

Природничо-математичний факультет
Кафедра інформатики і обчислювальної техніки

Кваліфікаційна робота

освітнього ступеня «бакалавр»

на тему

**РОЗРОБКА ЕЛЕКТРОННОГО МУЗЕЮ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ “ЧЕРНІГІВСЬКИЙ КОЛЕГІУМ”
ІМЕНІ Т. Г. ШЕВЧЕНКА**

Виконала:

студентка 4 курсу, 44-фмт групи
спеціальності

122 Комп’ютерні науки

Лебідь Олена Олександрівна

Науковий керівник:

к.п.н., доц. Цибко Г.Ю.

Чернігів – 2025

Роботу подано до розгляду « _____ » _____ 2025 року.

Студент (ка)

(підпис)

Лебідь О.О.
(прізвище та ініціали)

Науковий керівник

(підпис)

Цибко Г.Ю.
(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Кваліфікаційна робота розглянута на засіданні кафедри
Інформатики і обчислювальної техніки протокол № _____
від « _____ » _____ 202__ року.

Студент (ка) допускається до захисту даної роботи в екзаменаційній комісії.

Завідувач кафедри

(підпис)

Горошко Ю.В.
(прізвище та ініціали)

Анотація

У межах кваліфікаційної роботи здійснено аналіз і розробку вебсайту цифрового музею фізико-математичного факультету Чернігівського колегіуму ім. Т.Г. Шевченка.

Робота охоплює вирішення ключових завдань:

1. Системний аналіз інформації.
2. Проектування архітектури та дизайну.
3. Підбір засобів та середовища реалізації.
4. Розробка вебсайту з урахуванням адаптивності.
5. Забезпечення доступності цифрового музею.
6. Опис можливостей розширення функціоналу.

Ключові слова: електронний музей, цифровий архів, веброзробка.

Abstract

Within the framework of this qualification work, analysis and development of a digital museum website for the Faculty of Physics and Mathematics at T.H. Shevchenko National University «Chernihiv Colehium» was conducted.

The work encompasses resolution of key objectives:

1. Systematic information analysis.
2. Architecture and design planning.
3. Selection of implementation tools and environment.
4. Website development with adaptive design consideration.
5. Ensuring digital museum accessibility.
6. Description of functionality enhancement capabilities.

Keywords: electronic museum, digital archive, web development.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 ПЛАНУВАННЯ РЕАЛІЗАЦІЇ ЕЛЕКТРОННОГО МУЗЕЮ ТА ВИБІР СЕРЕДОВИЩА РОЗРОБКИ.....	7
1.1. Аналіз проблематики створення електронного музею.....	7
1.2. Обґрунтування вибору середовища та засобів розробки.....	9
РОЗДІЛ 2 СИСТЕМНЕ ПРОЕКТУВАННЯ ТА ВЕБДИЗАЙН ЦИФРОВОГО МУЗЕЮ.....	11
2.1. Розробка інформаційної моделі майбутнього вебсайту.....	11
2.2. Розробка візуального макету.....	13
2.3. Вдосконалення дизайну.....	14
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЕЛЕКТРОННОГО МУЗЕЮ.....	19
3.1. Побудова каркасу сторінки. Ідея структури.....	19
3.2. Імплементація загального функціоналу з використанням JavaScript	20
РОЗДІЛ 4. ВАЛІДАЦІЯ, ТЕСТУВАННЯ Й ОЦІНКА ЕФЕКТИВНОСТІ. ПУБЛІКАЦІЯ ТА ХОСТИНГ. ПІДТРИМКА ЖИТТЄВОГО ЦИКЛУ СТВОРЕНОГО ПРОДУКТУ.....	38
4.1. Тестування продукту.....	38
4.2. Загальний огляд та аналіз створеного проєкту, опис переваг та недоліків.....	38
4.3. Хостинг вебсайту. Використання платформи GitHub.....	39
4.4. Підтримка життєвого циклу та процедури оновлення контенту.....	40
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58

ВСТУП

У межах кваліфікаційної роботи здійснено дослідження у напрямку практичної розробки цифрового музею фізико-математичного факультету Чернігівського колегіуму імені Т.Г. Шевченка. Досліджувальний об'єкт охоплює чітко визначений історичний період й має конкретні часові межі існування, а привернута до нього увага має на меті збереження історичної спадщини навчального закладу. Реалізація проєкту відбулася у форматі розробки вебсайту, який структурує інформацію та надає вільний доступ до її використання.

Актуальність реалізації цього проєкту визначена кількома провідними чинниками, серед яких важливість переведення архівних даних у цифровий формат. Особливо критичного значення набуває факт нестабільної ситуації в країні, адже паперові документи та фото носії інформації знаходяться під більшим ризиком пошкодження чи знищення. Цифрове форматування усуває цю проблему, та, на цей час, гарантує довготривале збереження даних. Також, проєкт доступний незалежно від фізичної локації, роблячи його територіально необумовленим. Іншою важливою характеристикою є поширення доступності інформації. Конкретно визначена структура даних зменшує витрати часу на пошук і тим самим підвищує продуктивність у дослідницькій та науковій сферах.

Мета роботи полягає у збереженні історичної спадщини від забуття та створенні ресурсу зі зручно організованим матеріалом та вільним доступом до нього. Вебтехнології - найефективніший спосіб досягнення поставленої мети.

Завдання дослідження полягають в аналізі наявних підходів до розробки електронних музеїв; систематизації архівних матеріалів фізико-математичного факультету; створенні вебдизайну проєкту; проєктуванні зручного інтерфейсу; виборі відповідних середовищ реалізації; тестуванні працездатності створеного ресурсу.

Об'єкт дослідження – процес цифровізації документів та їх розміщення на вебресурсі.

Предмет дослідження – розробка вебсайту цифрового музею фізико-математичного факультету НУЧК.

Теоретичні підходи і методи. У роботі застосовано системний підхід до організації інформаційних ресурсів. Використано такі методи аналізу та синтезу для вивчення існуючих рішень у сфері цифрових музеїв: метод моделювання для створення структури вебресурсу й експериментальний метод для тестування функціональності системи.

Практична значущість роботи полягає у створенні функціонального цифрового архіву, який може бути використаний в освітньому процесі, науково-дослідницькій діяльності та для популяризації історії навчального закладу. Реалізований проєкт у форматі вебсайту, який структурує інформацію та забезпечує її збереження, виконуючи функцію цифрового архіву.

Апробація результатів дослідження

Деякі матеріали, які були задіяні у кваліфікаційній роботі, були обговорені на таких конференціях:

1. Лебідь О. О., Цибко Г. Ю. Графічний дизайн в ІТ-бізнесі: теоретичний і прикладний аспекти // Збірник матеріалів VII Міжнародної науково-практичної конференції студентів і молодих учених «Інформаційні технології в соціокультурній сфері, освіті та економіці». 2023. С. 140-142.
2. Цибко Г. Ю., Лебідь О. О. Цифрова доступність музеїв: вибір ефективних фреймворків // Збірник матеріалів міжнародної науково-практичної конференції здобувачів вищої освіти і молодих учених «Інформаційні технології в сучасному світі». Харків: ДБТУ, 2025. С. 59-60.
3. Лебідь О. О., Цибко Г. Ю. Цифрові музеї: інтеграція технологій, стандарти та стратегічні інструменти // Збірник матеріалів Міжнародної науково-практичної конференції студентів і молодих учених «Інформаційні технології в соціокультурній сфері, освіті та економіці» (23-24 квітня 2025 р.). С. 243-245.

Кваліфікаційна робота має практичне значення та наукову цінність, сприяючи архівуванню даних, підвищенню обізнаності та оптимізації у науково-дослідницькому середовищі.

РОЗДІЛ 1 ПЛАНУВАННЯ РЕАЛІЗАЦІЇ ЕЛЕКТРОННОГО МУЗЕЮ ТА ВИБІР СЕРЕДОВИЩА РОЗРОБКИ

1.1. Аналіз проблематики створення електронного музею.

Інформація як ресурс характеризується потребою у систематизації та належному збереженні. Особливо це стосується історичних даних, які мають підвищену цінність та потребують спеціального підходу. На тлі сучасного технологічного прогресу, ефективним рішенням є шлях відцифрування даних. Порухнене питання є критично актуальним. Зокрема, Чернігівський колегіум імені Т. Г. Шевченка є одним із характерних прикладів, де питання історичної інформаційності потребує комплексного рішення.

Об'єктом дослідження кваліфікаційної роботи став фізико-математичний факультет згаданого навчального закладу. Історія факультету охоплює 100-річний період функціонування: від тих часів, коли заклад мав назву Чернігівський інститут народної освіти у 1917 році й майбутній факультет був одним зі трьох відділень[1] до 2017 року, де внаслідок реорганізації відбулося злиття факультетів, трансформувались у природничо-математичний[2].

Чинні форми збереження академічної історії не забезпечують можливість швидкого й оперативного доступу до більшості історичних матеріалів. Наукові праці, присвячені історії університету, такі як монографія А. Боровика та М. Боровика "Історія Чернігівського національного педагогічного університету у біографіях його керівників"[1] та колективна праця В. С. Кролевця, В. У. Черноус, В. Ф. Шморгуна "Півстоліття невтомної праці"[3], хоч і мають окреслену структуру та описують перебіг подій становлення університету, проте не охоплюють всієї інформації з її науковими діячами. Слід зазначити, що інформація про фізико-математичний факультет у працях, наприклад, такій як Філон Л.Г. та Ященко О.М. "Маловідомі сторінки історії фізико-математичного факультету Чернігівського національного педагогічного університету імені Т. Г.

Шевченка (1916–1941 роки)”[4], висвітлена оглядово й доволі стисло; фрагментарно описана в інших джерелах.

Дослідження виявило недостатню увагу до 100-річного пласта історії факультету з діяльністю співробітників та педагогів, їхніми інноваційними методологіями та здобутками. Відсутність систематизованого підходу з представленням у цифровому форматі створює прогалину в історичному надбанні закладу.

Концепція електронного музею, як сучасного засобу збереження та подання історичних матеріалів відповідає сучасним методам представлення інформації та ефективно розв’язує поставлене завдання.

Питання співіснування цифрових та традиційних форм подання потребує окремого огляду. Електронний музей не замінює фізичну версію, а доповнює її, надаючи більше можливостей популяризації історичних матеріалів[5]. Відповідно до сучасного подання інформації, цифрові технології мають ряд певних переваг, як можливість мультимедійно презентувати вкладену інформацію або інтерактивно взаємодіяти з користувачами у необмеженому часопросторовому доступі до ресурсу.

Успішна реалізація описаної концепції цифрових музеїв демонструється провідними закладами України. Зокрема, електронний музей Київського національного економічного університету імені Вадима Гетьмана (КНЕУ)[6] демонструє комплексний підхід реалізації, використовуючи сучасні вебтехнології.

Згадані наукові праці стали фундаментом для реалізації поставленої мети. Та практична концепція збирання даних ґрунтувалася на поєднанні інформації отриманої у ході досліджень фізичного музею, державного архіву Чернігівської області та мережі інтернет. На підставі отриманих даних, проєкт має ширше охоплення.

З огляду на це, розробка електронного музею фізико-математичного факультету Чернігівського колегіуму ім. Т.Г. Шевченка являє собою науково

обґрунтований проєкт, спрямований на поширення і збереження академічної спадщини в умовах сучасного подання інформації.

1.2. Обґрунтування вибору середовища та засобів розробки.

Реалізація електронного музею у вигляді вебсторінки зумовлена рядом переваг. Таких як: доступність, кросплатформність та швидке оновлення матеріалу.

Перед початком вибору середовища розробки було сформовано вигляд бажаного результату проєкту. У ході системного аналізу накопиченого пласта інформації, виокремлено :часову шкалу, перелік біографічних даних, ряд фотокарток й список літератури. Такий підхід допоміг зробити оптимальний вибір, відповідний умовам проєкту.

З визначеною структурою, середовищем розробки було обрано Visual Studio Code[7] — багатофункціональний редактор коду від компанії Microsoft. Вибір обґрунтовується рядом переваг, до яких відноситься висока продуктивність, інтуїтивний інтерфейс, підтримка синтаксису з автоматичною підсвіткою коду, широкий спектр розширень для оптимізованої роботи, інтегрований термінал для виконання команд й засоби для попереднього перегляду результату та інше. Серед плагінів використано у цьому проєкті:

- Auto Close Tag авторством Jun Han[8]
- Auto Complete Tag авторством Jun Han[9]
- Auto Rename Tag авторством Jun Han[10]
- Image preview авторством Kiss Tamás[11]
- Live Server авторством Ritwick Dey[12]

Код реалізовано мовою розмітки HTML (HyperText Markup Language) з використанням таблиці стилів CSS (Cascading Style Sheets) та мовою програмування JavaScript для забезпечення інтерактивності й динамічності контенту.

Для оптимізації процесу розробки, було обрано фреймворку Bootstrap[13] - відкриту бібліотеку CSS та JavaScript. Обраний інструмент містить частини вже описаних компонентів інтерфейсу з вбудованими стилями, полегшуючи роботу над користувацьким інтерфейсом.

Не зважаючи на свою простоту й загальнодоступність Bootstrap - це потужний фреймворк, який дозволяє створювати професійні проєкти з незначними витратами часу.

Обрані засоби фігурують у створенні оптимальної моделі у рамках обраної кваліфікаційної роботи й виправдовують співвідношення між ефективністю, простотою та функціональністю.

РОЗДІЛ 2 СИСТЕМНЕ ПРОЕКТУВАННЯ ТА ВЕБДИЗАЙН ЦИФРОВОГО МУЗЕЮ

2.1. Розробка інформаційної моделі майбутнього вебсайту

Процес формування інформаційної архітектури електронного музею здійснювався на основі системного підходу[14], який передбачав комплексний підхід до структуризації наявного історичного масиву даних. Розглянемо пройдені етапи.

1. Збір та інвентаризація даних. Виокремлення необхідних архівних матеріалів, періодичних видань та наукових праць; відцифрування фотоматеріалів та їх корекція створили необхідний масив.
2. Визначення основних категорій. Класифікація зібраних матеріалів на тематичні групи (хронологія, історичні постаті, архіви) мали на меті розподілити інформацію на блоки.
3. Створення ієрархічної структури. Передбачає подачу інформації у вигляді багаторівневої системи, де кожен елемент займає певне місце у загальній структурі.

Послідовність виконання цих етапів й реалізують системність підходу.

В результаті критичного огляду на масив даних, було прийнято рішення реалізувати історію фізико-математичного факультету через призму персоналій — від керівників до випускників. Оскільки протягом всього існування саме людський фактор визначав характер на напрямок його розвиток.

Одним з основних завдань інформаційного проектування стало розробка логічної часової схеми всіх перетворень факультету, його розгалужень та численних реорганізацій. Найбільш доступним та зрозумілим форматом було обрано хронологічну шкалу. Подібна візуалізація дозволяє користувачу простежити весь час існування факультету в контексті загальних освітніх та соціально-політичних процесів. Хронологічна шкала розміщена в теці *img* під

назвою “hrono.png” і розроблена за допомогою програми Adobe Photoshop[15], що забезпечило високу якість графічного відображення історичних етапів та їх візуальну інтеграцію із загальним дизайном вебресурсу.

У принципі академічної ієрархії було виділено чотири основні категорії: декани, викладацький склад, наукові працівники та випускники. Окрім загального оглядового опису, кожна постать має окрему вебсторінку, якщо достатньо інформації. Вмістом є біографічні дані, наукові здобутки, публіцистична діяльність й посилання на використані джерела.

Викладацький склад, хоча не є вичерпним, проте має широкий перелік науково-педагогічних представників. Категорія співробітників охоплює лаборантів та інших спеціалістів, що брали участь у житті факультету. Блок випускників містить інформацію про перший випуск та постатей, які досягли успіхів у науковій чи підприємницькій діяльності.

Відповідно розділам, описано матеріали про 8 деканів, 62 викладачі, 5 співробітників й 6 випускників, на момент створення електронного музею.

Всі розділи, окрім деканського, не є повними через брак історичних даних. Проте вебсайт розрахований на доповнення інформації через бекенд. Детальна інструкція викладена у РОЗДІЛ 4.4.

Останні два розділи присвячені фотоархіву й документальним джерелам. Фотоархів електронного музею представлений у хронологічній послідовності фотографій, які відображають життя факультету: офіційні заходи, навчальний процес, наукові збори, студентське життя. Усі фотографії відцифровано та оброблено з використанням Adobe Photoshop[15]. Було виправлено технічні артефакти, покращено якість зображень та архівовано.

Документальний архів містить колекцію першоджерел, які стали основою всього контексту. До них відносяться: повні тексти наукових праць, архівні документи, тематичні публікації, архівні дані та періодичні видання, зокрема майже повна колекція студентського часопису "УНІВЕР СІТУ". Документи можливо як переглядати так і завантажувати.

Загальний огляд демонструє сім розділів електронного музею, чотири з яких присвячені ієрархічним категоріям персоналій, де майже у кожного є можливість переходу до індивідуальної бібліографічної сторінки. Реалізована структура надає персональний підхід до презентації історичної інформації через індивідуальні внески кожного науково-педагогічного представника з академічної спільноти у розвиток фізико-математичного факультету.

2.2.Розробка візуального макету

На основі опрацьованої інформації було створено перший дизайн. Реалізація шаблону була виконана за допомогою професійного інструменту Figma[16] — платформи, яку використовують у будіванні макетів та прототипів вебсторінок чи додатків. Перша версія дизайну (рис. 2.1. Перша версія електронного музею) була створена для розміру комп'ютера й демонструвала візуальну концепцію майбутнього електронного музею.

Колористична гамма вебресурсу базувалася на принципі контрастності з використанням збалансованої палітри кольорів та їх відтінків. Основну частину оформлення склали два ключові кольори: фоновий колір #D9D2CC (теплий нейтральний відтінок) та колір заголовкового блоку #28392F (темно-зелений). Комбінація цих тонів утворюють контраст для комфортного сприйняття текстової інформації. Основним шрифтом було обрано Open Sans через його високу читабельність в цифровому середовищі.

Характерною особливістю дизайну стало використання закруглених кутів для елементів інтерфейсу, що створює візуально приємне та сучасне оформлення. Застосована концепція узгоджується з принципами “user experience design” спрямоване на створення комфортного середовища для користувачів вебресурсу. М'які форми елементів та збалансована колористика знижують візуальну втому та покращують загальне сприйняття контенту.

оглядовою інформацією першого варіанту. Причиною зменшення кількості розділів біографії стала недостатність даних.

Причиною зменшення кількості розділів біографії стала недостатність даних.

Розділ з документами тепер містить загальну інформацію про кожну архівну довідку. Додано опис, автори, рік публікації та характерні особливості, кількість сторінок, вага документа. Реалізовано з можливістю переглядання і завантаження при натиску на відповідну кнопку.



Рис. 2.2. Друга версія електронного музею

Ця версія мала покращення проте мала недоліки. Тому у процесі розробки дизайн знову зазнав суттєвих змін. Третя версія стала фінальною (рис. 2.3. Третя версія електронного музею).

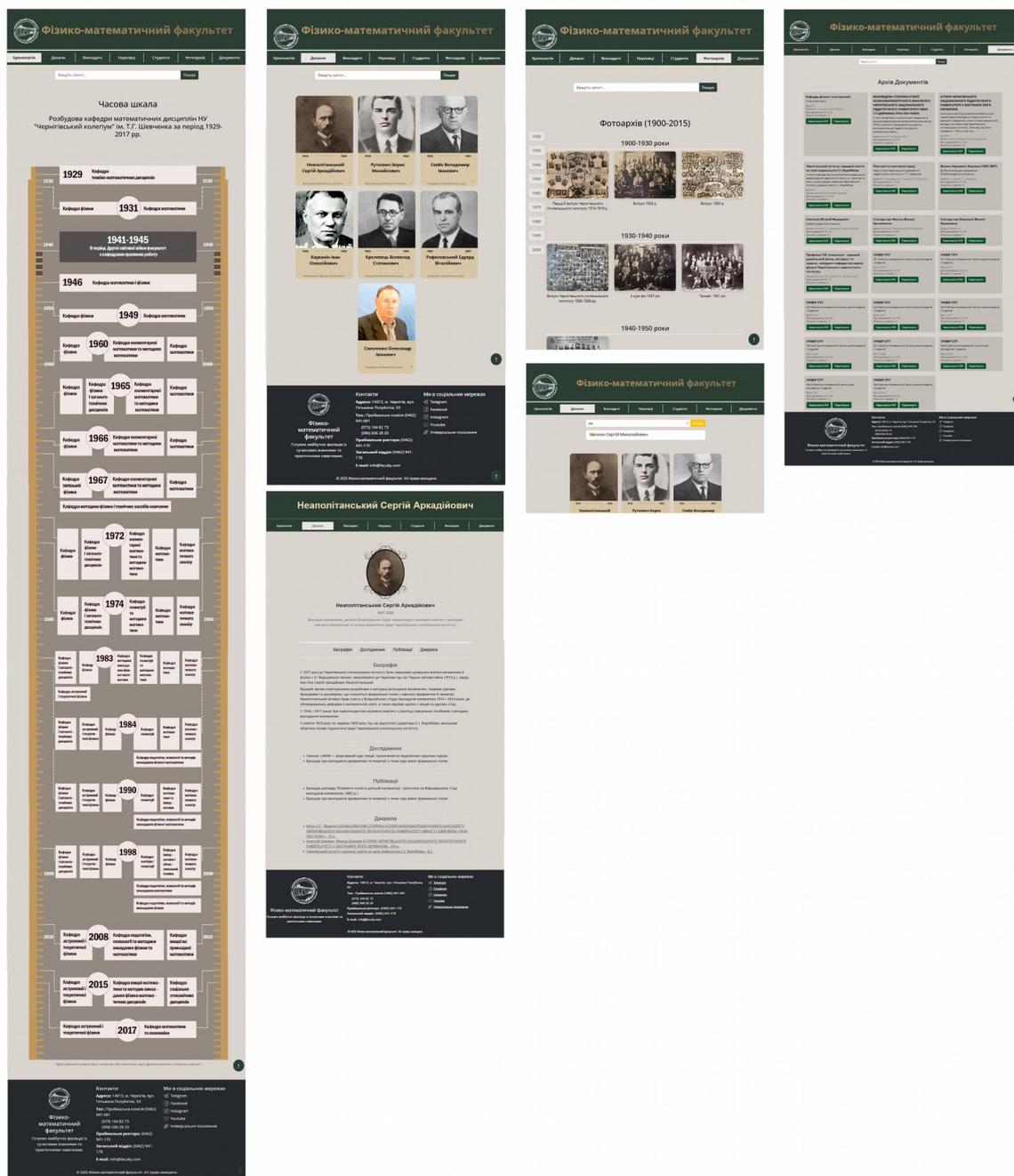


Рис. 2.3. Третя версія електронного музею

Тепер картки персонажів й документу розташовані три в ряд. У фотоархіві було додано бокову панель з роками для швидшого пошуку.

Відбулися перейменування меню, змінено дизайн карток й кольорів (рис. 2.4. Таблиця кольорів).

#212529	#28392F	#3B5242	#0D4E28
#4F4A4A	#8E847C	#D9D2CC	#F0E9E4
#C09457	#FFC107	#E9E5E1	#F8F9FA
#D7C4A3	#E8DCB6	#FFF3CD	#FFFACD

Рис. 2.4. Таблиця кольорів

Фінальна версія має адаптації сторінки до формату планшетів та телефонів (рис. 2.5 Адаптивність меню на прикладі планшета (580x1025)).

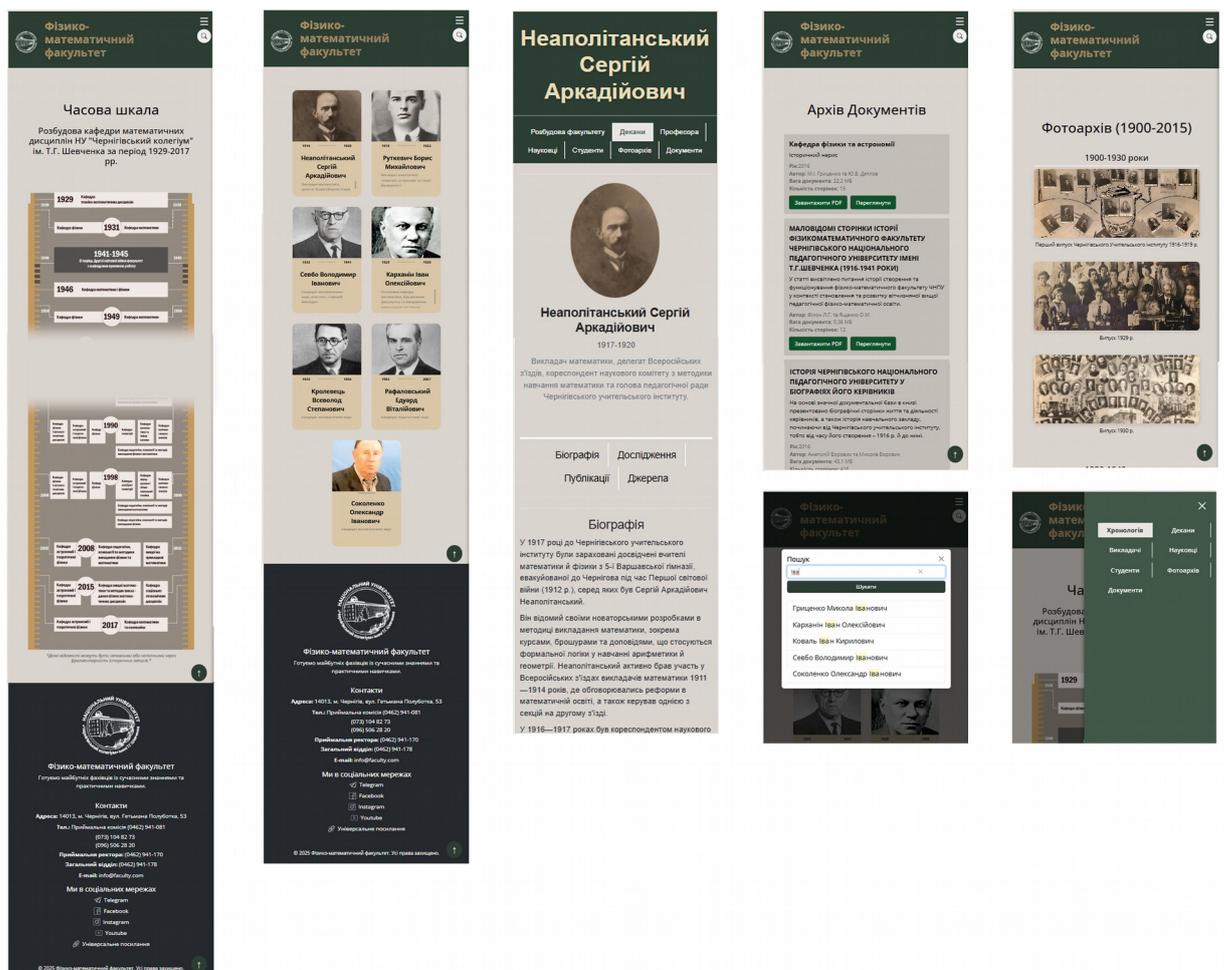


Рис. 2.5 Адаптивність меню на прикладі планшета (580x1025)

Для компактності, меню і пошукова стрічка стали висувними.

Інтуїтивно зрозумілі функції розміщуються у шапці профілю. Меню висувається справа від користувача й закривається при натисканні на хрестик, а також при кліку на будь-яку зону поза меню або при зміні розміру екрана.

Пошукова стрічка відтворюється в окремому модальному вікні посередині користувацького інтерфейсу.

Весь функціонал, попри різну подачу, залишився незмінним.

На прикладі екрана для планшета видно описаний дизайн. Аналогічний функціонал реалізовано й у мобільній версії, яка відрізняється від планшетної лише розміром об'єктів та елементів.

Остання версія вебресурсу демонструє оптимальне співвідношення між естетичними якостями та функціональними характеристиками, забезпечуючи інтуїтивно зрозумілу навігацію та візуальну привабливість контенту відповідно до специфіки електронного музею.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЕЛЕКТРОННОГО МУЗЕЮ

3.1. Побудова каркасу сторінки. Ідея структури.

Проект електронного музею включає основні вебфайли (HTML, CSS, JavaScript) та три тематичні директорії: "doc", "img" та "people".

Директорія "doc" призначена для зберігання документальних матеріалів та має таку структуру: тека "газета" містить файли часопису "УНІВЕР СІТІ", а "Спогади" — текстові документи зі спогадами про описаних персоналій.

Директорія "img" організована у 12 тек: 10 з них містять фотографії, систематизовані за десятиліттями, окремо виділені теки "icons" для іконок елементів інтерфейсу та "портрети" для зображень описаних осіб. Фотоматеріали збережено у форматі WebP для оптимізації швидкості завантаження та економії дискового простору. Додатково директорія містить три PNG-файли: "hrono.png" (часова шкала головної сторінки), "no_photo.png" (фотокартка для відсутніх портретів) та fimat_logo (логотип минулого факультету).

Директорія "people" містить індивідуальні сторінки персоналій, іменовані англійськими прізвищами. Базова структура описана у "name.html", а специфічний контент додається через JavaScript.

Структура вебресурсу побудована за модульним принципом з чітким розподілом на секції. Розділ "Хронологія" описаний в файлі "index.html". Персональні розділи ("Декани" — "decanu.html", "Викладачі" — "profesors.html", "Співробітники" — "employees.html", "Випускники" — "students.html") використовують уніфікований підхід до оформлення через спільний файл стилів "card.css". Розділи "Фотоархів" ("albom.html") та "Документи" ("doc.html") мають спеціалізовані файли стилів відповідно "albom.css" та "doc.css". Загальне оформлення та адаптивність забезпечується файлом "base.css", який відповідає за стилізацію меню, підвалу та адаптацію під

різні розміри екранів. Для оптимізації коду шапка з навігаційним меню та підвал сторінки винесені в окремі файли "menu.html" та "footer.html" й підключаються до сторінок через JavaScript. Такий підхід, запобігає дублюванню повторюваних елементів у структурі проекту й нагромадженню коду.

Файл "main_script.js" забезпечує динамічне завантаження компонентів карток, реалізацію системи пошуку з базою даних персоналій, функціонування адаптивного меню та інтеграцію модульних елементів сторінки.

3.2. Імплементация загального функціоналу з використанням JavaScript .

З огляду на наявність часто повторюваних фрагментів коду було прийнято рішення про використання методу сепарації таких елементів та реалізацію модульної організації інтерфейсних частин. Динамічне завантаження компонентів здійснюється за допомогою засобів асинхронного завантаження HTML через Fetch API.

Як підключається меню й футер.

Процес ініціалізації меню використовує обробник подій "DOMContentLoaded", який забезпечує виконання скрипту після повного завантаження DOM-структури документа:

```
document.addEventListener("DOMContentLoaded", () => {  
    // Код ініціалізації  
});
```

Виконуваний скрипт незалежний від порядку завантаження та перехоплює збої, пов'язані зі спробами взаємодії з неіснуючими DOM-елементами.

Асинхронність завантаження меню забезпечується методом `fetch()`, який створює запит до сервера:

```
fetch("menu.html")
  .then((response) => response.text())
  .then((html) => {
    document.getElementById("header-container").innerHTML =
html;
  })
```

Promise-based архітектура обробляє асинхронні операції з можливістю ланцюжкового виклику функцій обробки. Отримання HTML-даних у текстовому форматі реалізується методом `response.text()`, який через властивість `innerHTML` перетворює отримані дані на відповідні HTML-елементи.

Як тільки навігаційне меню завантажується активується каскад пост-процедурних функцій для забезпечення функціональності інтерфейсу:

```
highlightActiveMenuItem();
if (typeof populatePage === "function") {
  populatePage();
}
checkUrlForCardScroll();
```

Функція `highlightActiveMenuItem()` відповідає за виділення поточного пункту меню для орієнтування користувача. Умовний виклик `populatePage()` забезпечує гнучкість архітектури, не порушуючи загальну структуру сайту. Функція `checkUrlForCardScroll()` реалізує прокрутку до конкретних секцій сторінки за допомогою визначених ідентифікаторів.

Система обробки помилок реалізована через метод `catch()`, який перехоплює помилки під час виконання та ізолює проблемну частину, дозволяючи іншим компонентам програми продовжувати функціонування:

```
.catch((error) => console.error("Помилка завантаження шаблону меню:", error));
```

Підвал сайту використовує аналогічний принцип, але з спрощеною реалізацією без пост-процедурної обробки:

```
fetch('footer.html')
  .then(response => response.text())
  .then(data => {
    document.getElementById('footer-container').innerHTML =
data;
  });
```

Адаптивність меню.

Функція `toggleMenu()` керує станом адаптивного меню:

```
function toggleMenu(force = null) {
  const menu = document.getElementById('menu-container');
  const overlay = document.getElementById('menu-overlay');

  const isActive = menu.classList.contains('active');
  const shouldShow = force !== null ? force : !isActive;

  if (shouldShow) {
    menu.classList.add('active');
    overlay.classList.add('active');
    document.body.style.overflow = 'hidden';
  } else {
    menu.classList.remove('active');
  }
}
```

```
        overlay.classList.remove('active');
        document.body.style.overflow = '';
    }
}
```

Параметр `force` дозволяє встановити стан меню примусово (`true/false/null`), клас `active` використовується для перевірки поточного стану. Логіка перемикання реалізована за принципом інверсії: якщо поточний стан `true`, новий стан встановлюється `false`, і навпаки.

Ініціалізація системи автоматичного закриття відбувається після повного завантаження DOM:

```
document.addEventListener('DOMContentLoaded', () => {
    // код виконується після побудови структури сторінки
});
```

Спочатку визначаються необхідні елементи меню та напівпрозорий шар (оверлей):

```
const menu = document.getElementById('menu-container');
const overlay = document.getElementById('menu-overlay');
```

Примусове початкове закриття меню здійснюється через видалення класу `active`, оскільки бокове меню має викликатися виключно за кліком користувача на відповідну іконку:

```
menu.classList.remove('active');
overlay.classList.remove('active');
```

Закриття при кліці на оверлей:

```
overlay.addEventListener('click', () => toggleMenu(false));
```

Закриття при натисканні поза межами меню:

```
document.addEventListener('click', (e) => {  
  const isMenuOpen = menu.classList.contains('active');  
  const isClickInsideMenu = menu.contains(e.target);  
  const isMenuToggle = e.target.closest('.menu-toggle');  
  
  if (isMenuOpen && !isClickInsideMenu && !isMenuToggle) {  
    toggleMenu(false);  
  }  
});
```

Система перевіряє чи відкрито меню (`isMenuOpen`), чи був клік всередині меню (`isClickInsideMenu`) та чи не був натиснутий перемикач меню (`isMenuToggle`). Якщо меню відкрито, клік був поза межами меню та не по кнопці-перемикачу, викликається `toggleMenu(false)` для приховування меню.

Закриття при натисканні клавіші Escape:

```
document.addEventListener('keydown', (e) => {  
  if (e.key === 'Escape') toggleMenu(false);  
});
```

Закриття при зміні розміру вікна:

```
window.addEventListener('resize', () => {  
  toggleMenu(false);  
});
```

Описана реалізація забезпечує адаптивність меню відповідно до очікувань користувачів.

Як працює кнопка прокрутки “Нагору”.

Кнопка прокрутки "Нагору" призначена для швидкого повернення користувача до початку сторінки без необхідності ручного прокручування контенту.

Система відстеження позиції користувача для прокрутки реалізована наступним чином:

```
const scrollToTopBtn = document.getElementById('scrollToTopBtn');

window.onscroll = function () {
    if (document.body.scrollTop > 50 ||
document.documentElement.scrollTop > 50) {
        scrollToTopBtn.style.display = "block";
    } else {
        scrollToTopBtn.style.display = "none";
    }
};
```

Механізм відображення базується на відстеженні позиції прокрутки сторінки з пороговим значенням 50 пікселів. Кросбраузерна сумісність забезпечується подвійною перевіркою значень `document.body.scrollTop` та `document.documentElement.scrollTop`, оскільки різні браузери можуть використовувати різні властивості для визначення позиції прокрутки.

Функціональність повернення до початку сторінки реалізована через пряме присвоєння нульових значень:

```
scrollToTopBtn.onclick = function () {
```

```
document.body.scrollTop = 0;
document.documentElement.scrollTop = 0;
};
```

Такий підхід забезпечує миттєву прокрутку до початку документа з урахуванням особливостей різних браузерів.

Як здійснюється пошук.

Пошукова функція на вебсторінці працює у режимі реального часу. Вона реалізована за принципом прогресивного вебдизайну підтримуючи десктопну і мобільну версії інтерфейсів.

Інформація для пошуку організована у масив `searchData`, де кожен об'єкт містить три ключові властивості:

```
const searchData = [
    { title: 'Адаменко Валентин Васильович', url:
'People/Adamenko.html', cardId: 'Adamenko' },
    { title: 'Атросченко Ф.Г', url: 'profesors.html', cardId:
'Atroschenko' },
    // ... інші записи
];
```

Властивість `title` використовується для пошуку та відображення назви, `url` містить шлях до індивідуальної сторінки персоналій, а `cardId` представляє унікальний ідентифікатор для навігації до конкретного елемента на сторінці у випадку їх відсутності.

Центральна пошукова функція `findMatches()` реалізує алгоритм нечіткого пошуку:

```
function findMatches(query) {
    const trimmed = query.trim().toLowerCase();
    return searchData.filter(item =>
```

```
        item.title.toLowerCase().includes(trimmed)
    );
}
```

Функція `filter()` створює новий масив з елементами, що відповідають критеріям пошуку. Метод `includes()` здійснює перевірку після переведення запиту у нижній регістр.

Система підсвічування знайдених збігів реалізована функцією `highlight()`:

```
function highlight(text, term) {
    if (!term || term.trim() === '') return text;
    const regex = new RegExp(`(${term})`, 'gi');
    return text.replace(regex, '<mark>$1</mark>');
}
```

Вирази з прапорцями глобального пошуку `g` та нечутливості до регістру `i` використовуються для заміни входжень пошукового терміна HTML-елементом `<mark>`, що реалізує візуальне виділення знайдених фрагментів.

Формування URL-адрес здійснюється функцією `createUrl()`:

```
function createUrl(item) {
    if (item.cardId) {
        return `${item.url}#${item.cardId}`;
    }
    return item.url;
}
```

Функція формує URL-адреси, додаючи ідентифікатор `#cardId` для навігації до конкретного елемента на сторінці.

Динамічне подання результатів пошуку виконується функцією `showResults()`:

```

function showResults(resultsElement, matches, query) {
  if (matches.length === 0) {
    return false;
  }
  resultsElement.innerHTML = `

Метод map() перетворює масив знайдених елементів у HTML-розмітку, створюючи активний список посилань з підсвіченими збігами.



Пошук у реальному часі реалізований через обробник подій addEventListener:



```

mobileSearchInput.addEventListener('input', (e) => {
 const query = e.target.value.trim();
 mobileClearButton.style.display = query.length > 0 ? 'block' :
'none';

 if (query.length >= 2) {
 const matches = findMatches(query);
 showResults(mobileSearchResults, matches, query);
 } else {
 mobileSearchResults.innerHTML = '';
 }
});

```



28


```

Обробник подій `input` активується при кожній зміні вмісту пошукової стрічки. Валідація мінімального значення визначено мінімальною кількістю у 2 символи для оптимізації продуктивності системи.

Підтримка клавіатурних команд здійснюється наступним чином:

```
mobileSearchInput.addEventListener('keydown', (e) => {
  if (e.key === 'Enter') {
    performSearch(mobileSearchInput, mobileSearchResults);
  } else if (e.key === 'Escape') {
    closeSearch();
  }
});
```

Клавіша `Enter` ініціює пошук, а `Escape` закриває пошукове вікно.

Функція `performSearch()` централізує логіку пошуку та валідації:

```
function performSearch(input, resultsElement) {
  const query = input.value.trim();

  if (query.length < 2) {
    showMessage(resultsElement, 'Введіть щонайменше 2 символи
для пошуку');
    return;
  }

  const matches = findMatches(query);
  if (!showResults(resultsElement, matches, query)) {
    showMessage(resultsElement, `Нічого не знайдено за запитом
"${query}"`);
  }
}
```

Система відображає повідомлення користувачу та автоматично приховує їх через чотири секунди за допомогою функції `showMessage()`.

Підсвічування знайдених результатів реалізовано через динамічне створення стилів:

```
const style = document.createElement('style');
style.textContent = `
    .highlighted-card {
        animation: highlight-pulse 3s;
    }
    @keyframes highlight-pulse {
        0% { box-shadow: 0 0 0 0 rgba(66, 139, 202, 0.7); }
        50% { box-shadow: 0 0 0 20px rgba(66, 139, 202, 0); }
        100% { box-shadow: 0 0 0 0 rgba(66, 139, 202, 0); }
    }
`;
```

Анімація привертає увагу користувача до результатів пошуку. Адаптивний дизайн забезпечує коректне відображення на різних типах пристроїв через використання окремих наборів DOM-елементів та обробників подій з єдиною логікою функціонування.

Як заповнюються карти.

Створення інформаційних блоків у розділах "Декани", "Викладачі", "Співробітники" та "Випускники" реалізується однаково через динамічну генерацію DOM-елементів з інтерактивними компонентами на основі структурованих даних.

Основою системи є масив об'єктів `cards`, що містить інформацію про кожну персоналію:

```
const cards = [
    {
```

```
    link: 'People/Adamenko.html',
    img: 'img/no_photo.png',
    alt: 'Адаменко Валентин Васильович',
    cardId: 'Adamenko',
    date: { year1: "1920", year2: "1929" },
    name: { line1: 'Адаменко Валентин Васильович' },
    description: 'Викладав курси введення в аналіз...'
  }
];
```

Кожен об'єкт містить:

- Навігаційні властивості: `link` (посилання на детальну сторінку) та `cardId` (унікальний ідентифікатор)
- Візуальні елементи: `img` (шлях до зображення) та `alt` (альтернативний опис)
- Темпоральні дані: `date` (роки перебування в університеті)
- Ідентифікаційну інформацію: `name` (повне ім'я)
- Описову частину: `description` (посада або звання)

Центральна функція `createCards()` здійснює динамічне створення DOM-елементів: \

```
function createCards() {
    const container = document.getElementById("ucl-card-
container");

    if (!container) {
        console.error("Контейнер для карток не знайдено");
        return;
    }
}
```

Для кожного об'єкта в масиві створюється елемент-посилання `<a>` методом `createElement()`:

```
cards.forEach(card => {
  const a = document.createElement("a");

  if (card.link) {
    a.href = card.link;
    a.setAttribute("data-has-link", "true");
    a.setAttribute("data-link", card.link);
  } else {
    a.href = "#";
    a.setAttribute("data-has-link", "false");
  }
}
```

Система використовує `data`-атрибути для реалізації логіки посилань та збереження метаінформації про стан елемента.

Кожен елемент отримує унікальні ідентифікатори та ARIA-атрибути:

```
if (card.cardId) {
  a.id = card.cardId;
}
a.className = "ucl-card-link";
a.setAttribute("data-person-name", card.name.line1);
a.setAttribute("aria-label", `Details about ${card.name.line1}`);
```

Внутрішня структура картки створюється через `template literals` з використанням семантичної HTML-розмітки:

```
a.innerHTML = `
  <div class="ucl-card">
    <div class="ucl-image-container">
```

```

        
    </div>
    <div class="ucl-date-bar" aria-label="Years active">
        <span class="ucl-year-left">{card.date.year1}</span>
        <span class="ucl-dash"></span>
        <span class="ucl-year-right">{card.date.year2}</span>
    </div>
    <div class="ucl-info">
        <h class="ucl-info-title">
            {card.name.line1}
        </h>
    </div>
    <hr class="ucl-divider" />
    <div class="ucl-description-container">
        <div class="ucl-description" aria-label="Event
location and description">
            {card.description}
        </div>
    </div>
</div>
`;

```

Використовуються наступні принципи HTML-семантики:

- ARIA-атрибути для опису призначення елементів (aria-label)
- Альтернативний текст для зображень (alt)
- Логічна ієрархія заголовків та контейнерів
- Семантичні розділювачі (<hr>)

Реалізовано диференційну поведінку залежно від наявності детальної інформації:

```

a.addEventListener("click", function (event) {
    if (this.getAttribute("data-has-link") === "true") {
        return; // Стандартна поведінка переходу за посиланням
    }
});

```

```

    } else {
        event.preventDefault();
        showNoInfoModal(this.getAttribute("data-person-name"));
    }
});

```

При наявності посилання на індивідуальну сторінку виконується перехід, за його відсутності активується модальне вікно з відповідним повідомленням.

Завершальним етапом є додавання створеного елемента до контейнера:

Метод `appendChild()` забезпечує коректне вбудовування елементів в чинну DOM-структуру з автоматичним запуском процесу рендерингу браузером.

Стан керується через систему data-атрибутів:

- `data-has-link` — індикатор наявності активного посилання
- `data-link` — збереження URL для подальшого використання
- `data-person-name` — ім'я особи для відображення в модальних вікнах

Як влаштовані індивідуальні сторінки

Система побудована на модульному підході, що базується на використанні HTML-шаблонів та динамічного JavaScript-наповнення. Така архітектура забезпечує розділення структури (HTML) та логіки (JavaScript), що підтримує статичність дизайну при гнучкості наповнення контенту та уникає дублювання коду.

Основний шаблон `name.html` містить семантичну структуру сторінки з визначеними секціями:

```

<header class="text-center mb-4">
    <h1 id="page-title" class="mb-3"></h1>
    <p class="lead text-muted" id="page-subtitle"></p>
    <nav class="nav justify-content-center border-top pt-3 w-100
text-center">
        <a class="nav-link flex-fill"
href="../../index.html">Хронологія</a>

```

```
        <!-- інші навігаційні елементи -->
    </nav>
</header>
```

Шаблон включає глобальну навігаційну структуру, що забезпечує узгодженість між усіма сторінками персоналій.

Контейнер з основною інформацією про персоналію:

```
<section class="text-center mb-5">
    <div class="portrait-wrapper position-relative d-inline-block">
        <img id="profile-photo" alt="Портрет" class="rounded-circle" width="200">
        
    </div>
    <h2 id="person-name" class="fw-bold"></h2>
    <p id="person-years" class="text-muted"></p>
    <p id="person-description" class="text-secondary"></p>
</section>
```

Дизайнерське рішення включає декоративну рамку для підкреслення візуальної привабливості, реалізовану через систему позиціювання з використанням CSS-класів Bootstrap для responsive-дизайну та абсолютного позиціювання для overlay-ефектів.

Асинхронне завантаження через Fetch API.

```
document.addEventListener("DOMContentLoaded", () => {
    const container = document.getElementById("template-container");

    fetch("name.html")
```

```

.then((response) => response.text())
.then((html) => {
    container.innerHTML = html;

    if (typeof populatePage === "function") {
        populatePage();
    }
}))
.catch((error) => console.error("Помилка завантаження
шаблону:", error));
});

```

Автоматичне визначення активного розділу.

```

const activeMenu = "../decanu.html";

document.querySelectorAll('.nav-link').forEach(item => {
    if (item.getAttribute('href') === activeMenu) {
        item.classList.add('active');
    }
});

```

До відповідного елемента на основі поточного шляху динамічно додається CSS-клас `active`, створюючи візуальний індикатор поточного розділу.

Динамічне повернення здійснюється наступним чином:

```

function populatePage() {
    // Управління історією навігації
    if (document.referrer.includes('../decanu.html')) {
        sessionStorage.setItem('previousPage', '../decanu.html');
    } else if (document.referrer.includes('../employees.html')) {
        sessionStorage.setItem('previousPage',
'../employees.html');
    }
}

```

```

    }

    // Встановлення базової інформації
        document.getElementById("page-title").textContent =
"Неаполітанський Сергій Аркадійович";
        document.getElementById("profile-photo").src =
"../img/портрети/Неаполітанський.webp";
        document.getElementById("person-name").textContent =
"Неаполітанський Сергій Аркадійович";
        document.getElementById("person-years").textContent = "1917-
1920";
    }

```

Використання `sessionStorage` для збереження інформації про попередню сторінку забезпечує можливість повернення на крок назад.

Повернення на попередню сторінку реалізується так:

```

document.getElementById("page-title").addEventListener("click",
function () {
    var previousPage = sessionStorage.getItem('previousPage');

    if (previousPage) {
        window.location.href = previousPage;
    } else {
        window.location.href = "../decanu.html";
    }
});

```

Детальний процес додавання біографічної інформації описано у РОЗДІЛ 4.4.

РОЗДІЛ 4. ВАЛІДАЦІЯ, ТЕСТУВАННЯ Й ОЦІНКА ЕФЕКТИВНОСТІ. ПУБЛІКАЦІЯ ТА ХОСТИНГ. ПІДТРИМКА ЖИТТЄВОГО ЦИКЛУ СТВОРЕНОГО ПРОДУКТУ.

4.1. Тестування продукту

Процес валідації створеного електронного музею включав систему тестування, спрямовану на виявлення технічних недоліків та оцінку відповідності ресурсу поставленим цілям. Функціональне тестування охоплювало перевірку працездатності всіх інтерактивних елементів, коректності відображення контенту в різних браузерях та на різних пристроях, а також стабільності роботи навігаційної системи.

Тестування користувацького досвіду проводилося з залученням представників різних вікових категорій та рівнів технічної підготовки.

Результати показали високий рівень інтуїтивності інтерфейсу та позитивну реакцію на візуальне оформлення ресурсу. Особливо відзначалася ефективність хронологічної навігації та зручність доступу до біографічної інформації.

4.2. Загальний огляд та аналіз створеного проєкту, опис переваг та недоліків.

Створений електронний музей демонструє високий рівень відповідності початковим вимогам щодо збереження та популяризації історичної спадщини фізико-математичного факультету. Структурна організація контенту забезпечує логічну послідовність подачі інформації, а візуальне оформлення сприяє комфортному сприйняттю матеріалу.

Перевірка на відповідність стандартам вебдоступності (WCAG 2.1) показала дотримання основних принципів: достатній колірний контраст, семантична розмітка HTML, підтримка клавіатурної навігації. Адаптивний

дизайн забезпечує коректне відображення на мобільних пристроях, що розширює аудиторію потенційних користувачів.

Створений електронний музей успішно вирішує поставлене завдання збереження інституційної пам'яті через цифрову платформу. Ресурс забезпечує доступ до унікальних історичних матеріалів, що раніше були недоступні широкому загалу.

Основні переваги реалізованого проєкту включають: систематизацію великого масиву історичної інформації, зручну навігаційну систему, високу візуальну привабливість та технічну надійність. Серед обмежень можна відзначити неповноту деяких біографічних даних через об'єктивну недоступність архівних матеріалів й можливість їх доповнення лише через бекенд.

Перспективи подальшого розвитку включають розширення контентного наповнення й можливу інтеграцію додаткових мультимедійних елементів.

4.3. Хостинг вебсайту. Використання платформи GitHub

Для публікації та забезпечення онлайн-доступності створеного цифрового музею було обрано платформу GitHub як оптимальне рішення для хостингу вебресурсу. GitHub надає низку переваг: безплатний хостинг статичних сайтів через GitHub Pages, інтегровану систему контролю версій, надійність серверної інфраструктури та можливість колективної роботи над проєктом.

У процесі публікації було створено спеціалізований обліковий запис `physmath-nuchk` з відповідною електронною поштою `physmathnuchk@gmail.com`. На платформі GitHub було ініціалізовано репозиторій `nuchk-physmath-museum` з головною гілкою, що містить файл `index.html` як основну сторінку ресурсу.

Завдяки функціональності GitHub Pages отримано стабільне посилання та сформовано домен <https://physmath-nuchk.github.io/nuchk-physmath-museum/>, що

забезпечує постійний публічний доступ до цифрового музею. Така реалізація гарантує високу доступність ресурсу та можливість його подальшого розвитку.

4.4. Підтримка життєвого циклу та процедури оновлення контенту

Забезпечення довготривалої функціональності та актуальності цифрового ресурсу потребує систематичного технічного супроводу та регулярного оновлення контенту. Ефективна підтримка проєкту гарантує збереження його наукової цінності та відповідність сучасним технічним стандартам.

Процес внесення змін до коду проєкту на платформі GitHub здійснюється через вебінтерфейс платформи. Послідовність дій включає: авторизацію під обліковим записом **physmath-nuchk**, перехід до репозиторію **nuchk-physmath-museum**, вибір файлу для редагування, активацію режиму редагування через кнопку "Edit this file" (іконка олівця), внесення необхідних змін до коду та збереження результатів через кнопку "Commit changes". Після підтвердження змін вони автоматично публікуються на вебсайт.

Конкретна процедура додавання або редагування інформації залежить від типу контенту та структури даних.

Додавання нової постаті складається з 3 основних етапів:

1. Створення картки особи
2. Створення індивідуальної сторінки (за наявності достатньої інформації)
3. Додавання метаданих до масиву пошуку

Етап 1: Створення картки особи

Спочатку визначте, до якої категорії належить особа:

- Викладач → файл `profesors.html`
- Співробітник → файл `employees.html`
- Випускник → файл `students.html`

Відкрийте відповідний HTML-файл та знайдіть масив `const cards`. Вставте новий запис у хронологічному порядку (за роками перебування в університеті).

Шаблон коду:

```
{  
  link: 'People/name.html',  
  img: 'img/no_photo.png',  
  alt: 'name',  
  cardId: 'idNAME',  
  date: { year1: "0000", year2: "0000" },  
  name: { line1: 'my name' },  
  description: 'describe the position'  
},
```

Заповнення подів:

`link` - посилання на біографічну сторінку

- Формат: 'People/прізвище_англійською.html'
- Якщо даних недостатньо: залиште порожнім `link`: "

`img` - зображення

Вимоги до фото:

- Формат: .webp
- Розмір: 900×1100 пікселів
- Розміщення: `img/портрети/`

Приклад:

- З фото: 'img/портрети/Александров Павло Сергійович.webp'
- Без фото: 'img/no_photo.png' (за замовчуванням)

`alt` - альтернативний текст

Повне ім'я особи (відображається, якщо фото не завантажується)

`cardId` - унікальний ідентифікатор

Прізвище особи англійською мовою

date - роки роботи

- year1 - рік початку
- year2 - рік закінчення
- Якщо працює досі: year2 залишити порожнім
- Якщо рік невідомий: поставити -
- Якщо дати невідомі: залишити лапки порожніми

name - повне ім'я

Прізвище, ім'я, по батькові

description - опис

Посада та звання особи

Етап 2: Створення картки особи

Умова: тільки при наявності достатньої кількості біографічної та наукової інформації.

Створення HTML-файлу

1. Створіть новий HTML-файл у теці People
2. Назва файлу: прізвище_англійською.html
3. Назва має відповідати значенню з поля link з попереднього етапу

Скопіюйте наступний шаблон коду в новостворений файл:

```
<!DOCTYPE html>
<html lang="uk">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="icon" href="../img/icons/profile.png"
type="person">
  <title>ПІП</title>
```

```

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/
css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="style_p.css">
    <link rel="stylesheet" href="card.css">
</head>

<body>
    <!-- Контейнер для шаблону -->
    <div id="template-container"></div>
    <div id="footer-container"></div>

    <script>
        document.addEventListener("DOMContentLoaded", () => {
            const container = document.getElementById("template-
container");

            // Завантаження шаблону
            fetch("name.html")
                .then((response) => response.text())
                .then((html) => {
                    container.innerHTML = html;

                    // Викликаємо функцію для наповнення сторінки,
якщо така є
                    if (typeof populatePage === "function") {
                        populatePage();
                    }

                    // Вказуємо активний пункт меню вручну
                    const activeMenu = "../profesors.html"; //
Шлях для активного пункту меню

                    // Перебираємо всі посилання меню

```

```

        document.querySelectorAll('.nav-
link').forEach(item => {
        // Перевіряємо, чи збігаються href з
активним меню
        if (item.getAttribute('href') ===
activeMenu) {
        // Додаємо клас "active" до
відповідного пункту меню
        item.classList.add('active');
        }
        });
    })
    .catch((error) => console.error("Помилка
завантаження шаблону:", error));
});

function populatePage() {
    // При переході на сторінку, зберігаємо попередню
сторінку
    if (document.referrer.includes('../profesors.html')) {
        sessionStorage.setItem('previousPage',
'../profesors.html');
    } else if
(document.referrer.includes('../employees.html')) {
        sessionStorage.setItem('previousPage',
'../employees.html');
    }

    // Заголовки
    document.getElementById("page-title").textContent =
"ПІП";

    // Повернення на попередню сторінку при натисканні на
заголовок

```

```

        document.getElementById("page-
title").addEventListener("click", function () {
            var previousPage =
sessionStorage.getItem('previousPage');

            if (previousPage) {
                window.location.href = previousPage;
            } else {
                // Якщо немає попередньої сторінки, можна
встановити значення за замовчуванням
                window.location.href = "../decanu.html"; //
Наприклад, повернення на "Професора"
            }
        });

// Основна інформація
        document.getElementById("profile-photo").src =
"../img/no_photo.png
";
        document.getElementById("person-name").textContent =
"ПІП";
        document.getElementById("person-years").textContent =
"";

        document.getElementById("person-
description").textContent = "";

// Біографія
        document.getElementById("biography-content").innerHTML
= `
        <p></p>`;

// Дослідження
        const researchList =
document.getElementById("research-list");

```

```

["", ""].forEach((item) => {
    const li = document.createElement("li");
    li.textContent = item;
    researchList.appendChild(li);
});

// Публікації
const publicationsList =
document.getElementById("publications-list");
[""].forEach((item) => {
    const li = document.createElement("li");
    li.textContent = item;
    publicationsList.appendChild(li);
});

// Джерела
const sourcesList = document.getElementById("sources-
list");
[
    { text: "", url: "" }
].forEach((item) => {
    const li = document.createElement("li");
    const a = document.createElement("a");
    a.textContent = item.text; // Текст для посилання
    a.href = item.url; // URL для гіперпосилання
    a.target = "_blank"; // Відкривається у новій
вкладці
    li.appendChild(a); // Додаємо гіперпосилання до
елемента списку
    sourcesList.appendChild(li); // Додаємо елемент
списку до списку на сторінці
});
}

```

```

// Активувати клікабельність для всіх пунктів меню
document.addEventListener("DOMContentLoaded", () => {
    const menuItems = document.querySelectorAll('nav a');
    menuItems.forEach((item) => {
        item.addEventListener("click", (e) => {
            e.preventDefault(); // Забороняє дефолтний
перехід
            const targetId =
item.getAttribute("href").substring(1);
            const targetSection =
document.getElementById(targetId);
            if (targetSection) {
                targetSection.scrollIntoView({ behavior:
"smooth" }); // Скрол до секції
            }
        });
    });
});
</script>
</body>

</html>

```

В секції <head>:

- <title>: замінити "ППП" на повне ім'я персони

В коментарі "Активний пункт меню":

Вказати відповідну категорію:

- ../profesors.html (для викладачів)
- ../employees.html (для співробітників)
- ../students.html (для випускників)

В секції "Заголовки":

Замінити "ППП" на повне ім'я персони

Основна інформація :

```
// Основна інформація
document.getElementById("profile-photo").src =
"../img/портрети/Прізвище.webp"; // або "../img/no_photo.png"
document.getElementById("person-name").textContent = "Повне ПІП";
document.getElementById("person-years").textContent = "1967 -
1976"; // Роки роботи
document.getElementById("person-description").textContent =
"Посада та звання";
```

Біографічні дані :

```
document.getElementById("biography-content").innerHTML = `
  <p>Перший абзац біографії з основними даними...</p>
  <p>Другий абзац з освітою та кар'єрними досягненнями...</p>
`;
```

Приклад заповнення:

```
document.getElementById("biography-content").innerHTML = `
  <p>Іван Петренко (1975-2023) – український науковець...</p>
  <p>Народився в м. Київ. Закінчив Київський національний
університет...</p>
`;
```

Дослідження :

```
const researchList = document.getElementById("research-list");
["Назва дослідження 1", "Назва дослідження 2"].forEach((item) => {
```

Приклад:

```
["Квантова механіка в напівпровідниках", "Дослідження властивостей графену"].forEach((item) => {
```

Для приховування розділу (якщо даних немає):

```
// Дослідження (приховати, якщо не потрібно)
const researchSection = document.getElementById("research");
const researchLink = document.querySelector('nav a[href="#research"]');
const showResearch = false; // Змінити на true, щоб відображати

if (!showResearch) {
    researchSection.style.display = "none";
    researchLink.style.display = "none";
}
```

Публікації:

```
const publicationsList = document.getElementById("publications-list");
["Повна назва публікації 1", "Повна назва публікації 2"].forEach((item) => {
```

Приклад:

```
["Квантові ефекти в наноструктурах. Журнал фізики, 2022", "Новітні підходи до синтезу матеріалів. Наукові записки, 2023"].forEach((item) => {
```

Для приховування розділу:

```

// Публікації (приховати, якщо не потрібно)
const publicationsSection =
document.getElementById("publications");
const publicationsLink = document.querySelector('nav
a[href="#publications"]');
const showPublications = false;

if (!showPublications) {
    publicationsSection.style.display = "none";
    publicationsLink.style.display = "none";
}

```

Джерела:

```

const sourcesList = document.getElementById("sources-list");
[
    { text: "Назва джерела 1", url: "https://example.com" },
    { text: "Назва джерела 2", url: "https://example2.com" }
].forEach((item) => {

```

Приклад:

```

[
    { text: "Профіль у Google Scholar", url:
"https://scholar.google.com/citations?user=example" },
    { text: "Інституційна сторінка", url: "https://university.edu/
faculty/profile" }
].forEach((item) => {

```

Для приховування розділу:

```

// Джерела (приховати, якщо не потрібно)
const sourcesSection = document.getElementById("sources");

```

```
const sourcesLink = document.querySelector('nav
a[href="#sources"]');
const showSources = false;

if (!showSources) {
  sourcesSection.style.display = "none";
  if (sourcesLink) {
    sourcesLink.style.display = "none";
  }
}
```

Технічні рекомендації

- Перевірте відсутність граматичних та синтаксичних помилок
- Переконайтеся у відсутності помилок JavaScript
- Перевірте працездатність усіх URL-адрес
- Дотримуйтеся єдиного стилю написання
- Протестуйте коректність відображення сторінки в браузері

Етап 3: Додавання метаданих до пошуку

Редагування файлу main_script.js

1. Відкрийте файл main_script.js у головній теці проєкту
2. Знайдіть секцію з коментарем "Пошук"
3. Локалізуйте масив searchData

Структура масиву searchData

```
const searchData = [
  { title: 'Адаменко Валентин Васильович', url:
'People/Adamenko.html', cardId: 'Adamenko' },
  { title: 'Атросченко Ф.Г', url: 'profesors.html', cardId:
'Atroschenko' },
  { title: 'Біб О.С. ', url: 'profesors.html', cardId: 'Bib' },
  // Продовження списку...
```

];

Компоненти запису

- title — повне ім'я персони у форматі "Прізвище Ім'я По батькові"
- url — відносний шлях до HTML-файлу персони (або порожнє поле, якщо файлу немає)
- cardId — унікальний ідентифікатор картки персони

Додавання нового запису

1. Визначте алфавітну позицію за прізвищем
2. Знайдіть відповідне місце у масиві
3. Додайте новий рядок за шаблоном:

```
{ title: 'Прізвище Ім\'я По батькові', url: 'шлях/до/файлу.html',  
cardId: 'ІдентифікаторКартки' }
```

Примітка: Всі записи розташовані в алфавітному порядку для зручності орієнтації.

Додатковий розділ: Алгоритм додавання фотокарток

Підготовка зображення

- Конвертувати фото у формат .webp (для оптимізації розміру файлу)
- Назвати файл за принципом: назва_опису_рік.webp (наприклад: випускник_1965.webp)
- Розмістити файл у відповідній теці за періодом

Відкриття файлу album.html

1. Відкрити файл album.html у текстовому редакторі
2. Знайти коментар з відповідним періодом за допомогою пошуку (Ctrl+F)

Пошук відповідного розділу

Приклади коментарів у кодї:

`<!-- 1960-1970 роки -->`

`<!-- 1970-1980 роки -->`

`<!-- Фото без дати -->`

Алгоритм пошуку:

1. Якщо рік фото відомий → шукати коментар відповідного десятиліття
2. Якщо рік невідомий → шукати коментар `<!-- Фото без дати -->`
3. Якщо період не існує → створити новий розділ з коментарем

Додавання коду фотокартки

Вставити наступний код після знайденого коментаря:

```
<div class="col-12 col-md-4 mb-4">
  <div class="photo-item" data-bs-toggle="modal" data-bs-
target="#photoModal" data-photo="РІК р.">
  
</div>
<p class="caption">ПІДПИС ДО ФОТО</p>
</div>
```

Заповнення параметрів

Параметр `data-photo`

- Призначення: відображається у модальному вікні як заголовок
- Формат: "1965 р." або "1960-1970 рр." або " "

Параметр `src`

- Призначення: шлях до файлу зображення
- Формат: "img/60-70/випускник_1965.webp"

Параметр alt

- Призначення: альтернативний текст для доступності
- Формат: короткий опис змісту фото

Елемент <p class="caption">

- Призначення: підпис під фото
- Формат: детальний опис того, що зображено

Приклад заповненого коду:

```
<div class="col-12 col-md-4 mb-4">
    <div class="photo-item" data-bs-toggle="modal" data-bs-
target="#photoModal" data-photo="1965 p.">
        
    </div>
    <p class="caption">Випуск випускників математичного відділення
1965 p.</p>
</div>
```

Після внесення змін до коду важливо провести комплексну перевірку функціональності:

Технічна перевірка:

- Переконалися у коректності синтаксису HTML/CSS коду
- Перевірити валідність посилань на зображення та інші ресурси
- Протестувати адаптивність верстки на різних розмірах екрану

Функціональне тестування:

- Перевірити правильність відображення всіх елементів сторінки
- Протестувати роботу інтерактивних компонентів (модальні вікна, кнопки навігації)
- Переконалися у відповідності підписів та контенту зображень

Оптимізація та якість:

- Забезпечити унікальність назв файлів при додаванні нових зображень
- Дотримуватися єдиного стилю оформлення підписів та описів
- Організовувати контент за логічними принципами (хронологія, тематика)
- При значному обсязі контенту розглянути можливість створення тематичних підрозділів

Документування змін:

- GitHub автоматично веде історію всіх змін через систему комітів
- Важливо завжди додавати змістовні коментарі при збереженні змін (commit message)

ВИСНОВКИ

Поставлені у кваліфікаційній роботі цілі та завдання повністю досягнуті. Створено функціональний, естетично привабливий та технічно робочий вебресурс, що ефективно розв'язує проблему збереження та популяризації історичної спадщини фізико-математичного факультету. Цифровий музей успішно розгорнуто на платформі GitHub Pages за адресою <https://physmath-nuchk.github.io/nuchk-physmath-museum/>, що забезпечує стабільний онлайн-доступ та надійний хостинг ресурсу.

Розроблено гібридну архітектуру вебсайту, що поєднує переваги статичного та динамічного контенту через модульну шаблонну систему. Це забезпечує швидкодію статичних сайтів при гнучкості динамічних систем управління контентом. Обрана архітектура оптимально підходить для хостингу на GitHub Pages, гарантуючи безплатне та стабільне функціонування платформи.

Створено комплексну методологію цифровізації історичних матеріалів освітніх закладів, що інтегрує принципи digital humanities з практичними потребами інституційної історіографії. Методологія включає стандартизовані підходи до структурування біографічних даних, темпоральної візуалізації та інтерактивної навігації.

Запропоновано оригінальний підхід до реалізації адаптивного пошуку через багаторівневу систему фільтрації, що забезпечує високу точність релевантності без необхідності серверної інфраструктури. Алгоритм демонструє ефективність $O(n \log n)$ при обробці великих обсягів персональних даних та повністю сумісний з обмеженнями статичного хостингу.

Електронний музей створює новий стандарт представлення інституційної історії, що може бути адаптований іншими освітніми закладами. Система забезпечує довготривале збереження історичних даних у структурованому,

доступному форматі з можливістю регулярного оновлення через GitHub-репозиторій.

Розроблено детальну систему технічного супроводу проєкту, що включає процедури додавання нового контенту, редагування наявної інформації та управління структурою даних. Створена документація забезпечує можливість подальшого розвитку ресурсу навіть користувачами з базовими технічними навичками.

Розроблена система створює фундамент для подальших досліджень у напрямку створення загальної мережі цифрових архівів освітніх закладів. Технологічні рішення можуть бути масштабовані для створення інтегрованої платформи вивчення історії української освіти та науки.

Результати дослідження мають як теоретичне значення для розвитку цифрової гуманітаристики, так і практичну цінність для освітньої спільноти Національного університету "Чернігівський колегіум" імені Т.Г. Шевченка та широкого кола зацікавлених осіб. Методологічні принципи, апробовані в роботі, мають потенціал застосування в суміжних галузях digital humanities, зокрема у створенні цифрових музеїв, віртуальних виставок та інтерактивних історичних реконструкцій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Боровик А., Боровик М. *Історія Чернігівського національного педагогічного університету у біографіях його керівників*. – Київ, 2016. 416 с.
2. Природничо-математичний факультет Чернігівського національного педагогічного університету імені Т. Г. Шевченка. Офіційний сайт ЧНПУ.URL: <http://chnpu.edu.ua/faculties/physandmath-faculty> (дата звернення: 5.06.2025).
3. Кролевець В. С., Черноус В. У., Шморгун В. Ф. *Півстоліття невтомної праці*. — Київ, 1966. 51 с.
4. Філон Л. Г., Ященко О. М. Маловідомі сторінки історії фізико-математичного факультету Чернігівського національного педагогічного університету імені Т. Г. Шевченка (1916–1941 роки). — 2017. 12 с.
5. Цибко Г. Ю., Лебідь О. О. Цифрова доступність музеїв: вибір ефективних фреймворків // *Збірник матеріалів міжнародної науково-практичної конференції здобувачів вищої освіти і молодих учених «Інформаційні технології в сучасному світі»* Харків: ДБТУ, 2025. С. 59-60.
6. Музей історії Київського національного економічного університету імені Вадима Гетьмана // Офіційний сайт КНЕУ. — URL: https://kneu.edu.ua/ua/kneu_history/museum/ (дата звернення: 6 червня 2025 р.).
7. Visual Studio Code // Офіційний сайт Visual Studio Code. — URL: <https://code.visualstudio.com/> (дата звернення: 5 червня 2025 р.).
8. Han J. Auto Close Tag // Visual Studio Code Marketplace. — URL: <https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-close-tag> (дата звернення: 5 червня 2025 р.).
9. Han J. Auto Complete Tag // Visual Studio Code Marketplace. — URL: <https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-complete-tag> (дата звернення: 5 червня 2025 р.).

10. Han J. Auto Rename Tag // Visual Studio Code Marketplace. — URL: <https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-rename-tag> (дата звернення: 5 червня 2025 р.).
11. Kiss T. Image preview // Visual Studio Code Marketplace. — URL: <https://marketplace.visualstudio.com/items?itemName=kisstkondoros.vscode-gutter-preview> (дата звернення: 5 червня 2025 р.).
12. Dey R. Live Server // Visual Studio Code Marketplace. — URL: <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer> (дата звернення: 5 червня 2025 р.).
13. Bootstrap // Офіційний сайт Bootstrap. — URL: <https://getbootstrap.com/> (дата звернення: 5 червня 2025 р.).
14. Лебідь О. О., Цибко Г. Ю. Цифрові музеї: інтеграція технологій, стандарти та стратегічні інструменти // Збірник матеріалів Міжнародної науково-практичної конференції студентів і молодих учених «Інформаційні технології в соціокультурній сфері, освіті та економіці» (23-24 квітня 2025 р.). С. 243-245.
15. Лебідь О. О., Цибко Г. Ю. Графічний дизайн в ІТ-бізнесі: теоретичний і прикладний аспекти // Інформаційні технології в соціокультурній сфері, освіті та економіці : матеріали VII Міжнародної науково-практичної конференції студентів і молодих учених. / М-во освіти і науки України; Київ. нац. ун-т культури і мистецтв. Київ : Видавничий центр КНУКіМ, 2023. - С. 140-142.
16. Figma // Офіційний сайт Figma. — URL: <https://www.figma.com/> (дата звернення: 5 червня 2025 р.).