки в ізольованому середовищі, які називаються контейнерами. На відміну від віртуальних машин, контейнери не потребують окремої операційної системи, що дозволяє зменшити обсяг споживаних ресурсів та прискорити запуск.

Використання Docker у рамках даного проєкту забезпечує низку важливих переваг. По-перше, це портативність усі залежності, бібліотеками та конфігурації можна запустити у будь-якому середовищі, що підтримує Docker, без потреби попереднього налаштування. Це істотно спрощує розгортання інформаційної системи на різних серверах, включно з локальними машинами розробників, тестовими середовищами та серверами.

По-друге, Docker сприяє структурованості розробки. У проєкті було створено окремі контейнери для різних компонентів системи, зокрема для серверної логіки, бази даних PostgreSQL. Такий підхід наближає архітектуру системи до принципів мікросервісної організації, у якій кожен сервіс функціонує незалежно, що покращує масштабованість та спрощує супровід.

Окрему увагу слід приділити тому, що Docker підтримує інструменти автоматизації, такі як Docker Compose, які дають змогу визначити конфігурацію багатьох контейнерів у єдиному YAML-файлі. Це дозволяє швидко розгортати всю систему за допомогою однієї команди, що важливо при спільній роботі команди над проєктом або під час перенесення рішення на інший сервер.

Контейнеризація також сприяє підвищенню безпеки, оскільки кожен контейнер функціонує в ізольованому середовищі, що обмежує вплив можливих збоїв або вразливостей одного модуля на інші частини системи. Крім того, легко реалізується контроль версій контейнерів, що дозволяє зберігати попередні робочі конфігурації та відновлювати систему у разі непередбачених збоїв.

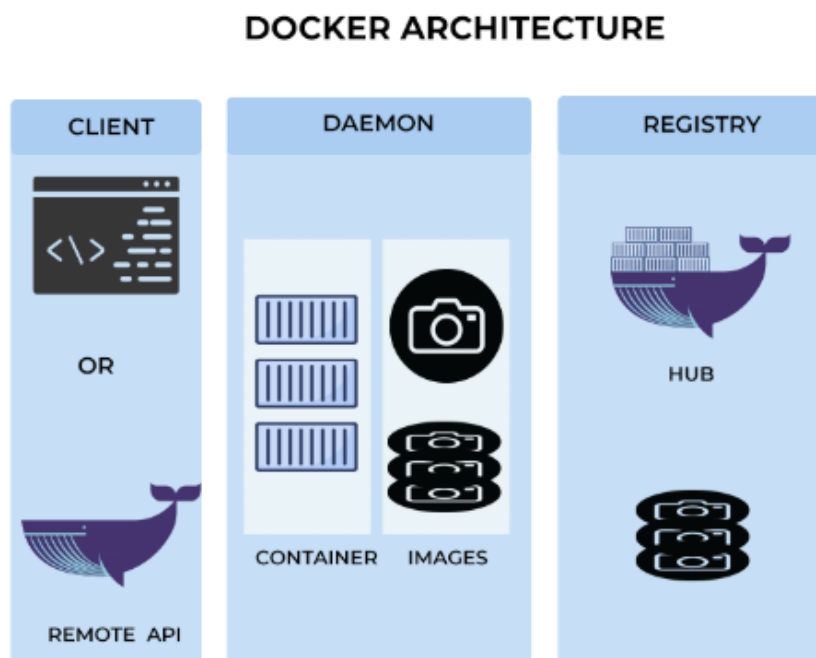


Рисунок 4.1 структура Docker

4.6 Мікросервісна архітектура

У процесі розробки інформаційної системи було застосовано підхід мікросервісної архітектури, який на сьогодні є одним з найбільш сучасних та ефективних для побудови гнучких, масштабованих програмних систем.

Мікросервісна архітектура передбачає розділення всієї системи на окремі сервіси (модулі), кожен з яких виконує чітко визначену функцію та працює автономно.

На відміну від монолітної архітектури, де всі компоненти системи розгортаються як єдине ціле, мікросервісний підхід дозволяє розробляти, тестувати, оновлювати та масштабувати окремі модулі незалежно один від одного. Це суттєво знижує ризики виникнення критичних збоїв, оскільки відмова одного сервісу не впливає на функціонування інших. Крім того, цей підхід забезпечує можливість використання різних технологій і мов програмування в межах однієї системи, що дозволяє обирати найбільш ефективні інструменти для кожного конкретного завдання.

У даному проєкті мікросервісна архітектура реалізована через поділ логіки системи на окремі сервіси, що працюють у контейнерах. Наприклад, обробка запитів користувача, збереження даних, формування індивідуального навчального плану, взаємодія з базою даних – усе це винесено в окремі контейнери, які можуть масштабуватися незалежно. Комунікація між сервісами відбувається через внутрішні мережеві протоколи з використанням REST API.

Однією з ключових переваг такої архітектури є її масштабованість. У разі зростання навантаження на окремих сервіс (наприклад, на модуль генерації планів), можливо розгорнути додаткові його екземпляри, не змінюючи інші частини системи. Також мікросервісна структура полегшує оновлення системи, оскільки нова версія окремого сервісу може бути впроваджена без зупинки всієї системи.

Крім того, застосування мікросервісної архітектури є доцільним з точки зору організації командної роботи: різні розробники або команди можуть працювати паралельно над різними мікросервісами, що підвищує ефективність процесу розробки.

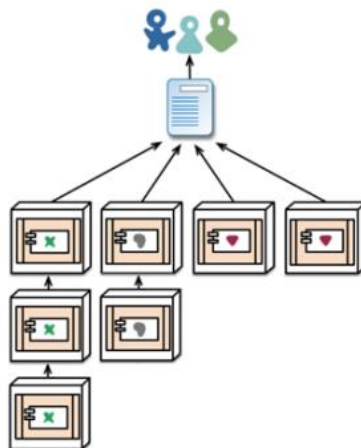


Рисунок 4.5 Можливість горизонтального та вертикального масштабування мікросервісної архітектури

РОЗДІЛ 5

ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ СЕПРОВОДЖЕННЯ ІНДИВІДУАЛЬНОГО НАВЧАЛЬНОГО ПЛАНУ

5.1 Розроблення баз даних

Розробка бази даних є одним із критичних етапів проектування інформаційної системи. Тому основною метою стало створення чіткої та логічно розширюваною бази даних, без повторів на рівні мікросервісів, яка могла б містити в собі всі дані про успішність здобувача вищої освіти, вибіркові компоненти зібрані по кафедрам та індивідуально обрані предмети здобувачем.

Нами було виокремлено такі сутності:

Сутність ***Студент*** містить лише один атрибут – це ідентифікатор здобувача освіти, він є унікальним, надходить із зовнішнього сервісу, який обслуговує ЄДЕБО, на підставі обраної вступником спеціальності під час зарахування до закладу вищої освіти.

Атрибути:

– **student_id** – унікальний ідентифікатор, наданий сервісом, який обслуговує ЄДЕБО.

Особливості:

- Не зберігає додаткових персональних даних (ім'я, прізвище, тощо).
- Використовується, як зовнішній ключ для зв'язку із іншими сутностями (індивідуальний навчальний план, група, обрана спеціальність, факультет, тощо).

Сутність ***Предмет*** відображає дисципліну в початковій програмі. Основним призначенням є збереження базової інформації про предмет та забезпечення зв'язків з іншими сутностями системи (індивідуальний навчальний план, результат навчання).

Атрибути:

– **id_subject** – ідентифікатор предмета (не унікальний), який використовується для ідентифікації предмета при зверненні до сервісу, який обслуговує навчальні плани, та нашого модуля, який обслуговує індивідуальний навчальний план, а саме до вибіркового компонента у ньому та встановленні зв'язків із іншими сутностями (оцінки).

– **subject_name** – назва предмету, яка використовується у парі з **id_subject** для відображення дисципліни в користувацьких програмах, звітах, навчальних планах, додатках до дипломів.

– **credits** – кількість кредитів ЄКТС.

– **semester** – семестр викладання предмета.

Особливості:

– Один предмет може бути пов'язаний із багатьма записами в індивідуальних навчальних планах.

– Один предмет може мати багато результатів навчання.

– Сутність **Кафедра** використовується для відображення структурного поділу в закладах вищої освіти, має зв'язок із вибічковими дисциплінами.

Атрибути:

– **id** – унікальний ідентифікатор кафедри.

– **name_cathedral** – назва кафедри.

Особливості:

– Кафедра має зв'язок із вибічковими предметами та індивідуальними планами через зовнішні ключі.

Сутність **Індивідуальний навчальний план** є основною, для якої і виокремлювалися всі сутності. Відображає перелік дисциплін, які обрав здобувач.

Атрибути:

– **id_student** – зовнішній ключ студента.

– **id_subject** – ідентифікатор предмета.

– **cathedral** – зовнішній ключ на таблицю кафедр.

– **name_subject** – назва предмета (збережена на момент вибору).

Особливості:

- Первинний ключ складається з пари `id_student` і `id_subject`.
- Забезпечує унікальність предмету в межах навчального плану здобувача.
- Служить точкою з'єднання між студентом, предметом та кафедрою.
- Сутність ***Рейтинг*** використовується для збереження результатів навчання студентів, повертається, при запиті, на користувацький інтерфейс.

Атрибути:

- **id_student** – зовнішній ключ студента.
- **id_subject** – ідентифікатор предмета.
- **rating** – числове значення оцінки, в межах 0–100 (може бути NULL – без оцінки).
- **last_update** – дата останнього оновлення оцінки.

Особливості:

- Первинний ключ складається з пари `id_student` і `id_subject`.
- Забезпечує зберігання оцінок для предметів, які відсутні у базі даних, тобто обов'язкових компонентів.

Сутність ***Вибіркові предмети*** зберігає дані, які здобувач може обрати в межах індивідуальної траєкторії навчання.

Атрибути:

- **name_subject** – назва вибіркового компонента.
- **kafedra_id** – унікальний ідентифікатор кафедри.
- **semester** – семестр у якому викладається предмет.

Особливості:

- Отримується із зовнішніх джерел, через парсинг даних.
- Використовується для побудов індивідуального плану, через інтерфейс користувача.

На основі вище вказаних сутностей було сформовано базу даних, яка має вигляд на рис 5.1.

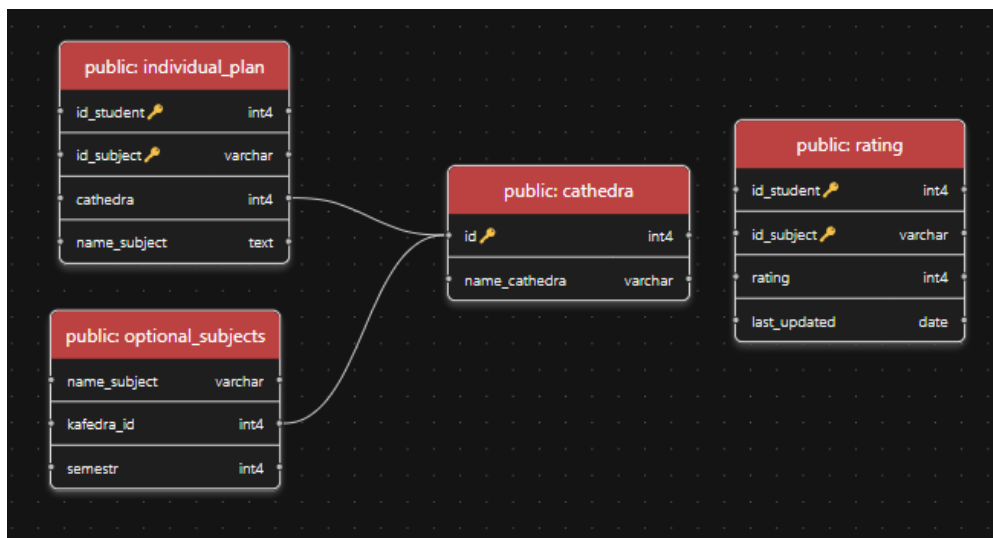


Рисунок 5.1 Схема бази даних

Дані до бази даних надходять через запити ззовні, тобто через API, або відбувається збір даних через внутрішні модулі мікросервісу.

В майбутньому можливе впровадження нормалізації бази даних, приведення до третьої нормальної форми, що дасть змогу уникнути надлишкового дублювання даних, підвищити цілісність даних.

Описана вище база даних є надійним фундаментом для подальшої розробки функціоналу та забезпечує гнучкість і масштабованість системи.

5.2 Модуль обслуговування індивідуального навчального плану здобувача освіти

Модуль обслуговування індивідуального навчального плану здобувача освіти є ключовим компонентом інформаційної системи підтримки освітнього процесу в закладі вищої освіти. Його основною метою є забезпечення гнучкості у навчанні студента, що дозволяє реалізувати право на вибір навчальних дисциплін відповідно до особистих освітніх потреб та компетентностей у навчальній програмі. Модуль надає інтерфейс для вибору дисциплін із запропонованого

переліку, що відповідає навчальній програмі та нормативним вимогам. В ньому реалізовано механізм перевірки коректності навчального плану, який забезпечує відповідність кредитам, уникнення конфліктів за розкладом та дотримання послідовності вивчення дисциплін.

Здобувач має можливість редагувати свій навчальний план протягом встановленого періоду, а працівники деканату та адміністрація можуть переглядати план, вносити коментарі та затверджувати його. Для користувачів реалізована система повідомлень, що інформує про статус затвердження або необхідність доопрацювання. Усі зміни в навчальному плані зберігаються із позначенням часу та користувача, який вніс корективи, що забезпечує логування і перегляду змін у разі надзвичайних ситуацій.

Для підвищення ефективності роботи модуля забезпечується інтеграція з іншими інформаційними системами закладу та зовнішніми сервісами. Зокрема, модуль обмінюється даними з Єдиною державною електронною базою з питань освіти (ЄДЕБО), через мікросервіс-проксі, для актуалізації інформації про студентів та освітні програми, взаємодіє з модулем формування додатку до диплома, що дозволяє автоматично формувати офіційні документи на основі затвердженого навчального плану, а також синхронізується з централізованою платформою для вибору освітніх компонентів.

OK30	Класифікація та розпізнавання образів / Classification of Pattern Recognition	4	94
OK31	Розподілені системи та паралельні обчислення / Distributed Systems and Parallel Calculation	4	93
OK32	Технології хмарних обчислень / Cloud Computing Technologies	4	95
OK33	Моделювання систем / Modeling of Systems	4	95
OK34	Проектування інформаційних систем / Information Systems Design	4	92
OK35	Методи та системи штучного інтелекту / Methods and Systems of Artificial Intellect	3	95
OK36	Захист інформації / Data Protection	4	75
BK01	Методи та засоби комп'ютерних інформаційних технологій / Methods and Means of Computer Information Technologies	3	95
BK04	Веб-технології та веб-дизайн / Web technologies and Web design	4	100
BK07	Програмування на базі технології .net / Technology – Based .net Programming	4	85
BK10	Веб-програмування / Web programming	4	90
BK13	Тривимірна та анімаційна графіка / Three-dimensional and Animation Graphics	3	94
BK16	Стохастичне моделювання / Stochastic Modeling	4	82
BK19	Сховища та простори даних / Repositories and Data Spaces	4	91
BK22	Крос-платформенне програмування / Cross-platform Programming	4	91
BK25	Інтерфейси користувача та системні інтерфейси / User Interfaces and System Interfaces	4	90
BK28	Основи наукових досліджень / Basics of Scientific Research	3	78
BK31	Менеджмент колективних проектів / Management of Collective Projects	4	90
BK34	Нейронні мережі / Neural Networks	4	80
BK38	Сучасна теорія управління / Modern Theory of Management	4	95
BK40	Інтернет речей / Internet of Things	4	96
BK43	Програмування мобільних пристроїв / Programming of Mobile Devices	4	90
BK47	Основи кібербезпеки / Fundamentals of Cyber Security	3	75
OK37	Курсова робота з дисциплін кафедри інформаційних технологій та моделювання / Academic year paper in disciplines of the Department of Information Technology and Modeling	3	96
OK38	Обчислювальна практика / Computational Practice	3	90
OK39	Технологічна практика / Technological Practice	3	93
OK40	Виробнича практика / Training Practice	6	100
OK41	Захист кваліфікаційної роботи «Створення та оптимізація моделі нейронної мережі для автоматичного розпізнавання дорожніх знаків на зображеннях» / Defence of Bachelor's thesis «Creating and optimizing a neural network model for automatic recognition of road signs in images»	6	98
Загальна кількість кредитів Європейської кредитної трансферно- накопичувальної системи / Total number of ECTS credits		240	

Рисунок 5.2 Приклад фрагменту згенерованого додатку до диплома

Архітектура модуля побудована за принципами мікросервісного підходу, що забезпечує масштабованість, зручність підтримки та оперативне впровадження змін. Основними компонентами системи є клієнтський інтерфейс, який забезпечує зручну взаємодію користувачів із системою, сервіс логіки навчального плану, що відповідає за правила формування та перевірки, сервіс інтеграції для обміну даними з зовнішніми системами, а також сервіс повідомлень і сповіщень, який автоматично інформує користувачів про статуси та зміни.

З метою захисту персональних даних та забезпечення конфіденційності в модулі впроваджено систему контролю доступу з визначенням ролей користувачів, адміністратор, гарант освітньої програми та викладачі. Для аутентифікації та авторизації застосовуються сучасні протоколи, а комунікації між компонентами захищені шифруванням.

Технічно реалізація модуля базується на сучасних технологіях, які забезпечують надійність, продуктивність і масштабованість. Серверна частина розробляється з використанням Flask.

5.3 Прикладний програмний інтерфейс

Прикладний програмний інтерфейс (API) створений для забезпечення зручного та ефективного обміну даними між фронтендом, модулями бази даних та зовнішніми сервісами. Його основною метою є стандартизація взаємодії між клієнтом та серверною частиною модуля формування індивідуального навчального плану.

Прикладний програмний інтерфейс реалізовано із використанням REST – архітектури, що дозволяє використовувати стандартні HTTP методи (GET, POST, PUT, DELETE), для виконання основних CRUD операцій. В поєднанні зі Swagger досягається максимальна читабельність документації та забезпечується очікувана структура відповідей.

Основні ендпоїнти API:

PUT /optional_subject/<id_student> – додає вибірккові дисципліни до індивідуального навчального плану здобувача. Приймається JSON–запит зі списком обраних дисциплін, що містить ідентифікатор, назву предмета та кафедру, з якої обраний предмет.

GET /subjects/<id_student> – повертає поточний індивідуальний план здобувача, включаючи перелік дисциплін, оцінки, назву предмета та семестр впровадження предмета.

GET /subjects/optional/<speciality> – повертає перелік дисциплін, які здобувач може обрати на новий навчальний рік. Містить в собі дві дисципліни, які пропонуються гарантом навчальної програми та вільний вибір предмета серед інших кафедр закладу освіти.

GET /diploma/<id_student> – повертає дані, для формування додатку до диплома в ньому містяться оцінки, ідентифікатор предмета в початковій програмі, кількість кредитів. Для формування JSON відбувається запит до сервісу, який обслуговує навчальні плани, у відповідь на запит отримуємо обов’язкові

компоненти для спеціальності, на які навчається здобувач освіти, вибіркові компоненти отримуються із внутрішньої бази даних нашого сервісу.

GET /optional_subjects – повертає перелік дисциплін, які здобувач може обрати на новий навчальний рік. Містить в собі дисципліни, із загально університетського вибору, які доступні для кожного студента не залежно від кафедри.

GET /individual_plan – повертає готовий до підписання індивідуальний навчальний план, який формується за допомогою запитів, які описані вище.

В усіх маршрутах використовують JSON формат подання даних, у відповідь API завжди повертає стандартизовану структуру даних: статус операції, код відповіді, коротка інформація у разі невдачі обробки запиту.

Також важливо додати, що всі запити проходять валідацію на рівні серверної логіки, та в майбутньому є можливість додати аутентифікацію користувача через токени (JWT або OAuth2).

5.4 Інтеграція з іншими мікросервісами інформаційної системи

Інтеграція з іншими мікросервісами інформаційної системи є невід’ємною складовою побудови сучасної масштабованої платформи для підтримки освітнього процесу у закладі вищої освіти. Завдяки використанню мікросервісної архітектури кожен сервіс відповідає за чітко визначену функціональність, що підвищує гнучкість, надійність і зручність розробки, супроводу та масштабування системи. Взаємодія між цими сервісами здійснюється через стандартизовані протоколи обміну даними, що забезпечує узгодженість і цілісність інформації в межах всієї платформи.

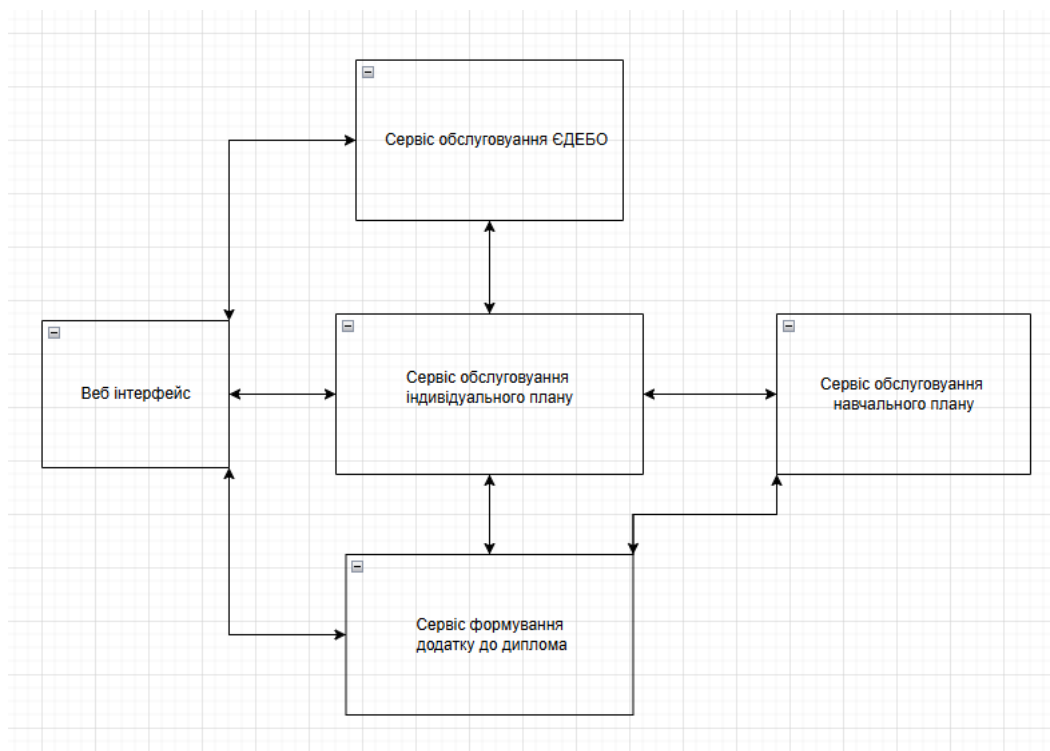


Рисунок 5.3 Взаємодія компонентів інформаційної системи

Модуль обслуговування індивідуального навчального плану, як один із центральних компонентів системи, інтегрується з низкою інших мікросервісів, кожен з яких відповідає за специфічні аспекти освітнього процесу. Насамперед, важливою є інтеграція з веб-інтерфейсом, який забезпечує користувачам зручний доступ до функціоналу модуля. Через веб-інтерфейс користувачі – викладачі та адміністративний персонал – можуть формувати, редагувати, переглядати і затверджувати індивідуальні навчальні плани.

Для підтримки актуальності і достовірності даних модуль інтегрується з сервісом, який обслуговує ЄДЕБО. Цей сервіс надає актуальну інформацію про студентів, освітні програми, нормативні документи та інші дані, необхідні для формування індивідуальних навчальних планів відповідно до державних стандартів. Обмін даними між модулем та сервісом ЄДЕБО здійснюється через захищені API, що дозволяє автоматично оновлювати інформацію про студента, його академічний статус та навчальні програми без участі користувача, мінімізуючи ризики помилок та дублювання даних.

Іншим важливим напрямом інтеграції є взаємодія з сервісом навчальних планів, який відповідає за зберігання і підтримку базових освітніх програм, стандартів навчання та структур дисциплін. Цей сервіс надає модулю обслуговування ІНП необхідні шаблони і нормативні рамки, за якими студенти можуть формувати свої індивідуальні плани. Інтеграція забезпечує не лише доступ до актуальних версій навчальних планів, а й можливість оперативного оновлення та внесення змін у разі змін нормативної бази чи внутрішніх правил закладу. Це дозволяє модулю динамічно реагувати на зміни та підтримувати відповідність освітньої траєкторії здобувача вимогам.

Також модуль тісно інтегрується із сервісом генерації додатку до диплома, що виконує автоматичне формування офіційних документів на основі затвердженого індивідуального навчального плану та результатів навчання студента. Завдяки цій інтеграції процес оформлення додатку до диплома суттєво спрощується і автоматизується: всі необхідні дані передаються в сервіс генерації у структурованому вигляді, де формуються документи, готові до підпису і видачі. Такий підхід зменшує людський фактор, підвищує точність і скорочує час підготовки документів.

Для забезпечення надійної та ефективної взаємодії між цими сервісами застосовується комплекс технологічних рішень. Основним способом комунікації є використання RESTful API, що гарантує простоту, стандартизованість і сумісність у обміні даними.

Уніфікація форматів даних є вирішальним аспектом. Для обміну інформацією застосовуються стандартизовані формати, зокрема JSON. Це дозволяє забезпечити простоту обробки даних і сумісність між різними мовами програмування та технологіями, що використовуються у розробці мікросервісів.

Особливу увагу приділено питанням безпеки інтеграції. Для захисту від несанкціонованого доступу застосовується багаторівнева система автентифікації і авторизації, включно з використанням токенів доступу (зокрема JWT) та протоколів OAuth 2.0. Всі дані передаються через захищені канали зв'язку (HTTPS), що запобігає можливості їх перехоплення або модифікації. Крім того,

реалізовано механізми аудиту та логування взаємодії між сервісами, що дозволяє відстежувати та аналізувати всі операції, виявляти і оперативно усувати потенційні загрози.

Інтеграція також забезпечує централізований моніторинг і трасування запитів, що суттєво спрощує підтримку системи, виявлення і виправлення помилок, а також дозволяє оцінювати продуктивність окремих сервісів і системи загалом.

ВИСНОВКИ

У результаті проведеного дослідження було розроблено інформаційну систему підтримки електронного документообігу закладу вищої освіти із супроводженням індивідуального навчального плану здобувача. Метою роботи було створення програмного рішення, яке забезпечує ефективне управління навчальними планами, інтеграцію з існуючими інформаційними системами навчального закладу, а також полегшує взаємодію між здобувачами освіти, викладачами і адміністративним персоналом. Поставлені завдання з проектування зручного користувацького інтерфейсу, формування динамічного вибору дисциплін, а також створення безпечної та масштабованої серверної частини були успішно виконані.

Отримані результати мають практичне значення для освітніх закладів, оскільки автоматизація процесів супроводу індивідуальних навчальних планів дозволяє підвищити якість управління освітнім процесом. Запропонована система значно скорочує час на обробку та погодження навчальних планів, знижує ризик помилок і втрати даних, що особливо важливо в умовах швидких змін освітніх програм і нормативних вимог. Впровадження єдиної цифрової платформи забезпечує узгодженість інформації, відкриває можливості для аналітики й формування звітності, що допомагає керівництву навчального закладу приймати вірні рішення, що до управління ЗВО.

Архітектура системи, заснована на мікросервісному підході, дозволяє гнучко масштабувати рішення та легко адаптувати його під специфіку різних навчальних закладів, враховуючи їхні організаційні особливості та потреби. Використання сучасних технологій захисту інформації гарантує конфіденційність персональних даних і стабільну роботу системи у різних середовищах експлуатації. Крім того, наявність відкритого API забезпечує інтеграцію із зовнішніми системами, що є важливим фактором для побудови єдиного цифрового простору закладу.

Значною перевагою розробленого рішення є зручний інтерфейс, розроблений із урахуванням специфіки роботи різних користувачів – від здобувачів освіти, які

можуть самостійно формувати індивідуальний навчальний план, до викладачів і адміністрації, які здійснюють контроль і погодження. Такий підхід підвищує ефективність комунікації та знижує навантаження на кадрові ресурси.

Перспективними напрямками подальших досліджень є впровадження інтелектуальних алгоритмів для рекомендацій вибору дисциплін з урахуванням інтересів і академічних результатів студентів, розробка механізмів автоматичного аналізу відповідності навчальних планів стандартам та нормативам, а також інтеграція із системами дистанційного навчання і електронними бібліотеками. Також актуальним є розвиток модулів для збору й обробки статистичних даних, що дозволить удосконалювати освітній процес на основі аналітики.

Отже, виконана робота робить вагомий внесок у цифрову трансформацію систем управління вищою освітою, сприяє підвищенню прозорості, ефективності та адаптивності освітнього процесу. Запропоноване програмне рішення має потенціал до масштабування й адаптації, що відкриває широкі можливості для впровадження в різних навчальних закладах і подальшого розвитку сучасних освітніх технологій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про електронні документи та електронний документообіг : Закон України від 22.05.2003 № 851-IV : станом на 1 січ. 2024 р. URL: <https://zakon.rada.gov.ua/laws/show/851-15#Text> (дата звернення: 02.04.2025).
2. Про електронний цифровий підпис : Закон України від 22.05.2003 № 852-IV : станом на 7 листопад 2018 р. URL: <https://zakon.rada.gov.ua/laws/show/852-15#Text> (дата звернення: 03.04.2025).
3. Цивільний кодекс України. (2003). Статті 205, 207. Закон України від 16 січня 2003 року № 435-IV. URL: <https://zakon.rada.gov.ua/laws/show/435-15#Text> (дата звернення: 04.04.2025).
4. Закон України «Про інформацію». (1992). Закон України від 02 жовтня 1992 року № 2657-XII. URL: <https://zakon.rada.gov.ua/laws/show/2657-12#Text> (дата звернення: 04.04.2025).
5. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах». (1994). Закон України від 05 липня 1994 року № 80/94-ВР. URL: <https://zakon.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80#Text> (дата звернення: 04.04.2025).
6. ДСТУ 4163:2020 «Уніфікована система організаційно-розпорядчої документації. Вимоги до оформлення документів». Національний стандарт України, 2020. URL: <https://zakon.rada.gov.ua/rada/show/v0144774-20#Text> (дата звернення: 05.04.2025).
7. Закон України «Про Національний архівний фонд та архівні установи». (1998). Закон України від 19 листопада 1998 року № 544/98-ВР. URL: <https://zakon.rada.gov.ua/laws/show/3814-12#Text> (дата звернення: 05.04.2025).
8. Закон України «Про захист персональних даних». (2010). Закон України від 01 червня 2010 року № 2297-VI. URL: <https://zakon.rada.gov.ua/laws/show/2297-17#Text> (дата звернення: 06.04.2025).

9. Кабінет Міністрів України. (2011, 13 липня). Постанова № 752. Київ. URL: <https://zakon.rada.gov.ua/laws/show/752-2011-%D0%BF#Text> (дата звернення: 07.04.2025).

10. Круковський М.Ю. Рішення електронного документообігу / М. Ю. Круковський. – К. : Азимут–Україна, 2006. – 112 с. (дата звернення: 05.05.2025)

11. Грейнер Г. А., Римар С. І. Автоматизована система "Електронний деканат" : thesis. 2018. URL: <http://repository.kpi.kharkov.ua/handle/KhPI-Press/46111> (дата звернення: 06.05.2025).

12. Мова програмування Python для інженерів і науковців: навчальний посібник. Ivano–Frankivsk, Ukraine : ІФНТУНГ, 2019. 275 с. (дата звернення: 07.05.2025).

13. Про затвердження Вимог з ядерної та радіаційної безпеки до інформаційних та керуючих систем, важливих для безпеки атомних станцій : Наказ Держ. інспекції ядер. регулювання України від 22.07.2015 № 140 : станом на 9 січ. 2020 р. URL: <https://zakon.rada.gov.ua/laws/show/z0954-15#Text> (дата звернення: 08.05.2025).

14. Про затвердження Порядку взаємодії органів виконавчої влади з питань захисту державних інформаційних ресурсів в інформаційних та телекомунікаційних системах : Постанова Каб. Міністрів України від 16.11.2002 № 1772 : станом на 7 верес. 2022 р. URL: <https://zakon.rada.gov.ua/laws/show/1772-2002-п#Text> (дата звернення: 09.05.2025).

15. Про Положення про технічний захист інформації в Україні : Указ Президента України від 27.09.1999 № 1229/99 : станом на 4 трав. 2008 р. URL: <https://zakon.rada.gov.ua/laws/show/1229/99#Text> (дата звернення: 09.05.2025).

16. Про освіту: Закон України від 05.09.2017 р. №2145-VIII. Дата оновлення: 27.03.2025. URL: <https://zakon.rada.gov.ua/laws/show/2145-19#Text> (дата звернення: 08.05.2025).

17. Про вищу освіту: Закон України від 01.07.2014 р. №1556-VII. Дата оновлення: 19.12.2024. URL: <https://zakon.rada.gov.ua/laws/show/1556-18#Text> (дата звернення: 08.05.2025).

18. Про внесення змін до деяких законів України щодо розвитку індивідуальних освітніх траєкторій та вдосконалення освітнього процесу: Закон України від 23.04.2024 р. №3642-IX. URL: <https://zakon.rada.gov.ua/laws/show/3642-20#Text> (дата звернення: 08.05.2025).

19. Flask Documentation. Flask 3.1.1 documentation. URL: <https://flask.palletsprojects.com/en/3.1.x/> (дата звернення: 15.05.2025).

20. Requests: HTTP for Humans. Документація Requests. URL: <https://requests.readthedocs.io/en/latest/> (дата звернення: 16.05.2025).