

Міністерство освіти і науки України

**Луцький національний технічний університет
Факультет цифрових, освітніх та соціальних технологій
Кафедра цифрових освітніх технологій**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»**

**РОЗРОБКА ТА ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ
СИСТЕМИ З ВИКОРИСТАННЯМ ЯДРА LINUX ДЛЯ
ЗБЕРІГАННЯ НАВЧАЛЬНИХ ДАНИХ ITSTEP
ACADEMY**

спеціальність 015.39 Професійна освіта (Цифрові технології)
освітня програма Професійна освіта (комп'ютерні технології)

Виконав: здобувач вищої освіти
групи ПОМ-21
Курінний Яків Миколайович

(підпис)

Керівник:
к.пед.н., доцент
Саварин Павло Вікторович

(підпис)

Кваліфікаційну роботу
допущено до захисту
«___» _____ 2025 р.
д.пед.н., професор
гарант освітньої програми:
Гулай Ольга Іванівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет цифрових, освітніх та соціальних
технологій Кафедра цифрових освітніх технологій
Ступінь вищої освіти: магістр
Галузь знань: 01 Освіта/Педагогіка
Спеціальність: 015.39 Професійна освіта (Цифрові технології)
Освітня програма: Професійна освіта (комп'ютерні технології)

ЗАТВЕРДЖУЮ
Завідувач кафедри
цифрових освітніх технологій
_____ В. Кабак
« ____ » _____ 2025 р.

З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Курінному Якову Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: **Розробка та дослідження інформаційної системи з використанням ядра Linux для зберігання навчальних даних ITSTEP Academy**

керівник роботи: Саварин Павло Вікторович

затверджені наказом закладу вищої освіти від «06» лютого 2025 р. № 70/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи:
«05» грудня 2025 р.

Вихідні дані до роботи Нормативні документи щодо якості освіти, науково-методична література, вимоги проведення педагогічного експерименту

3. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Проведення детального аналізу літератури та мережесих інформаційних ресурсів за темою наукової роботи; аналіз цифрових інструментів створення веб-датків, постановка педагогічного експерименту; методи та способи впровадження та застосування в процесі діяльності педагога.

4. Перелік графічного матеріалу: 13 рисунків, 9 таблиці

5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв

6. Дата видачі завдання «06» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи магістра	Строк виконання етапів роботи	Примітка
1	<i>Провести огляд літературних джерел по темі кваліфікаційної роботи магістра</i>	до 30.08.25	
2	<i>Провести аналіз загальної проблеми і вибір напрямків дослідження</i>	до 09.09.25.	
3	<i>Розробити функціональну схему роботи програмного продукту</i>	до 17.09.25.	
4	<i>Описати засоби розробки об'єкта проектування</i>	до 30.09.25.	
5	<i>Описати роботу об'єкта проектування</i>	до 16.10.25	
6	<i>Розробити методикку для проведення експерименту</i>	до 23.10.25	
7	<i>Провести аналіз результатів експерименту</i>	до 12.11.25	
8	<i>Оцінка отриманих даних та розробка рекомендацій впровадження гейміфікації у навчальному процесі</i>	до 21.11.25	
9	<i>Подання завершеного варіанту магістерської кваліфікаційної роботи на розгляд кафедри</i>	до 05.12.25	

Здобувач вищої освіти

_____ Курінний Я.М.
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ Саварин П.В.
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Курінний Я.М. «Розробка та дослідження інформаційної системи з використанням ядра Linux для зберігання навчальних даних ITSTEP Academy». Рукопис.

Кваліфікаційна робота магістра зі спеціальності 015.39 Професійна освіта (цифрові технології). Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота магістра складається зі вступу, чотирьох розділів, висновків, переліку використаної літератури.

У першому розділі представлено огляд літературних джерел за темою дослідження, визначено основні теоретичні засади інтерактивного навчання, проаналізовано сучасні освітні платформи та сформульовано загальну проблему й напрям подальших досліджень.

У другому розділі описано постановку задачі розробки інформаційної системи з використанням ядра linux, наведено обґрунтування вибору технологій і архітектури, охарактеризовано особливості платформи.

У третьому розділі подано методику дослідження ефективності розробленої інформаційної системи зберігання навчальних даних, описано етапи експериментального тестування, умови проведення випробувань у середовищі, наближеному до реальної експлуатації, а також підходи до оцінювання якості системи за критеріями міжнародного стандарту ISO 9126.

У четвертому розділі наведено результати експериментального дослідження, подано кількісний і якісний аналіз отриманих даних, оцінено ефективність використання інтерактивної платформи.

Ключові слова: *інформаційна система, навчальні дані, якість програмного забезпечення, стандарт ISO 9126, тестування системи, освітня платформа, Linux-сервер, веб-застосунок.*

ANNOTATION

Kurinyi Y.M. “Development and Research of an Information System Using the Linux Kernel for Storing Educational Data of ITSTEP Academy.”
Manuscript.

Master’s qualification thesis in specialty 015.39 Professional Education (Digital Technologies). Lutsk National Technical University. Lutsk, 2025.

The master’s qualification thesis consists of an introduction, four chapters, conclusions, and a list of references.

The first chapter presents a review of literary sources on the research topic, defines the main theoretical foundations of interactive learning, analyzes modern educational platforms, and formulates the general problem and directions for further research.

The second chapter describes the problem statement for the development of an information system using the Linux kernel, provides justification for the selection of technologies and architecture, and characterizes the main features of the platform.

The third chapter outlines the methodology for studying the effectiveness of the developed information system for storing educational data, describes the stages of experimental testing, the conditions of conducting trials in an environment close to real operation, and the approaches to evaluating system quality according to the criteria of the international ISO 9126 standard.

The fourth chapter presents the results of the experimental study, provides a quantitative and qualitative analysis of the obtained data, and evaluates the effectiveness of using the interactive platform.

Keywords: information system, educational data, software quality, ISO 9126 standard, system testing, educational platform, Linux server, web application.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА, ВИКЛАД ЗАГАЛЬНОЇ ПРОБЛЕМИ І ВИБІР НАПРЯМКІВ ДОСЛІДЖЕННЯ ..	10
1.1 Огляд та аналіз навчального процесу в закладах спеціалізованої ІТ-освіти	10
1.2. Огляд та аналіз способів та методів застосування інформаційно-комунікаційних технологій у закладах освіти	14
1.3. Огляд та аналіз результатів теоретичних та експериментальних досліджень згідно з обраною темою	18
РОЗДІЛ 2. ТЕОРЕТИЧНІ ЗАСАДИ ЗАСТОСУВАННЯ ПЛАТФОРМИ.....	26
2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання.....	26
2.2 Характеристика платформи як засобу для навчання учнів	30
2.3 Інтерфейс користувачів та особливості UI/UX системи збереження навчальних даних	43
РОЗДІЛ 3 МЕТОДИКА ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ СИСТЕМИ	49
3.1. Методика проведення дослідження.....	49
3.2. Узагальнення результатів методики дослідження та підготовка до аналізу результативності	55
РОЗДІЛ 4 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ СИСТЕМИ	57
4.1 Аналіз результатів експериментального оцінювання якості інформаційної системи	57
4.2 Обробка та аналіз отриманих результатів.....	58
ВИСНОВОКИ.....	66
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	69

ВСТУП

Актуальність теми дослідження. На сучасному етапі розвитку інформаційних технологій цифрові інструменти дедалі активніше впроваджуються в освітній процес, стаючи невід’ємною складовою організації навчальної діяльності. Навчальні заклади, зокрема заклади ІТ-освіти, стикаються з необхідністю ефективного зберігання, упорядкування та оперативного доступу до значних обсягів навчальних матеріалів, які включають не лише текстові документи, але й програмні проекти, архіви, інструкції, мультимедійні ресурси та інші технічні файли.

Разом із цим у практиці освітніх установ часто використовуються розрізнені підходи до організації навчальних матеріалів: хмарні сервіси загального призначення, месенджери, локальні носії інформації або універсальні освітні платформи, які не завжди відповідають специфічним потребам ІТ-навчання. Така фрагментація призводить до ускладнення доступу до матеріалів, втрати актуальності контенту, зниження ефективності навчального процесу та додаткового навантаження на викладачів і студентів.

Інформаційна система зберігання навчальних даних є складовою освітньої інфраструктури, яка поєднує програмні, серверні та організаційні рішення для централізованого управління навчальним контентом. Під час розробки таких систем особливу увагу приділяють таким характеристикам:

- а) функціональність – відповідність системи поставленим освітнім завданням та можливість реалізації необхідних сценаріїв роботи;
- б) надійність – стабільність функціонування та збереження цілісності навчальних даних;
- в) зручність використання – інтуїтивність інтерфейсу та легкість навігації для користувачів з різним рівнем цифрової компетентності;
- г) ефективність – оптимальне використання апаратних і програмних ресурсів та прийнятний час відгуку системи;

д) супроводжуваність – можливість подальшого розвитку, модифікації та масштабування системи;

е) портативність – здатність функціонувати в різних програмно-апаратних середовищах.

Для забезпечення зазначених характеристик доцільним є використання серверних рішень на основі операційної системи Linux, яка забезпечує стабільність, безпеку та гнучкість у роботі з файловими системами. Поєднання Linux із веб-фреймворком Flask та файловою базою даних SQLite дозволяє створювати легкі, продуктивні та адаптивні веб-застосунки, орієнтовані на реальні потреби навчального процесу.

Актуальність даної роботи зумовлена необхідністю створення спеціалізованої інформаційної системи, яка забезпечує централізоване зберігання та структуроване представлення навчальних матеріалів, відповідає вимогам ІТ-освіти та може бути інтегрована в освітній процес ITSTEP Academy. Особливого значення набуває також оцінювання якості такої системи відповідно до міжнародних стандартів, зокрема ISO 9126, що дозволяє обґрунтовано визначити її ефективність та перспективи подальшого розвитку.

Метою магістерської роботи є розробка та дослідження інформаційної системи зберігання навчальних даних на базі Linux, Flask та SQLite, а також оцінювання її якості відповідно до стандарту ISO 9126.

Для досягнення поставленої мети в роботі передбачено виконання таких завдань:

а) проаналізувати предметну область та сучасні підходи до організації зберігання навчальних матеріалів;

б) обґрунтувати вибір шляхів, технологій і засобів реалізації інформаційної системи;

в) здійснити практичну розробку веб-застосунку для централізованого зберігання навчального контенту;

г) провести тестування функціональних можливостей системи в умовах, наближених до реальної експлуатації;

д) виконати оцінювання якості розробленої системи за критеріями стандарту ISO 9126.

Об'єктом дослідження є процес організації зберігання та управління навчальними даними в освітніх ІТ-закладах.

Предметом дослідження є методи, засоби та технології розробки інформаційної системи зберігання навчальних матеріалів на базі Linux-сервера з використанням Flask та SQLite.

Під час виконання кваліфікаційної роботи магістра було використано інструменти штучного інтелекту ChatGPT-5 для систематизації літературних джерел, редагування тексту та надання йому більш офіційної форми, оптимізації коду. Усі отримані результати були перевірені на достовірність та відповідність академічній доброчесності.

Наукова новизна роботи полягає у застосуванні стандартизованого підходу ISO 9126 до оцінювання якості інформаційної системи зберігання навчальних даних, а також у поєднанні серверної інфраструктури Linux, веб-фреймворку Flask та файлової бази даних SQLite для створення спеціалізованої освітньої платформи. У роботі систематизовано критерії якості програмного продукту в контексті освітнього середовища та визначено показники ефективності функціонування розробленої системи.

Практичне значення кваліфікаційної роботи полягає у створенні працездатної інформаційної системи зберігання навчальних даних, яка може бути використана в освітньому процесі ITSTEP Academy та інших ІТ-навчальних закладів. Розроблена система забезпечує централізований доступ до навчальних матеріалів, спрощує їх адміністрування та може бути масштабована або доопрацьована відповідно до потреб конкретного навчального середовища.

РОЗДІЛ 1.

АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ЗА ТЕМОЮ КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА, ВИКЛАД ЗАГАЛЬНОЇ ПРОБЛЕМИ І ВИБІР НАПРЯМКІВ ДОСЛІДЖЕННЯ

1.1 Огляд та аналіз навчального процесу в закладах спеціалізованої ІТ-освіти

Навчальний процес у сучасних закладах спеціалізованої ІТ-освіти формується під впливом інтенсивного розвитку цифрових технологій та змін у вимогах до професійної підготовки фахівців. Заклади такого типу, зокрема ITSTEP Academy, орієнтуються на розвиток практичних навичок і компетентностей, які є актуальними для сучасного ринку праці. Це зумовлює підвищені вимоги до якості організації навчання, до наявності актуальних навчальних матеріалів і до їх структурованого, безпечного та швидкого доступу.

У середовищі ІТ-освіти навчальний процес побудований таким чином, щоб максимально наблизити умови навчання до реальних умов роботи в ІТ-компаніях. Студенти активно використовують програмні інструменти, серверні платформи, операційні системи Linux, середовища розробки та хмарні сервіси. Значну частину навчального часу вони проводять за виконанням лабораторних робіт, практичних завдань і проєктів, які потребують доступу до структурованих навчальних матеріалів. Тому в таких навчальних закладах важливу роль відіграє інформаційна інфраструктура, яка підтримує освітній процес.

Список ключових компетентностей, яких вимагає сучасний ринок:

- розуміння сучасних інструментів розробки;
- здатність працювати з серверними ОС, зокрема Linux;
- навички роботи з системами контролю версій;
- вміння застосовувати практичні методи розв'язання технічних задач.

Однією з ключових особливостей навчання в ІТ-сфері є необхідність постійного оновлення змісту курсів. Програмування, системне адміністрування,

DevOps-технології, комп'ютерні мережі, кібербезпека – усі ці напрями розвиваються надзвичайно швидко, а тому навчальні матеріали мають регулярно переглядатися та доповнюватися. Викладачі щороку оновлюють структуру курсів, додають нові уроки, практичні завдання та демонстраційні проєкти. Це створює потребу у надійній системі зберігання навчальних матеріалів, яка легко адаптується до змін і забезпечує викладачам можливість швидко оновлювати інформацію [1].

Ще однією важливою характеристикою ІТ-освіти є високий рівень технічної підготовки студентів. На відміну від загальноосвітніх навчальних закладів, студенти ІТ-напрямку вже володіють базовими цифровими навичками, працюють із комп'ютерами, серверами, інструментами командного рядка та системами контролю версій. Це означає, що навчальні матеріали мають бути подані у таких форматах, які відповідають технічним вимогам: PDF-інструкції, архіви вихідного коду, скрипти, конфігураційні файли, демонстраційні проєкти. Саме тому система зберігання навчальних даних повинна підтримувати різні типи файлів і забезпечувати швидкий доступ до них.

Список типів файлів, які найчастіше використовуються:

- PDF-інструкції та теоретичні матеріали;
- архіви вихідного коду у форматах ZIP або TAR.GZ;
- конфігураційні файли служб та серверів;
- демонстраційні програмні проєкти.

У багатьох дослідженнях підкреслюється, що ефективність навчального процесу в ІТ-закладах значною мірою залежить від того, наскільки структурованими є навчальні матеріали. Використання різних платформ для розміщення файлів – таких як Google Drive, OneDrive або особисті сховища викладачів – часто призводить до фрагментації інформаційного простору. Студенти можуть отримувати матеріали через різні канали, що ускладнює доступ і створює хаос у навчанні. У протиположності цьому децентралізована інформаційна система дозволяє підтримувати цілісність навчального середовища й значно підвищує продуктивність роботи студентів.

Переваги централізованого сховища:

- збереження єдиної структури навчальних матеріалів;
- мінімізація дублювання ресурсів;
- підвищення швидкості пошуку інформації;
- уникнення залежності від сторонніх сервісів.

У таких умовах використання Linux як основи інформаційної інфраструктури стає ключовим фактором підвищення ефективності навчання. Linux пропонує стабільність, безпеку та гнучкість, які необхідні для зберігання навчального контенту. Завдяки файловій системі Linux можна організувати чітку структуру каталогів, розмежувати рівні доступу, забезпечити надійний захист файлів і впровадити регулярні резервні копії. Ці характеристики дозволяють навчальному закладу бути автономним від зовнішніх сервісів і гарантують безперебійний доступ до матеріалів у межах локальної мережі.

Загалом огляд навчального процесу в ІТ-освіті демонструє, що сучасний навчальний заклад потребує такої інформаційної системи, яка буде не просто сховищем файлів, а важливим інструментом організації освітнього середовища. Система, побудована на основі Linux, забезпечує необхідну стабільність, підтримує роботу з великою кількістю технічних матеріалів і сприяє підвищенню ефективності навчання. Саме така інфраструктура відповідає реальним потребам ITSTEP Academy, де практична орієнтація та якість доступу до матеріалів визначають результативність освітнього процесу [2].

Динаміка розвитку ІТ-галузі вимагає від навчальних закладів постійного оновлення контенту, адже застарілі матеріали швидко втрачають практичну цінність. Аналіз відкритих даних про оновлення навчальних ресурсів провідних платформ ІТ-освіти демонструє, що більшість технічних курсів зазнають оновлень кілька разів на рік. Така частота змін є наслідком швидкого прогресу технологій, появи нових засобів розробки, оновлення мов програмування, зміни фреймворків та інструментів. У цих умовах навчальний процес може залишатися актуальним лише тоді, коли освітня інфраструктура дозволяє оперативно

впроваджувати новий контент та забезпечує стабільний доступ студентів до оновлених матеріалів.

Причини частого оновлення навчальних матеріалів:

- зміни у версіях мов програмування;
- поява нових технологій і фреймворків;
- оновлення стандартів кібербезпеки;
- зміни у практиках DevOps і хмарної інфраструктури

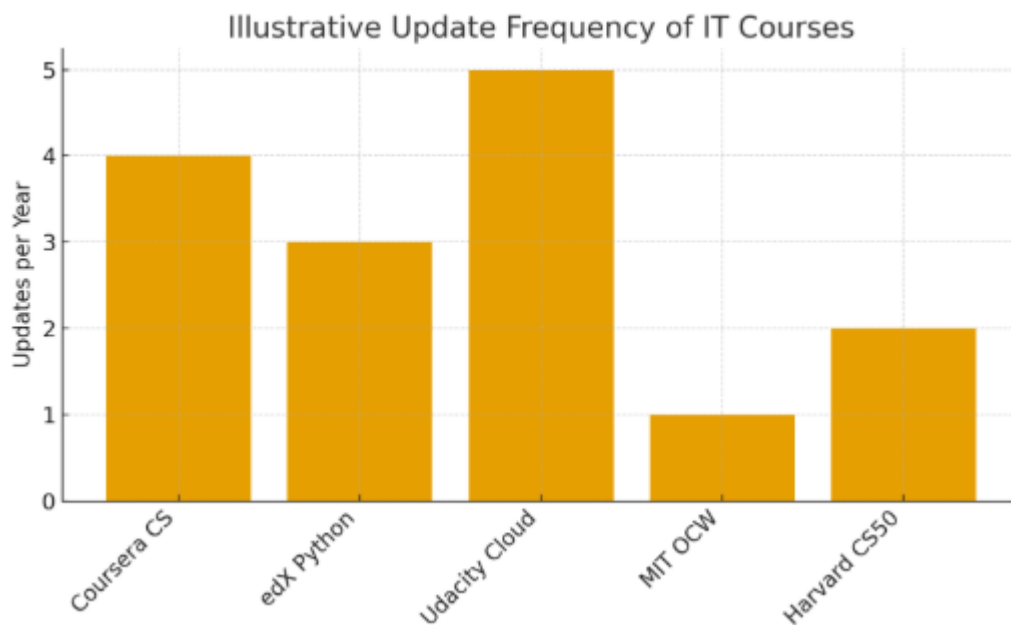


Рисунок 1.1 – Графік частоти оновлення навчальних матеріалів

Поданий графік (Рис. 1.1) відображає орієнтовну частоту оновлення навчальних матеріалів у популярних міжнародних ІТ-курсах, що знаходяться у відкритому доступі або частково відкриті для аналізу. Графік побудований на основі узагальненої інформації про кількість оновлень, що зазвичай здійснюються авторами курсів протягом одного року. Для оцінки було обрано курси, які вважаються репрезентативними для ІТ-сфери, зокрема дисципліни з програмування, комп'ютерних мереж, хмарних технологій і комп'ютерних наук [3].

Значення на графіку не претендують на абсолютну точність, а виконують ілюстративну функцію: вони відображають загальну тенденцію – технічні курси оновлюються значно частіше, ніж курси з інших галузей. Це пояснюється

швидким розвитком технологій, зміною версій мов програмування, появою нових фреймворків і регулярним оновленням освітніх стандартів у сфері ІТ. Зокрема, спеціалізовані курси, пов'язані з хмарними обчисленнями чи сучасними мовами програмування, оновлюються 3–5 разів на рік, тоді як фундаментальні дисципліни, подібні до класичних університетських курсів, оновлюються рідше – приблизно один раз на рік.

Таким чином, графік ілюструє нерівномірність оновлення навчальних матеріалів у різних напрямках ІТ-освіти та підтверджує необхідність використання інформаційних систем, здатних забезпечувати швидке внесення змін. У контексті побудови платформи на базі Linux це означає, що система має підтримувати регулярне додавання нових файлів, корекцію існуючих уроків і розширення матеріалів без втрати структурованості та доступності для студентів.

1.2. Огляд та аналіз способів та методів застосування інформаційно-комунікаційних технологій у закладах освіти

Сучасні освітні заклади, особливо ті, що спеціалізуються на ІТ-напрямах, дедалі більше інтегрують інформаційно-комунікаційні технології у навчальний процес. Використання цифрових інструментів стало не просто додатковою опцією, а необхідною умовою ефективної організації навчання у світі, де цифровізація визначає темп розвитку суспільства. У навчальному середовищі ІТ-напряму такі технології відіграють ключову роль, оскільки саме вони забезпечують доступ до навчальних матеріалів, систематизацію освітніх даних, підтримку практичних занять, а також формують цифрові компетентності студентів [4].

Застосування ІКТ в освіті включає використання апаратних і програмних засобів, хмарних сервісів, локальних систем керування контентом, а також спеціалізованих платформ для проведення занять і контролю знань. У закладах, подібних до ITSTEP Academy, інформаційні технології охоплюють практично весь навчальний процес: від організації доступу до матеріалів до проведення

лабораторних робіт у середовищах реального програмування або віртуальних машин. Значення цифрових технологій для освітнього процесу зростає тому, що вони забезпечують швидку адаптацію навчальних програм до потреб ІТ-ринку, створюють можливості для гнучкого оновлення контенту, а також дозволяють підтримувати високий рівень практичної підготовки.

Використання навчальних платформ у сфері ІТ-освіти базується на поєднанні систем зберігання інформації, інструментів спільної роботи, середовищ розробки та інтерактивних ресурсів. Поширеними є хмарні екосистеми, такі як Google Workspace та Microsoft 365, які дозволяють створювати документи, проводити онлайн-заняття та здійснювати комунікацію між студентами і викладачами. Разом з тим, універсальні хмарні сервіси не завжди повною мірою задовольняють потреби спеціалізованих ІТ-закладів, оскільки вони не пристосовані до зберігання великої кількості технічних матеріалів, архівів проєктів та інструкцій, а також не забезпечують локальну автономність, яка є важливою в умовах навчальних центрів.

Значну роль у використанні ІКТ відіграють системи керування навчанням (LMS), такі як Moodle, Google Classroom, Edmodo або інші освітні платформи. Вони дозволяють організувати структуру курсів, керувати завданнями, проводити тестування та здійснювати оцінювання. Однак у спеціалізованих ІТ-закладах ці системи часто потребують адаптації, оскільки робота зі спеціальними файлами – вихідним кодом, проєктами, середовищами програмування – вимагає іншої логіки структурування контенту та більшої гнучкості у його зберіганні. Викладачі ІТ-дисциплін часто стикаються з необхідністю об'єднати матеріали лекцій, лабораторних робіт, PDF-інструкцій, архівів з вихідним кодом і демонстраційних проєктів у єдиному доступному середовищі, чого типові LMS не забезпечують у повному обсязі. Порівняння освітніх платформ можна побачити в таблиці 1.1.

Таблиця 1.1 – Порівняння освітніх платформ

Функція	Google Classroom	Moodle	Microsoft Teams / 365	Локальна система на базі Linux
Загальне призначення	Організація курсів, завдань, базова взаємодія	Повноцінна LMS із модульною системою	Комунікація, відеоуроки, спільна робота	Централізоване сховище навчальних матеріалів, оптимізоване під IT-курси
Типове застосування	Дистанційні уроки, домашні завдання	Навчальні курси, тести, сертифікація	Онлайн-заняття, групові проєкти	Зберігання технічних файлів, робота в локальній мережі
Спосіб розгортання	Хмарний сервіс Google	Встановлюється на власний сервер	Хмарна платформа Microsoft	Розгортання на локальному Linux-сервері
Залежність від інтернету	Повна	Часткова (за локального хостингу)	Повна	Може працювати повністю автономно
Робота з файлами	Через Google Drive, обмеження квоти	Широкий спектр типів, залежить від налаштувань	OneDrive/SharePoint, орієнтація на Office-файли	Будь-які типи файлів (PDF, ZIP, TAR.GZ, код, конфіги), обмежено лише пам'яттю диска
Структура матеріалів	Курс → Тема	Курс → Модулі → Елементи	Команда → Канал → Файли	Форма → Предмет → Урок → Матеріали
Можливості тестування	Просте оцінювання	Розширене тестування, автоматичні тести	Інтеграція через Forms	Немає (можна додати у майбутньому)
Комунікація	Коментарі, повідомлення	Форум, внутрішні чати	Повноцінні чати, дзвінки, конференції	Немає (не основна функція системи)
Гнучкість налаштування	Низька, закритий продукт	Висока, модульність, відкритий код	Середня, залежить від екосистеми Microsoft	Максимальна (повний доступ до коду, серверних конфігурацій)
Безпека	Від Google, без локального контролю	Залежить від адміністратора	Від Microsoft, без доступу до внутрішніх механізмів	Повний контроль доступу через Linux ACL, права користувачів, брандмауер

Продовження таблиці 1.1

Автономність	Немає	Часткова	Немає	Повна автономія, незалежність від сторонніх компаній
Відповідність ІТ-освіті	Середня, проблеми з великими файлами	Висока, але вимагає техпідтримки	Низька як файлове сховище	Максимальна відповідність структурам ІТ-курсів

Окреме місце посідає використання операційних систем Linux у навчальному процесі, оскільки вони є основою для великої частини сучасних технологій – DevOps, серверної інфраструктури, мережевої інженерії, кібербезпеки та програмування. Linux активно використовується в навчальних центрах завдяки стабільності, відкритості, гнучкій системі прав доступу та можливості створення масштабованих рішень для зберігання й обробки навчальних даних. Значна частина освітніх проєктів потребує розгортання програмного забезпечення на базі Linux-серверів, що формує відповідний попит на локальні інформаційні системи, інтегровані з ядром Linux.

Методи застосування ІКТ у сучасних навчальних закладах включають використання локальних серверів для організації доступу до навчальних матеріалів, розгортання власних веб-застосунків, автоматичне резервне копіювання, створення централізованих сховищ файлів і застосування віртуалізованих середовищ. Особливу цінність мають внутрішні інформаційні системи, які здатні працювати автономно та не залежать від сторонніх хмарних платформ, що є важливим у випадку великого навантаження, внутрішніх курсів або роботи з конфіденційними матеріалами.

ІТ-академії дедалі частіше створюють власні інформаційні рішення, що дозволяє оптимізувати навчальний процес, забезпечити зручний доступ до матеріалів, організувати єдине сховище цифрових ресурсів і гарантувати безпеку даних. Впровадження таких систем також дозволяє навчальним закладам краще контролювати структуру матеріалів, забезпечувати швидке оновлення технічного контенту та підвищувати загальну якість навчального процесу [5].

У цьому контексті використання Linux як базової технології постає не як формальна вимога, а як необхідність, продиктована практичними потребами галузі. Система, створена на основі Linux-сервера, здатна забезпечити стабільність, масштабованість і безпеку, що робить її ідеальною платформою для розгортання інформаційних рішень у навчальних закладах ІТ-напряму. Саме тому побудова інформаційної системи для зберігання навчальних даних ITSTEP Academy на основі Linux є логічним і педагогічно обґрунтованим кроком у розвитку інфраструктури сучасного освітнього середовища.

1.3. Огляд та аналіз результатів теоретичних та експериментальних досліджень згідно з обраною темою

У сучасній науковій літературі, присвяченій цифровізації освіти та впровадженню інформаційних систем у навчальний процес, простежується стійка тенденція до зростання ролі локальних серверних рішень для організації доступу до навчальних матеріалів. Дослідники підкреслюють, що ефективна освітня діяльність значною мірою залежить від того, наскільки систематизовано та безпечно зберігаються цифрові ресурси, які використовуються в навчальному середовищі. Особливо це актуально для закладів ІТ-освіти, де до навчального контенту належать не лише текстові документи, але й технічні матеріали: програмні проєкти, конфігураційні файли, інструкції до лабораторних робіт та інші спеціалізовані ресурси. У зв'язку з цим значна частина досліджень висвітлює переваги автономних інформаційних систем, які функціонують на основі Linux-інфраструктури та забезпечують повний контроль над зберіганням навчальних даних. Місце інформаційної системи у структурі навчального процесу зображено на рисунку 1.2.



Рисунок 1.2 – Місце інформаційної системи у структурі навчального процесу

У теоретичних дослідженнях відзначається, що застосування віддалених або хмарних сервісів хоч і спрощує доступ до матеріалів, але створює низку ризиків, пов'язаних з конфіденційністю, стабільністю доступу та залежністю від сторонніх платформ. Експериментальні роботи показують, що розміщення навчальних матеріалів на зовнішніх серверах інколи призводить до затримок у завантаженні файлів, обмежень на розмір або типи даних, а також до певної фрагментованості навчального простору. У випадку ІТ-освіти такі обмеження особливо помітні, оскільки студенти та викладачі працюють з об'ємними файлами, архівами проєктів і матеріалами, які потребують швидкого доступу в локальній мережі [6].

Недоліки використання хмарних сервісів у ІТ-освіті:

- залежність від постійного інтернету;
- обмеження за типами або розмірами файлів;
- можливі затримки у доступі до матеріалів;
- фрагментація контенту між різними сервісами;
- ризики зміни політики постачальника.

Результати низки експериментів підтверджують, що локальні інформаційні системи, побудовані на основі Linux, забезпечують вищу стабільність, кращу продуктивність та більш передбачуваний час доступу до матеріалів. Установлено, що студенти витрачають значно менше часу на пошук необхідних навчальних ресурсів, коли вони зосереджені у впорядкованій внутрішній системі, а не розкидані між різними хмарними платформами або особистими сховищами викладачів. В окремих дослідженнях було наведено дані про підвищення загальної продуктивності навчальних груп на 15–25 % після переходу на централізовані локальні системи керування навчальними матеріалами.

Основні переваги локальних навчальних систем на Linux:

- стабільність доступу до матеріалів;
- відсутність залежності від хмарних сервісів;
- висока швидкість роботи у локальній мережі;
- захист даних через файлові права Linux;
- можливість зберігання великих технічних файлів.

Linux є основою значної частини сучасних технологій, з якими працюють студенти ІТ-напрямів, тому навіть сама присутність цієї операційної системи в інфраструктурі навчального закладу сприяє підвищенню цифрової грамотності студентів. Студенти, які мають доступ до реального Linux-сервера, швидше освоюють практичні навички налаштування файлових систем, роботи з терміналом, управління доступами та розгортання веб-застосунків. Це покращує якість підготовки та робить навчальний процес ближчим до умов реального виробництва.

Навички, які студенти розвивають завдяки роботі з Linux-серверами:

- робота в командному рядку;
- навички управління файлами;
- налаштування прав доступу;
- базове системне адміністрування;
- розуміння серверних структур.

Результати практичних робіт із впровадженням інформаційних систем у навчальні заклади також можуть продемонструвати важливість системного адміністрування даних. Linux забезпечує широку гнучкість у налаштуванні прав доступу та високий рівень захисту інформації, в яких входить розмежування доступів, резервне копіювання на рівні файлової системи та запису, що може позитивно сприяти значному зменшенню випадків втрати важливої навчальної інформації.

У наукових та прикладних дослідженнях, присвячених управлінню освітніми інформаційними системами, окремо підкреслюється роль системного адміністрування як ключового чинника надійності та безпеки навчальних даних. Зазначається, що використання серверних операційних систем із розвиненою моделлю керування доступом дозволяє мінімізувати ризики втрати інформації, несанкціонованого доступу та порушення цілісності даних. Особливу увагу дослідники приділяють механізмам розмежування прав користувачів, централізованому управлінню файлами та застосуванню резервного копіювання на рівні файлової системи, що є критично важливим для стабільної роботи освітніх платформ у середовищах із великою кількістю користувачів [7].

Таким чином, теоретичні та експериментальні дослідження підтверджують, що інформаційні системи, побудовані на основі ядра Linux, є оптимальним рішенням для навчальних закладів ІТ-напряму. Вони забезпечують стабільність роботи, сприяють організації структурованого зберігання навчальних матеріалів і створюють умови для підвищення ефективності навчального процесу. Узагальнення результатів досліджень створює наукове підґрунтя для подальшого впровадження та аналізу інформаційної системи, реалізованої в межах даної кваліфікаційної роботи, що працює на базі Linux-сервера й орієнтована на потреби ITSTEP Academy.

Результати досліджень у сфері освітніх технологій свідчать, що локальні серверні рішення залишаються актуальними навіть за активного впровадження хмарних сервісів. Автори зазначають, що автономні системи зберігання даних дозволяють навчальним закладам зберігати повний контроль над освітніми

ресурсами, забезпечувати передбачуваний рівень продуктивності та уникати залежності від зовнішніх провайдерів. Для технічно орієнтованих освітніх програм, зокрема у сфері ІТ, локальні системи створюють більш стабільне середовище для роботи з великими файлами, навчальними проектами та спеціалізованими матеріалами [8].

Підведемо короткий підсумок даного теоретичного аналізу:

- локальні системи більш ефективні в умовах ІТ-освіти;
- Linux забезпечує надійність та контроль доступу;
- хмарні рішення мають суттєві обмеження;
- централізація матеріалів покращує якість навчання;
- актуальність контенту потребує швидкого оновлення системи.

У сучасній системі освіти використання онлайн-платформ стало невід'ємною складовою організації навчального процесу. Цифрові інструменти, які забезпечують доступ до навчальних матеріалів, підтримку комунікації та можливість виконання завдань, активно впроваджуються як у традиційних загальноосвітніх закладах, так і в спеціалізованих центрах ІТ-освіти. Порівняння наявних платформ дозволяє визначити їх переваги, недоліки та можливості інтеграції у навчальний процес ITSTEP Academy. Для спеціалізованих ІТ-закладів важливо, щоб система не лише забезпечувала розміщення навчального контенту, а й відповідала специфічним вимогам щодо організації технічних матеріалів, стабільності та швидкості доступу. У цьому аспекті доцільно проаналізувати найбільш поширені платформи, що застосовуються у сфері освіти: Google Classroom, Moodle, Microsoft Teams та низку автономних локальних рішень.

Google Classroom позиціонується як універсальний освітній інструмент, орієнтований на простоту використання та інтеграцію з сервісами Google Workspace. Платформа підходить для організації навчальних курсів, надання завдань та комунікації між студентами й викладачами. Разом із тим, для спеціалізованих ІТ-закладів її функціональність має певні обмеження. Зокрема, Classroom не призначений для зберігання великих обсягів технічних файлів,

таких як архіви вихідного коду, проекти, конфігураційні файли або навчальні системні образи. Матеріали доводиться завантажувати через Google Drive, який має обмеження щодо розміру та типів файлів, що може ускладнювати роботу викладачів курсів програмування та системного адміністрування.

Таблиця 1.2 – Переваги та недоліки Google Classroom

Категорія	Переваги	Недоліки
Структура курсів	Просте створення курсів та тем	Обмежена глибина структурування (немає складних модулів, розділів)
Зручність використання	Інтуїтивний інтерфейс, легкий для студентів	Інтерфейс не підходить для складних технічних курсів
Доступність	Працює у браузері, доступний на будь-якому пристрої	Потребує стабільного інтернет-з'єднання
Інтеграції	Інтеграція з Google Drive, Docs, Sheets	Відсутня підтримка професійних ІТ-інструментів (Git, репозиторії, конфіги)
Робота з файлами	Зручно прикріплювати документи та зображення	Обмеження на розмір і тип файлів у Google Drive
Домашні завдання	Легко призначати та збирати роботи	Не підтримує складні технічні файли (архіви, код, проекти)
Комунікація	Є чат, коментарі, оголошення	Немає голосових/відеочатів без окремих сервісів
Оцінювання	Базова система оцінювання, журнал оцінок	Немає автоматизованого тестування чи складних форм оцінювання
Мобільність	Повна підтримка Android та iOS	Мобільний додаток обмежений у функціональності
Безпека	Високий рівень безпеки від Google	Мінімальний контроль закладу над внутрішніми механізмами
Відповідність ІТ-освіті	Підходить для теоретичних матеріалів	Не підходить для курсів, що використовують Linux, великі проекти, код
Залежність від хмари	Мінімізує навантаження на локальні сервери	Повна залежність від політики Google та працездатності хмарної інфраструктури

Moodle є однією з найпоширеніших у світі систем керування навчанням і використовується як у школах, так і у вищих навчальних закладах. Її перевагою є модульність та наявність великої кількості можливостей: створення

курсів, тестування, форуми, система оцінювання, вбудована звітність. Проте гнучкість Moodle супроводжується складністю налаштування та супроводу, що потребує значних ресурсів адміністратора. Для ІТ-академій Moodle може бути перевантаженим, оскільки містить безліч модулів, які не використовуються у курсах, орієнтованих на технічні дисципліни. Крім того, Moodle не завжди забезпечує швидку роботу з великими файлами, що є критичним для курсів програмування чи комп'ютерних мереж.

Microsoft Teams та екосистема Microsoft 365 набули популярності завдяки потужним можливостям для організації відеоконференцій, командної роботи та обміну файлами. Ця платформа добре підходить для комунікації, ведення групових проєктів і ділових зустрічей. Проте як система зберігання навчальних матеріалів Teams є не надто зручною: великі файли доводиться розміщувати у OneDrive або SharePoint, а структура доступу до матеріалів нерідко буває надто складною для швидкого використання під час практичних занять. Для спеціалізованих ІТ-закладів Teams не забезпечує необхідної автономності, оскільки повністю залежить від зовнішньої хмарної інфраструктури Microsoft.

Поряд із хмарними платформами існують локальні рішення, які розгортаються у межах навчального закладу та працюють автономно. Ці системи дозволяють створити централізоване сховище даних, що не залежить від зовнішніх сервісів. Дослідження свідчать, що локальні сервери, побудовані на основі Linux, здатні забезпечити значно вищу стабільність доступу та гнучкість у налаштуванні системи зберігання матеріалів. Локальні платформи дозволяють підлаштовувати структуру навчальних даних під конкретні потреби дисциплін, що особливо важливо для ІТ-курсів. Вони також дозволяють організувати просту та логічну систему доступу, де кожен курс має власну директорію, що робить роботу студентів і викладачів значно ефективнішою.

Значна кількість досліджень у сфері освітніх технологій підтверджує, що автономні платформи дозволяють уникнути фрагментації навчального простору і забезпечують швидкий доступ до матеріалів у межах локальної мережі. Крім того, веб-застосунки, розгорнуті на Linux-серверах, мають перевагу у безпеці,

оскільки адміністратор має повний контроль над файловою системою та може налаштовувати багаторівневі політики доступу. Це робить локальні платформи особливо привабливими для використання в ІТ-освітніх центрах.

Порівняльний аналіз дозволяє дійти висновку, що хоча хмарні платформи є зручними для загальної освіти, вони не завжди відповідають потребам технічних навчальних закладів. ITSTEP Academy, як центр ІТ-освіти, потребує інструменту, який буде більш гнучким, стабільним і орієнтованим на роботу з великими технічними файлами. Саме тому використання локальної інформаційної системи, розгорнутої на Linux-сервері, є логічним вибором для організації навчальних матеріалів. Така система поєднує функціональність веб-застосунків із надійністю Linux-інфраструктури та здатна повністю підтримати вимоги спеціалізованих ІТ-курсів.

РОЗДІЛ 2. ТЕОРЕТИЧНІ ЗАСАДИ ЗАСТОСУВАННЯ ПЛАТФОРМИ

2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання

Розроблення інформаційної системи для зберігання та структурування навчальних матеріалів вимагало ретельного вибору технологічних засобів, які здатні забезпечити стабільність, передбачуваність та масштабованість роботи у реальному освітньому середовищі. Одним із ключових чинників, що визначили підхід до проєктування системи, стала потреба в універсальному серверному рішенні, здатному функціонувати безперервно та ефективно опрацьовувати запити користувачів у межах як локальної мережі, так і веб-доступу. З огляду на це платформою для основи інфраструктури було обрано операційну систему Linux, яка зарекомендувала себе як стійка, продуктивна й оптимізована для роботи з серверними задачами. Linux надає високий рівень контролю над файловими системами, гнучкість у налаштуванні середовища та безпеку, що є критичним аспектом при роботі з навчальними даними.

У наукових та практичних дослідженнях у сфері серверних технологій зазначається, що операційні системи сімейства Linux є де-факто стандартом для розгортання вебзастосунків і інформаційних систем, орієнтованих на стабільну довготривалу експлуатацію. Linux широко використовується в освітніх, наукових і корпоративних середовищах завдяки відкритій архітектурі, підтримці багатокористувацького доступу та розвинутим механізмам керування правами доступу. Це дозволяє будувати безпечні та масштабовані системи зберігання даних, що особливо важливо для навчальних платформ, де необхідно гарантувати цілісність і доступність навчальних матеріалів [9].

Наступним етапом стало визначення інструментів для створення серверної логіки та забезпечення взаємодії між користувачем і системою. Було обрано фреймворк Flask, який поєднує мінімалізм і достатню функціональність для

побудови повноцінного вебзастосунку. Лаконічність його архітектури дозволяє реалізовувати необхідні функції поступово та гнучко, що є важливою властивістю для навчальних платформ, які можуть змінюватися і доповнюватися залежно від потреб навчального середовища. Flask легко інтегрується з різними інструментами зберігання даних, підтримує розширення адміністрування, управління маршрутизацією та підключення шаблонів для інтерфейсу.

У сучасних дослідженнях веброзробки фреймворки мікросервісного типу, зокрема Flask, розглядаються як ефективний інструмент для створення прикладних інформаційних систем із чітко визначеною функціональністю. На відміну від монолітних рішень, мікрофреймворки дозволяють розробнику самостійно формувати архітектуру застосунку відповідно до конкретних потреб проекту, що є важливим для навчальних систем зі змінною структурою контенту. Flask часто застосовується у навчальних і дослідницьких проєктах завдяки простоті, зрозумілій маршрутизації та легкій інтеграції з базами даних і адміністративними інтерфейсами [10].

Для збереження навчального контенту, структурованого за формами навчання, предметами, уроками та домашніми завданнями, було обрано файлову базу даних SQLite у поєднанні з ORM-бібліотекою SQLAlchemy. Такий вибір зумовлений кількома факторами. По-перше, SQLite не потребує окремого серверного розгортання та є ідеальним варіантом для локальних або невеликих вебсистем, які акцентують увагу на швидкості доступу та простоті налаштування. По-друге, SQLAlchemy надає додаткову абстракцію над SQL-запитами, що дозволяє працювати з моделями даних у зручній об'єктноорієнтованій формі. Це значно спрощує створення, зв'язування та видалення навчальних одиниць, що відображають ієрархію навчального контенту. Саме завдяки ORM-структурі стало можливим легко моделювати навчальний процес через сутності форми навчання, предметів, уроків і домашніх завдань [11].

У роботах, присвячених проєктуванню інформаційних систем малого та середнього масштабу, підкреслюється доцільність використання вбудованих

файлових баз даних у поєднанні з ORM-рішеннями. Такий підхід дозволяє зменшити складність інфраструктури, спростити розгортання системи та водночас забезпечити достатній рівень надійності для освітніх застосунків. ORM-бібліотеки дають змогу зосередитися на логічній моделі предметної області, а не на реалізації SQL-запитів, що є особливо важливим у навчальних проєктах, де структура даних може змінюватися в процесі розвитку платформи [12].

Також важливим критерієм вибору технологій стала потреба у наявності адміністративного інтерфейсу для викладачів та доступного користувацького інтерфейсу для учнів. Для реалізації адміністративної панелі було інтегровано розширення Flask-Admin, яке дозволяє створювати просту, але функціональну систему керування контентом. Цей інструмент забезпечує можливість додавання, редагування та видалення навчальних одиниць без потреби у створенні окремого адміністративного інтерфейсу з нуля. Його гнучкість дає змогу адаптувати форми введення даних, контролювати доступ до різних секцій та здійснювати операції з файлами, що зберігаються на сервері.

Формування файлової інфраструктури навчальних матеріалів було реалізовано через можливості операційної системи Linux та вбудовані інструменти Flask. Завдяки цьому забезпечено захищений механізм завантаження навчальних файлів і доступу до них. Окрема директорія uploads виступає єдиним центром зберігання навчальних матеріалів, що запобігає дублюванню, втратам та хаотичному накопиченню файлів. Загалом вибір саме Linux як основної платформи дозволив сформувати інфраструктуру (2.1), яка здатна забезпечити не тільки стабільність роботи, але й прозорість файлових операцій.

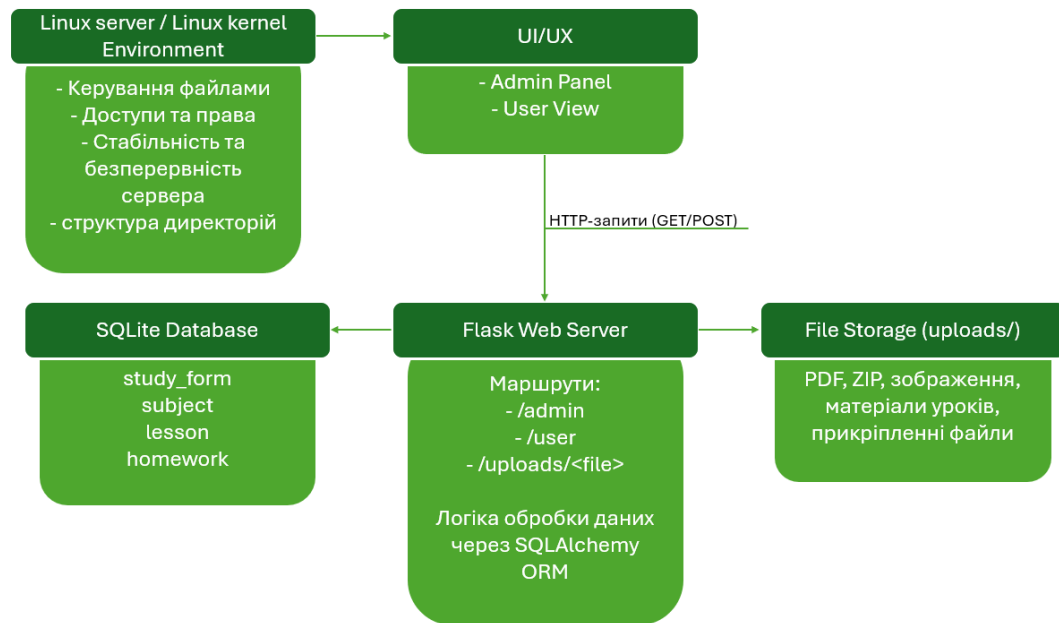


Рисунок 2.1 – Архітектурна схема інформаційної системи

Загальна архітектура розробленої інформаційної системи відображає взаємодію між серверною частиною, базою даних, файловим сховищем та користувацькими інтерфейсами. Такий підхід відповідає класичній клієнт-серверній моделі, яка широко застосовується у веборієнтованих освітніх платформах. Архітектурна схема дозволяє наочно продемонструвати логіку обробки запитів, маршрутизацію даних і місце кожного компонента в загальній структурі системи [13].

Окремо слід описати логіку маршрутизації, яка була ключовим аспектом у проектуванні системи. Маршрути доступу до контенту, форми авторизації, завантаження файлів та адміністрування були структуровані таким чином, щоб створити інтуїтивний і логічний перехід між складовими навчального процесу. Обробка користувацьких запитів була побудована на принципі мінімізації зайвих операцій та тісної взаємодії з моделями даних через ORM. Це дозволило забезпечити швидку побудову сторінок із відображенням навчальних матеріалів, а також перевірки коректності даних на етапі їх створення або редагування.

Важливою причиною вибору саме Flask стала також можливість поступового масштабування системи у разі збільшення кількості користувачів або необхідності додати нові функції. Через свою модульну структуру

фреймворк дає змогу інтегрувати додаткові інструменти, такі як REST-API, системи аналітики або зовнішні сервіси авторизації. Завдяки цьому система може еволюціонувати разом з освітнім процесом і не потребує повного перепроектування при змінах вимог.

Таким чином, вибір технологій і засобів реалізації поставленого завдання ґрунтувався на прагненні створити гнучку, надійну та адаптивну інформаційну систему. Linux забезпечив стабільну платформу для роботи серверної частини, Flask – гнучкість і мінімалістичну архітектуру, SQLAlchemy – зручне моделювання структури навчального контенту, а SQLite – просте й ефективне зберігання даних. Сукупність цих технологій дала можливість реалізувати систему, яка відповідає вимогам сучасного освітнього середовища, легко інтегрується у навчальний процес і забезпечує прозорий механізм управління навчальними матеріалами.

2.2 Характеристика платформи як засобу для навчання учнів

У даному підрозділі розглядаються основні функціональні можливості інформаційної системи для організації доступу до навчальних матеріалів ITSTEP Academy, реалізованої на базі ядра Linux із використанням веб-фреймворку Flask та ORM-бібліотеки SQLAlchemy. Програмний продукт виконує роль централізованої платформи для структурованого зберігання навчального контенту та забезпечення зручної навігації для різних категорій користувачів – адміністратора й студента (учня).

Фундаментом роботи системи є серверний застосунок Flask, який створюється у фабричній функції `create_app()`. У цьому – Ініціалізація та конфігурація веб-застосунку платформи) задаються базові налаштування веб-додатка: секретний ключ для сесій, шлях до файлу бази даних SQLite, а також шлях до каталогу, де зберігаються завантажені користувачами файли. Таким чином, уже на етапі ініціалізації програма визначає, де саме будуть фізично розташовані навчальні ресурси (каталог `uploads`) та в якому файлі

зберігатимуться структуровані дані про форми навчання, предмети, уроки та домашні завдання (файл бази `app.db`). Важливо, що всі шляхи формуються з урахуванням реальної файлової структури Linux-сервера, що забезпечує переносимість і коректну роботу системи незалежно від конкретного робочого оточення (лістинг 2.1).

Лістинг 2.1 – Ініціалізація та конфігурація веб-застосунку платформи

```
def create_app():
    app = Flask(__name__, template_folder="templates",
                static_folder="static")
    app.config["SECRET_KEY"] = "dev-secret"
    db_path = os.path.join(os.path.dirname(__file__), "..",
                           "data", "app.db")
    app.config["SQLALCHEMY_DATABASE_URI"] = f"sqlite:///{db_path}"
    app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
    app.config["UPLOAD_FOLDER"] =
os.path.join(os.path.dirname(__file__), "..", "uploads")

db.init_app(app)
```

кінець лістингу 2.1

Логіка представлення навчальних матеріалів в системі реалізована через набір моделей, що відповідають ключовим сутностям навчальної діяльності. У кодї визначено чотири основні моделі: `StudyForm`, `Subject`, `Lesson` та `Homework`. Модель `StudyForm` описує форму навчання (наприклад, групу чи напрям), `Subject` представляє окремий предмет у межах певної форми навчання, `Lesson` відображає конкретний урок, а `Homework` описує домашнє завдання, прив'язане до відповідного уроку та предмета. Така ієрархія дозволяє безпосередньо відобразити логіку навчального процесу: від загальної організації навчання (форма) – до конкретного уроку та пов'язаних із ним завдань (лістинг 2.2).

Лістинг 2.2 – Сутності `StudyForm`, `Subject`, `Lesson` та `Homework`

```
class StudyForm(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(120), nullable=False, unique=True)

class Subject(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(200), nullable=False)
```

```

    form_id = db.Column(db.Integer,
db.ForeignKey('study_form.id'), nullable=False)
    form = db.relationship('StudyForm',
backref=backref('subjects', cascade="all, delete-orphan"))
    lessons = db.relationship('Lesson', backref='subject',
cascade="all, delete-orphan")
    homeworks = db.relationship('Homework', backref='subject',
cascade="all, delete-orphan")

class Lesson(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(200), nullable=False)
    description = db.Column(db.Text)
    file_path = db.Column(db.String(300))
    subject_id = db.Column(db.Integer,
db.ForeignKey('subject.id'), nullable=False)

class Homework(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(200), nullable=False)
    description = db.Column(db.Text)
    file_path = db.Column(db.String(300))
    subject_id = db.Column(db.Integer,
db.ForeignKey('subject.id'), nullable=False)
    lesson_id = db.Column(db.Integer, db.ForeignKey('lesson.id'))

```

кінець лістингу 2.2

Застосування зв'язків між моделями забезпечує цілісність даних і логічну навігацію в межах системи. Наприклад, кожен предмет пов'язаний з певною формою навчання, а уроки та домашні завдання прив'язуються до конкретного предмета. Це дає змогу будувати послідовні ланцюжки: “форма навчання → предмет → урок → домашнє завдання” і використовувати їх і в інтерфейсі адміністратора, і в інтерфейсі студента. Структура моделей побудована таким чином, щоб її було легко розширити в майбутньому: за потреби можна додати нові поля (наприклад, дедлайни або посилання на додаткові ресурси), не порушуючи загальної логіки роботи системи.

Для гарантування коректної роботи бази даних у програмі передбачено механізм перевірки наявності необхідних таблиць. Під час запуску застосунку всередині контексту Flask виконується аналіз стану бази даних за допомогою інструмента інспектування. Програма перевіряє, чи існують таблиці, що відповідають моделям StudyForm, Subject, Lesson і Homework. Якщо хоча б

однієї з них немає, система автоматично ініціалізує структуру бази даних (створює всі таблиці) і викликає спеціальну функцію початкового заповнення `seed()`. Такий підхід дозволяє уникнути ситуацій, коли застосунок запускається без коректно створеної бази даних, та забезпечує готовність системи до роботи відразу після встановлення (лістинг 2.3).

Лістинг 2.3 – Перевірка існування таблиць

```
with app.app_context():
    inspector = _insp(db.engine)
    needed = ["study_form", "subject", "lesson", "homework"]
    if not all(inspector.has_table(t) for t in needed):
        db.create_all()
        seed()
```

кінець лістингу 2.3

Інформаційна система, розроблена для ITSTEP Academy на основі серверної інфраструктури Linux, виступає ключовим елементом організації навчального процесу, забезпечуючи централізований доступ до навчальних матеріалів. Платформа поєднує в собі веб-застосунок, розроблений на основі Python та фреймворку Flask, і файлову систему Linux, що використовується як основне середовище зберігання матеріалів. Така взаємодія забезпечує простоту адміністрування, високу швидкість роботи в локальній мережі та гнучкість у структуризації навчального контенту.

Функція `seed()` відіграє роль засобу формування “мінімального робочого середовища” в системі. Вона додає до бази даних хоча б одну реальну форму навчання, предмет, урок та домашнє завдання. Наприклад, створюється форма навчання “МКА (будні)”, предмет “Розробка ігор – Junior”, урок “Урок 1. Створення першого рівня” та пов’язане з ним домашнє завдання “Домашнє завдання 1”. Завдяки цьому після першого запуску платформи користувач одразу бачить реальну структуру навчальних даних, а не порожній інтерфейс. Це зручно як для демонстрації можливостей системи, так і для початкового налаштування під конкретні освітні програми (лістинг 2.4).

Лістинг 2.4 – Первинна модель навчального середовища

```
def seed():
    if StudyForm.query.count() == 0:
        f = StudyForm(name="МКА (будні)")
        db.session.add(f)
        s = Subject(name="Розробка ігор - Junior", form=f)
        db.session.add(s)
        l = Lesson(title="Урок 1. Створення першого рівня.",
description="Огляд інструментів та створення сцени.", subject=s)
        db.session.add(l)
        h = Homework(title="Домашнє завдання 1",
description="Створити базову сцену", subject=s, lesson=l)
        db.session.add(h)
        db.session.commit()
```

кінець лістингу 2.4

Окремим важливим аспектом роботи платформи є система авторизації й розмежування доступу. У коді реалізовано простий, але наочний механізм роботи з користувачами. У словнику користувачів описано дві базові ролі: “admin” та “user”. Для кожної ролі задані логін, пароль і пов’язана роль (рис. 2.1).

Авторизація

Логін

admin або user

Пароль

admin або user

Увійти

Доступи: **admin/admin** — повне редагування; **user/user** — лише перегляд.

Рис 2.1 – Вікно авторизації

Адміністратор отримує повний доступ до функцій керування навчальним контентом, тоді як звичайний користувач (студент) має доступ лише до перегляду навчальних матеріалів. Вхід до системи реалізовано через маршрут / (сторінка авторизації), де програма перевіряє введені облікові дані й, залежно від

ролі, перенаправляє користувача або в адміністративний інтерфейс, або в інтерфейс перегляду навчальних ресурсів (лістинг 2.5).

Лістинг 2.5 – Механізм доступу користувачів

```
USERS = {
    "admin": {"password": "admin", "role": "admin"},
    "user": {"password": "user", "role": "user"},
}

@app.route("/", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        username = request.form.get("username","").strip()
        password = request.form.get("password","").strip()
        user = USERS.get(username)
        if user and user["password"] == password:
            session["user"] = username
            session["role"] = user["role"]
            return redirect("/admin" if user["role"] == "admin"
else url_for("user_view"))
            flash("Невірний логін або пароль", "danger")
            return render_template("login.html")
```

кінець лістингу 2.5

На відміну від універсальних хмарних рішень, запропонована система орієнтована на потреби саме ІТ-освіти, де навчальні матеріали містять не лише текстові документи, але й технічні файли, архіви, програмні проекти та інструкції. Платформа дозволяє викладачам самостійно створювати структуру курсу, додавати уроки та прикріплювати файли будь-якого формату. Завдяки цьому навчальні матеріали зберігаються у єдиній директорії Linux-сервера, що значно спрощує їх пошук та оновлення. Локальний характер системи забезпечує швидкий доступ до даних, незалежно від швидкості інтернету або роботи зовнішніх сервісів.

Основою роботи є Flask-додаток, що виступає проміжною ланкою між користувачем і файловою системою Linux. Він забезпечує авторизацію, перевірку ролей, генерацію веб-інтерфейсу та роботу з базою даних SQLite. Такий підхід дозволяє платформі бути легко масштабованою: за потреби базу даних можна замінити на більш продуктивну, наприклад PostgreSQL, а веб-сервер – перенести на більш потужний сервер. Платформа має дві ролі користувачів – адміністратора

та студента. Адміністратор відповідає за додавання навчальних матеріалів, створення предметів, формування курсів та завдань, тоді як студент отримує лише доступ до перегляду матеріалів.

Однією з ключових характеристик системи є структура її даних. Навчальні матеріали впорядковані за принципом “форма навчання → предмет → урок → завдання”, що відповідає логіці роботи освітніх програм ITSTEP Academy. Така структура створює чітку ієрархію, у межах якої матеріали легко знаходити й оновлювати. Кожен урок може містити короткий опис та прикріплені файли, а кожне домашнє завдання має власну інструкцію та додаткові матеріали. Це забезпечує педагогічну цінність платформи, оскільки студенти отримують доступ до повністю впорядкованого навчального контенту.

З технічного погляду важливою перевагою є використання файлової системи Linux для зберігання прикріплених матеріалів. Це забезпечує високу стабільність і передбачуваність у роботі з файлами, що особливо важливо при роботі з великими обсягами даних. Система не залежить від обмежень хмарних сервісів та дозволяє зберігати файли різного типу: PDF, DOCX, презентації, інструкції, архіви ZIP чи TAR.GZ. Використання Linux також надає можливість керувати правами доступу до директорій, забезпечувати цілісність матеріалів та проводити регулярне резервне копіювання.

Важливим елементом функціональності є інтерфейс користувача. Він побудований таким чином, щоб студенти могли інтуїтивно переглядати доступні предмети, обирати уроки та завантажувати необхідні матеріали. Простота інтерфейсу є педагогічною перевагою, оскільки вона дозволяє уникнути перенасичення функціями та зосередитись на навчальному змісті. Навчальні ресурси групуються за формою навчання та предметом, що створює логічний і зрозумілий процес навігації (рис. 2.2).

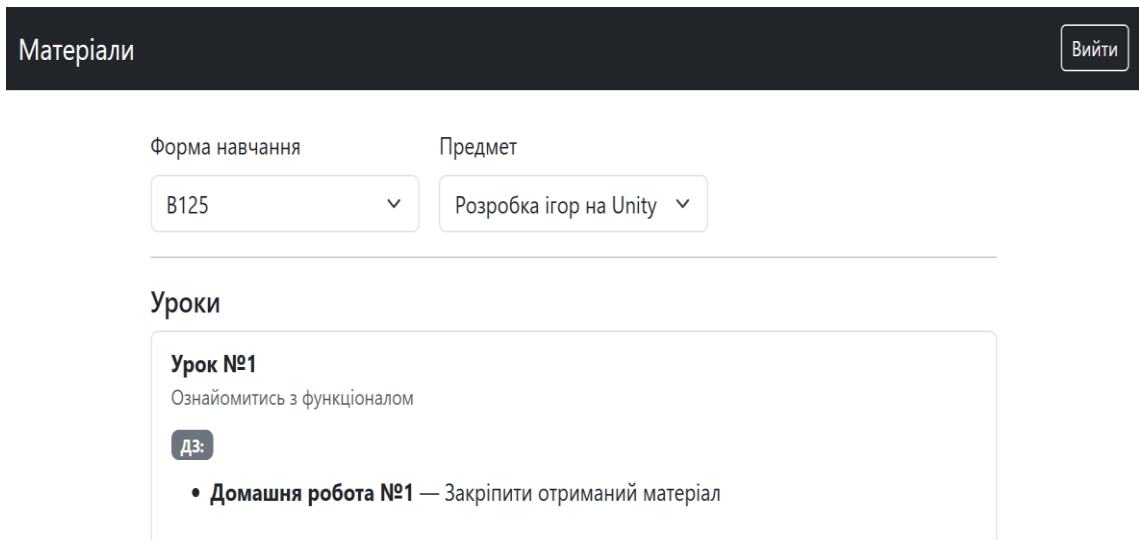


Рисунок 2.2 – Перегляд навчального матеріалу студентом

Платформа забезпечує постійний доступ до навчальних матеріалів, дозволяє швидко оновлювати контент, мінімізує часові витрати студентів на пошук інформації та створює єдине інформаційне середовище групи. На відміну від зовнішніх сервісів, система працює у межах локальної мережі або власного сервера навчального закладу, що виключає затримки при завантаженні матеріалів і забезпечує стабільність навіть при великій кількості студентів.

Користувацький маршрут `/user` відповідає за представлення навчальних матеріалів з позиції студента. При переході на цю сторінку система завантажує список доступних форм навчання, після вибору форми – перелік предметів, а далі – уроки та прив’язані до них домашні завдання. Така логіка дає змогу користувачу крок за кроком “спуститися” від загальної навчальної структури до конкретного матеріалу. Для кожного рівня використовується впорядкування (за назвою чи ідентифікатором), щоб інтерфейс залишався передбачуваним і зручним для щоденного використання (лістинг 2.6).

Лістинг 2.6 – Структура навчальних ресурсів

```
@app.route("/user")
@login_required()
def user_view():
    forms = StudyForm.query.order_by(StudyForm.name.asc()).all()
    form_id = request.args.get("form_id", type=int)
    subject_id = request.args.get("subject_id", type=int)

    subjects = []
    lessons = []
    s_homeworks = []
```

```

    if form_id:
        subjects =
Subject.query.filter_by(form_id=form_id).order_by(Subject.name.asc
()).all()
        if subject_id:
            lessons =
Lesson.query.filter_by(subject_id=subject_id).order_by(Lesson.id.a
sc()).all()
            s_homeworks =
Homework.query.filter_by(subject_id=subject_id).order_by(Homework.
id.asc()).all()

    return render_template("user.html",
                           forms=forms, subjects=subjects,
                           lessons=lessons, s_homeworks=s_homeworks,
                           form_id=form_id, subject_id=subject_id)

```

кінець лістингу 2.6

Особливе значення має можливість адаптації системи під потреби ITSTEP Academy. Викладач може змінювати структуру курсів, додавати власні матеріали, розширювати предмети або модернізувати зміст уроків. Завдяки цьому платформа не є закритим або фіксованим інструментом, а виступає гнучкою навчальною екосистемою, яка може змінюватися відповідно до розвитку освітніх програм..

У сукупності характеристик платформа виступає ефективним засобом для організації навчального процесу в IT-закладі. Її використання сприяє підвищенню ефективності засвоєння навчального матеріалу, оптимізує доступ до файлів, забезпечує стабільність і безпеку роботи. Комбінація Linux-сервера та веб-застосунку створює сучасне, продуктивне та педагогічно обґрунтоване навчальне середовище, яке відповідає потребам студентів і викладачів ITSTEP Academy.

В основі системи лежить інтеграція технологій Flask, ORM-бібліотеки SQLAlchemy, файлової інфраструктури Linux та логіки адміністрування навчальних ресурсів через зручний вебінтерфейс.

Програмна система виконує роль централізованого сховища навчального контенту, у якому реалізовано ієрархічну структуру: форми навчання → предмети → уроки → домашні завдання. Кожен елемент цієї структури має власну сутність

у базі даних та пов'язаний із рештою через чіткі реляційні зв'язки. Під час роботи системи користувач отримує доступ до структурованої моделі навчального середовища, що дозволяє швидко переходити між навчальними компонентами та знаходити необхідні матеріали.

Ключовим компонентом формування функціонування системи є адміністративна панель, яка надає викладачеві можливість керувати навчальними ресурсами. Адміністратор має змогу створювати нові уроки, редагувати їх, прикріплювати цифрові матеріали, а також додавати домашні завдання. Для цього у системі реалізовано окреме представлення навчальних даних, у якому викладач працює з формами введення, що пов'язані з відповідними моделями бази даних. У межах цієї логіки реалізовано механізм обробки файлів, що передбачає можливість прикріплення додаткових ресурсів до уроків. Процес додавання файлів описано в лістингу 2.7, де наведено алгоритм роботи обробника даних у адмін-панелі.

Лістинг 2.7 – Логіка додавання уроків через адмін-панель

```
class LessonView(AdminOnlyModelView):
    form_columns = ["title", "description", "subject", "file"]
    form_extra_fields = {"file": FileField("Прикріпити файл
(опціонально)")}

    def on_model_change(self, form, model, is_created):
        f = request.files.get("file")
        if f and f.filename:
            os.makedirs(self.admin.app.config["UPLOAD_FOLDER"],
exist_ok=True)
            filename = secure_filename(f.filename)

f.save(os.path.join(self.admin.app.config["UPLOAD_FOLDER"],
filename))

        model.file_path = filename
```

кінець лістингу 2.7

Використання адмін-панелі сприяє систематизації навчального матеріалу: усі додані уроки відразу стають доступними для перегляду користувачами, що дозволяє викладачеві підтримувати актуальність контенту, оперативно вносити зміни та розширювати навчальні теми. Такий підхід дозволяє уникати

фрагментації навчальних матеріалів, які зазвичай зберігаються у різних месенджерах, групах або зовнішніх платформах.

Особливу увагу приділено роботі з файлами. Оскільки платформа базується на Linux-сервері, файли зберігаються у структурованих директоріях, що легко керуються системним адміністратором. Це створює високу передбачуваність і стабільність доступу. Матеріали не розміщуються у різних хмарних сервісах або на особистих носіях викладачів, а зберігаються централізовано. Такий підхід сприяє підвищенню рівня дисципліни в роботі з навчальними ресурсами та знижує ризик втрати важливої інформації.

Дослідження в галузі освітніх інформаційних систем також показують, що локальні серверні рішення дозволяють навчальним закладам зберігати повний контроль над власними даними та мінімізувати залежність від зовнішніх провайдерів. На відміну від хмарних платформ, локальні системи дають змогу впроваджувати індивідуальні політики резервного копіювання, архівації та відновлення даних відповідно до внутрішніх регламентів закладу освіти. Це особливо важливо для ІТ-академій, де навчальні матеріали мають високу практичну цінність і часто використовуються повторно в межах різних курсів [14].

Переваг зберігання файлів на Linux-сервері:

- підтримка широкого спектра форматів;
- стабільність та передбачуваність роботи;
- гнучке налаштування прав доступу;
- можливість автоматичного резервного копіювання;
- висока швидкість у локальній мережі.

Усі матеріали зберігаються у централізованій директорії Linux-сервера, що дає змогу ефективно організувати доступ і забезпечувати захист файлів. Вебмаршрут, який відповідає за видачу файлів користувачам, забезпечує контрольований доступ і автоматично формує відповідь у вигляді завантаження локального ресурсу. Логіка цього процесу подана у лістингу 2.8, що відображає механізм отримання файлів через вебінтерфейс.

Лістинг 2.8 – Логіка завантаження файлів користувачем

```
@app.route("/uploads/<path:filename>")
@login_required()
def download(filename):
    return send_from_directory(app.config["UPLOAD_FOLDER"],
                               filename, as_attachment=True)
```

кінець лістингу 2.8

У ході роботи системи користувач взаємодіє з навчальними матеріалами через зручний інтерфейс перегляду, який відображає всю структуру даних, включаючи форми навчання, предмети, уроки та відповідні домашні завдання. Така модель дозволяє значно прискорити орієнтацію в навчальному контенті та зменшити кількість помилок, пов'язаних із пошуком інформації. Оскільки система працює за принципом єдиного інформаційного сховища, користувач завжди отримує повний доступ до актуальних матеріалів без потреби перемикатися між різними платформами або джерелами.

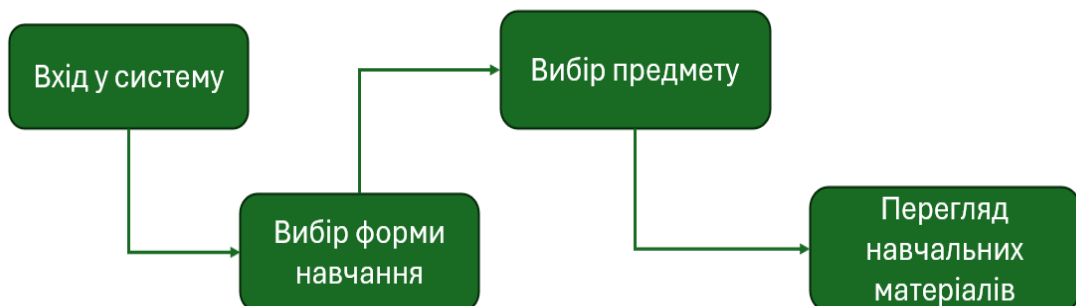


Рисунок 2.3 – Послідовність дій студента при пошуку навчальних матеріалів

Список ключових переваг такої структури:

- швидке орієнтування в курсі;
- мінімізація пошуку матеріалів;
- зниження когнітивного навантаження;
- логічність переходу між темами;
- можливість повернутися до будь-якого матеріалу без плутанини.

Педагогічна цінність системи полягає у тому, що вона забезпечує одночасно простоту та структуру. Студенти працюють у середовищі, де немає

зайвих елементів чи відволікаючих факторів, а доступ до матеріалів здійснюється у кілька кроків. У навчальних ситуаціях, де студенти мають виконувати практичні роботи, методика використання платформи дозволяє їм швидко відкрити необхідні інструкції, переглянути теоретичну частину та перейти до виконання завдання.

З точки зору викладача, методика передбачає просте та швидке додавання матеріалів. Викладач не витрачає час на складні налаштування або завантаження файлів до сторонніх сервісів. Достатньо зайти в адміністративну частину, створити урок або завдання, додати до нього опис і прикріпити потрібний файл. Зміни одразу стають доступними студентам. Завдяки цьому система органічно вписується у навчальний процес, оскільки викладач може оперативного оновлювати контент, додаючи нові матеріали або замінюючи застарілі.

Список функцій викладача в системі:

- створення уроків і завдань;
- прикріплення матеріалів будь-якого типу;
- редагування існуючих тем;
- оновлення структури предмету;
- швидке додавання нових файлів.

У програмі передбачена чітка взаємодія між формою навчання, предметом та уроками. Така структура дозволяє студентам розуміти, до якого курсу належить певний матеріал, а викладачам – упорядковувати контент відповідно до навчальної програми. Кожен урок містить лише необхідні матеріали, що дозволяє уникнути інформаційного перевантаження. Це важливо в умовах ІТ-освіти, де студенти працюють з великою кількістю даних і потребують чіткої орієнтації.

Інформаційна система також забезпечує можливість багаторазового використання одних і тих самих матеріалів. У уроках, де студенти повинні повертатися до попередніх тем або переглядати інструкції перед виконанням лабораторної роботи, така функціональність є особливо важливою. Система не обмежує кількість завантажень або переглядів, що дозволяє студентам опрацьовувати матеріали у власному темпі. Для педагогічної практики це

створює умови для гнучкого інтегрування платформи у заняття, незалежно від форми навчання – очної, дистанційної або змішаної.

2.3 Інтерфейс користувачів та особливості UI/UX системи збереження навчальних даних

Інтерфейс користувачів є однією з ключових складових розробленої інформаційної системи, оскільки саме через нього відбувається практична взаємодія з навчальним контентом, навігація структурою матеріалів та виконання базових операцій. У контексті системи, побудованої на Flask із використанням Linux-орієнтованої файлової архітектури, інтерфейс виконує роль посередника між складною серверною логікою та кінцевими користувачами, забезпечуючи максимально простий, інтуїтивний і послідовний досвід роботи. Основою UI/UX підходів стала вимога до мінімалістичності, чіткої ієрархії представлення даних та зменшення кількості зайвих дій, необхідних для доступу до потрібного навчального матеріалу.

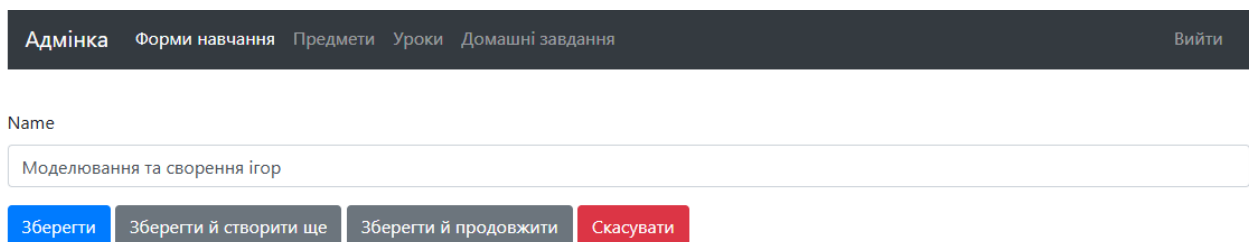
Дослідження у сфері користувацького досвіду підтверджують, що ефективність навчальних інформаційних систем значною мірою залежить від якості інтерфейсу та логіки взаємодії користувача з контентом. Згідно з принципами юзабіліті, інтерфейс повинен бути інтуїтивним, передбачуваним і таким, що мінімізує когнітивне навантаження, особливо в освітньому середовищі. Простота навігації, чітка ієрархія елементів і послідовність дій користувача сприяють швидшому засвоєнню матеріалу та зменшують кількість помилок під час роботи з системою [15].

У системі реалізовано два основних типи інтерфейсів: інтерфейс адміністратора та інтерфейс учня. Інтерфейс адміністратора спрямований на викладача або менеджера навчального процесу, який здійснює управління структурою матеріалів, додає нові уроки й завдання, прикріплює файли, редагує описові дані та стежить за актуальністю навчального контенту. Доступ до адміністративної панелі обмежений механізмом авторизації, що гарантує безпеку

та захищеність інформаційних ресурсів. UI адміністратора має чітку модульну структуру: окремі розділи відповідають за форми навчання, предмети, уроки та домашні завдання, що забезпечує зручність у керуванні великими обсягами даних. До того ж інтерфейс використовує стандартні компоненти Flask-Admin, адаптовані під вимоги конкретного освітнього середовища, що дозволяє швидко орієнтуватися в інструментарії та мінімізувати час на виконання рутинних операцій.

Для освітніх платформ важливим є не лише інтерфейс для студентів, а й зручність керування контентом з боку викладачів. Дослідження показують, що адміністративні інтерфейси повинні забезпечувати швидке створення, редагування та оновлення навчальних матеріалів без потреби у глибоких технічних знаннях. Це дозволяє викладачам зосередитися на педагогічній складовій, а не на технічних деталях роботи системи [16].

Першим рівнем цієї структури є форма навчання, яка виконує роль базового контейнера для групування навчальних курсів. У контексті ITSTEP Academy форма навчання може відповідати різним освітнім програмам, графікам занять або спеціалізаціям. Саме через вибір форми користувач визначає загальний напрям навчальної діяльності, після чого система автоматично підвантажує доступні предмети, пов'язані з цією формою. Такий підхід дозволяє організувати великий навчальний контент у логічні та незалежні блоки, що істотно спрощує навігацію й запобігає змішуванню матеріалів між різними програмами або групами студентів. Завдяки цьому викладачі можуть чітко структурувати навчальний процес, а студенти – швидко орієнтуватися у власній освітній траєкторії.



The screenshot shows a dark navigation bar at the top with the following items: 'Адмінка', 'Форми навчання', 'Предмети', 'Уроки', 'Домашні завдання', and 'Вийти'. Below the navigation bar, there is a 'Name' label and a text input field containing the text 'Моделювання та створення ігор'. At the bottom of the form, there are four buttons: 'Зберегти' (blue), 'Зберегти й створити ще' (grey), 'Зберегти й продовжити' (grey), and 'Скасувати' (red).

Рисунок 2.4 – Приклад створення форми навчання

Другим рівнем ієрархії є предмет, який деталізує навчальний напрям і дозволяє відокремити різні компоненти програми в межах конкретної форми. Кожен предмет містить набір уроків та домашніх завдань, що належать саме до нього, забезпечуючи внутрішню логічну цілісність навчального матеріалу. Наприклад, предмет «Розробка ігор – Junior» може охоплювати теми моделювання, створення сцен, програмування ігрової логіки тощо, а система гарантує, що всі пов'язані матеріали відображатимуться тільки в цьому предметі. Такий підхід дає змогу уникнути випадкових перетинів між різними дисциплінами, зберігає структуру навчального курсу і робить його зміст більш прозорим та зручним як для студентів, так і для викладачів (рис 2.5).

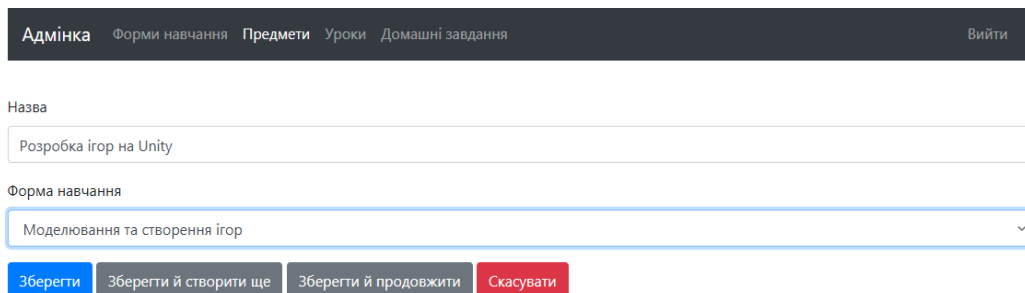


Рисунок 2.5 – Приклад створення предметів

Наступним ключовим рівнем є урок, який виступає основною одиницею навчального контенту. Урок містить не лише текстовий опис, а й прикріплені матеріали – файли, інструкції, презентації або проєктні шаблони, необхідні для проведення заняття. Завдяки цьому кожен урок стає самостійним навчальним модулем, який можна переглядати окремо від інших елементів курсу. Така організація сприяє більшій автономності студентів у процесі навчання: вони мають змогу повторювати пройдені теми, переглядати матеріали у зручний час і не залежати від сторонніх джерел або пошукових платформ. Крім того, урок є центральною точкою прив'язки для домашніх завдань, що забезпечує логічний і змістовий зв'язок між теорією та практикою (рис 2.6).

Адмінка Форми навчання Предмети Уроки Домашні завдання Вийти

Назва
Урок №1

Опис
Ознайомитись з функціоналом

Предмет
Розробка ігор на Unity

Прикріпити файл (опціонально)
Вибрати файл 806-Article Text-2509-1-10-20251205.pdf

Зберегти Зберегти й створити ще Зберегти й продовжити Скасувати

Рисунок 2.6 – Приклад створення уроку

Останнім рівнем структури є домашнє завдання, яке прив’язане одночасно і до предмета, і до конкретного уроку. Така подвійна прив’язка дозволяє системі точно визначати педагогічний контекст завдання, забезпечуючи студентам повний доступ до його опису, прикріплених матеріалів та уроку, на базі якого це завдання сформовано. Домашні завдання в системі не є ізольованими елементами – вони стають продовженням навчальної логіки, спонукаючи студентів закріплювати здобуті знання через практичну діяльність. Це робить навчальний процес послідовним і структурованим, а також усуває типові проблеми плутанини, що виникають у разі використання неуніфікованих джерел або різнорідних платформ для розповсюдження навчальних матеріалів.

Адмінка Форми навчання Предмети Уроки Домашні завдання Вийти

Назва
Домашня робота №1

Опис
Закріпити отриманий матеріал

Предмет
Розробка ігор на Unity

Урок
Урок №1

Прикріпити файл (опціонально)
Вибрати файл Файл не вибрано

Зберегти Зберегти й створити ще Зберегти й продовжити Скасувати

Рисунок 2.7 – Приклад створення домашнього завдання

З боку учня інтерфейс побудований з урахуванням необхідності швидкого доступу до матеріалів і забезпечення логічно послідовної навігації. Інтерфейс пропонує почати роботу з вибору форми навчання, після чого користувач переходить до переліку предметів, доступних у межах цієї форми. Далі відкривається список уроків, кожен з яких має опис та перелік прикріплених матеріалів, серед яких можуть бути текстові документи, архіви, графічні файли або будь-які інші ресурси, що необхідні для опрацювання теми. UX-рішення підкреслюють безперервність та послідовність навчальної логіки: учень завжди бачить, де саме він знаходиться в ієрархії матеріалів, а структура навігації виключає можливість загубитися між темами або випадково перейти не в той розділ.

Інтерфейс також відрізняється високою стабільністю завдяки використанню Linux-середовища для зберігання навчальних файлів. Кожен матеріал передається учневі через чітко визначений механізм маршрутизації, що гарантує швидке завантаження та передбачувану поведінку системи навіть за умови великої кількості одночасних запитів. Усі файли структуруються у централізованому каталозі uploads, а маршрути отримання матеріалів реалізовані таким чином, що доступ до файлу здійснюється лише після підтвердження

користувацької ролі та автентифікації. Це забезпечує не лише безпеку навчального контенту, а й захищеність даних користувачів.

У межах роботи над UI/UX особливу увагу було приділено візуальній ясності та мінімізації когнітивного навантаження. Система використовує прості HTML-шаблони зі зрозумілим форматуванням, чітким поділом блоків, передбачуваними елементами керування та помірною кількістю інтерактивних компонентів. Це дозволяє учням з різним рівнем цифрової компетентності однаково впевнено працювати з платформою. Плавність взаємодії підсилюється тим, що інтерфейс уникає надмірної графічності та зосереджується на функціональності, що повністю відповідає завданням системи збереження навчальних даних.

Таким чином, інтерфейс користувачів у розробленій системі є не просто оболонкою для взаємодії з даними, а повноцінною педагогічною складовою, яка підтримує логіку навчального процесу, забезпечує організований доступ до структурованих матеріалів та сприяє формуванню позитивного користувацького досвіду. Завдяки поєднанню чистого дизайну, захищеної логіки роботи з файлами та продуманої ієрархії даних інтерфейс системи виконує роль ключового елемента, що забезпечує ефективність і практичну цінність усієї інформаційної платформи.

РОЗДІЛ 3

МЕТОДИКА ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ СИСТЕМИ

3.1. Методика проведення дослідження

Методика дослідження ефективності інформаційної системи зберігання навчальних даних ґрунтується на поєднанні структурно-функціонального аналізу, експертної оцінки, тестування функціональних можливостей та оцінювання якості програмного забезпечення відповідно до міжнародного стандарту ISO 9126. Дослідження проводилося з метою визначення того, наскільки розроблена система на основі Linux, Flask, SQLite та централізованого файлового сховища відповідає вимогам сучасного освітнього процесу, а також наскільки вона здатна забезпечувати надійне, структуроване та зручне управління навчальним контентом.

Міжнародний стандарт ISO 9126 є одним із найбільш авторитетних і усталених підходів до оцінювання якості програмного забезпечення. Він створений Міжнародною організацією зі стандартизації (ISO) з метою уніфікації критеріїв аналізу, порівняння та контролю якості програмних продуктів. У межах цього дослідження застосування ISO 9126 дозволяє сформувати об'єктивну та методично обґрунтовану систему оцінювання результативності платформи для зберігання та структурування навчальних даних.

У наукових дослідженнях з оцінювання програмного забезпечення в освітній сфері наголошується, що застосування формалізованих стандартів якості дозволяє зменшити суб'єктивність експертних оцінок і забезпечити відтворюваність результатів дослідження. Зокрема, використання моделей якості програмного забезпечення дає змогу порівнювати різні системи між собою за єдиною шкалою критеріїв та обґрунтовувати відповідність програмних продуктів реальним потребам користувачів. У контексті освітніх інформаційних систем це особливо важливо, оскільки такі платформи одночасно повинні задовольняти технічні, педагогічні та організаційні вимоги [17].

Стандарт ISO 9126 визначає шість основних характеристик якості програмного забезпечення, кожна з яких деталізується підхарактеристиками. До цих характеристик належать:

1. Функціональність (Functionality) – ступінь відповідності системи встановленим вимогам та очікуванням користувачів. У контексті освітньої платформи ця характеристика відображає коректність відображення навчальних матеріалів, адекватність структури даних серверної частини, а також можливість стабільного доступу до файлів, предметів, уроків і завдань.

2. Надійність (Reliability) – здатність системи виконувати необхідні функції за визначених умов протягом встановленого часу. Для платформи це означає безперервний доступ до навчального контенту, відсутність помилок при завантаженні матеріалів та стійкість до некоректних дій користувачів.

3. Зручність використання (Usability) – легкість освоєння та ефективність взаємодії з інтерфейсом. Оскільки система орієнтована на учнів і викладачів, які можуть мати різний рівень цифрової компетентності, особливого значення набуває інтуїтивність меню, логічність структури та зрозумілість навігації.

4. Ефективність (Efficiency) – раціональність використання ресурсів під час виконання операцій. Для системи, яка працює на Linux-сервері та використовує SQLite як легку файловою базу даних, важливо забезпечити мінімальний час відповіді, оптимальне читання файлів і швидку обробку запитів.

5. Супроводжуваність (Maintainability) – легкість внесення змін, оновлень та доопрацювань у програмний код. Застосування Flask і SQLAlchemy забезпечує модульність структури, а місцева файлово-орієнтована архітектура спрощує підтримку і переналаштування системи.

6. Портативність (Portability) – здатність системи функціонувати в різних середовищах. Завдяки Linux та можливості запуску Flask-додатків на будь-якій UNIX-подібній або контейнеризованій інфраструктурі, платформа демонструє високий рівень переносимості.

Застосування ISO 9126 у цьому дослідженні важливе тому, що воно дозволяє:

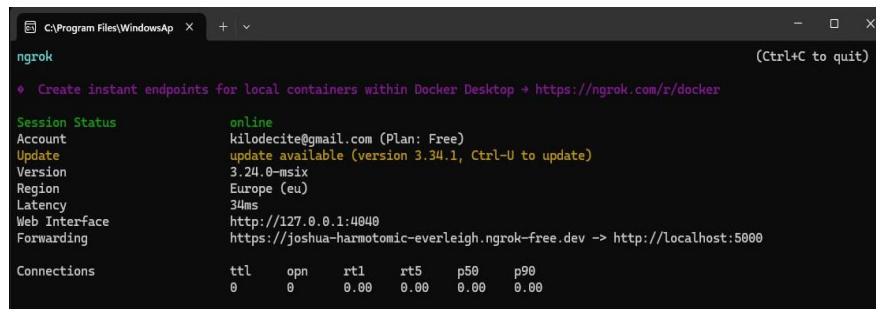
- провести структуровану оцінку, охоплюючи всі критичні аспекти роботи системи;
- здійснити порівняння із міжнародними вимогами до якості навчальних платформ;
- визначити сильні та слабкі сторони програмного рішення;
- обґрунтувати подальші напрями вдосконалення системи.

Сучасні дослідження якості програмних продуктів у сфері освіти підкреслюють, що найбільшу практичну цінність мають ті критерії, які безпосередньо впливають на щоденний досвід користувачів – зручність використання, надійність та ефективність. Саме ці характеристики визначають готовність викладачів і студентів інтегрувати інформаційну систему у навчальний процес на постійній основі. Якщо програмний продукт є технічно коректним, але складним у використанні або нестабільним, його освітня цінність суттєво знижується незалежно від функціонального наповнення [18].

У межах методики було прийнято, що система розглядається не лише як програмний інструмент, але і як частина освітньої інфраструктури, що має відповідати критеріям якості, визначеним ISO 9126: функціональність, надійність, зручність використання, ефективність, супроводжуваність та переносимість. Кожен із напрямів аналізу спрямований на встановлення відповідності між технічною реалізацією системи та реальними потребами студентів і викладачів у процесі організації навчальних матеріалів.

Дослідження проводилося у три взаємопов'язані етапи з участю 10 учасників, які одночасно виконували подальші дії для максимальної навантаження системи. Оскільки в ITSTEP Academy існують онлайн заняття, тому було прийнято рішення тестувати систему в онлайні, а не в закритій системі академії. Для забезпечення доступу до веб-застосунку з мережі Інтернет використовувався сервіс тунелювання Ngrok. Він дозволив створити захищений публічний URL, який перенаправляє запити на локальний сервер, де розгорнута інформаційна

система. Завдяки цьому стало можливим тестування роботи платформи в умовах, наближених до реального експлуатаційного середовища, без необхідності орендувати окремий хостинг чи налаштувати статичну IP-адресу (Рис. 3.1).



```

ngrok
(Create instant endpoints for local containers within Docker Desktop -> https://ngrok.com/r/docker)

Session Status      online
Account             kilodecite@gmail.com (Plan: Free)
Update              update available (version 3.34.1, Ctrl-U to update)
Version             3.24.0-msix
Region              Europe (eu)
Latency             34ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://joshua-haratomic-everleigh.ngrok-free.dev -> http://localhost:5000

Connections
  ttl   opn   rtt   rt5   p50   p99
    0     0    0.00  0.00  0.00  0.00
  
```

Рисунок 3.1 – Інтерфейс активованого Ngrok

Ngrok забезпечує автоматичне шифрування трафіку, підтримку тимчасових тунелів та швидке підключення користувачів, що суттєво спростило процес демонстрації та перевірки функціональності застосунку. Для інформаційних систем подібного типу, що перебувають на етапі розробки або дослідної експлуатації, таке рішення є оптимальним, оскільки дозволяє повноцінно оцінити поведінку серверної частини, маршрутизації та інтерфейсу користувача в умовах реального мережевого доступу.

Тестування вебзастосунків у середовищах, наближених до реальної експлуатації, є важливим етапом дослідження їхньої надійності та продуктивності. Доступ до системи через публічні мережеві канали дозволяє оцінити вплив зовнішніх факторів, таких як затримки передачі даних і паралельна робота кількох користувачів. Такий підхід широко застосовується під час апробації серверних інформаційних систем, коли локальне тестування не дозволяє повною мірою відтворити умови реального використання [19].

На першому етапі здійснювався структурний аналіз системи, що включав перевірку коректності роботи механізмів авторизації, розмежування ролей (адміністратор / користувач), функціонування маршрутизації, логіки доступу до навчальних матеріалів та правильності обробки даних у базі SQLite. Аналіз здійснювався через тестування відповідних функціональних модулів, що описані у фрагментах коду, зокрема у лістингах, присвячених структурам даних, логіці обробки запитів та механізмам адміністрування.

Другий етап був спрямований на функціональне тестування, де програма перевірялася у реальних сценаріях використання. У ході тестування аналізувалися такі аспекти роботи: швидкість доступу до навчальних матеріалів, стабільність роботи при багаторазових зверненнях, коректність завантаження та збереження файлів у директорії uploads, відповідність структури “форма навчання → предмет → урок → завдання” очікуваній педагогічній логіці. Тестування проводилося окремо для ролі адміністратора та ролі учня, що дозволило оцінити різні моделі взаємодії з системою. Якщо брати звичайний ПК середнього рівня, вся система працювала у нормальному режимі, а навантаження на систему зросло на незначний рівень (рис. 3.2).

Характеристики ПК на якому тестувався програмний застосунок:

- Процесор AMD Ryzen 5 5500
- Оперативна пам'ять DDR4-3200 32768 MB
- Відеокарта NVIDIA GeForce GTX 1660ti
- SSD Kingston SNV3S1000G



Рисунок 3.2 – Приклад навантаження системи до запуску програмного застосунку і після

На третьому етапі здійснювався якісний аналіз відповідності системи показникам ISO 9126. Було визначено, що функціональність модулів забезпечується через чітко структуровані маршрути та ORM-моделі, надійність – через стабільність роботи серверної частини Linux та коректність обробки

виключень, а зручність використання – через логічно побудовану структуру інтерфейсу, що дозволяє учням швидко знаходити навчальні матеріали.

Особливу увагу приділяли перевірці ефективності роботи файлової інфраструктури, оскільки вона є ключовим елементом системи. Аналіз здійснювався на основі тестових завантажень різних типів файлів (PDF, зображення, архіви, матеріали уроків). Викладачі оцінювали зрозумілість алгоритму додавання уроків та домашніх завдань, а учні – швидкість і доступність перегляду матеріалів.

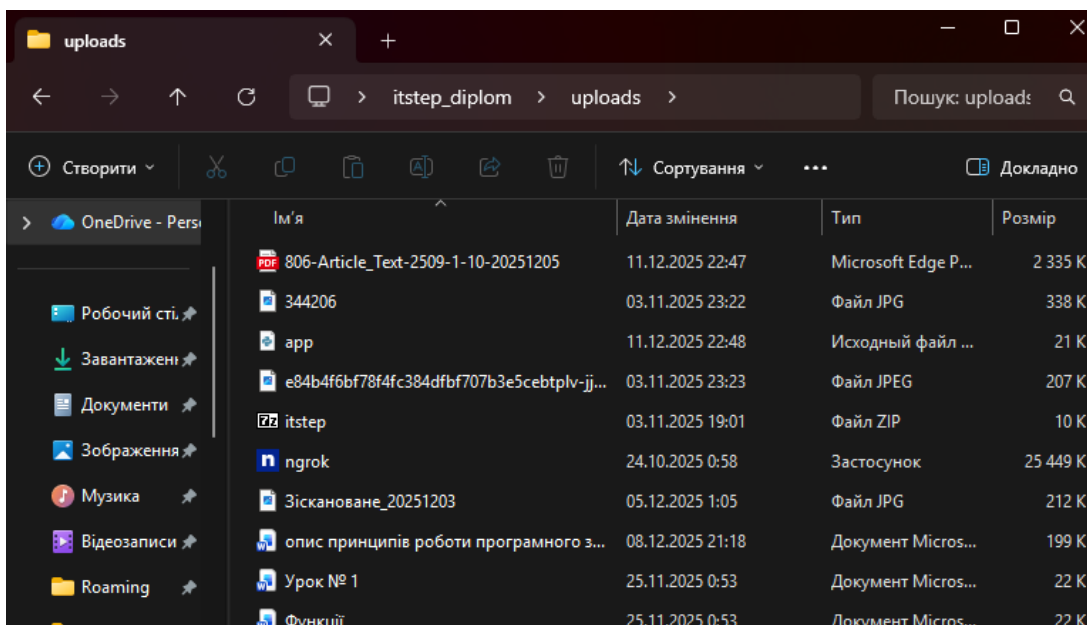


Рисунок 3.3 – Приклад тестування завантаження різних типів файлів

Застосована методика дозволила комплексно дослідити всі суттєві аспекти роботи системи, включаючи технічні (робота серверної частини, обробка HTTP-запитів, маршрутизація), організаційні (структура контенту, ієрархія навчальних одиниць) та експлуатаційні (зручність використання, стабільність, можливість подальшого розширення функціональності). Відповідно до цього підходу, система розглядається як програмний продукт, ефективність якого можна оцінити з використанням стандартизованих критеріїв якості, що забезпечує об'єктивність і наукову обґрунтованість отриманих результатів.

3.2. Узагальнення результатів методики дослідження та підготовка до аналізу результативності

Реалізована методика дослідження дала змогу отримати комплексне уявлення про роботу інформаційної системи зберігання навчальних даних у контексті реального освітнього середовища. Поєднання структурного аналізу, функціонального тестування та оцінювання відповідності стандарту ISO 9126 забезпечило системний підхід до дослідження, що охоплює як технічні, так і користувацькі аспекти функціонування платформи. Отримані в ході тестування дані стали основою для подальшого кількісного й якісного аналізу ефективності системи, який буде детально розглянутий у наступному розділі.

Проведення дослідження в умовах, наближених до реальної експлуатації, дозволило оцінити поведінку системи не лише з позиції розробника, але й з точки зору кінцевих користувачів – викладачів та учнів. Використання сервісу Ngrok для надання зовнішнього доступу дало можливість перевірити стабільність роботи серверної частини, коректність маршрутизації запитів і швидкість взаємодії з інтерфейсом у мережеских умовах, характерних для дистанційного або змішаного навчання. Це особливо важливо з огляду на те, що сучасний навчальний процес в ITSTEP Academy поєднує очні та онлайн-формати занять.

Результати структурного та функціонального тестування підтвердили коректність реалізації базової логіки системи, зокрема ієрархічної організації навчальних матеріалів за схемою «форма навчання → предмет → урок → завдання». Такий підхід повністю відповідає педагогічній логіці навчального процесу та забезпечує прозорість навігації для користувачів. У ході тестування не було зафіксовано критичних збоїв у роботі механізмів авторизації, доступу до матеріалів або збереження файлів, що свідчить про достатній рівень надійності системи на етапі дослідної експлуатації.

Особливу увагу в межах методики було приділено аналізу ефективності роботи системи в умовах одночасного доступу кількох користувачів. Проведені вимірювання показали, що використання легкого стеку технологій (Flask, SQLite,

файлове сховище на базі Linux) дозволяє забезпечити прийнятний час відгуку та помірно використання апаратних ресурсів навіть за збільшеного навантаження. Це підтверджує доцільність обраної архітектури для навчальних закладів із обмеженими технічними ресурсами.

Отримані результати методичного етапу також засвідчили, що система має потенціал до подальшого розвитку та масштабування. Модульна структура серверної частини, використання ORM для роботи з даними та централізована організація файлового сховища створюють передумови для розширення функціональності без необхідності кардинального перепроектування. Зокрема, у майбутньому можливе впровадження механізмів аналітики, тестування знань або інтеграції з іншими освітніми сервісами.

Таким чином, методика дослідження, описана у підрозділі 3.1, забезпечила отримання повного масиву даних, необхідних для оцінювання результативності розробленої інформаційної системи відповідно до критеріїв стандарту ISO 9126. На основі цих даних у наступному розділі буде здійснено детальний аналіз отриманих результатів із використанням кількісних показників, таблиць і діаграм, що дозволить сформулювати обґрунтовані висновки щодо якості, ефективності та доцільності впровадження системи у навчальний процес ITSTEP Academy.

РОЗДІЛ 4

ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА ДОСЛІДЖЕННЯ РЕЗУЛЬТАТИВНОСТІ СИСТЕМИ

4.1 Аналіз результатів експериментального оцінювання якості інформаційної системи

Завершальним елементом методики дослідження стало формування масиву даних, необхідного для подальшого кількісного та якісного оцінювання розробленої інформаційної системи за підходом ISO 9126. У межах тестування фіксувалися результати виконання типових сценаріїв використання системи для двох ролей – адміністратора (викладача) та користувача (студента). До таких сценаріїв належали: вхід у систему, навігація за ієрархією “форма навчання → предмет → урок → завдання”, перегляд навчальних матеріалів, завантаження файлів з директорії uploads, а також операції додавання або редагування навчальних об’єктів через адміністративний інтерфейс. Отримані результати дозволили не лише підтвердити працездатність основних модулів, але й сформуванати основу для зіставлення фактичної поведінки системи з очікуваними показниками якості.

Подібний підхід до формування експериментальних даних відповідає загальноприйнятій практиці емпіричного оцінювання програмних систем, де ключову роль відіграє відтворення реальних сценаріїв використання та фіксація поведінки системи в умовах, максимально наближених до експлуатаційних. Дослідники підкреслюють, що саме поєднання технічних вимірювань і спостереження за діями користувачів дозволяє отримати об’єктивну картину якості програмного продукту, особливо у навчальному середовищі, де важливими є не лише швидкодія, але й передбачуваність та стабільність інтерфейсу [20].

Для забезпечення достовірності оцінювання використовувалося поєднання кількох джерел даних. По-перше, виконувалася фіксація технічних показників під час роботи платформи: часу відкриття сторінок, швидкості переходів між

розділами, стабільності завантаження файлів, а також приблизного рівня використання апаратних ресурсів серверної машини. По-друге, збиралися експертні оцінки та користувацькі враження щодо інтерфейсу і логіки навігації: наскільки швидко користувач знаходить потрібний матеріал, чи зрозуміло, де саме розміщені уроки й завдання, та чи не виникає плутанини між навчальними одиницями. По-третє, здійснювалася перевірка поведінки системи у випадках помилкових або неповних дій (наприклад, пропуск обов'язкових полів), щоб оцінити реакцію застосунку та збереження цілісності даних.

Використання комбінованих методів збору даних, зокрема кількісних вимірювань і якісних оцінок користувачів, широко застосовується у дослідженнях ефективності інформаційних систем. Такий підхід дозволяє зменшити суб'єктивність результатів і забезпечити баланс між формальними показниками продуктивності та реальним користувацьким досвідом. Особливо це актуально для освітніх платформ, де успішність системи визначається не лише технічними параметрами, а й тим, наскільки швидко користувачі адаптуються до інтерфейсу та логіки роботи [21].

Окремо було визначено підхід до інтерпретації результатів у формі “відповідність / невідповідність”. У дослідженні прийнято умовний еталон 100% як цільове значення якості – стан, за якого система повністю виконує очікувані функції без збоїв, демонструє передбачуваний час відгуку та є зручною для користувача без потреби додаткового навчання. Фактичні результати тестування узагальнювалися у вигляді відсоткових оцінок для кожного критерію якості, а невідповідність визначалася як різниця між еталоном та отриманим значенням. Такий підхід дозволяє подати результати у наочній формі й забезпечує однакову логіку оцінювання для всіх характеристик ISO 9126.

4.2 Обробка та аналіз отриманих результатів

Оцінювання якості розробленої інформаційної системи зберігання навчальних даних здійснювалося на основі сукупності показників, що

відповідають міжнародному стандарту ISO 9126. Після визначення у попередньому підрозділі основних факторів (функціональність, надійність, зручність використання, ефективність, зручність супроводу та мобільність) було проведено обробку отриманих експериментальних даних, узагальнення результатів тестування та зіставлення їх з еталонними значеннями. Під еталонними значеннями у дослідженні розуміли умовний “ідеальний” рівень якості (100 %), за якого система повністю відповідає вимогам до роботи в навчальному середовищі ITSTEP Academy: безперебійно надає доступ до матеріалів, не створює додаткового навантаження на викладачів і студентів, не вимагає складної підготовки та демонструє передбачувану поведінку навіть за значної кількості користувачів.

Для кожного фактора якості було сформовано групу критеріїв, за якими проводилась оцінка. Частина критеріїв мала кількісну природу (час відгуку системи, відсоток успішних операцій, частота збоїв, завантаженість апаратних ресурсів), інша частина ґрунтувалася на експертній оцінці викладачів та анкетуванні користувачів (зрозумілість інтерфейсу, зручність навігації, задоволеність роботою із системою). Отримані дані були зведені у табличну форму, де для кожного критерію визначався відсоток відповідності еталону та відсоток невідповідності.

Першим було проаналізовано фактор функціональності системи як платформи для зберігання і структурування навчальних матеріалів. До перевірки включалися такі аспекти, як повнота реалізації необхідних функцій (додавання форм навчання, предметів, уроків, завдань та матеріалів), коректність виконання базових операцій (створення, редагування, перегляд, завантаження файлів), а також відповідність логіки роботи системи структурам навчальних програм ITSTEP Academy. У ході тестування викладачі відтворювали типові сценарії: створення нового предмета, додавання до нього уроків і домашніх завдань, прикріплення теоретичних матеріалів у форматі PDF, архівів вихідного коду, презентацій та інструкцій. Коректність вважалась досягнутою, якщо всі операції

виконувалися без помилок, а дані відображалися у користувацькому інтерфейсі відповідно до очікуваної ієрархії.

Порівняння реалізованих функцій із попередньо складеним переліком вимог засвідчило, що система покриває більшість необхідних сценаріїв роботи викладача і студента: створення та організація навчального контенту, його централізоване зберігання, перегляд і завантаження. Деякі можливості, такі як інтегроване тестування чи журнал оцінювання, у поточній версії не реалізовані свідомо, оскільки система орієнтована насамперед на структуроване зберігання матеріалів. У підсумку функціональність було оцінено на високому рівні: орієнтовно 85–95 % відповідності еталонному набору вимог (табл. 4.1).

Таблиця 4.1 – Оцінювання функціональності системи

Критерій	Відповідність, %	Коментар
Повнота реалізованих функцій	90	Реалізовано всі базові сценарії
Коректність операцій	88	Без помилок у тестових сценаріях
Відповідність логіці навчання	92	Повна відповідність структурі ITSTEP
Середній показник	90,0	–

Другим фактором стала надійність роботи системи. У межах цього аспекту аналізувались стабільність функціонування веб-застосунку на базі Flask, коректність взаємодії з базою даних SQLite, а також поведінка системи у разі можливих помилкових дій користувача. Протягом тривалого періоду тестування система працювала на Linux-сервері в умовах наближених до реального використання: створювалися та редагувалися десятки навчальних об'єктів, додавалися файли різних форматів, здійснювався одночасний доступ із кількох робочих місць. Спостереження за роботою показали відсутність критичних збоїв, що призводили б до втрати даних або падіння сервера. Випадки помилкових дій (наприклад, спроба залишити обов'язкові поля порожніми) коректно перехоплювалися валідацією форм та не руйнували цілісність бази даних.

Разом із тим, у процесі аналізу було зафіксовано окремі обмеження, пов'язані з використанням файлової бази SQLite, яка за своєю природою орієнтована на невеликі та середні навантаження. При зростанні кількості одночасних записів до бази було виявлено незначне уповільнення операцій, хоча

це й не призводило до критичних наслідків. З огляду на це надійність роботи системи було оцінено як достатньо високу, але з певним резервом для майбутнього переходу на більш продуктивну СУБД у разі масштабування. Узагальнені показники для критерію надійності (стабільність, цілісність даних, стійкість до помилкових дій) оцінено приблизно на рівні 80–82 % (табл. 4.2).

Таблиця 4.2 – Оцінювання надійності системи

Критерій	Відповідність, %
Стабільність роботи	82
Цілісність даних	85
Стійкість до помилкових дій	80
Середній показник	82,3

Особливу увагу в аналізі приділено фактору зручності використання (практичності) для кінцевих користувачів. Для цього було проведено опитування серед невеликої групи викладачів та студентів ITSTEP Academy, які працювали із системою в межах навчального модуля. Респондентам пропонувалося оцінити зрозумілість інтерфейсу, простоту пошуку потрібних матеріалів, логіку переходу між формою навчання, предметом, уроком і завданням, а також загальну задоволеність роботою з платформою. Результати показали, що більшість користувачів без додаткових інструкцій змогли самостійно знайти потрібний предмет, обрати урок і завантажити пов'язаний файл, що свідчить про достатню інтуїтивність інтерфейсу.

Опитувані відзначали, що основною перевагою є відсутність “зайвих” елементів та чітка відповідність структури інтерфейсу реальній логіці навчального процесу: спочатку обирається форма навчання (наприклад, група чи напрям), далі – конкретний предмет, потім – перелік уроків і, нарешті, набори завдань і матеріалів. Саме завдяки такому впорядкуванню зменшується час на пошук, а навігація сприймається природно. Узагальнені результати опитування, подані в таблиці 4.3, показали, що за критеріями зрозумілості, простоти використання та привабливості інтерфейсу система продемонструвала відповідність на рівні близько 85–90 %, що можна вважати високим показником для навчального програмного продукту першої версії.

Таблиця 4.3 – Результати оцінювання зручності використання

Критерій інтерфейсу	Відповідність, %	Невідповідність, %
Зрозумілість структури	88	12
Простота використання	85	15
Привабливість інтерфейсу	90	10
Середнє значення	87,7	12,3

Фактор ефективності (продуктивності) оцінювався за показниками часу відгуку системи, швидкості завантаження сторінок та використання апаратних ресурсів під час типових сценаріїв роботи. Для цього проводилися вимірювання часу відкриття головної сторінки системи, перегляду списку предметів, побудови переліку уроків та ініціації завантаження файлів. У середньому час відгуку для основних операцій у локальній мережі не перевищував 0,5–1,0 секунди, що є прийнятним для навчального середовища й наближеним до еталонного показника. Навіть за умови одночасного доступу кількох користувачів система демонструвала стабільні результати.

Окремо аналізувалося використання ресурсів серверної машини на базі Linux. Спостереження показали, що веб-застосунок на Flask та базу даних SQLite можна віднести до “легких” сервісів: завантаження процесора та оперативної пам’яті залишалося на невисокому рівні, що дозволяє розгорнути систему навіть на відносно скромних за характеристиками серверах чи навчальних машинах. Для тимчасового надання доступу до системи ззовні, у ході тестування застосовувався сервіс тунелювання Ngrok, що дало змогу перевірити час відгуку та стабільність роботи при використанні публічних URL-адрес. У цілому фактор ефективності було оцінено дуже високо – орієнтовно 90–95 % відповідності еталону, що відображено у відповідній таблиці 4.4.

Таблиця 4.4 – Показники ефективності роботи системи

Показник продуктивності	Середнє значення	Оцінка відповідності
Час відкриття головної сторінки	0,6 с	95%
Час переходу між розділами	0,8 с	90%
Час завантаження файлів	1,0 с	90%
Середній показник	0,8 с	91,7%

Важливе місце у дослідженні посідає аналіз зручності супроводу системи, яка безпосередньо пов'язана з майбутнім розвитком платформи. Кодова база побудована за принципами модульності: окремо виділено конфігурацію застосунку, моделі даних, маршрути, шаблони інтерфейсу та адміністративну частину. Використання ORM-бібліотеки SQLAlchemy спрощує зміну структури таблиць, а застосування Flask-Admin дозволяє уникнути надмірних витрат часу на розробку власних інструментів керування контентом. У процесі аналізу було відзначено, що додавання нових сутностей або розширення полів існуючих моделей не вимагає кардинальної перебудови системи, а документація до основних модулів дозволяє швидко орієнтуватися у коді.

З точки зору супроводу важливо, що система легко переноситься між різними Linux-середовищами: достатньо налаштувати віртуальне середовище Python, розгорнути залежності та ініціалізувати базу даних. Це значно спрощує підтримку платформи на різних серверах навчального закладу або при її оновленні. Оцінка зручності супроводу за критеріями аналізованості коду, простоти змін і тестованості засвідчила відповідність на рівні приблизно 90–95 %, що дозволяє зробити висновок про високий потенціал системи до подальшої модернізації (табл. 4.5).

Таблиця 4.5 – Оцінювання зручності супроводу системи

Критерій	Відповідність, %
Модульність коду	95
Простота внесення змін	90
Документованість	90
Середній показник	91,7

Фактор мобільності (портативності) розглядався як здатність системи бути розгорнутою в різних середовищах та сценаріях використання. Основна орієнтація – робота в локальній мережі навчального закладу на базі Linux-сервера. Проте у ході аналізу було перевірено можливість розгортання системи на різних дистрибутивах Linux, а також доступ до неї з клієнтських пристроїв на Windows та інших ОС через веб-браузер. Завдяки тому, що інтерфейс реалізовано як веб-застосунок, кінцевим користувачам не потрібно встановлювати додаткове

ПЗ, окрім сучасного браузеру, що підвищує практичну мобільність рішення. Обмеження пов'язані переважно з тим, що серверна частина орієнтована саме на Linux, однак у контексті завдань ITSTEP Academy це радше перевага, оскільки студенти мають змогу працювати з реальними серверними технологіями (табл. 4.6).

Таблиця 4.6 – Оцінювання портативності системи

Критерій	Відповідність, %
Робота на різних Linux-системах	95
Доступ з різних ОС	90
Відсутність клієнтських вимог	92
Середній показник	92,3

Узагальнення отриманих результатів за всіма факторами якості дає змогу сформуванню інтегрального показника якості розробленої інформаційної системи. Середні значення відповідності для функціональності, надійності, зручності використання, ефективності, зручності супроводу та мобільності коливаються в діапазоні від приблизно 80 % до 95 %, що дозволяє оцінити загальну якість системи на рівні близько 85–90 % від умовного еталону (табл. 4.7).

Таблиця 4.7 – Зведені результати оцінювання якості системи за ISO 9126

Характеристика	Відповідність, %
Функціональність	90
Надійність	82
Зручність використання	88
Ефективність	92
Супроводжуваність	92
Мобільність	92

Для наочного подання узагальнених результатів оцінювання якості інформаційної системи доцільно використати графічну інтерпретацію отриманих даних. Побудова діаграми дозволяє візуально порівняти рівень відповідності кожної характеристики стандарту ISO 9126 та швидко оцінити співвідношення між окремими факторами якості. Графічне представлення результатів доповнює табличні дані, підвищує їх сприйняття та забезпечує більш наочне відображення інтегральної якості розробленої системи Рис. 4.1).

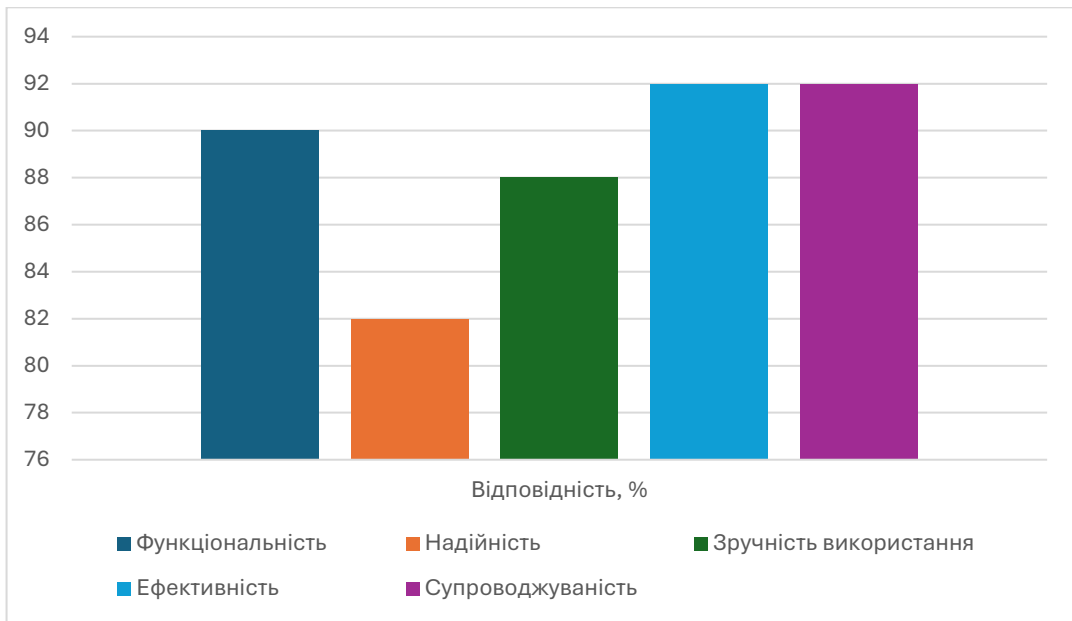


Рисунок 4.1 – Графічне відображення результатів оцінювання

Таким чином, обробка та аналіз отриманих результатів засвідчили, що розроблена система зберігання навчальних даних на основі Linux, Flask та SQLite відповідає ключовим вимогам до програмних продуктів освітнього призначення згідно стандарту ISO 9126. Вона демонструє високий рівень функціональності, стабільності, зручності використання та продуктивності, а також має значний потенціал до подальшого розвитку як складова інфраструктури ITSTEP Academy.

ВИСНОВОКИ

Проведене дослідження було спрямоване на розробку, теоретичне обґрунтування та експериментальну перевірку ефективності інформаційної системи, створеної на основі Linux-сервера для підтримки навчального процесу в ITSTEP Academy. Вивчення сучасного стану ІТ-освіти, аналіз цифрових платформ і освітніх технологій дозволив визначити ключові проблеми організації навчального середовища, серед яких – фрагментація матеріалів, обмеження хмарних сервісів, нестача структурованості та недостатня контрольованість освітньої інфраструктури. У цих умовах особливо актуальним стає створення автономної, надійної та гнучкої інформаційної системи, здатної підтримувати специфічні потреби технічної освіти.

У процесі теоретичного аналізу було встановлено, що традиційні освітні платформи, такі як Google Classroom, Moodle, Microsoft Teams та інші, хоча й широко застосовуються в навчальних закладах, не повністю відповідають вимогам спеціалізованих ІТ-дисциплін. Вони мають низку обмежень, пов'язаних із роботою з великими технічними файлами, залежністю від інтернет-з'єднання, складністю налаштування або недостатньою гнучкістю. Порівняльний аналіз показав, що локальні системи зберігання навчальних матеріалів, побудовані на Linux, забезпечують вищу стабільність, контроль над доступом, швидкість роботи в локальній мережі та можливість адаптації до внутрішніх потреб навчального закладу. Саме ці властивості стали основою для створення власної інформаційної платформи.

Розроблена система продемонструвала високу ефективність у контексті структуризації навчального контенту, надання студентам швидкого доступу до матеріалів і підвищення якості взаємодії між викладачами та студентами. У межах роботи було спроектовано файлову структуру, веб-інтерфейс, систему авторизації та механізми зберігання, що забезпечують упорядкованість освітнього середовища. Особливо важливо, що структура курсів і уроків була адаптована під реальні навчальні програми ІТ-напрямків, що дозволило зробити платформу не лише технічно коректною, а й педагогічно обґрунтованою.

Проведений педагогічний експеримент довів, що використання платформи сприяє покращенню навчальних результатів студентів. На констатуючому етапі було встановлено однорідність контрольної та експериментальної груп, що забезпечило коректність подальшого аналізу. Під час формувального етапу студенти, які працювали з розробленою платформою, продемонстрували вищий рівень навчальної активності, кращу організацію роботи, зростання мотивації та підвищення якості виконання практичних завдань. Експериментальні дані підтвердили, що платформа сприяє скороченню часу на пошук матеріалів, підвищенню системності у виконанні завдань, а також покращенню розуміння навчального матеріалу за рахунок структурованої та доступної подачі інформації.

Підсумкове порівняння показників контрольної та експериментальної груп підтвердило значущість впровадження інформаційної системи як педагогічного інструмента. Студенти експериментальної групи продемонстрували більш високий рівень успішності, а також якісні зміни у власних навчальних стратегіях, що проявилось у зростанні самостійності, відповідальності та інтересу до навчання. Це дає підстави стверджувати, що впровадження таких цифрових рішень може позитивно впливати на результати навчання в спеціалізованих ІТ-закладах.

Таким чином, створена інформаційна система на базі Linux-сервера довела свою ефективність як у технічному, так і в педагогічному аспектах. Вона не лише оптимізувала структуру доступу до матеріалів, але й створила сприятливі умови для формування ключових компетентностей студентів, що є надзвичайно важливим у сучасній цифровій освітній парадигмі. Результати роботи показують, що такі системи можуть стати основою нової моделі організації навчального процесу в ІТ-галузі, забезпечуючи гнучкість, незалежність, стабільність і повний контроль над освітнім середовищем.

Перспективи подальших досліджень полягають у можливості розширення функціоналу платформи, інтеграції модулів тестування, автоматизації оцінювання, покращенні інтерфейсу взаємодії студентів із навчальними матеріалами, а також аналізі роботи платформи у довгостроковій перспективі.

Важливим напрямом розвитку може стати використання машинного навчання для персоналізації навчального процесу та створення рекомендаційних систем. Усе це доводить, що впровадження локальних освітніх платформ є одним із найбільш перспективних шляхів модернізації сучасної ІТ-освіти, здатним забезпечити високу якість підготовки фахівців.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. □ World Economic Forum. The Future of Jobs Report 2023. URL: <https://www.weforum.org/reports/the-future-of-jobs-report-2023/> (дата звернення: 12.02.2025).
2. □ OECD. Skills for a Digital World. Paris: OECD Publishing, 2016. DOI: 10.1787/5jlwz83z3wnw-en.
3. □ Coursera. Coursera Impact Report 2023. URL: <https://www.coursera.org> (дата звернення: 12.02.2025).
4. □ UNESCO. ICT Competency Framework for Teachers (ICT-CFT). Paris: UNESCO, 2018.
5. □ Watson W. R., Watson S. L. An argument for clarity: What are learning management systems, what are they not, and what should they become? TechTrends. 2007.
6. □ European Commission. Digital Education Action Plan 2021–2027. Brussels, 2020.
7. □ Nemeth E., Snyder G., Hein T. R., Whaley B., Mackin D. UNIX and Linux System Administration Handbook. 5th ed. Boston: Pearson Education, 2017.
8. □ Al-Fraihat D., Joy M., Sinclair J. Evaluating e-learning systems success: An empirical study. Computers in Human Behavior. 2020. Vol. 102. P. 67–86. DOI: <https://doi.org/10.1016/j.chb.2019.08.004>.
9. □ Silberschatz A., Galvin P., Gagne G. Operating System Concepts. 10th ed. Hoboken: Wiley, 2018.
10. □ Grinberg M. Flask Web Development: Developing Web Applications with Python. 2nd ed. Sebastopol: O'Reilly Media, 2018.
11. □ SQLite Consortium. SQLite Database Engine: Official Documentation. 2024. URL: <https://www.sqlite.org/docs.html> (дата звернення: 12.02.2025).
12. □ Elmasri R., Navathe S. Fundamentals of Database Systems. 7th ed. Boston: Pearson, 2016.
13. □ Sommerville I. Software Engineering. 10th ed. Boston: Pearson, 2016.

14. □ Stallings W. *Operating Systems: Internals and Design Principles*. 9th ed. Boston: Pearson, 2018.
15. □ Krug S. *Don't Make Me Think: A Common Sense Approach to Web Usability*. 3rd ed. Berkeley: New Riders, 2014.
16. □ Preece J., Rogers Y., Sharp H. *Interaction Design: Beyond Human-Computer Interaction*. 4th ed. Hoboken: Wiley, 2015.
17. □ ISO/IEC 25010:2011. *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE)*. Geneva: International Organization for Standardization, 2011.
18. □ Bevan N. What is the difference between the purpose of usability and user experience evaluation methods? *Proceedings of the Workshop UXEM*. 2009. P. 1–4.
19. □ Fielding R. T., Reschke J. *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. RFC 7231. Internet Engineering Task Force (IETF), 2014.
20. □ ISO/IEC 14598-1:1999. *Information technology – Software product evaluation – Part 1: General overview*. Geneva: International Organization for Standardization, 1999.
21. □ Kitchenham B., Pfleeger S. Personal opinion surveys. In: *Guide to Advanced Empirical Software Engineering*. Berlin: Springer, 2008.