**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 ОСВІТНЬОГО ВЕБ-РЕСУРСУ ДЛЯ ВИВЧЕННЯ МОВ ПРОГРАМУВАННЯ

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

Виконав: здобувач вищої освіти
групи ПОм-21
Кротюк Артур Віталійович

_____
(підпис)

Керівник:
к.пед.н., доцент
Редько Ольга Іванівна

_____
(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

**Кротюк Артур Віталійович**
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка та дослідження освітнього веб-ресурсу для вивчення мов програмування
керівник роботи: к.пед.н., доцент Редько Ольга Іванівна
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.*

3. Вихідні дані до роботи *Технічне та програмне забезпечення, вимоги до організації навчального процесу, ергономічні вимоги до функціонування програмного засобу.*

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):
*Аналіз літературних джерел за темою кваліфікаційної роботи магістра, виклад загальної проблеми і вибір напрямків дослідження; опис рішення загальної проблеми та основних методів дослідження; методика для проведення експерименту.*

5. Перелік графічного матеріалу:

6. Консультанти розділів роботи

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

7. Дата видачі завдання «06» *лютого* 2025 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи магістра | Строк виконання етапів роботи | Примітка |
|---|---|---|---|
| *1* | *Провести огляд літературних джерел по темі кваліфікаційної роботи магістра* | *до 30.08.25р.* | |
| *2* | *Провести аналіз загальної проблеми і вибір напрямків дослідження* | *до 09.09.25р.* | |
| *3* | *Розробити функціональну схему роботи програмного продукту* | *до 17.09.25р.* | |
| *4* | *Описати засоби розробки об'єкта проектування* | *до 30.09.25р.* | |
| *5* | *Описати роботу об'єкта проектування* | *до 16.10.25р.* | |
| *6* | *Розробити методику для проведення експерименту* | *до 23.10.25р.* | |
| *7* | *Провести аналіз результатів експерименту* | *до 12.11.25р.* | |
| *8* | *Здача чистового варіанту кваліфікаційної роботи магістра на кафедру* | *до 05.12.25р.* | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Здобувач вищої освіти _____ Кротюк А.В.
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ Редько О.І.
(підпис) (прізвище, ініціали)

# АНОТАЦІЯ

**Кротюк А.В. Розробка та дослідження освітнього веб-ресурсу для вивчення мов програмування.** Рукопис.

Кваліфікаційна робота магістра ОП «Професійна освіта (комп'ютерні технології)» спеціальності 015.39 Професійна освіта (Цифрові технології). Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота магістра складається зі вступу, чотирьох розділів, висновків, списку використаних джерел

У роботі досліджено підходи до організації навчання програмуванню в умовах цифрового освітнього середовища та обґрунтовано доцільність використання хмарних платформ для підтримки навчального процесу. Основний акцент зроблено на розробці й апробації електронного курсу для вивчення мов програмування Python і JavaScript із застосуванням Google Classroom.

У теоретичній частині розкрито сучасні тенденції ІТ-освіти, специфіку навчання програмуванню та роль електронних платформ у формуванні практичних навичок. Наведено порівняльну характеристику Python і JavaScript у навчальному контексті, визначено їхні переваги та особливості використання під час вивчення базових тем програмування. Окремо описано методологічні основи дослідження, що поєднують педагогічні методи аналізу результатів навчання та елементи статистичного опрацювання даних.

Практична частина роботи спрямована на проєктування структури курсу в Google Classroom, підготовку теоретичних матеріалів із посиланнями на мультимедійні ресурси, розробку практичних завдань, інтерактивних прикладів і засобів контролю знань. Також описано особливості адаптації змісту курсу для двох мов програмування та подано приклади реалізованих елементів курсу.

Ключові слова: електронне навчання, *освітній веб-ресурс, Google Classroom, програмування, Python, JavaScript, інтерактивні завдання, контроль знань, педагогічний експеримент*.

# ANOTATION

**Krotiuk A.V. Development and research of an educational web resource for learning programming languages.** Manuscript.

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

The thesis investigates approaches to teaching programming within a digital educational environment and substantiates the feasibility of using cloud-based platforms to support the learning process. The main focus is placed on the design and pilot implementation of an e-learning course for Python and JavaScript using Google Classroom.

The theoretical part outlines current trends in IT education, the specifics of learning programming, and the role of e-learning platforms in developing practical skills. A comparative analysis of Python and JavaScript in an educational context is provided, highlighting their advantages and distinctive features for studying fundamental programming topics. The methodological framework combines pedagogical methods for evaluating learning outcomes with elements of statistical data processing.

The practical part is aimed at designing the course structure in Google Classroom, preparing theoretical materials with links to multimedia resources, developing practical tasks, interactive examples, and knowledge assessment tools. The thesis also describes how the course content is adapted for two programming languages and presents examples of implemented course elements.

Key words: *e-learning, educational web resource, Google Classroom, programming, Python, JavaScript, interactive tasks, knowledge assessment, pedagogical experimen*

**ЗМІСТ**

# ВСТУП

Сучасний етап розвитку інформаційних технологій зумовлює підвищення вимог до підготовки фахівців у галузі програмування. Умови цифровізації освіти вимагають впровадження таких форм і методів навчання, які забезпечують не лише засвоєння теоретичних знань, а й формування практичних навичок, необхідних для професійної діяльності. У цьому контексті особливої актуальності набуває використання освітніх веб-ресурсів і хмарних платформ, що дозволяють організувати гнучкий, інтерактивний та доступний навчальний процес.

Навчання мов програмування є складним і багатокомпонентним процесом, який потребує поєднання теоретичного матеріалу з практичною діяльністю. Традиційні підходи до викладання програмування не завжди забезпечують достатній рівень залученості студентів і не повною мірою відображають реальні умови розробки програмного забезпечення. У зв’язку з цим доцільним є впровадження проєктно-орієнтованих методик навчання, що передбачають використання декількох мов програмування в межах одного навчального завдання або проєкту.

Одним із ефективних інструментів організації електронного та змішаного навчання є освітня платформа Google Classroom. Вона забезпечує зручне управління навчальними матеріалами, організацію практичних завдань, комунікацію між викладачем і студентами, а також інтеграцію з іншими сервісами Google. Простота використання та мінімальні технічні вимоги роблять Google Classroom доцільним вибором для швидкого розгортання навчальних курсів у закладах вищої освіти.

У даній магістерській роботі розглядається підхід до навчання програмуванню, що базується на інтегрованому використанні мов Python і JavaScript. Python застосовується для реалізації серверної логіки та обробки

даних, тоді як JavaScript використовується для створення клієнтської частини та забезпечення інтерактивної взаємодії з користувачем.

**Метою магістерської роботи** є розробка інтерактивного навчального курсу на платформі Google Classroom, спрямованого на формування міжмовних компетентностей програмування шляхом реалізації єдиного навчального проєкту з використанням мов Python та JavaScript. У межах курсу Python розглядається як інструмент реалізації серверної логіки та обробки даних, тоді як JavaScript використовується для створення клієнтської частини та інтерактивної взаємодії з користувачем.

**Об'єктом дослідження** є процес навчання мов програмування в умовах цифрового освітнього середовища. У межах роботи розглядається організація освітнього процесу підготовки майбутніх IT-фахівців або вчителів інформатики з використанням електронних навчальних ресурсів.

**Предметом дослідження** є методика розробки та використання електронного навчального курсу на платформі Google Classroom для вивчення мов програмування Python і JavaScript.

Для досягнення поставленої мети в роботі визначено такі основні завдання дослідження:

– проаналізувати роль і місце мов програмування Python та JavaScript у сучасній IT-освіті, визначити їхні особливості, переваги та сфери застосування;

– розробити структуру електронного курсу на платформі Google Classroom;

– розробити структуру електронного навчального курсу, що поєднує теоретичний матеріал, практичні завдання та контроль знань;

– розробити методику педагогічного експерименту для перевірки ефективності створеного курсу;

– проаналізувати результати експериментального навчання та оцінити доцільність запропонованого підходу.

**Методологічну та теоретичну основу дослідження** становлять загальнонаукові та спеціальні методи пізнання. У роботі застосовано системний підхід, який дозволяє розглядати навчання програмуванню як цілісну систему, що поєднує технічні засоби та педагогічні методи. Також використано процесний підхід, що забезпечив поетапну організацію розробки курсу — від аналізу вимог і проєктування структури до реалізації та впровадження. Серед методів педагогічного дослідження застосовано аналіз наукових джерел, порівняльний аналіз, анкетування, експертне оцінювання та елементи статистичного аналізу результатів навчання.

Крім того, у процесі виконання роботи були задіяні інструменти штучного інтелекту, а саме ChatGPT-5, для формування структури курсу, систематизації джерел, удосконалення формулювань тексту та перевірки логіки викладу матеріалу. Усі результати, отримані з використанням таких засобів, пройшли перевірку на достовірність і відповідність вимогам академічної доброчесності.

**Наукова новизна** роботи полягає у впровадженні підходу до навчання програмування, що базується не лише на порівняльному аналізі мов Python і JavaScript, а й на їх інтегрованому використанні в межах одного навчального проєкту.

**Практичне значення** роботи полягає у створенні готового електронного навчального курсу, який може бути використаний у навчальному процесі закладів вищої освіти для підготовки фахівців у галузі інформаційних технологій. Матеріали курсу та описана методика його розробки можуть слугувати основою для створення аналогічних курсів з інших дисциплін програмування або ІТ-напрямів.

**Апробація результатів** дослідження здійснювалася шляхом експериментального впровадження розробленого курсу на базі платформи Google Classroom. В експерименті брали участь добровольча група, результати навчання яких були проаналізовані за допомогою тестування та анкетування.

Отримані дані засвідчили позитивний вплив використання електронного курсу на рівень засвоєння матеріалу та навчальну мотивацію студентів.

# РОЗДІЛ 1. АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ЗА ТЕМОЮ МАГІСТРАТУРСЬКОЇ РОБОТИ, ВКЛАД ЗАГАЛЬНОЇ ПРОБЛЕМИ І ВИБІР НАПРЯМКІВ ДОСЛІДЖЕННЯ

## 1.1. Сучасні підходи до навчання програмуванню та використання електронних платформ

Навчання програмуванню традиційно поєднує теорію з великою кількістю практики. Останніми роками у світі набули поширення платформи електронного навчання (Learning Management Systems, LMS), що дозволяють створювати інтерактивні онлайн-курси з програмування. Moodle, Canvas, Blackboard, Google Classroom та інші системи управління навчанням активно використовуються для підтримки освітнього процесу у закладах освіти. Результати численних досліджень свідчать, що за умови якісно спроєктованого курсу електронне навчання може бути не менш ефективним, ніж традиційне аудиторне. При цьому ключову роль відіграє чітка структура курсу, поступове ускладнення завдань, поєднання теорії з практикою та можливість експериментувати з кодом у безпечному середовищі.

До основних труднощів, з якими стикаються студенти під час вивчення програмування, належать підтримання навчальної мотивації, необхідність регулярної практики, своєчасне отримання зворотного зв’язку та врахування індивідуальних темпів засвоєння матеріалу. В умовах масового навчання ці проблеми стають особливо відчутними, оскільки викладач не завжди має змогу оперативно реагувати на помилки кожного студента. У відповідь на ці виклики почали активно застосовуватися електронні освітні ресурси, онлайн-курси, інтерактивні середовища програмування (Codeacademy, freeCodeCamp тощо), а також системи управління навчанням (LMS), які дозволяють частково автоматизувати процес навчання і контролю знань.

Однією з найпоширеніших LMS є Moodle (Modular Object-Oriented Dynamic Learning Environment) — система управління навчанням з відкритим програмним кодом, яка дає змогу організовувати повноцінні електронні курси з різних дисциплін. Moodle забезпечує гнучке структурування навчального матеріалу за темами або модулями, підтримує розміщення текстових і мультимедійних матеріалів, створення тестів і практичних завдань, організацію форумів для обговорень та ведення електронного журналу оцінок. Завдяки цим можливостям платформа широко використовується для викладання дисциплін ІТ-спрямування, зокрема мов програмування.

Для курсів з програмування Moodle пропонує низку корисних інструментів і додаткових плагінів. Зокрема, плагіни для автоматичного оцінювання коду, такі як CodeRunner, дозволяють інтегрувати в навчальний процес виконання та перевірку програмного коду безпосередньо в середовищі курсу. Студент отримує можливість одразу побачити результат виконання своєї програми, а система автоматично оцінює правильність розв’язання на основі заздалегідь підготовлених тестів. Такий підхід є ефективним для формування практичних навичок програмування та зменшення навантаження на викладача.

Ще однією важливою можливістю Moodle є інтеграція з інтерактивними середовищами виконання коду. Для мови Python це можуть бути Jupyter Notebook або зовнішні онлайн-сервіси, що дозволяють виконувати код покроково та візуалізувати результати. Для JavaScript широко застосовуються сервіси на кшталт JSFiddle або CodePen, які надають змогу демонструвати роботу скриптів у браузері. Використання таких інструментів у навчальному процесі сприяє кращому розумінню принципів роботи програм і підвищує залученість студентів.

Поряд із Moodle все більшого значення набувають хмарні платформи, зокрема Google Classroom. Google Classroom – це безкоштовна веб-служба для навчання, що входить до складу Google Workspace for Education. Вона дозволяє викладачам створювати класи, розповсюджувати завдання, спілкуватися зі

студентами та надавати зворотний зв’язок – усе на єдиній інтуїтивно зрозумілій платформі. На відміну від Moodle, що потребує самостійного хостингу або інсталяції, Google Classroom є повністю хмарним рішенням: для роботи достатньо Інтернет-з’єднання та облікового запису Google. Ця платформа стрімко набула популярності під час пандемії COVID-19, ставши одним із основних інструментів дистанційного навчання. Так, станом на березень 2020 року додаток Google Classroom перевищив позначку у 50 мільйонів завантажень і очолив рейтинги освітніх застосунків. У глобальному вимірі кількість активних користувачів Google Classroom сягнула десятків мільйонів, а в Україні платформа стала широко використовуватися під час вимушеного переходу на дистанційне та змішане навчання (зокрема, у період карантину та воєнних дій). Безкоштовний характер сервісу, відсутність реклами та тісна інтеграція з іншими продуктами Google (Google Drive, Docs, Gmail, Calendar тощо) роблять Classroom привабливим вибором для навчальних закладів.

Таким чином, сучасні електронні платформи здатні забезпечити ефективне навчання програмуванню за умови правильної методичної організації. Використання LMS сприяє індивідуалізації навчального процесу: студенти мають можливість опановувати матеріал у власному темпі, повертатися до складних тем і повторювати їх за потреби. Такий підхід є особливо важливим у навчанні програмуванню, де практика відіграє ключову роль. Дослідники відзначають переваги змішаного навчання (blended learning), коли електронні курси доповнюють аудиторні заняття, поєднуючи гнучкість онлайн-доступу з перевагами живого спілкування.

Враховуючи тенденції розвитку IT-освіти, розробка інтерактивного курсу з програмування мовами Python і JavaScript із використанням платформи Google Classroom є одним із кращих напрямком. Такий підхід дозволяє інтегрувати теоретичні знання та практичні навички, забезпечуючи гнучкість навчального процесу.

## 1.2. Особливості та порівняння мов Python і JavaScript у навчальному контексті

Python і JavaScript – дві популярні мови програмування, що належать до різних сімейств і мають різне призначення. Python (створений у 1991 році) – мова загального призначення, яка вирізняється простою синтаксичною структурою і читаємістю коду. Вона широко використовується у наукових обчисленнях, аналізі даних, машинному навчанні, бекенд-розробці веб-сервісів тощо. JavaScript (з'явився у 1995 році) – основна мова для розробки інтерфейсу веб-сторінок; вона забезпечує динамічну взаємодію з користувачем у браузері і є незамінною для фронтенд-програмування веб-застосунків. Обидві мови входять до переліку найпопулярніших у світі: за даними різних рейтингів, Python і JavaScript стабільно посідають провідні місця завдяки своїй широкій сфері застосування та активній спільноті розробників.

Попри спільну мету, в управлінні поведінкою програм, Python і JavaScript мають суттєві концептуальні та синтаксичні відмінності. Python є мовою з динамічною типізацією, але вимагає дотримання відступів для структурування коду, підтримує множинне успадкування, має розвинену систему роботи з бібліотеками (через pip) та орієнтований на швидку розробку прототипів. JavaScript також динамічно типізований, однак його синтаксис ближчий до мов C-подібної родини; він спочатку створювався для браузера, тому містить ряд особливостей, що визначають його використання на клієнтській стороні. Історично Python розвивався як інструмент для автоматизації та обчислень, а JavaScript – як скриптова мова веб-клієнта, проте нині обидві мови розширили свої сфери: з появою Node.js JavaScript вийшов на сервер, а Python, завдяки проектам на кшталт Brython чи Transcrypt, може виконуватися і на стороні клієнта.

| Конструкція | Python | JavaScript |
|---|---|---|
| Виведення даних | print("Hello, world!") | console.log("Hello, world!"); |
| Оголошення змінної | x = 5 (неявний тип) | let x = 5; (явне ключове слово для змінної) |
| Умовний оператор | if x > 0: ... else: ... | if (x > 0) { ... } else { ... } |
| Цикл for | for i in range(5): print(i) | for (let i = 0; i < 5; i++) { console.log(i); } |
| Функція | def add(a, b): return a + b | function add(a, b) { return a + b; } |
| Коментарі | # коментар; '''багаторядковий коментар''' | // коментар; /* багаторядковий коментар */ |
| Список/масив | my_list = [1, 2, 3] | let myArray = [1, 2, 3]; |
| Словник/об’єкт | my_dict = {"key": "value"} | let obj = { "key": "value" }; |
| Обробка винятків | try: ... except Exception as e: ... | try { ... } catch (e) { ... } |

Таблиця 1.1 – Приклади синтаксичних відмінностей між Python та JavaScript

Для викладання програмування важливо врахувати ці особливості. Студенти повинні розуміти, в яких випадках певна мова є доцільнішою, які парадигми притаманні Python та JavaScript, і як їх комбінувати. Наприклад, Python краще підходить для реалізації складної логіки «на сервері» або обробки даних (через простоту синтаксису і наявність наукових бібліотек), тоді як JavaScript оптимальний для створення інтерактивного інтерфейсу користувача у веб-застосунках.

У межах нашого курсу передбачено поступове порівняння двох мов на прикладах: одна й та сама задача (наприклад, обчислення факторіалу, сортування списку чи реалізація рекурсії) реалізується спочатку на Python, потім на JavaScript. Це дозволяє студентам побачити відмінності у синтаксисі та підходах (наприклад, робота з масивами і структурами даних, обробка винятків, визначення функцій тощо) і сприяє формуванню міжмовного мислення – вміння обирати оптимальні інструменти під конкретну задачу.

Об’єднання в навчанні двох мов у межах одного курсу може підвищити гнучкість мислення початківців-програмістів. Студенти вчаться не прив’язуватися до однієї конкретної технології, а розуміти концепції програмування на глибшому рівні, щоб побачити їх реалізацію в різних синтаксичних формах. Випускник курсу буде готовий швидше опановувати нові мови й технології в майбутньому, адже він вже має досвід перемикання між Python і JavaScript у процесі навчання.

### 1.3. Теоретичні основи та методи дослідження в контексті роботи

У даному підрозділі узагальнено теоретичні положення та методологічні підходи, що покладені в основу дослідження, а також обґрунтовано вибір методів оцінювання ефективності розробленого електронного навчального курсу. Особлива увага приділяється поєднанню педагогічних і технологічних аспектів у процесі навчання програмуванню.

Електронний навчальний курс або навчально-методичний посібник доцільно розглядати з позицій теорії систем. Відповідно до системного підходу, курс являє собою цілісну систему, що складається з взаємопов’язаних компонентів: навчального контенту, практичних завдань, тестових матеріалів, інтерфейсу платформи та учасників освітнього процесу. Ефективність функціонування такої системи залежить від узгодженості її елементів і правильності організації взаємодії між ними.

Застосування системного підходу в межах даної роботи дозволило забезпечити цілісність проєктування курсу та відповідність усіх його складових загальній меті навчання. Кожен модуль курсу було розроблено з урахуванням конкретних навчальних цілей, а система оцінювання — узгоджена зі змістом теоретичних і практичних матеріалів. Такий підхід сприяв підвищенню логічності структури курсу та прозорості навчального процесу для студентів.

Окрім системного підходу, у роботі використано процесний підхід, який передбачає поетапну організацію діяльності з розробки електронного навчального ресурсу. Зокрема, процес створення курсу включав етапи аналізу вимог, проєктування структури, наповнення навчальним контентом, тестування функціональності та впровадження у навчальний процес. Така послідовність дій дозволила систематизувати роботу та своєчасно виявляти й усувати недоліки.

Для оцінювання результативності створеного курсу було сплановано педагогічний експеримент, теоретичною основою якого є концепція формувального експерименту. Суть даного підходу полягає в порівнянні навчальних досягнень студентів, які навчаються за традиційною методикою, з результатами студентів, що опановують матеріал за допомогою нового електронного курсу. Такий підхід дозволяє оцінити вплив запропонованої методики на рівень засвоєння навчального матеріалу.

У процесі дослідження застосовувалися різні методи педагогічного аналізу. До них належать аналіз наукових і методичних джерел, порівняльний аналіз (для зіставлення особливостей мов програмування Python і JavaScript), анкетування студентів, а також експертне оцінювання якості електронного курсу. Застосування комплексу методів дозволило отримати як кількісні, так і якісні результати дослідження.

Для кількісного аналізу результатів експериментального навчання використовувалися елементи математичної статистики. Зокрема, обчислювалися середні показники успішності студентів, аналізувався розподіл оцінок.

Поряд із кількісними показниками важливе значення мало якісне оцінювання результатів навчання. Анкетування студентів дозволило виявити рівень їхньої навчальної мотивації, суб’єктивну оцінку зручності використання платформи Google Classroom та корисності окремих елементів курсу. Отримані результати дали змогу комплексно оцінити ефективність запропонованого підходу до навчання програмуванню

Обрані теоретичні підходи та методи дослідження забезпечили наукову обґрунтованість магістерської роботи та створили основу для подальшої практичної реалізації електронного навчального курсу і аналізу результатів його впровадження.

# РОЗДІЛ 2. ОПИС РІШЕННЯ ЗАГАЛЬНОЇ ПРОБЛЕМИ ТА ОСНОВНИХ МЕТОДІВ ДОСЛІДЖЕННЯ

## 2.1. Технологічні аспекти проєктування електронного навчального ресурсу

Розробка електронного навчального курсу передбачає поєднання педагогічного проєктування та технічної реалізації. На початковому етапі було визначено загальну концепцію курсу, його цільову аудиторію та очікувані результати навчання. Це дозволило сформувати чітке уявлення про структуру курсу та вимоги до його функціональних можливостей.

Електронний курс «Мови програмування Python і JavaScript» орієнтований на студентів старших курсів бакалаврату або магістратури спеціальностей, пов’язаних з інформаційними технологіями та комп’ютерними науками. Курс може бути інтегрований у навчальний план як складова дисципліни з програмування. Передбачається, що слухачі володіють базовими навичками роботи з комп’ютером та мають початкові знання з алгоритмізації. Попередній досвід програмування іншими мовами є бажаним, але не обов’язковим, оскільки курс починається з базових понять.

Основною метою курсу є формування у студентів практичних навичок програмування мовами Python і JavaScript, а також розуміння концептуальних особливостей цих мов і сфер їх застосування. У результаті проходження курсу студенти повинні вміти аналізувати поставлену задачу та обирати доцільну мову програмування для її розв’язання.

Для досягнення поставленої мети було сформульовано низку конкретних навчальних цілей. Зокрема, курс передбачає вивчення базового синтаксису та основних конструкцій мов Python і JavaScript, засвоєння принципів роботи з функціями, структурами даних і об’єктами, а також формування навичок налагодження програмного коду та пошуку помилок. Окрему увагу приділено

розумінню відмінностей у підходах до розробки програм мовами Python і JavaScript.

Відповідно до визначених цілей було розроблено структуру курсу, що складається з тематичних модулів. Загалом курс поділено на дві логічні частини: перша присвячена вивченню мови програмування Python, друга — мови JavaScript. Кожен модуль містить теоретичний матеріал, приклади програмного коду, практичні завдання та елементи контролю знань, що забезпечує послідовне засвоєння навчального матеріалу.

Принциповою особливістю проєктування курсу є його орієнтація на реалізацію єдиного навчального проєкту, у межах якого поєднується використання двох мов програмування. Python застосовується для реалізації серверної частини проєкту, зокрема обробки даних і бізнес-логіки, тоді як JavaScript використовується для створення клієнтського інтерфейсу та забезпечення взаємодії з користувачем. Така інтеграція дозволяє студентам зрозуміти практичну доцільність використання різних мов програмування в межах одного програмного продукту.

*Одна частина* курсу, присвячена мові Python, охоплює основи синтаксису, роботу з типами даних, умовними операторами та циклами, визначення функцій і використання стандартної бібліотеки. Окремий модуль присвячено об’єктно-орієнтованому програмуванню та роботі з файлами.

*Друга частина* курсу зосереджена на вивченні мови JavaScript та її застосуванні у веб-розробці. У межах цієї частини розглядаються основи синтаксису JavaScript, робота з функціями й об’єктами, взаємодія з елементами DOM, а також базові принципи асинхронного програмування. Практичні завдання орієнтовані на створення простих інтерактивних веб-сторінок і веб-застосунків.

*Завершальним елементом* курсу є інтеграційне практичне завдання, яке передбачає розробку навчального мініпроєкту з використанням Python і JavaScript. Виконання такого завдання сприяє закріпленню міжмовних зв’язків,

формуванню навичок проєктного мислення та розумінню архітектури простого веб-застосунку.

Кожен модуль курсу побудовано за єдиною логікою: спочатку подається теоретичний матеріал, далі — приклади застосування відповідних конструкцій, після чого студенти виконують практичні завдання та проходять тест для самоперевірки. Такий підхід забезпечує поступове ускладнення матеріалу та сприяє закріпленню знань на практиці.

Таким чином, технологічні аспекти проєктування електронного навчального ресурсу були спрямовані на створення логічно структурованого, інтерактивного та зручного у використанні курсу, який відповідає сучасним вимогам до навчання програмуванню та забезпечує ефективну взаємодію студентів із навчальним матеріалом.

## 2.2. Вибір засобів розробки та налаштування середовища

З-поміж доступних платформ обрано Google Classroom через її простоту в розгортанні та широкі можливості інтеграції з інструментами Google. Google Classroom, будучи хмарним сервісом, не вимагає встановлення спеціалізованого програмного забезпечення чи налаштування серверної інфраструктури – для роботи достатньо браузера з доступом до Інтернету. Це суттєво знизило поріг технічного запуску курсу: викладач може створити новий онлайн-клас буквально за кілька хвилин, запросивши студентів через код або електронну пошту. Крім того, Classroom безкоштовний для освітнього використання і не містить реклами, що усуває фінансові та інформаційні бар’єри для студентів.

Оскільки Google Classroom має дещо інший функціонал порівняно з Moodle (який передбачає встановлення плагінів), при проєктуванні курсу було зроблено акцент на зовнішніх сервісах, що доповнюють можливості Classroom. Зокрема, для забезпечення виконання коду і миттєвого зворотного зв’язку було вирішено інтегрувати сторонні інструменти. Для мови Python обрано

використання середовища Google Colab, що дозволяє студентам запускати код на сервері і одразу отримувати результати. Для JavaScript було передбачено застосування онлайн-редакторів на кшталт JSFiddle або CodePen – вони дають змогу виконувати клієнтський JS-код у браузері.

Ще одним завданням було забезпечити автоматизоване тестування знань (тестові завдання, вікторини). У Moodle для цього існує модуль Quiz, натомість у Google Classroom аналогічну функцію виконує зв'язка з Google Forms. Були створені тести у вигляді Google Form із увімкненим режимом "Тест" (Quiz Mode), що дозволяє призначати правильні відповіді й бали за питання. Такі форми були впроваджені в Classroom через опцію "Quiz Assignment" – студенти переходили за посиланням на форму, відповідали на питання й одразу після відправлення отримували результати та правильні відповіді (за умови відповідних налаштувань форми) – тобто реалізовано миттєвий фідбек за тестами. Оцінки з форми автоматично переносилися в журнал оцінок Google Classroom, що полегшило відстеження успішності.

З технічної точки зору, розробка курсу передбачала також підготовку навчальних матеріалів у зручному форматі. Теоретичні відомості (лекції, конспекти) оформлені у вигляді документів у веб-ресурсах. Ці матеріали були розбиті на тематичні модулі і завантажені до Classroom як ресурси. Таким чином, весь контент курсу зберігається у хмарі (Google Drive), що забезпечує його доступність і актуальність (викладач може оновлювати матеріали й студенти одразу бачать зміни).

Вибір інструментів розробки і налаштування середовища Google Classroom забезпечили необхідні технічні умови для реалізації інтегрованого навчального курсу, орієнтованого на паралельне вивчення Python і JavaScript та практичне застосування знань. Використання хмарної платформи дозволило сконцентруватися на наповненні курсу, мінімізувавши зусилля на підтримку інфраструктури.

## 2.3. Реалізація структури курсу в Google Classroom

Реалізація структури електронного курсу в системі Google Classroom здійснювалася відповідно до попередньо розробленої концепції та навчальних цілей. Основним організаційним елементом курсу став онлайн-клас “Programming Languages: Python & JavaScript” у Google Classroom. Після створення класу було розподілено основні навчальний матеріал за темами.

Структура курсу поділена на тематичні модулі, кожен з яких присвячений певному блоку знань. Наприклад, перша тема – “Вступ. Основи Python”, далі “Основи JavaScript”, “Структури даних у Python”, “DOM і події в JavaScript”, та інтерактивні сторінки (JS)”, і наприкінці – інтеграційний проєкт, що поєднує обидві мови. Ця послідовність тем забезпечує чергування матеріалу з Python і JavaScript, дозволяючи студентам поступово переходити від однієї мови до іншої і назад.

В межах кожного тематичного розділу розміщено:

- Теоретичні матеріали – у форматі прикріплених посилань на Google Colab та NodePen. Наприклад, конспект лекції з теми “Структури даних та функцій” додано як посилання на Colab-записник, а для JavaScript – посилання на NodePen.

- Приклади коду – короткі демонстрації, які інтегровані через посилання на Colab. Для Python: Colab з прикладом реалізації циклу for і його результатами; для JavaScript: готовий приклад на NodePen, що демонструє зміну вмісту HTML-сторінки через DOM.

- Практичні завдання – створені як Assignments (завдання). Кожне завдання містить опис проблеми, яку треба розв’язати програмним шляхом, і поля для здачі роботи (студенти можуть прикріпити файли .py, .js або посилання на свій код). Деякі завдання мають вкладені шаблони: наприклад, шаблон Colab із заготовленими тестами для функції, яку студент має написати.

- Тести для самоконтролю – реалізовані через Quiz Assignment, що прив'язаний до форми Google Forms із питаннями по темі. Студенти проходять тест після вивчення теорії та практики, щоб закріпити матеріал. Результати автоматично фіксуються для викладача.

Особливу увагу було приділено використанню можливостей Google Classroom для підтримки навчання програмуванню. Хоча Classroom не має плагінів, що виконують код всередині платформи, можна було обійти це обмеження шляхом тісної інтеграції зовнішніх ресурсів. Так, у описі деяких завдань з Python додано кнопку "Open in Colab" – при натисканні студент переходить до відповідного записника, де може написати код і запустити перевірочні тести. Для завдань з JavaScript, які потребують взаємодії з веб-сторінками, надано HTML-шаблони та пропонується використати live-середовища.

Стандартні модулі Google Classroom були задіяні повною мірою:

- Stream (Стрічка) використовувався для оголошень та організації дискусій. Викладач публікував у стрічці повідомлення про початок нового модуля, корисні посилання, нагадування про дедлайни. Студенти могли коментувати ці дописи або ставити запитання. Таким чином, Stream фактично виконував роль форуму для курсу. Спілкування в реальному часі відбувалося також через коментарі до конкретних завдань – студенти часто запитували уточнення до умов задач, а викладач давав підказки або пояснення для всієї групи.

- People (Люди) – цей розділ використовувався для керування списком студентів і додавання співвикладачів. На початку експериментального навчання всі учасники були успішно приєднані до класу за допомогою електронних адрес. Студентам було надано інструкцію щодо налаштування своїх профілів, щоб їх імена відображалися коректно (уникаючи ніків, для офіційності в навчанні).

- Grades (Оцінки) – журнал оцінок Google Classroom автоматично агрегував результати студентів за всіма завданнями і тестами. Завдяки цьому студенти бачили свою успішність і могли відстежувати прогрес

Наприкінці реалізації всі компоненти курсу були ретельно протестовані. Посилання на зовнішні ресурси працюють (записники Colab відкриваються з передбаченими правами доступу, форми Google Forms приймають відповіді і показують результати, завдання коректно приймають файли). Курс успішно реалізовано у середовищі Google Classroom: структура тем, ресурси і активності налаштовані згідно з розробленим планом. Технічні випробування показали працездатність усіх компонентів курсу; курс оптимізовано для доступу як з ПК, так і з мобільних пристроїв (через застосунок Classroom студенти можуть переглядати матеріали і здавати завдання зі смартфонів).

## 2.4. Фрагменти реалізованого курсу

На рисунку 2.1 зображена головна сторінка курсу в Google Classroom із переліком тем та завдань. Проста і зрозуміла навігація, що дозволяє швидко переходити до потрібного матеріалу.

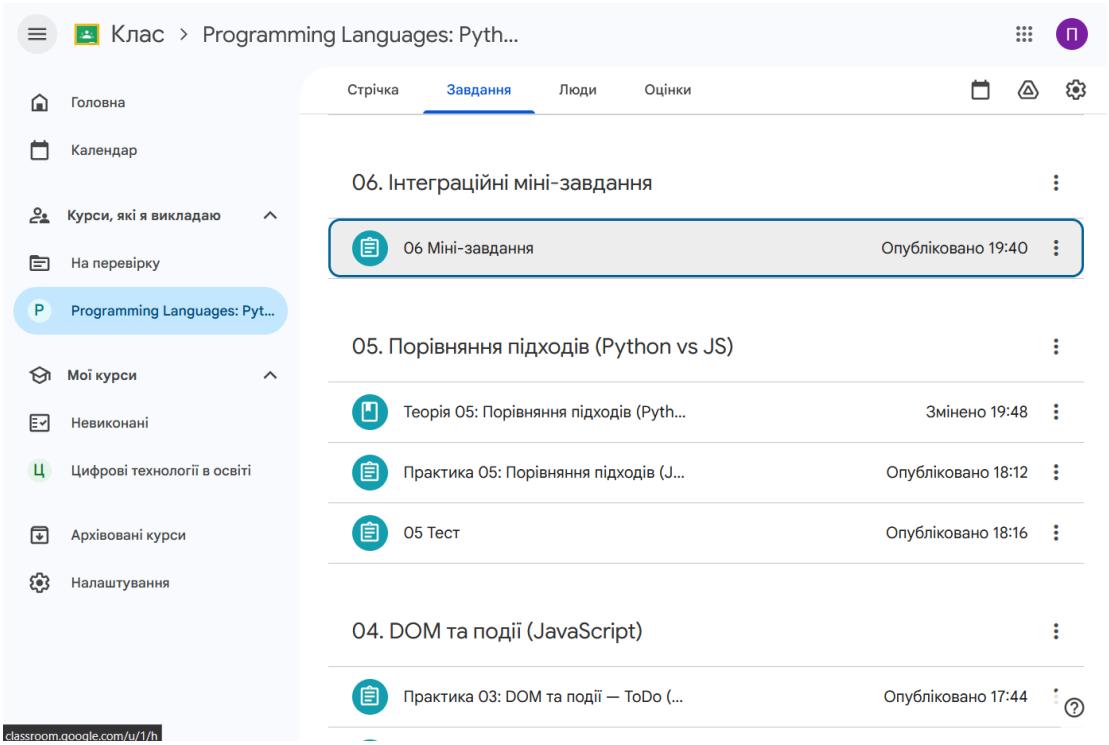


Рисунок 2.1. - Головна сторінка з завданнями

Для теоретичного матеріалу використовувався веб-ресурс NodePen (рисунок 2.2), якщо це JavaScript, та Google Colab, якщо це Python. Подібним підходом можна було продемонструвати можливості мов програмування прямо в середовищі ресурсу, наприклад запустити невелику частину коду, щоб наглядно побачити результат прикладу.

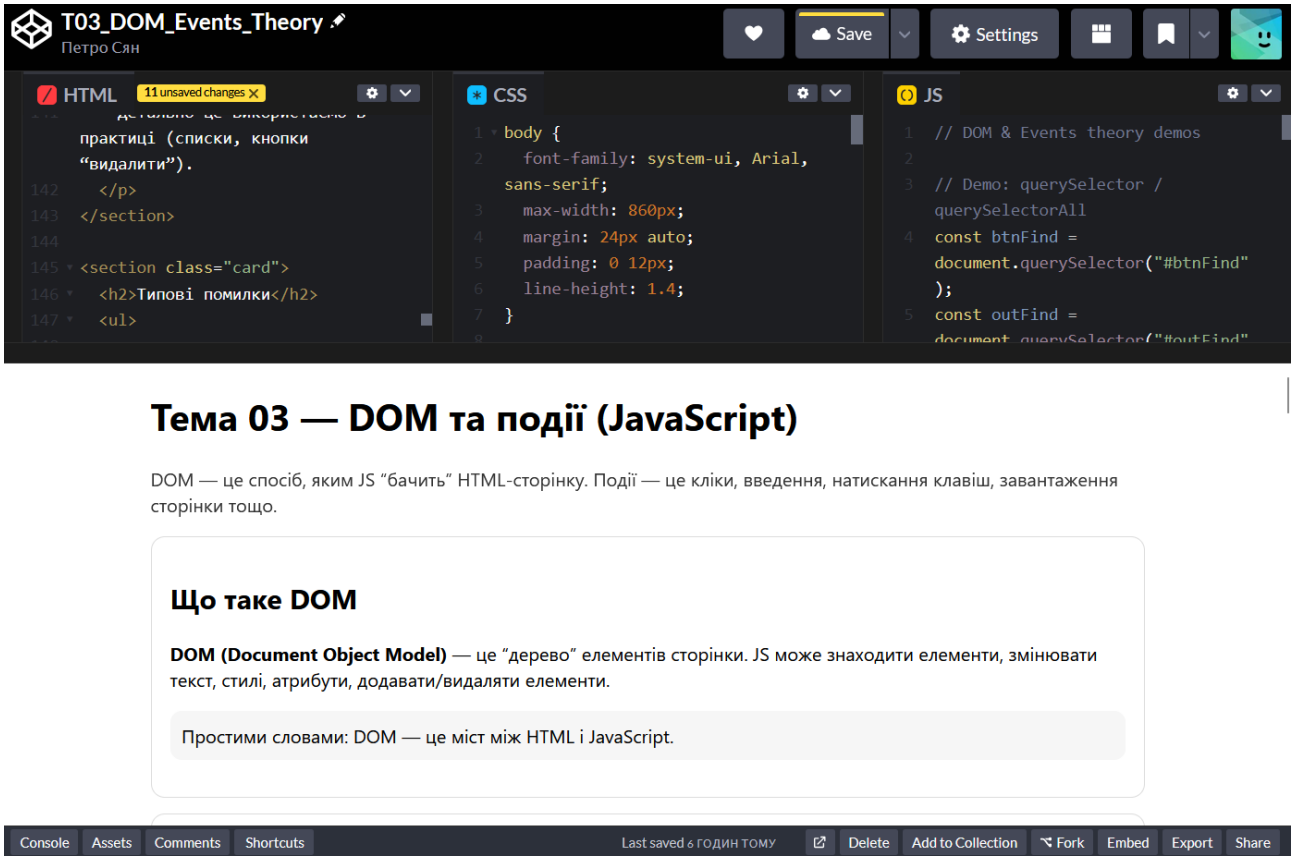


Рисунок 2.2. - Приклад теоретичного матеріал

Для контролю засвоєння знань використовувались Google Form по 10 питань (рисунок 2.3). В основному це були варіанти з одною правильною відповіддю. Проте, підсумкове тестування, що було реалізовано там само, налічувало вже 20 питань, з яких 3 з короткою відповіддю, а також декілька питань з кількома правильними відповідями.

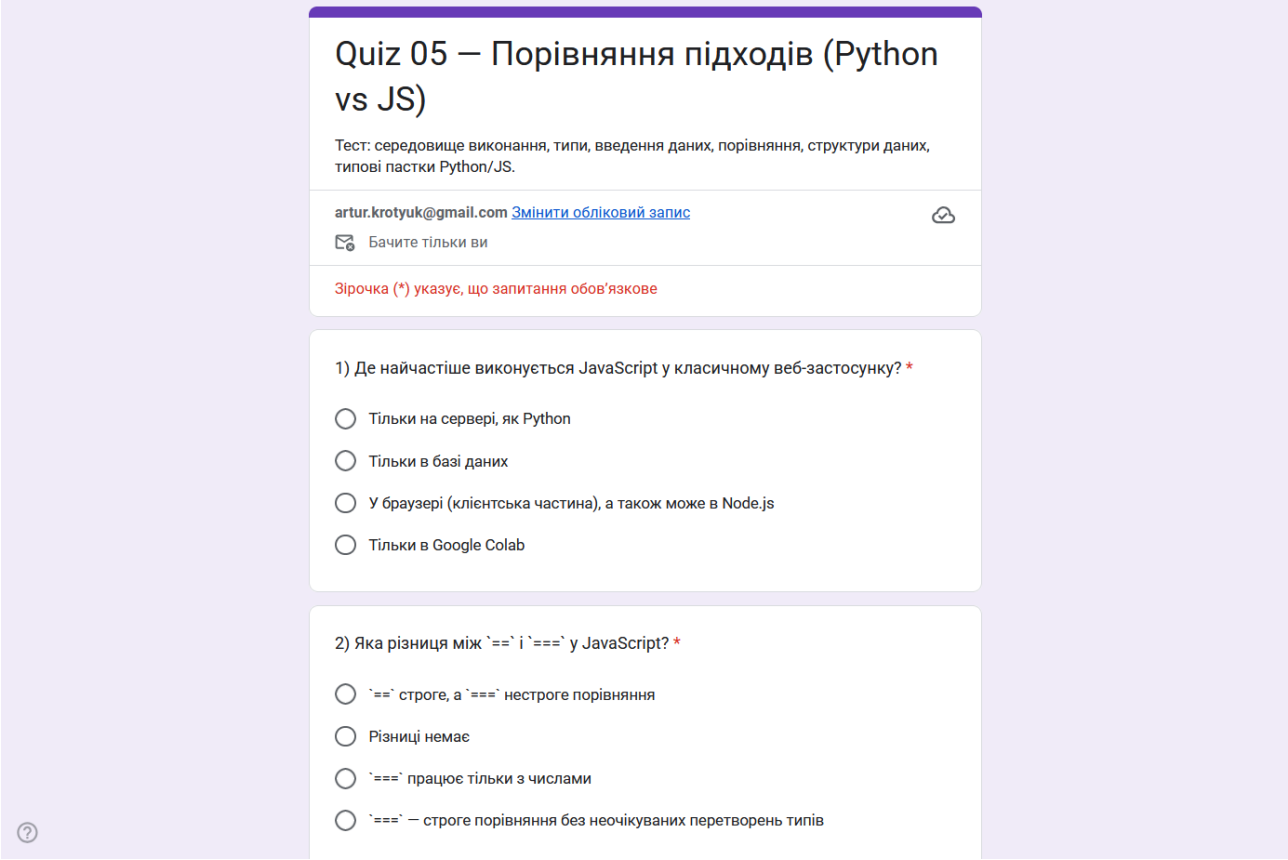


Рисунок 2.3. - Приклад тестів

Щодо практичної роботи, на рисунку зображений Colab-записник (у випадку практичних JavaScript використовувався NodePen), що містить формулювання задачі та шаблон коду, який потрібно дописати (рисунок 2.4). Вбудовані тестові виклики (набори вхідних даних і очікуваних виходів) запускаються для перевірки розв'язку; студенти одразу бачать, які тести пройдено, а які ні.

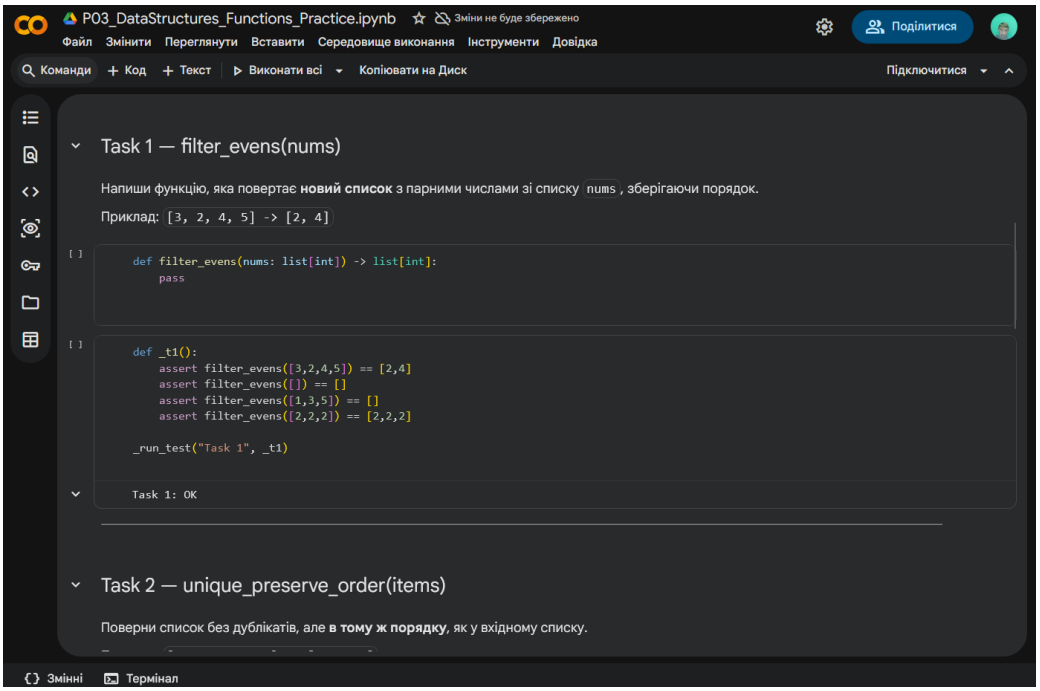


Рисунок - 2.4. Приклад практичних

Фінальний проєкт представлено у вигляді інтернет-магазину (рисунок 2.5). Студентам пропонується розробити цілісний застосунок, в якому серверна логіка (Python) інтегрована з клієнтським інтерфейсом (JavaScript).

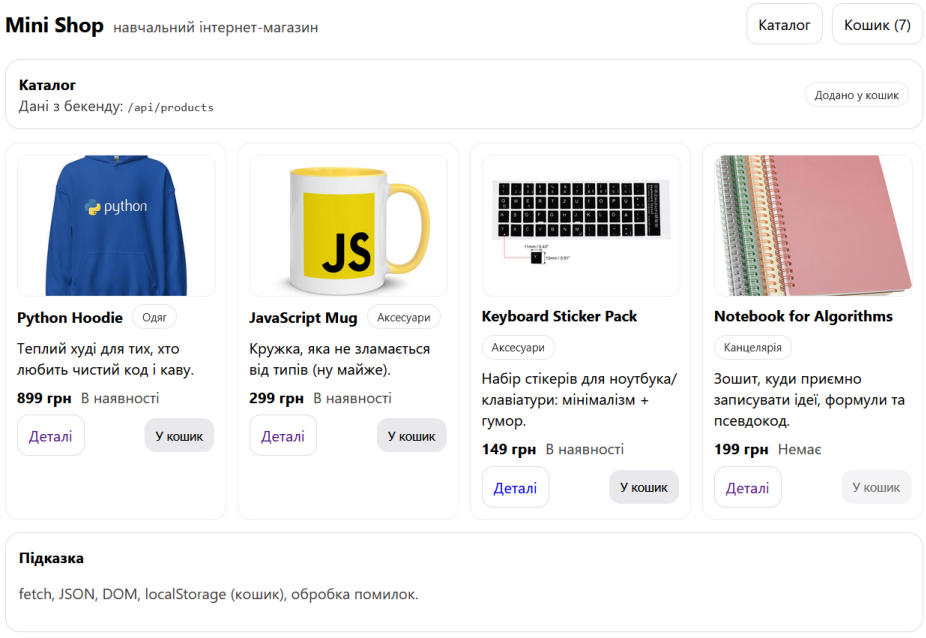


Рисунок 2.5. - Фінальний проєкт

## 2.5. Відмінності реалізації курсу для Python vs JavaScript

При розробці курсу враховано специфіку двох мов програмування, що вивчаються, і це відобразилося в особливостях реалізації окремих компонентів курсу.

Найбільші технічні відмінності стосувалися автоматизації перевірки практичних завдань. Для Python платформа Google Classroom (разом із інтегрованими інструментами) дозволила реалізувати майже повну автоматичну перевірку програмних робіт. Зокрема, було налаштовано завдань записники Colab з попередньо прописаними тестами. Наприклад, після вивчення теми про функції студенти отримували завдання реалізувати функцію factorial(n). До завдання додавався Colab-записник із шаблоном: порожня функція factorial та декілька тестових викликів (для n=0, 1, 5, 10 тощо). Студент писав реалізацію прямо у браузері і запускав тести, які показували, чи правильний отримано результат. Такий підхід забезпечив оперативний зворотний зв’язок і сприяв самостійному виправленню помилок студентами – вони могли одразу побачити, де їхній код працює неправильно, і спробувати знову.

Для JavaScript автоматизація перевірки виявилася складнішою. Завдання з JavaScript у курсі нерідко пов’язані з маніпулюванням веб-сторінкою (DOM) або реакцією на події користувача. Через відсутність у Classroom вбудованого середовища виконання JS з DOM (Moodle, наприклад, теж не мав такої можливості – його плагін CodeRunner запускає JS-код у Node.js без DOM) було прийняте рішення перевіряти частину JS-завдань у ручному режимі. Це стосується переважно тих задач, де результат важко автоматично верифікувати без браузерного оточення. Хоча це додає навантаження на викладача, зате дозволяє оцінити не лише коректність синтаксису, а й якість реалізації інтерфейсу – чи зручно користуватися формою, чи повністю вона відповідає вимогам. Таким чином, частина завдань з JavaScript свідомо залишились з

ручним оцінюванням, щоб охопити ті аспекти, які неможливо автоматизовано протестувати (візуальні ефекти, взаємодію з користувачем тощо).

Варто зазначити, що навіть у рамках автоматизованих завдань виникали цікаві нюанси. Так, у Python виконання студентського коду через Colab пройшло без проблем – Python-скрипти запускалися коректно на сервері Google. У випадку JavaScript довелось експериментувати з можливістю використання Node.js для автоперевірки алгоритмічних задач (не пов'язаних з DOM). Було підготовлено декілька завдань, де студентам пропонувалося написати чисту функцію на JS (наприклад, для генерації послідовності Фібоначчі), а перевірка виконувалася шляхом запуску їхнього коду на Node.js-середовищі. Ці завдання вдалося автоматизувати, тож студент міг отримати бал за правильність реалізації логіки. Проте завдання, що потребували доступу до браузерного API (маніпуляції з HTML), все одно вимагали вручну переглянути роботу.

У процесі впровадження курсу, студенти по-різному реагували на дві мови. Деякі активніше бралися за Python-завдання. Інші, навпаки, більше захоплювалися JavaScript, коли бачили результат своєї роботи безпосередньо на веб-сторінці (динамічна взаємодія). Щоб вирівняти залученість, завдання чергувались видами діяльності: після блоку з автоматизованим Python-завданням йшов блок з творчим JS-завданням, і навпаки. Це підтримувало інтерес: алгоритмічні задачі сприймалися як своєрідні "головоломки" (один зі студентів відгукнувся, що завдання з кодом нагадують «ігровий і цікавий», а фронтенд-завдання давали простір для креативності в дизайні.

Підсумовуючи, можна сказати, що реалізація структури курсу в Google Classroom забезпечила цілісність навчального процесу, поєднання теоретичних і практичних компонентів та створила умови для ефективного освоєння двох мов програмування паралельно. Відмінності у підходах до автоматизації перевірки для Python і JavaScript не стали перешкодою, а навпаки – продемонстрували студентам реальні особливості роботи з кожною з технологій. Вони побачили, що не все можна віддати "машині" – інколи потрібен ручний огляд і аналіз,

особливо коли йдеться про користувацький інтерфейс. Це теж важливий урок, який підготував їх до практичних умов розробки, де автоматизоване тестування поєднується з ручним.

# РОЗДІЛ 3. МЕТОДИКА ДЛЯ ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

## 3.1. Організація експериментального навчання

Для перевірки ефективності розробленого електронного навчального курсу було організовано педагогічний експеримент на базі закладу вищої освіти. Метою експерименту стало оцінювання впливу інтегрованого підходу до навчання програмуванню, що передбачає спільне використання мов Python і JavaScript у межах одного навчального проєкту, на рівень сформованості практичних навичок студентів.

Учасниками експериментального навчання стали студенти, які добровільно виявили зацікавленість у поглибленому вивченні програмування та використанні електронного навчального курсу. Загальна кількість учасників була обмеженою, що дозволило забезпечити індивідуальний підхід до навчання та якісний зворотний зв’язок у процесі виконання практичних завдань.

Перед початком основного етапу експерименту було проведено вхідне оцінювання рівня підготовки учасників з основ програмування. Результати цього оцінювання засвідчили, що студенти мали приблизно однаковий базовий рівень знань, що створило порівнювані умови для подальшого аналізу результатів навчання.

Основний етап експериментального навчання тривав протягом кількох навчальних тижнів і здійснювався з використанням розробленого електронного курсу. Навчальний процес було організовано таким чином, щоб студенти поступово опановували матеріал з мов Python і JavaScript, виконуючи теоретичні та практичні завдання, орієнтовані на реалізацію інтеграційного навчального проєкту.

У процесі навчання учасники експерименту працювали з навчальними матеріалами курсу, виконували практичні завдання з автоматичною перевіркою, проходили тести для самоконтролю та брали участь в обговореннях на форумі

курсу. Окремі завдання передбачали поєднання серверної логіки, реалізованої мовою Python, з клієнтською частиною, створеною засобами JavaScript, що сприяло формуванню міжмовних компетентностей програмування.

Для підтримки учасників експериментального навчання протягом курсу надавалися консультації та рекомендації щодо виконання завдань. Це дозволяло своєчасно виявляти труднощі, що виникали у студентів, і коригувати навчальний процес відповідно до їхніх потреб та рівня підготовки.

Після завершення навчання було проведено підсумкове оцінювання результатів експериментального навчання, яке включало тестування та виконання практичного завдання з програмування. Особлива увага приділялася здатності студентів застосовувати мови Python і JavaScript у межах одного навчального проєкту та розумінню принципів взаємодії між різними компонентами програмного продукту.

Додатково було проведено анкетування учасників експерименту з метою виявлення їхнього ставлення до електронного курсу та інтегрованого підходу до навчання програмуванню. Отримані результати використовувалися для якісного аналізу ефективності запропонованої методики.

Таким чином, організація експериментального навчання дала змогу оцінити практичну доцільність використання інтегрованого електронного курсу та визначити його вплив на формування міжмовних компетентностей програмування у студентів.

### 3.2. Інструменти оцінювання та критерії успішності

Для об’єктивного оцінювання ефективності експериментального навчання було використано комплекс інструментів, що дозволяють отримати як кількісні, так і якісні показники результатів навчальної діяльності студентів. Застосування різних інструментів оцінювання дало змогу всебічно проаналізувати рівень

сформованості знань і практичних навичок у процесі вивчення мов програмування Python і JavaScript.

Основним інструментом контролю теоретичних знань у межах електронного курсу були тести. Тестові завдання включали запитання закритого типу, завдання на встановлення відповідностей, а також питання з короткою відповіддю. Такі тести використовувалися як для поточного контролю знань, так і для підсумкового оцінювання результатів навчання.

Для оцінювання практичних умінь студентів застосовувалися практичні завдання з програмування, виконання яких передбачало написання програмного коду мовами Python і JavaScript. Частина завдань перевірялася автоматично, що забезпечувало оперативний зворотний зв’язок і сприяло самостійному виправленню помилок студентами.

Завдання, пов’язані з розробкою клієнтської частини навчального проєкту мовою JavaScript та взаємодією з елементами веб-сторінки, оцінювалися шляхом аналізу наданого програмного коду та результатів його виконання у браузері. При цьому враховувалася не лише синтаксична коректність коду, а й логічна правильність реалізації та працездатність створеного інтерфейсу.

Важливим елементом оцінювання стало виконання підсумкового інтеграційного завдання, яке передбачало поєднання використання мов Python і JavaScript у межах одного навчального проєкту. Оцінювання цього завдання здійснювалося за низкою критеріїв, серед яких коректність реалізації серверної логіки, ефективність взаємодії між компонентами проєкту, а також зручність і функціональність користувацького інтерфейсу.

Для формалізації результатів навчання було визначено критерії успішності, що дозволяють віднести рівень підготовки студентів до одного з умовних рівнів: низького, середнього або достатнього. Критерії враховували результати тестування, якість виконання практичних завдань та здатність студентів застосовувати знання з Python і JavaScript у комплексі.

Окрім кількісних показників, у процесі оцінювання враховувалися й якісні характеристики навчальної діяльності студентів. Зокрема, аналізувалася активність студентів у курсі, їх участь в обговореннях, самостійність під час виконання завдань і здатність аргументувати прийняті програмні рішення.

Для виявлення суб'єктивного сприйняття електронного курсу та інтегрованого підходу до навчання програмуванню було використано анкетування. Запитання анкети стосувалися зручності використання платформи Google Classroom, зрозумілості навчальних матеріалів, рівня складності завдань і корисності виконання інтеграційного навчального проєкту.

Використання комплексу інструментів оцінювання та чітко визначених критеріїв успішності забезпечило об'єктивність і надійність результатів педагогічного експерименту та дозволило оцінити ефективність розробленого електронного навчального курсу.

### 3.3. Аналіз результатів педагогічного експерименту

Аналіз результатів педагогічного експерименту було спрямовано на оцінювання змін у рівні сформованості знань і практичних навичок студентів після завершення навчання з використанням розробленого електронного курсу.

Порівняння результатів вхідного та підсумкового оцінювання засвідчило загальну позитивну динаміку навчальних досягнень учасників експериментального навчання. Зокрема, було зафіксовано зменшення кількості помилок у практичних завданнях, підвищення впевненості студентів під час написання програмного коду та кращу орієнтацію в структурі програмних проєктів.

Аналіз виконання практичних завдань показав, що студенти поступово почали більш усвідомлено підходити до вибору засобів реалізації програмних рішень. Під час виконання інтеграційного завдання більшість учасників експерименту змогли коректно розмежувати серверну логіку, реалізовану мовою

Python, та клієнтську частину, створену засобами JavaScript. Це свідчить про формування у студентів розуміння ролей різних мов програмування в межах одного програмного продукту.

Результати тестування з теоретичних питань також засвідчили покращення рівня засвоєння навчального матеріалу. Студенти демонстрували більш системне розуміння базових концепцій програмування, принципів роботи з функціями, структурами даних і об’єктами, а також особливостей асинхронного виконання коду в JavaScript.

Важливим показником ефективності курсу стала якість виконання інтеграційного навчального проєкту. Більшість студентів змогли реалізувати працездатні програмні рішення, які включали обробку даних на стороні сервера та їх відображення в користувацькому інтерфейсі. В окремих роботах спостерігалися недоліки, пов’язані з оптимізацією коду або оформленням інтерфейсу, що є типовим для початкового етапу формування проєктних навичок.

Аналіз анкетування учасників експерименту показав загалом позитивне ставлення студентів до запропонованого формату навчання. Студенти відзначали зручність використання платформи Google Classroom, корисність автоматизованого зворотного зв’язку та практичну цінність виконання завдань, орієнтованих на розробку єдиного навчального проєкту. Разом із тим частина респондентів зазначила, що інтеграційні завдання вимагали більше часу та зусиль порівняно з традиційними вправами.

Отримані результати свідчать про те, що використання інтегрованого електронного курсу сприяє підвищенню практичної підготовки студентів та формуванню міжмовних компетентностей програмування. Водночас результати експерименту мають пілотний характер, що зумовлює доцільність подальших досліджень із розширенням вибірки та вдосконаленням методики навчання.

Таким чином, аналіз результатів педагогічного експерименту підтвердив доцільність використання розробленого електронного курсу та обґрунтував

ефективність інтегрованого підходу до навчання програмуванню з використанням мов Python і JavaScript.

# РОЗДІЛ 4. ЕКСПЕРЕМЕНТАЛЬНА ЧАСТИНА ДОСЛІДЖЕННЯ ТА ОБРОБКА, АНАЛІЗ І СПІВСТАВЛЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

## 4.1. Узагальнені результати експериментального навчання

У межах даного розділу представлено узагальнені результати педагогічного експерименту, проведеного з метою оцінювання ефективності розробленого електронного навчального курсу на платформі Google Classroom. Основна увага приділяється аналізу змін у рівні сформованості теоретичних знань і практичних навичок студентів у процесі навчання з використанням інтегрованого підходу до вивчення мов програмування Python і JavaScript.

У результаті експериментального навчання було зафіксовано загальну позитивну динаміку навчальних досягнень студентів. Порівняння результатів вхідного та підсумкового контролю засвідчило покращення розуміння базових концепцій програмування, а також підвищення впевненості студентів під час виконання практичних завдань.

Особливо помітні зміни спостерігалися у здатності студентів застосовувати знання на практиці. У процесі виконання навчальних завдань студенти демонстрували більш усвідомлений підхід до розв’язання задач, коректніше структурували програмний код та ефективніше використовували можливості мов програмування Python і JavaScript відповідно до їхніх ролей у навчальному проєкті.

Результати виконання інтеграційного навчального проєкту показали, що більшість студентів змогли реалізувати працездатні програмні рішення, які поєднують серверну логіку та клієнтський інтерфейс. Це свідчить про формування у студентів цілісного уявлення про архітектуру простого програмного продукту та принципи взаємодії між його компонентами.

## 4.2. Аналіз та педагогічна інтерпретація отриманих результатів

Аналіз отриманих результатів здійснювався з позицій педагогічної доцільності та відповідності поставленим навчальним цілям. Особливу увагу було приділено не лише кількісним показникам успішності, а й якісним характеристикам навчальної діяльності студентів у процесі роботи з електронним курсом.

Використання інтегрованого підходу до навчання програмуванню сприяло формуванню у студентів системного бачення процесу розробки програмного забезпечення. Студенти почали краще розуміти функціональне призначення різних мов програмування та усвідомлювати доцільність їх використання в межах одного програмного проєкту.

Аналіз практичних робіт показав, що інтеграція Python і JavaScript у межах навчального курсу позитивно вплинула на розвиток алгоритмічного мислення та навичок аналізу задач. Студенти більш впевнено розмежовували серверну та клієнтську частини застосунку, а також демонстрували здатність адаптувати отримані знання до нових умов і завдань.

Для уточнення суб’єктивного сприйняття курсу та окремих його компонентів було проведено анкетування учасників експериментального навчання за допомогою сервісу Google Forms (Рисунок 4.1). Результати анкетування подано на рисунках 4.2–4.5 і використано для педагогічної інтерпретації ефективності курсу та визначення напрямів його вдосконалення.

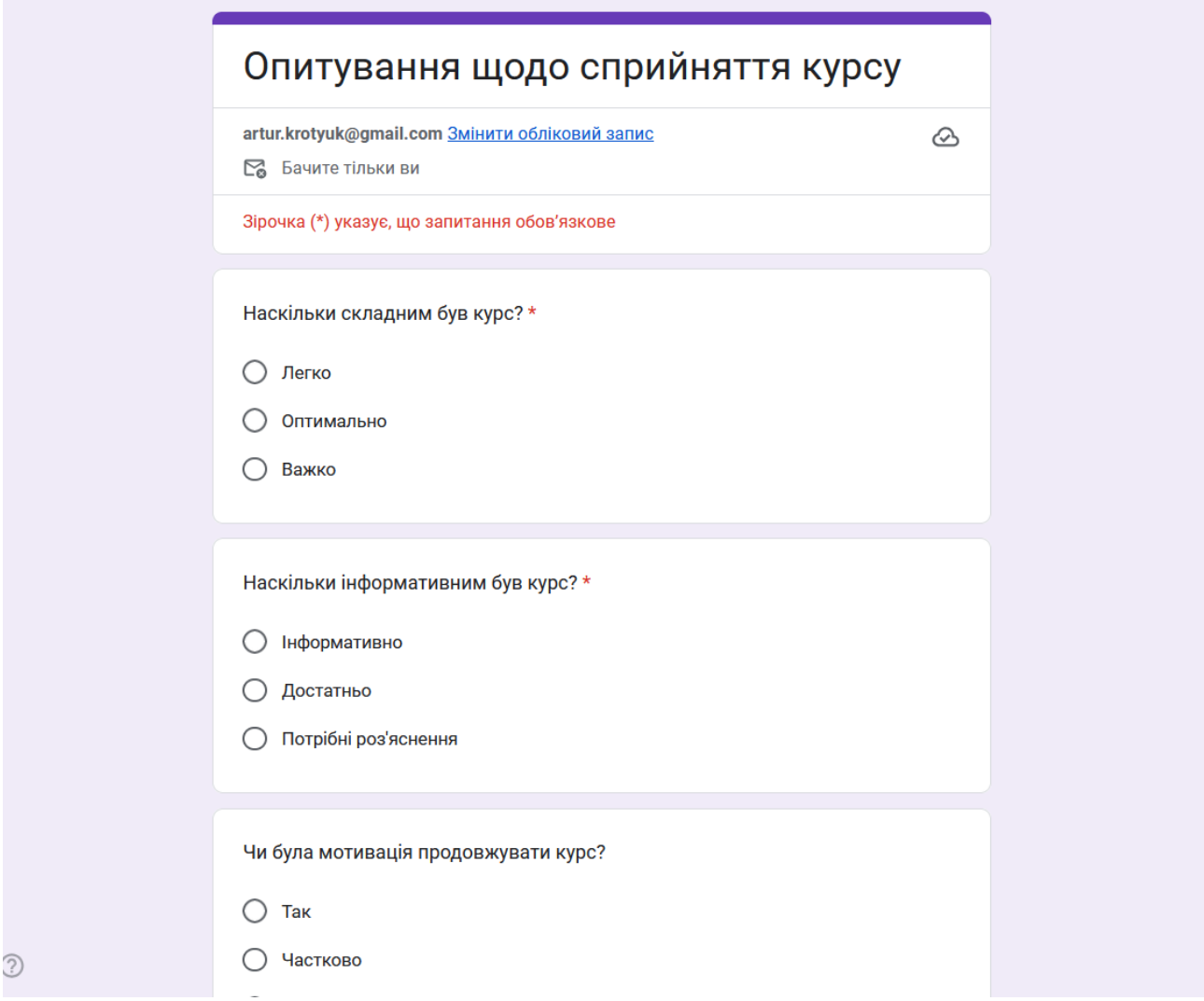


Рисунок 4.1 – Google Form для анкетування студентів стосовно курсу

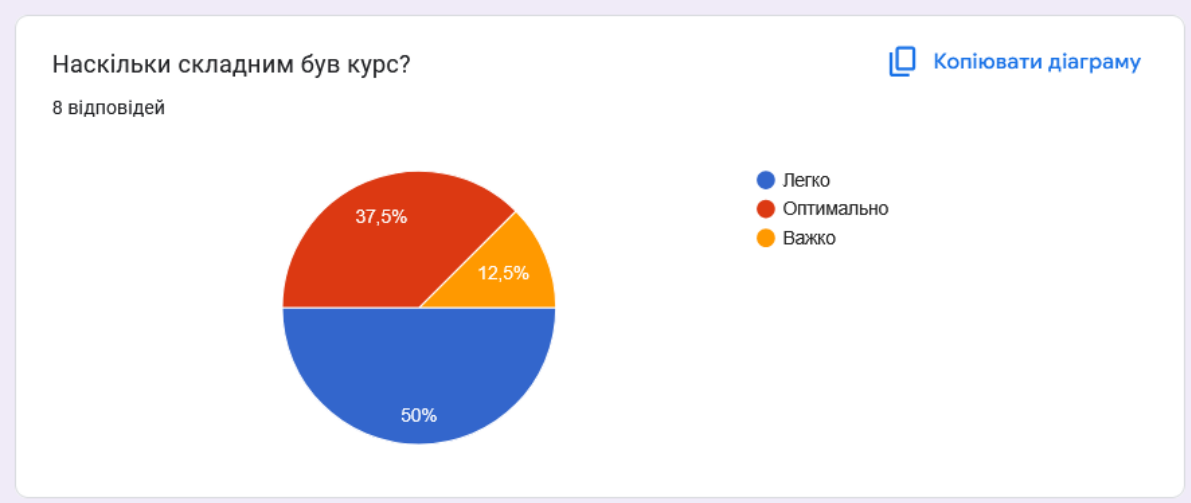


Рисунок 4.2 – Діаграма складності курсу

За показником складності курсу половина респондентів оцінила навчання як «легке» (50%), ще 37,5% визначили рівень складності як «оптимальний», і лише 12,5% відзначили, що курс був «важким». Такий розподіл свідчить, що для більшості слухачів темп і рівень завдань були педагогічно прийнятними та не створювали надмірного когнітивного навантаження.

Водночас наявність респонденів, які зазначали високий рівень складності, вказує на потребу диференціації навчального матеріалу: доцільно передбачати короткі пояснювальні мікрокроки для складних тем (наприклад, асинхронність у JavaScript або робота з файлами в Python), а також додаткові приклади та необов'язкові завдання підвищеної складності для студентів, які рухаються швидше.


Рисунок 4.3 – Діаграма інформативності курсу

Оцінювання інформативності показало більш рівномірний розподіл: 37,5% респондентів вважають курс «інформативним», ще 37,5% визначили його як «достатньо інформативний», а 25% зазначили, що потрібні роз'яснення. Це означає, що основна частина матеріалу сприймається як корисна й структурована, проте для окремих тем або завдань варто посилити пояснення та приклади застосування на практиці.

Як методичний висновок, доцільно доповнити курс короткими підсумками після кожного модуля (ключові поняття, типові помилки, мінішпаргалки з

синтаксису), а також збільшити частку демонстраційних прикладів з коментарями, що пояснюють логіку розв’язання задач.

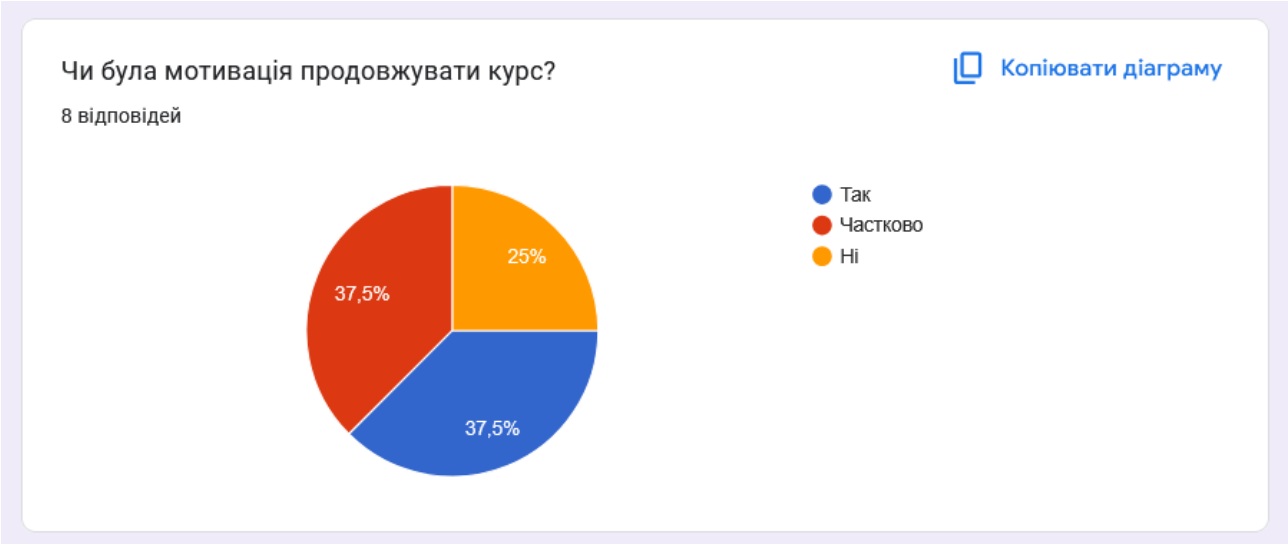


Рисунок 4.4 – Діаграма навчальної мотивації курсу

Рівень навчальної мотивації до продовження курсу характеризується змішаними відповідями: 37,5% опитаних відповіли «так», 37,5% — «частково», і 25% — «ні». Отже, приблизно дві третини учасників зберігають позитивну або помірно позитивну мотивацію, однак наявність частки негативних відповідей підкреслює необхідність додаткових мотиваційних механізмів.

Практичними засобами підсилення мотивації можуть бути: розбиття інтеграційних завдань на коротші етапи з проміжним результатом, регулярний швидкий зворотний зв’язок (коментарі до коду, приклади правильних розв’язків після дедлайну), а також елементи видимого прогресу (чек-листи виконаних тем, невеликі підсумкові мініпроєкти наприкінці модулів).

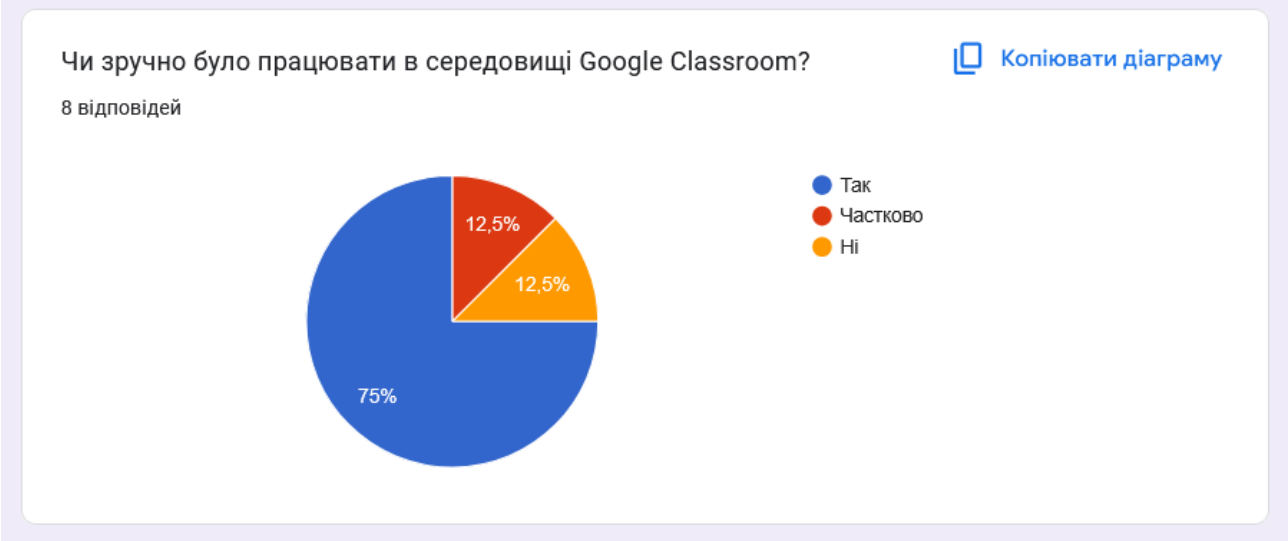


Рисунок 4.5 – Діаграма зручності Google Classroom

Щодо зручності роботи в середовищі Google Classroom, 75% респондентів оцінили роботу як зручну, по 12,5% обрали варіанти «частково» та «ні». Домінування позитивних оцінок підтверджує, що обрана платформа не створювала істотних організаційних бар'єрів і підтримувала доступність матеріалів, завдань і комунікації.

Негативні або частково негативні оцінки можуть бути пов'язані з індивідуальними труднощами навігації, налаштуванням сповіщень або роботою з прикріпленими ресурсами (Colab, онлайн-редактори коду). З огляду на це доцільно додати коротку інструкцію із рекомендаціями щодо організації робочого простору, правил здачі робіт і типових дій у Classroom (перегляд коментарів, перевірка дедлайнів, доступ до матеріалів).

Узагальнюючи результати анкетування, можна зробити висновок, що курс загалом сприймається як доступний за рівнем складності та достатньо інформативний, а платформа Google Classroom забезпечує зручну організацію навчального процесу. Виявлені зауваження щодо потреби в додаткових роз'ясненнях і часткової втрати мотивації є підставою для подальшого методичного вдосконалення курсу, насамперед через розширення пояснювальних матеріалів і посилення підтримки студентів на складних етапах.

Важливим результатом експерименту стало підвищення навчальної мотивації студентів. За результатами анкетування більшість учасників зазначили, що виконання практичних завдань, орієнтованих на створення реального програмного продукту, є більш цікавим і корисним порівняно з ізольованими вправами з окремих мов програмування.

Разом із тим у процесі аналізу було виявлено окремі труднощі, пов’язані з підвищеною складністю інтеграційних завдань і необхідністю опанування двох мов програмування одночасно. Це підтверджує доцільність поступового ускладнення навчального матеріалу та надання додаткових консультацій на початкових етапах навчання.

Педагогічна інтерпретація отриманих результатів свідчить про ефективність інтегрованого підходу до навчання програмуванню та підтверджує відповідність розробленого курсу сучасним вимогам до підготовки фахівців у сфері інформаційних технологій.

Для поглибленої педагогічної інтерпретації важливо враховувати, що оцінка складності курсу відображає не лише об’єктивний рівень завдань, а й попередній досвід слухачів, наявність прогалин у базових темах та особливості сприйняття матеріалу в дистанційному форматі. В онлайн-курсі навіть невеликі організаційні труднощі (наприклад, помилка під час налаштування середовища, незрозуміле повідомлення про помилку або відсутність швидкого уточнення від викладача) можуть підвищувати суб’єктивне відчуття складності і впливати на впевненість студентів під час виконання практичних завдань.

У випадку інтегрованого навчання двох мов програмування додатковим чинником виступає перемикання контексту: студентам потрібно тримати в пам’яті синтаксичні правила та домовленості кожної мови, швидко переходити від одного стилю запису до іншого, а також співвідносити типові конструкції (умовні оператори, цикли, робота зі строками, масивами або списками). Така особливість підсилює когнітивне навантаження і потребує більш чіткої побудови маршруту навчання.

Щоб полегшити міжмовну інтеграцію, варто створити карту відповідностей між Python і JavaScript: короткі таблиці або пам’ятки, які показують, як у кожній мові записуються базові операції (вивід, введення, умови, цикли, функції, робота з колекціями). Наявність таких матеріалів знижує час на згадування синтаксису та дає можливість зосередитися на алгоритмічному мисленні й розумінні задачі, а не на пошуку правильного написання конструкцій.

Результати анкетування не лише підтверджують загальну ефективність курсу, а й уточнюють конкретні напрямки методичного вдосконалення: диференціація складності, посилення пояснювальних матеріалів, поетапна підтримка у великих завданнях, активніша комунікація та стандартизація організаційних процесів у Google Classroom. Реалізація цих кроків потенційно підвищить стабільність навчальної мотивації та зменшить частку студентів, які відчувають труднощі на інтеграційних етапах.

### 4.3. Співставлення отриманих результатів з результатами аналогічних досліджень

Отримані в межах даної магістерської роботи результати узгоджуються з висновками попередніх досліджень у галузі електронного навчання та методики викладання програмування. У наукових працях вітчизняних і зарубіжних авторів наголошується на доцільності використання проєктно-орієнтованого та інтегрованого підходів до навчання програмуванню.

Зокрема, результати досліджень, у навчанні програмуванню, свідчать про позитивний вплив інтерактивних завдань і автоматизованого зворотного зв’язку на навчальні досягнення студентів. Отримані в межах даної роботи дані підтверджують ці висновки та доповнюють їх аспектом міжмовної інтеграції.

Порівняння з результатами досліджень, у яких розглядається навчання програмуванню на основі одного інструмента або мови, показує, що інтегроване використання Python і JavaScript сприяє формуванню більш глибокого розуміння

принципів розробки програмного забезпечення. Такий підхід дозволяє студентам краще орієнтуватися у сучасних тенденціях веб-розробки та практичної діяльності програміста.

Отримані результати також узгоджуються з ідеями компетентнісного підходу в освіті, згідно з яким важливим є не лише засвоєння окремих знань, а й здатність комплексно застосовувати їх для розв'язання практичних завдань. Реалізація інтеграційного навчального проєкту в межах курсу сприяла формуванню саме таких компетентностей.

Співставлення результатів даного дослідження з результатами аналогічних наукових праць підтверджує актуальність обраного напряму дослідження та доцільність використання інтегрованого підходу до навчання мов програмування Python і JavaScript у середовищі Google Classroom.

У багатьох дослідженнях електронного навчання програмуванню наголошується, що одним із найсильніших чинників успішності є швидкий зворотний зв'язок щодо помилок і прогресу. Студентам важливо не лише отримати оцінку, а й побачити пояснення: у якому місці допущено помилку, як її виправити, і які принципи стоять за правильним розв'язанням. У межах розробленого курсу ця ідея реалізовувалася через поєднання практичних завдань та коментарів до виконаних робіт у Google Classroom.

Отримані результати також свідчать, що навіть організаційно просте середовище може бути ефективним за умови правильної дидактичної побудови курсу: чітких дедлайнів, зрозумілих критеріїв оцінювання, послідовності тем і логічного зв'язку між теорією та практикою. У цьому контексті Google Classroom виступає не стільки як програмне середовище, скільки як системоутворювальна платформа для керування навчальним процесом і підтримки комунікації.

Порівняно з курсами, що фокусуються на одній мові програмування, інтегрований підхід має подвійний ефект. З одного боку, він сприяє перенесенню знань і формуванню гнучкого мислення: студент бачить, що одна й та сама ідея може реалізовуватися різними синтаксичними засобами. З іншого боку,

зростають вимоги до структурування матеріалу та підтримки, оскільки ризик перевантаження на певних етапах є вищим. Оцінки складності, отримані в анкетуванні, підтверджують актуальність цього балансу.

Водночас практика міжмовних інтеграційних проєктів відповідає сучасним вимогам до підготовки ІТ-фахівців, адже реальні задачі часто передбачають взаємодію клієнтської та серверної частин, роботу з API, обробку даних і представлення результатів у веб-інтерфейсі. Саме тому підхід, що поєднує Python і JavaScript, може розглядатися як більш наближений до професійного контексту, ніж ізольоване вивчення кожної мови окремо.

Узагальнюючи, можна стверджувати, що результати даного дослідження вписуються у загальну тенденцію розвитку цифрової освіти: поєднання проєктно-орієнтованого навчання, інтерактивних завдань і хмарних платформ для організації взаємодії. Специфікою роботи є акцент на міжмовній інтеграції та практичній реалізації курсу в Google Classroom, що доповнює існуючі підходи та демонструє можливість застосування інтегрованої методики в умовах обмежених інструментальних ресурсів.

# ВИСНОВКИ

У магістерській роботі розглянуто проблему розробки та дослідження освітнього веб-ресурсу для вивчення мов програмування в умовах цифрового освітнього середовища. Актуальність теми зумовлена зростанням ролі програмування у професійній діяльності фахівців інформаційних технологій, а також необхідністю впровадження сучасних інтерактивних підходів до навчання в закладах вищої освіти.

У межах дослідження обґрунтовано доцільність інтегрованого підходу до вивчення мов програмування Python і JavaScript. Показано, що використання цих мов у межах одного навчального проєкту дозволяє сформувати у студентів цілісне уявлення про процес розробки програмного забезпечення, зокрема про взаємодію серверної та клієнтської частин веб-застосунку. Такий підхід сприяє розвитку міжмовних компетентностей програмування та системного мислення.

У ході практичної реалізації було розроблено електронний навчальний курс на платформі Google Classroom, який структуровано відповідно до навчальних цілей і логіки поступового ускладнення матеріалу. Курс включає теоретичні матеріали, інтерактивні приклади коду, практичні завдання з автоматичною перевіркою та підсумкове інтеграційне завдання, орієнтоване на застосування мов Python і JavaScript у комплексі.

Педагогічний експеримент, проведений у межах дослідження, засвідчив позитивний вплив використання розробленого електронного курсу на рівень сформованості знань і практичних навичок студентів. Узагальнені результати експериментального навчання свідчать про підвищення якості виконання практичних завдань, зростання навчальної мотивації та кращу орієнтацію студентів у структурі програмних проєктів.

Аналіз та педагогічна інтерпретація отриманих результатів підтвердили ефективність інтегрованого підходу до навчання програмуванню. Використання єдиного навчального проєкту сприяє активнішому залученню студентів до

навчального процесу та наближає освітній контекст до реальних умов професійної діяльності програміста.

Результати дослідження узгоджуються з висновками вітчизняних і зарубіжних науковців щодо доцільності проєктно-орієнтованого та компетентнісного підходів у навчанні програмуванню. Це підтверджує наукову обґрунтованість обраного напряму дослідження та практичну цінність запропонованої методики.

Разом із тим слід відзначити певні обмеження проведеного педагогічного експерименту. По-перше, результати ґрунтуються на даних пілотного впровадження курсу та анкетування учасників, що відображає насамперед суб'єктивне сприйняття складності, інформативності та мотивації. По-друге, на ефективність навчання могли впливати зовнішні чинники (попередній досвід програмування, різний рівень самоорганізації, технічні умови доступу до інтернету). Тому отримані висновки доцільно розглядати як обґрунтовані для описаних умов та як основу для подальшої верифікації на більшій вибірці.

На основі результатів експерименту можна сформулювати практичні рекомендації щодо вдосконалення курсу: запровадити вступний модуль з налаштуванням робочого середовища та правилами взаємодії у Classroom; збільшити кількість прикладів із поясненнями та мініпідсумків після тем; додати диференційовані завдання (базовий і поглиблений рівні) та шаблони для інтеграційних проєктів; посилити регулярний зворотний зв'язок через коментарі до робіт і короткі оголошення; передбачити механізми самоперевірки та контрольні точки в межах складних практичних робіт.

Перспективним напрямом є розвиток системи навчальної аналітики курсу: відстеження динаміки виконання завдань, типових помилок, часу на проходження тем, активності взаємодії в коментарях. Поєднання таких даних із результатами тестування та анкетування дозволить точніше оцінювати ефективність окремих методичних рішень і приймати рішення щодо зміни структури курсу на основі об'єктивних показників. У практичному вимірі це

може привести до створення більш адаптивного курсу, який враховує різні стартові рівні підготовки та підтримує індивідуальні траєкторії навчання.

Практичне значення магістерської роботи полягає у можливості використання розробленого електронного курсу в освітньому процесі закладів вищої освіти, а також у можливості адаптації запропонованої методики для навчання інших мов програмування або ІТ-дисциплін. Матеріали роботи можуть бути корисними викладачам, які впроваджують електронне та змішане навчання.

Перспективи подальших досліджень полягають у розширенні функціональних можливостей електронного курсу, збільшенні кількості інтеграційних проєктів, а також у проведенні більш масштабних педагогічних експериментів з метою детальнішого аналізу ефективності інтегрованого навчання програмуванню.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Бабенко В. А. Роль інтелектуальної власності у формуванні фахових компетентностей здобувачів вищої економічної освіти в Україні. *Економічний простір*. 2019. № 152. С. 204–213. DOI: 10.32782/2224-6282/152-16.
2. Бугайчук К. Л. Електронний підручник: поняття, структура, вимоги. *Інформаційні технології і засоби навчання*. 2011. Т. 22, № 2. DOI: 10.33407/itlt.v22i2.437.
3. Воробець О. Інформаційні технології у контексті формування цифрової компетентності майбутніх учителів. *Відкрите освітнє е-середовище сучасного університету*. 2019. Спецвипуск «Нові педагогічні підходи в STEAM освіті». С. 398–404. DOI: 10.28925/2414-0325.2019s36.
4. Воротникова І. Досвід використання е-підручників і електронних засобів навчального призначення в умовах цифровізації загальної середньої освіти України. *Інформаційні технології і засоби навчання*. 2019. Т. 71, № 3. С. 23–39. DOI: 10.33407/itlt.v71i3.2552.
5. Женченко М., Мельник О., Мірошниченко В., Женченко І. Electronic Textbooks for Ukrainian Education: Statistics, Models of Development, Quality Problems. *Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications*. 2020. Vol. 2732. P. 721–733. URL: https://ceur-ws.org/Vol-2732/20200721.pdf (дата звернення: 01.09.2025).
6. Ілійчук Л. Сучасні вимоги щодо розробки та впровадження електронних підручників в освітній процес початкової школи. *Відкрите освітнє е-середовище сучасного університету*. 2019. Спецвипуск «Нові педагогічні підходи в STEAM освіті». С. 123–132.
7. Кухарський В., Осередчук О. Електронний підручник в українській вищій освіті: від ідеї створення до реалізації. *Вісник Національного університету «Львівська політехніка»*. 2017. № 879. С. 84–90.

8. Лапінський В. В. Electronic educational resources – didactic requirements and classification. *Нові інформаційні технології в освіті для всіх: навчання протягом життя : тези VIII міжнар. конф.* Київ : ІППО «Нова школа», 2013. URL: https://lib.iitta.gov.ua/id/eprint/2004/ (дата звернення: 14.09.2025).

9. Леонтьєва Л. В. Особливості викладання дисципліни «Інтелектуальна власність» під час підготовки фахівців вищої школи. Харків : ХНАДУ, 2021. С. 40–44. URL: https://dspace.khadi.kharkov.ua/server/api/core/bitstreams/88942c61-4d92-4040-b063-085886df0e9b/content (дата звернення: 20.09.2025).

10. Лунячек В. Е., Рубан Н. П., Тіманюк В. М., Фесенко Н. С., Черненко Ю. Ю. Освіта в сфері інтелектуальної власності як умова сталого розвитку держави. *ScienceRise. Pedagogical Education*. 2017. № 11. С. 4–9. DOI: 10.15587/2519-4984.2017.116197.

11. Національна академія внутрішніх справ. Положення про охорону інтелектуальної власності в Національній академії внутрішніх справ : наказ № 1270 від 29.12.2020. URL: https://www.navs.edu.ua/naukova-diyalnist/intelektualna-vlasnist/polozhennya-pro-ohoronu-intelektualnoyi-vlasnosti-v-navs.pdf (дата звернення: 21.09.2025).

12. Перспективи використання електронних підручників під час навчання морської англійської мови. *Наукові записки НаУ «Острозька академія». Серія «Філологія»*. 2024. Т. 21, № 89. С. 195–200. URL: https://journals.oa.edu.ua/Philology/article/view/4089 (дата звернення: 28.09.2025).

13. Смаглюк Л. В., Карасюнок А. Є., Ляховська А. В., Воронкова Г. В. Електронний навчальний посібник «Propaedeutics of Orthodontics» : свідоцтво про реєстрацію авторського права № 128534. 2024.

14. Стахів М. Е-підручники для української школи: проблеми підготовки. *Записки Львівської національної наукової бібліотеки України імені В. Стефаника*. 2019. Т. 11, № 27. С. 169–180.

15. Python (programming language). Wikipedia. URL: https://en.wikipedia.org/wiki/Python. (дата звернення: 02.10.2025).

16. JavaScript. Wikipedia. URL: https://en.wikipedia.org/wiki/JavaScript. (дата звернення: 04.10.2025).

17. Babaev N. Z., Mazmayeva A. A., Dikeninskaya E. N. Education in the area of intellectual property as a condition of the stable state development. URL: https://www.researchgate.net/publication/324962137 (дата звернення: 05.10.2025).

18. Chung K. S., Byun H. W., Kim S., Yu H. C. Interactive digital textbook development methodology for higher education. *International Journal on Advanced Science, Engineering and Information Technology*. 2018. Vol. 8, No. 4–2. P. 1534–1539. URL: https://www.researchgate.net/publication/328920019 (дата звернення: 12.10.2025).

19. Navone E. C. Python vs JavaScript – Key Differences Between Two Popular Programming Languages. *freeCodeCamp*. URL: https://www.freecodecamp.org. (дата звернення: 15.10.2025).

20. Difference between Python and JavaScript. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org. (дата звернення: 17.10.2025).

21. Guth S. Electronic Rights Enforcement for Learning Media. *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2002)*. Kazan, USA, 2002. URL: https://40hz.org/Papers/icalt45.pdf (дата звернення: 26.10.2025).

22. Kline D., Kappos D. Introduction to Intellectual Property : open textbook. Minneapolis : OpenStax, 2021. ISBN 978-1-951693-35-0. URL: https://open.umn.edu/opentextbooks/textbooks/1336 (дата звернення: 26.10.2025).

23. Balalaieva O. Y. Analysis of theoretical and methodological approaches to design of electronic textbooks for students of higher agricultural educational institutions. *Information Technologies and Learning Tools*. 2017. Vol. 59, No. 3. P. 39–50. DOI: 10.33407/itlt.v59i3.1663.

24. Onaifo D. Alternate Academy: Investigating the Use of Open Educational Resources by Students at the University of Lagos in Nigeria : electronic thesis. 2016. URL: https://ir.lib.uwo.ca/etd/4086. (дата звернення: 31.10.2025)

25. Collis B., Strijker A. Technology and human issues in reusing learning objects. *Journal of Interactive Media in Education*. 2004. No. 4. Article 4. URL: https://jime.open.ac.uk/articles/10.5334/2004-4-collis. (дата звернення: 02.11.2025)

26. Valieiev R. Electronic textbooks as a factor for improving online learning in higher education. URL: https://www.researchgate.net/publication/376262085. (дата звернення: 03.11.2025)