

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ШАФОРСТ ВЛАДИСЛАВ ВЯЧЕСЛАВОВИЧ

Допускається до захисту:
в.о.завідувача кафедри
інформаційних технологій,
канд. техн. наук, доцент
_____ О.В. Зелінська
«_____» _____ 2024 р.

РОЗРОБКА ВЕБСАЙТУ «ІСТОРІЯ ЗАКЛАДУ ОСВІТИ»

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

О. В. Зелінська, доцент кафедри
інформаційних технологій,
к.т.н., доцент

Оцінка: _____ / _____ / _____

(бали за шкалою ЄКТС / за національною шкалою)

Голова ЕК: _____

(підпис)

Вінниця – 2024

АНОТАЦІЯ

Шафорост В.В. Розробка вебсайту «Історія закладу освіти». Спеціальність 122 «Комп'ютерні науки» освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2024.

Основною метою кваліфікаційної роботи є розробка вебсайту «Історія закладу освіти», що зробить доступ до інформації більш зручним та матиме конкурентоспроможний дизайн і функціонал.

У вступі розглянуто питання актуальності розробки вебсайту саме на цю тематику. В першому розділі проаналізовано та проведено огляд існуючих аналогів сайту. У другому розділі розглянуто основні інструменти та застосунки, які використовують для зручного створення вебзастосунків. В третьому розділі, виконана програмна реалізація вебсайту.

Ключові слова: вебсайт, вебдизайн, адаптивний дизайн, аналіз конкурентів, комп'ютерні науки, історія освітнього закладу, інтеграція контенту, front-end розробка, User Experience (UX), User Interface (UI), JavaScript фреймворки

Робота містить: 54 с., 1 табл., 45 рис., 11 дод., 40 джерел.

ABSTRACT

Shaforost V.V. Development of the "History of the educational institution" website. Specialty 122 "Computer Science". Vasyl Stus Donetsk National University, Vinnytsia, 2024.

The main goal of the qualification work is the development of the "History of the Educational Institution" website, which will make access to information more convenient and have a competitive design and functionality.

In the introduction, the issue of the relevance of developing a website on this topic is considered. In the first section, the existing analogues of the site are analyzed and reviewed. The second section covers the main tools and applications

that are used to conveniently create web applications. In the third section, the software implementation of the website is completed.

Keywords: Website, Web design, Responsive design, Competitor analysis, Computer science, History of an educational institution, Content integration, Front-end development, User Experience (UX), User Interface (UI), JavaScript frameworks.

The work contains: 54 pages, 1 table, 45 figures, 11 appendices, 40 sources.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ВЕБСАЙТІВ «ЗАКЛАДІВ ОСВІТИ»	7
1.1. Огляд існуючих аналогів вебсайту	7
1.2. Переваги та недоліки існуючих сайтів закладів освіти	10
1.3. Постановка задачі	11
Висновок до розділу 1	11
РОЗДІЛ 2. ОПИС ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ.....	13
2.1 Інструменти для створення сайтів.....	13
2.2 Технології розробки вебсайтів.....	18
2.3 Проектування бази даних	31
Висновок до розділу 2.....	34
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ САЙТУ «ІСТОРІЯ ЗАКЛАДУ ОСВІТИ»	35
3.1. Дизайн сайту.....	35
3.2 Програмна реалізація.....	38
3.3 Функціоналу сайту.....	42
Висновок до розділу 3.....	46
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	49
ДОДАТКИ.....	52

ВСТУП

Сьогодні інформація є ключовим ресурсом, вебсайти стають невід'ємною частиною нашого життя. Заклади освіти також ідуть у ногу із часом, вони створюють сайти де кожен може дізнатись про навчальні програми, події, досягнення та історію закладу. Створення вебсайту з історією закладу освіти є способом збереження та передачі його цінної спадщини. Адже історія закладу освіти є не лише послідовністю подій, а й унікальним досвідом, у якому присутні досягнення багатьох поколінь.

Під час створення та розробки вебсайту, потрібно приділяти особливу увагу не лише фактичним даним, а й відображати дух спільноти, що пройшла через роки. Кожні події, перемоги чи негаразди мають своє певне місце на сторінках віртуального архіву.

Розробка вебсайту «Історія закладу освіти» дозволить зберегти цінність минулого та відкрити нові можливості майбутнього.

Актуальність: Створення вебсайту «Історія закладу освіти» є актуальним у сучасних умовах, оскільки в наш час майже всю корисну та потрібну інформацію можна знайти у мережі. Наявність інтернет мережі спростила доступ до нової та корисної інформації, що у свою чергу дозволяє швидко та оперативно моніторити різні джерела інформації, що мають у собі певну історичну цінність та вагомість.

У рамках моєї бакалаврської роботи **мета** полягає у наступному:

1. Поглиблене вивчення практик розробки вебсайту.
2. Огляд та аналіз уже існуючих аналогів, що відповідають тематиці роботи.
3. Написання клієнтської частини сайту.

Задачею дослідження є аналіз існуючих аналогів вебсайту та виявлення їх особливостей і, насамперед, створення вебсайту, що допоможе дізнатись історію закладу освіти.

Об'єкт дослідження: вебсайт на тему «Історія закладу освіти».

Предмет дослідження: методи та процеси створення вебсайтів.

Структура роботи: робота складається зі вступу, трьох основних розділів, висновку та списку використаних літературних джерел.



РОЗДІЛ 1

АНАЛІЗ ВЕБСАЙТІВ «ЗАКЛАДІВ ОСВІТИ»

1.1. Огляд існуючих аналогів вебсайту

З давніх часів люди завжди прагнули здобувати нові знання, а історія освітніх установ має глибокі корені. Вони не лише місця для передачі знань, але й культурні центри, де формуються цінності та світогляд. Розвиток інформаційних технологій впливає на сферу освіти, вносячи зміни у методи навчання та доступ до знань.

Сучасне суспільство, що переповнене різноманітними вебресурсами, пропонує безліч альтернатив сайтам, спрямованим на полегшення процесу навчання. Ці ресурси стають не лише джерелом інформації, але й знаряддям для активної освіти та самовдосконалення. Платформи, що створюють заклади освіти дозволяють користувачам отримувати знання за індивідуальним графіком, розширюючи можливості дистанційного навчання та самоосвіти.

Проводячи огляд альтернатив освітньому вебсайту, ми поглиблюємо своє розуміння того, як ефективно набирати знання за допомогою інтернет-ресурсів. Цей огляд не лише сприяє визначенню найкращих ресурсів для навчання, але й пробуджує подальший прогрес у сфері освіти в цифровий час.

Розглянемо декілька аналогів нашого сайту від інших закладів освіти:

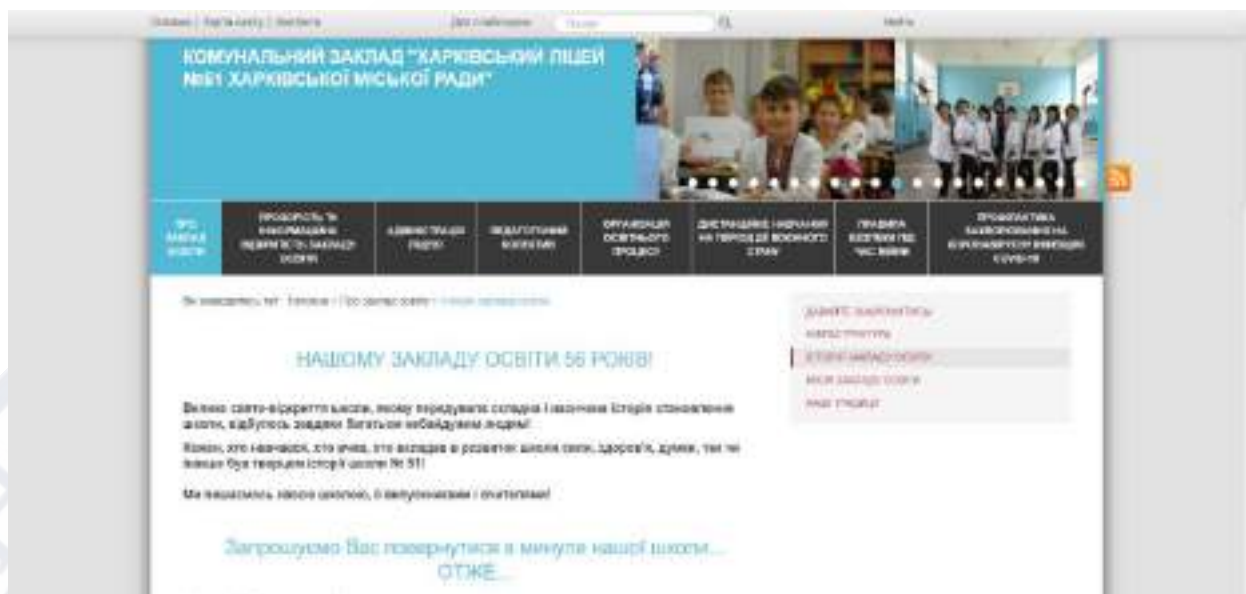


Рисунок 1.1 – Головна сторінка Комунального закладу «Харківський ліцей №51 Харківської міської ради» <http://school51.edu.kh.ua/> [1]

Комунальний заклад «Харківський ліцей №51 Харківської міської ради» - є одним з кращих та провідних закладів освіти Харківській області. Ліцей пропонує якісну освіту, включаючи дистанційне навчання, що в наш час є дуже важливим. На сайті можна побачити інформацію про його відкриття та важливі дати в його історії. Також на сайті присутні розділи, де можна побачити весь педагогічний склад, його кваліфікацію та досвід роботи з здобувачами освіти, нагороди та педагогічні звання. З цікавих нововведень можна побачити розділ з описом кожної зустрічі випускників, які з часом не забувають своє навчання у цьому закладі освіти та радо йдуть туди, щоб знову відчувати тепло рідних стін.



Рисунок 1.2 – Головна сторінка Київського ліцею «ЕКО» №198

[http://www.lyceum198.edukit.kiev.ua/\[2\]](http://www.lyceum198.edukit.kiev.ua/[2])

Київський ліцей «ЕКО» № 198 – є дуже важливим для Київського освітнього простору, він не тільки створює якісні умови для навчання, а й дуже активно впроваджує інноваційні підходи. Прикладом цього може бути інтегрування сучасних технологій для покращення навчального процесу та підвищення цікавості здобувачів освіти до отримання нових знань. На сайті цього навчального закладу можна побачити знайомі нам розділи, у яких є інформація про педагогів, що там викладають, їх кваліфікацію, нагороди та випускників, яких все ще пам'ятають і чекають на зустрічі випускників. З незвичних розділів можна виділити стратегію розвитку на 2020-2025 роки, у якій описані усі плани навчального закладу, що спрямовані на залучення нових абітурієнтів та покращення навчального процесу.

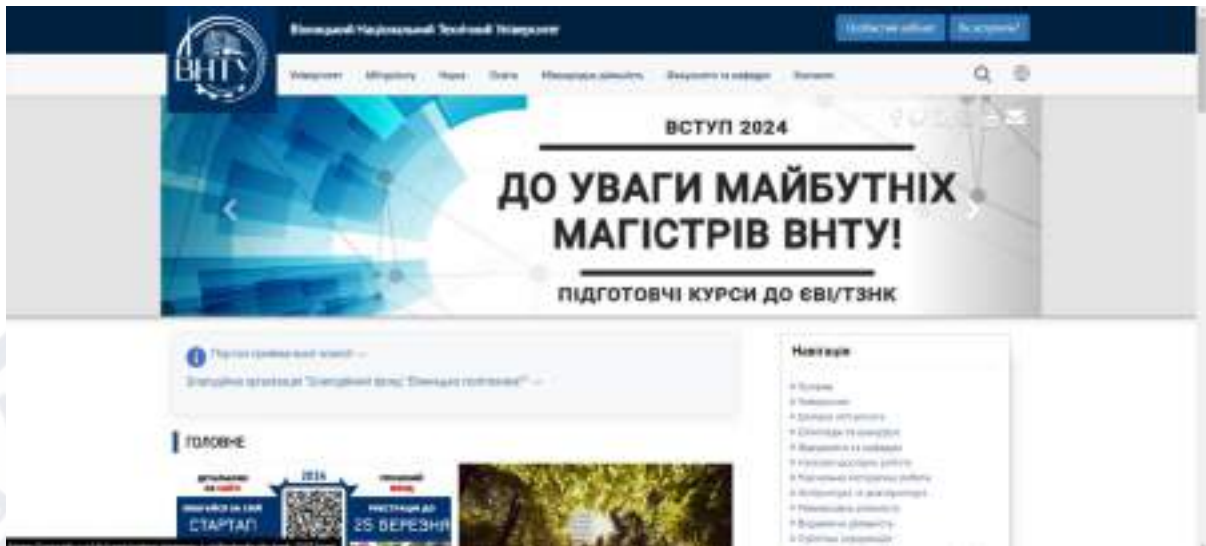


Рисунок 1.3 – Головна сторінка Вінницького національного технічного університету. <https://vntu.edu.ua> [3]

Вінницький національний технічний університет – є відомим навчальним закладом в Україні. Сьогодні ВНТУ є центром освіти і науки всього Подільського регіону. Він надає якісні умови для навчання та кар’єрного росту. На його сайті можна знайти повний перелік спеціальностей та правила вступу. Також присутні розділи для студентів, на яких є інформація про цікаві події та самодіяльність. З незвичайних розділів можна відмітити благодійний фонд. У цьому розділі можна побачити, куди витрачаються гроші з благодійного фонду та самому долучитись до певних зборів.

Отже, враховуючи огляд існуючих аналогів нашого вебсайту можна сказати, що у контексті розвитку освітнього середовища важливо враховувати не лише педагогічний аспект, але й доступність освітніх ресурсів через Інтернет. За допомогою сайтів освітніх закладів, що відкриваються перед сучасними учнями та студентами, і навіть викладачами, відбувається перегляд традиційних форм навчання та відкриття нових можливостей для здобуття знань.

1.2. Переваги та недоліки існуючих сайтів закладів освіти

Здійснивши огляд уже існуючих аналогів нашого вебсайту ми можемо здійснити коротке порівняння їх функціоналу, та визначити їх сильні та слабкі сторони.

Таблиця 1.1 – Порівняння аналогів нашого вебсайту

	school51.edu.kh.ua	lyceum198.edukit.kiev.ua	vntu.edu.ua
Дизайн	Сучасний і привабливий, але менш інтуїтивний у використанні.	Менш привабливий, але все ще зручний для навігації.	Привабливий, зручний для навігації.
Інформація	Чітка та структурована, але може бути трохи менше деталей.	Повна та легка для знаходження.	Доступна але трохи розпорошена.
Швидкість	Помірний час завантаження сторінок.	Помірний час завантаження сторінок.	Швидкий час завантаження сторінок.
Мобільна сумісність	Працює на всіх типах пристроїв.	Добре пристосований для мобільних пристроїв, але можуть виникати проблеми зі старими браузерми.	Працює добре на смартфонах, але можуть виникати проблеми при використанні планшетів.

Отже ми дійшли до висновку, що кожен з вище згаданий вебсайтів має як свої сильні сторони так і слабкі. Наприклад, доступ з різних пристроїв, деякі сайти можуть не так коректно працювати на планшетах чи старих браузерах, що в свою чергу може ускладнити їх використання.

Проте це не заважає їм виконувати поставлену задачу і доносити до громадськості всю головну інформацію про життя навчальних закладів.

1.3 Постановка задачі

Враховуючи сучасні події, інформація про створення університету є дуже актуальною, оскільки в наш час майже всю корисну та потрібну інформацію можна знайти у мережі.

Використовуючи інформацію з попередніх розділів можна зробити висновок, що наш сайт має мати певні критерії, а саме:

1. Сучасний дизайн.
2. Мати правильну постановку інформації.
3. Інтуїтивне управління.
4. Бути повністю адаптивним та крос браузерним.

Це основні характеристики, які роблять сайт привабливим та ефективним для користувачів. Ці критерії допомагають забезпечити зручний та ефективний досвід використання, незалежно від пристрою чи браузера. Таким чином, урахування цих аспектів є найбільш важливими для створення успішного та конкурентоспроможного вебсайту в сучасному інтернет-середовищі.

Висновки до розділу 1

У першій частині роботи було проаналізовано сайти освітніх організацій. Проведено дослідження існуючих аналогів вебсайтів, щоб оцінити їх особливості, переваги та недоліки. Далі проведено порівняльний аналіз, який стосувався ключових факторів оцінки ефективності сайтів. Цей порівняльний аналіз полегшив розуміння того, які аспекти веб-сайтів є найважливішими для користувачів, а які можна покращити. Завершуючи дослідження розділу 1 були відокремлені важливі аспекти, які впливають із результатів аналізу, забезпеченню створення оптимізованого та конкурентоспроможного вебсайту, пов'язаного з навчальним закладом.

РОЗДІЛ 2

ОПИС ВИКОРИСТАНИХ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ РОЗРОБКИ

2.1 Інструменти для створення сайтів

Вебпрограмування стало одним з найбільш популярних напрямків серед розробників, що призвело до великого попиту на програмне забезпечення для створення та редагування вебсайтів. Різні компанії присвячують свої зусилля розробці програмного забезпечення, яке полегшує завдання розробників, надаючи більший набір функцій. Тому розглянемо основні додатки, які спеціально розроблені для створення вебдодатків.

Bootstrap - це потужний фреймворк для розробки веб-інтерфейсів та мобільних додатків. Він заснований на HTML, CSS та JavaScript і пропонує набір готових стилів, компонентів та інструментів, що значно полегшує процес створення сучасних та респонсивних вебсайтів.

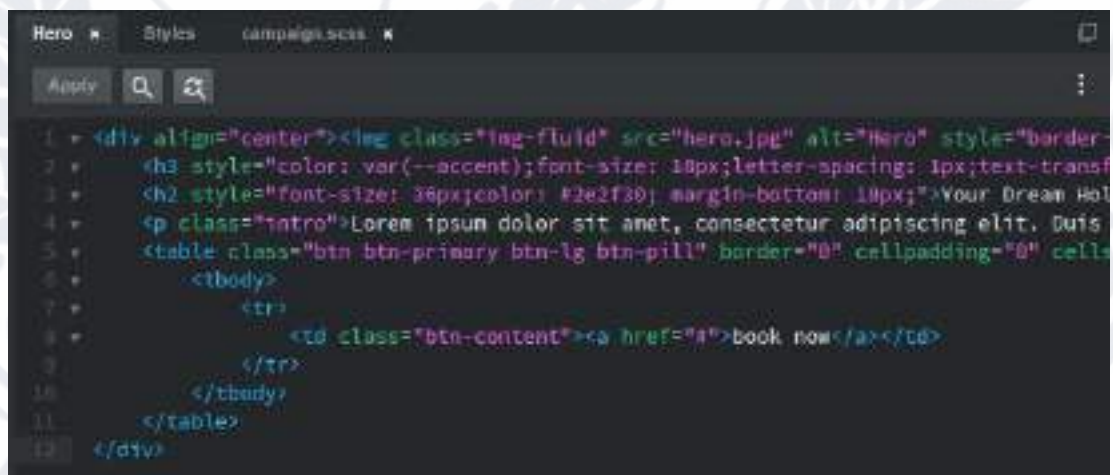


Рисунок 2.1 – Приклад написання коду використовуючи середовище Bootstrap.

Bootstrap надає гнучку сітку (grid system), яка автоматично адаптується до різних розмірів екранів, що дозволяє створювати сайти, що виглядають добре як на комп'ютерах, так і на мобільних пристроях. Також Bootstrap

містить широкий набір готових компонентів, таких як кнопки, форми, картки, модальні вікна тощо. Ці компоненти можна легко використовувати та налаштовувати за допомогою класів CSS або JavaScript.

Використання Bootstrap дозволяє значно прискорити процес розробки веб-сайтів завдяки готовим компонентам і стилям, що зменшує необхідність писати багато коду з нуля [5].

Bootstrap є одним з найпопулярніших фреймворків для веб-розробки завдяки своїй простоті використання, гнучкості та розширюваності. Він використовується мільйонами розробників по всьому світу для створення сучасних та ефективних веб-інтерфейсів.

Visual Studio Code (VS Code) - це безкоштовний і легкий текстовий редактор, який розроблений компанією Microsoft. Він призначений для редагування коду та розробки програм на різних мовах програмування, включаючи HTML, CSS, JavaScript, Python, Java, PHP та багато інших [4].



Рисунок 2.2 - Приклад написання коду використовуючи середовище VS Code.

VS Code має широкі можливості розширення завдяки великому екосистемі розширень. Розширення дозволяють розширити функціональність редактора, додавши нові можливості, підтримку різних мов програмування та інші корисні інструменти. Редактор працює швидко та ефективно, навіть при роботі з великими проектами, і має вбудовану підтримку системи контролю версій Git. Він надає розширені можливості редагування коду, такі як автоматичне доповнення коду, перевірка синтаксису, вбудовані термінали для виконання команд, відстеження помилок та багато іншого. Крім того, редактор підтримується на різних операційних системах, включаючи Windows, macOS та Linux. Visual Studio Code є одним з найпопулярніших редакторів коду завдяки своїй простоті використання, потужному функціоналу та активній спільноті розробників, яка постійно розвивається та додає нові можливості до редактора.

WebStorm - це інтегроване середовище розробки (IDE), призначене для веб-розробки, яке розроблене компанією JetBrains. Воно спрямоване на підтримку роботи з різними веб-технологіями, такими як HTML, CSS, JavaScript, TypeScript та іншими.

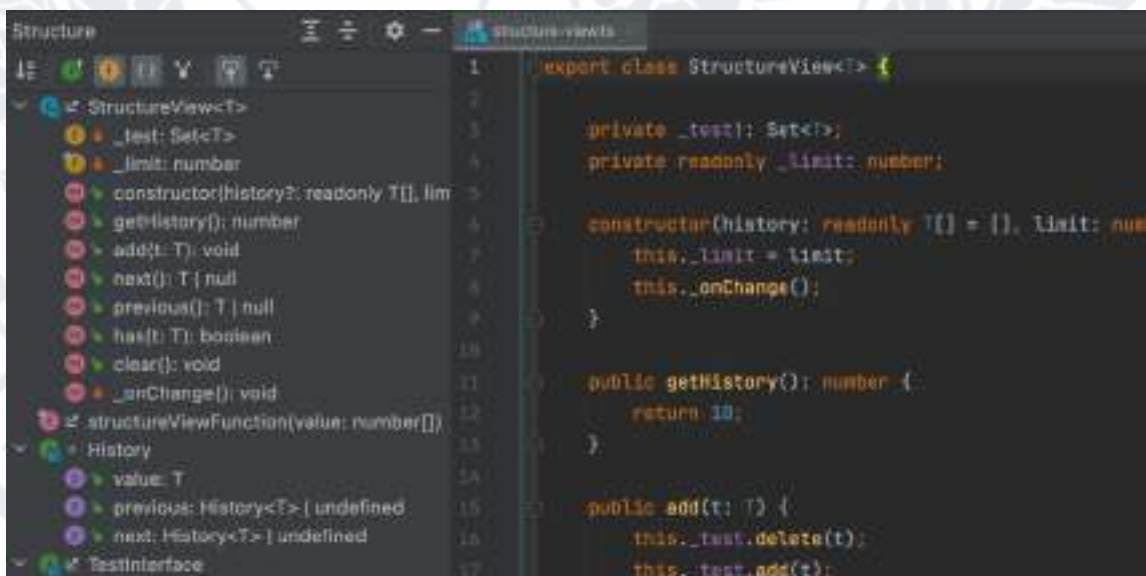


Рисунок 2.3 - Приклад написання коду використовуючи середовище WebStorm.

WebStorm інтегрований з Git, що значно полегшує роботу з системою контролю версій. Він дозволяє здійснювати коміти, злиття гілок та переглядати історію змін прямо з інтерфейсу редактора. Крім того, WebStorm має розширюваний функціонал за допомогою сторонніх плагінів, що робить його вкрай гнучким і налаштовуваним. Це ефективний та ресурсозберігаючий інструмент, що забезпечує високу продуктивність розробки веб-додатків на будь-яких мовах та фреймворках.

Sublime Text - це один з найпопулярніших текстових редакторів серед розробників програмного забезпечення та веб-розробників. Цей редактор відомий своєю простотою, швидкістю та багатифункціональністю.

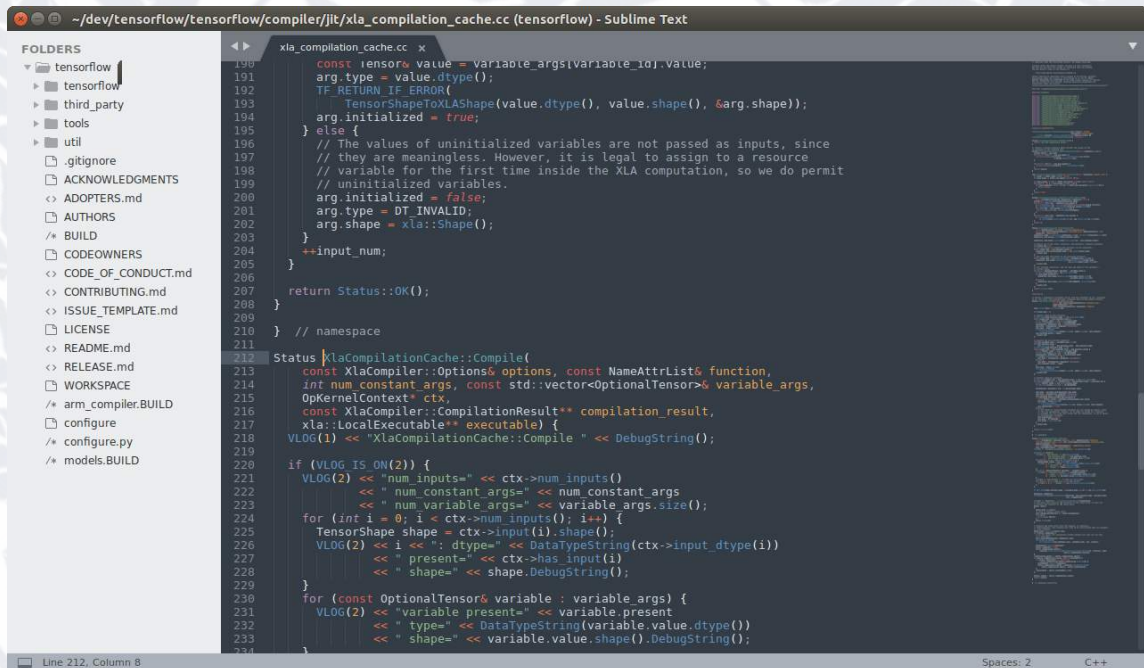


Рисунок 2.4 - Приклад написання коду використовуючи середовище Sublime Text.

Sublime Text підтримує роботу з багатьма мовами програмування, такими як HTML, CSS, JavaScript, Python, PHP та іншими, що робить його універсальним інструментом для різноманітних проектів. Крім того, Sublime Text має розширену функціональність завдяки можливості встановлення різних плагінів, які дозволяють розширити його можливості та налаштувати під конкретні потреби розробника. Цей редактор також забезпечує швидкий

доступ до різних функцій через клавіатурні комбінації, що сприяє підвищенню продуктивності роботи. Sublime Text є крос платформним рішенням і працює на різних операційних системах, таких як Windows, macOS та Linux. Загалом, він вважається одним з найефективніших редакторів коду для роботи над будь-якими проектами завдяки своїй швидкості, простоті та багатофункціональності.

Atom - це відкритий та безкоштовний текстовий редактор, розроблений компанією GitHub. Він відомий своєю розширюваністю та спрощеною інтеграцією з іншими інструментами для розробки.

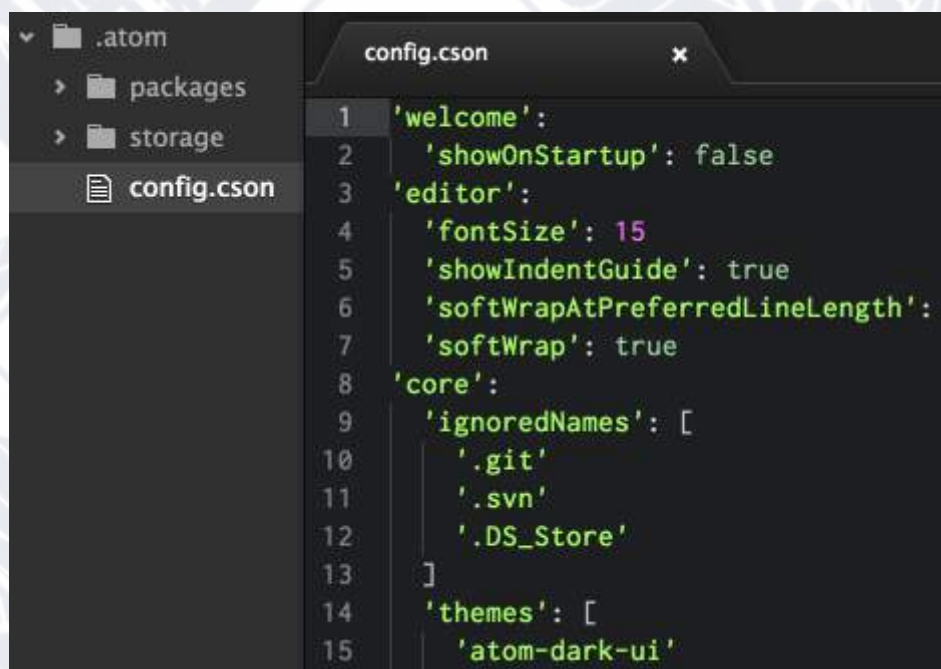


Рисунок 2.5 - Приклад написання коду використовуючи середовище Atom.

Atom має широкий вибір плагінів, що дозволяють розширити його функціональність та адаптувати під конкретні потреби розробника. Цей редактор підтримує роботу з багатьма мовами програмування, включаючи HTML, CSS, JavaScript, Python, PHP, Ruby та інші, що робить його універсальним інструментом для розробки різноманітних проектів. Atom також володіє рядом корисних функцій, таких як автоматичне доповнення коду, підтримка Git для керування версіями, можливість роботи з різними вкладками і проектами одночасно. Завдяки своїй відкритості і активній

спільноті розробників, Atom постійно розвивається та вдосконалюється, надаючи користувачам нові можливості та інструменти для зручної роботи [10].

2.2 Технології розробки вебсайтів

JavaScript є основною мовою програмування для веб-розробки, але через його недоліки з'явилися популярні фреймворки, такі як React.js, Angular.js та Vue.js [32].

Фреймворки JavaScript - це інструменти, які використовуються для створення сучасних веб-додатків. Раніше, коли інтерфейс застосунка був простішим, багато розробників легко писали код на чистому JavaScript. Проте з часом інтерфейси та їх логіка стали складнішими, що зробило використання чистого JavaScript більш вимогливим. Це призвело до появи фреймворків, які стали важливим інструментом для розробки сучасних веб-додатків.

JavaScript фреймворки надають розробникам значні переваги, що сприяють створенню якісного, продуктивного та ефективного коду. Ось основні переваги використання JavaScript фреймворків:

1. Сучасні фреймворки дозволяють використовувати систему навігації без перезавантаження сторінок. Це дає змогу створювати багатосторінкові сайти з усіма перевагами одно-сторінкового сайту, відомого як SPA (Single Page Application).



```

app.get((req, res) => {
  const id = req.params.id // отримання параметрів

  if (!id) {
    res.status(400).send('Id is require') // валідація
  }

  const queryMsg = `SELECT * FROM users WHERE id = ${id}` // створення SQL запиту
  const [ rows ] = db.query(queryMsg, [ id ]) // запитання до бази даних

  if (!rows || !rows.length) res.status(404).send('User not found') // повернення даних з бази
  res.status(200).send(rows) // відправлення успішного результату
})
  
```

Рисунок 2.6 – Приклад маршрутизації на фреймворках

2. Веб-компоненти. Компонентний підхід до розробки широко використовується у багатьох мовах програмування, включаючи JavaScript.

Замість повторного написання коду для кожної сторінки, наприклад, коду шапки сайту, можна створити компонент цієї шапки і підключити його однією рядком у всіх необхідних файлах. Це дозволяє працювати з будь-якими частинами сайту, які можна повторно використовувати.

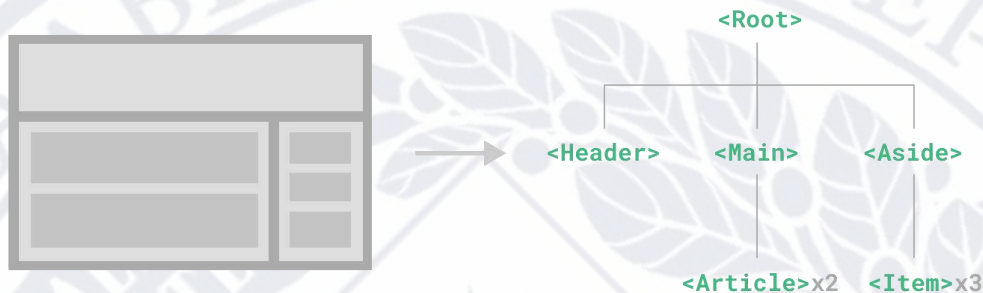


Рисунок 2.7 - Приклад основних компонентів на Vue.js

3. Зручні інструменти для збірки. Компонентний підхід вимагає інструменту для об'єднання всіх файлів в один та компіляції їх у розмітку, доступну для браузера. Основні фреймворки мають власні збірники, які легко встановити і які розширюють певний функціонал.

```

Nuxt.js 0 v2.13.3
  • Environment: development
  • Rendering:   server-side
  • Target:      server
  Listening: http://localhost:3000/

i Preparing project for development
i Initial build may take a while
/ Builder initialized
/ Nuxt files generated

/ Client
  Compiled successfully in 8.17s

/ Server
  Compiled successfully in 5.70s

i Starting type checking service...
i No type errors found
i Version: typescript 3.8.3
i Time: 6211 ms
i Waiting for file changes
i Memory usage: 671 MB (RSS: 783 MB)
i Listening on: http://localhost:3000/
  
```

Рисунок 2.8 - Приклад процесу збірки компонентів

React.js - це інструмент для розробки інтерфейсів користувача.

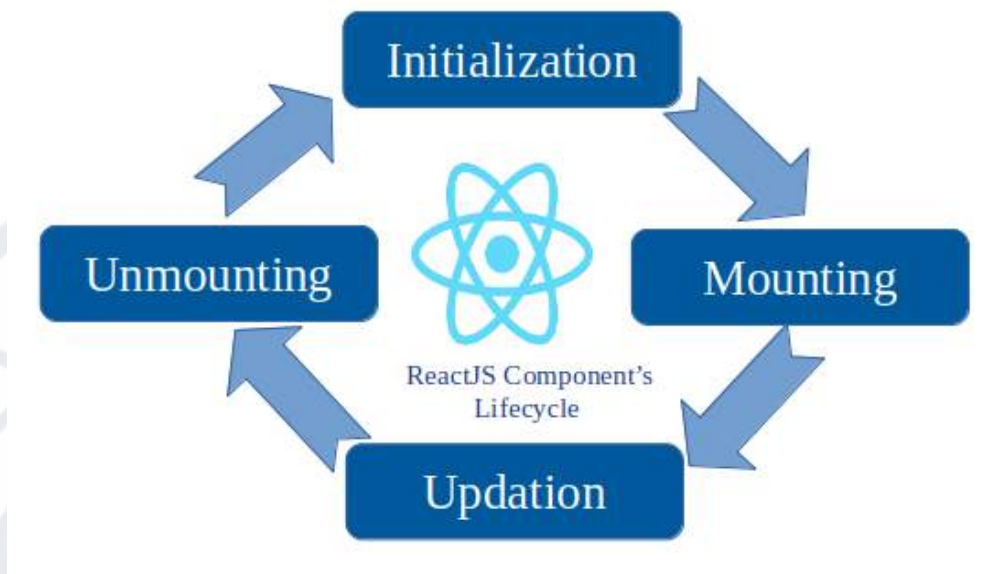


Рисунок 2.9 – Приклад компонентів React.js

Це популярна JavaScript-бібліотека для створення інтерфейсів користувача. Вона дозволяє розробникам будувати швидкі, інтерактивні та модульні веб-додатки. React.js розроблений Facebook і базується на концепції компонентів, які можна багаторазово використовувати та комбінувати для створення інтерфейсу. Використовуючи віртуальний DOM, React оптимізує оновлення інтерфейсу, забезпечуючи високу продуктивність додатків [31].

Переваги React.js.

React.js має декілька основних переваг, які роблять його популярним вибором для розробників веб-додатків:

1. Компонентна архітектура: React дозволяє розробникам створювати інтерфейси користувача з окремих, багаторазових компонентів. Це сприяє кращому управлінню кодом, його підтримці та повторному використанню.
2. Віртуальний DOM: React використовує віртуальний DOM для оптимізації оновлень інтерфейсу. Це забезпечує високу продуктивність, оскільки зміни порівнюються з віртуальним DOM перед оновленням реального DOM, що мінімізує переробку інтерфейсу [9].

3. Односпрямований потік даних: React використовує односпрямований потік даних для кращого контролю за станом додатку та його передбачуваності. Це спрощує розуміння та відладку коду.

4. Велика спільнота та екосистема: React має велику та активну спільноту розробників, а також багату екосистему бібліотек та інструментів, які допомагають розширити функціональність додатку [33].

5. Підтримка універсальних додатків: React може бути використаний як для створення односторінкових додатків (SPA), так і для серверного рендерингу (SSR) за допомогою таких технологій, як Next.js.

6. Висока продуктивність розробки: React забезпечує ефективний процес розробки завдяки можливості швидко збирати та тестувати компоненти.

7. Доступність та SEO: React може бути налаштований для покращення доступності та SEO, особливо в поєднанні з серверним рендерингом.

Недоліки React.js.

React.js має свої недоліки, які можуть створити певні складнощі для розробників:

1. Крива навчання: React.js має досить круту криву навчання, особливо для тих, хто не має досвіду роботи з JavaScript або ES6. Поняття компонентів, стану та життєвого циклу компонентів можуть бути складними для початківців.

2. Швидкий темп розвитку: React постійно оновлюється, з новими версіями та функціями, що можуть бути не сумісними з попередніми версіями. Це може призводити до проблем з підтримкою коду та ускладнювати підтримку проєктів.

3. Екосистема бібліотек: Хоча велика кількість бібліотек є перевагою, це також може бути джерелом плутанини. Вибір правильної бібліотеки для певного завдання може бути складним через надмірну кількість можливостей.

4. Відсутність стандартної архітектури: React не нав'язує певної архітектури додатку, що може призвести до того, що різні команди розробників реалізують функціональність по-різному. Це може ускладнювати підтримку та розуміння коду [34].

5. Великі розміри пакетів: Різні бібліотеки та інструменти в екосистемі React можуть спричиняти збільшення розмірів пакетів, що може вплинути на час завантаження додатків і продуктивність.

6. Питання SEO для SPA: Односторінкові додатки (SPA) можуть зіткнутися з труднощами в оптимізації для пошукових систем, оскільки більшість контенту динамічно створюється на клієнті. Хоча такі рішення, як серверний рендеринг, можуть допомогти, це вимагає додаткових зусиль [17].

7. Потреба у правильній оптимізації: Для досягнення максимальної продуктивності додатку потрібно правильно оптимізувати код. Це включає в себе оптимізацію використання компонентів, стану та інших аспектів.

8. Великі вимоги до пам'яті: Додатки на React можуть споживати багато ресурсів пам'яті, особливо якщо не застосовувати практики оптимізації або використовувати великі та складні компоненти.

Отже, React.js — це потужна та ефективна бібліотека для створення інтерфейсів користувача, яка базується на компонентній архітектурі та використовує віртуальний DOM для оптимізації оновлень. Завдяки великій спільноті розробників та багатій екосистемі інструментів, React.js є популярним вибором для створення різноманітних веб-додатків. Проте розробникам слід враховувати потенційні недоліки, такі як крута крива навчання та швидкий темп розвитку, під час роботи з цією бібліотекою [8].

Vue.js - це прогресивний JavaScript-фреймворк, розроблений для створення інтерфейсів користувача та односторінкових додатків (SPA). Його можна описати як простий у використанні, добре документований та ефективний інструмент для розробки веб-додатків [6].

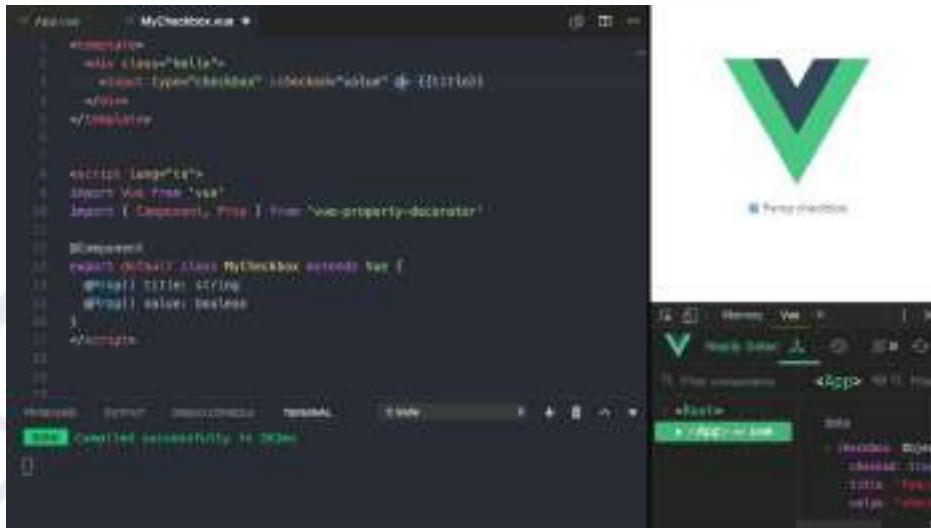


Рисунок 2.10 - Демонстрація роботи з Vue.js.

Переваги Vue.js

Vue.js пропонує кілька переваг, які роблять його популярним вибором серед розробників веб-додатків:

1. Легкість у вивченні та використанні: Vue.js має простий та інтуїтивно зрозумілий синтаксис, що полегшує навчання новачкам у порівнянні з іншими фреймворками, такими як Angular або React.
2. Компонентна архітектура: Vue.js використовує компонентний підхід до розробки інтерфейсів користувача. Кожен компонент має свій власний HTML, CSS і JavaScript, що сприяє організації та повторному використанню коду.
3. Двостороннє зв'язування даних: Vue.js дозволяє легко встановлювати зв'язок між даними та елементами інтерфейсу користувача, що забезпечує автоматичне оновлення компонентів при зміні даних.
4. Реактивність: У Vue.js є вбудована підтримка реактивності даних, що спрощує керування динамічними інтерфейсами.
5. Гнучкість: Vue.js можна використовувати як для невеликих проєктів, так і для великих додатків. Ви можете розширювати його за допомогою плагінів і додаткових бібліотек [7].

6. Масштабованість: Vue.js добре підходить для масштабування проектів за рахунок використання компонентів і модулів.
7. Інтеграція з іншими технологіями: Vue.js легко інтегрується з іншими бібліотеками та фреймворками, такими як Vuex (для керування станом) або Vue Router (для керування маршрутами).
8. Спільнота та документація: Велика та активна спільнота Vue.js забезпечує підтримку та обмін досвідом, а також розробку численних плагінів і бібліотек. Документація Vue.js є однією з найкращих у світі фронтенду.
9. Ефективність: Vue.js оптимізований для швидкої роботи і має низький розмір бібліотеки, що сприяє швидкому завантаженню сторінок.

Недоліки Vue.js

Vue.js, як і будь-яка технологія, має свої недоліки, які можуть впливати на вибір розробників під час створення веб-додатків. Ось деякі з них:

1. Екосистема: Хоча Vue.js має велику спільноту, вона все ще може бути менш розвинутою, ніж у інших популярних фреймворків, таких як React або Angular. Це може впливати на наявність плагінів та інших інструментів.
2. Проблеми масштабування: Для дуже великих проектів, які потребують високого рівня масштабування та управління, Vue.js може не бути таким ефективним, як інші фреймворки. Для таких проектів може знадобитися додаткові інструменти або фреймворки, наприклад, Vuex для управління станом [35].
3. Гнучкість може спричинити проблеми: Хоча гнучкість Vue.js є перевагою, вона також може призвести до меншої структури в проектах, особливо якщо не дотримуватися рекомендацій з організації коду.
4. Менше корпоративних проектів: У порівнянні з Angular або React, Vue.js може бути менш популярним серед великих підприємств, що може призвести до меншої кількості готових рішень і бібліотек для корпоративних проектів.

5. Складність переходу між версіями: Зміни у нових версіях Vue.js можуть бути значними, що може викликати проблеми під час оновлення великих проектів.

6. Суб'єктивний стиль коду: Хоча Vue.js надає багато можливостей для реалізації функціональності, це також може призвести до того, що різні розробники можуть писати код по-різному, що ускладнює підтримку та розробку проекту.

7. Обмежена підтримка TypeScript: Хоча Vue.js підтримує TypeScript, ця підтримка може бути не такою сильною, як у інших фреймворках, таких як Angular, що може бути недоліком для тих, хто вважає TypeScript важливим для своїх проектів [13].

Загалом, Vue.js є чудовим вибором для розробників, які шукають гнучкий та ефективний інструмент для створення сучасних веб-додатків. Однак для великих або корпоративних проектів, можливо, варто розглянути інші фреймворки, які мають ширшу підтримку та більше готових рішень. Вибір фреймворку залежить від специфіки кожного проекту та особистих уподобань розробника.

Angular.js - Це базовий відкритий фреймворк для веб-додатків на основі JavaScript, який спочатку був випущений у 2010 році компанією Google [16].

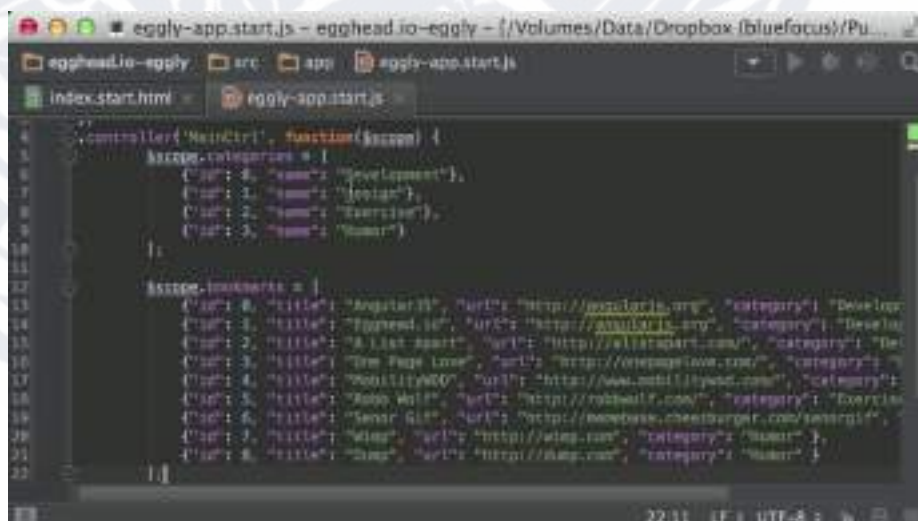


Рисунок 2.11 – Демонстрація роботи Angular.js

Angular.js призначений для спрощення розробки односторінкових додатків (SPAs) та покращення структури та підтримки веб-додатків, забезпечуючи архітектуру MVW (Model-View-Whatever). AngularJS надає різноманітні функції, які полегшують розробку динамічних і чуйних веб-додатків [19].

Переваги Angular.js

AngularJS має ряд переваг, які роблять його популярним вибором серед розробників для створення односторінкових веб-додатків:

1. Двостороннє зв'язування даних: AngularJS пропонує автоматичну синхронізацію між моделлю та представленням. Це означає, що будь-які зміни в даних моделі відображаються у користувацькому інтерфейсі майже миттєво, і навпаки. Це спрощує розробку динамічних та інтерактивних веб-додатків[29].

2. Ін'єкція залежностей: Фреймворк надає можливість управління залежностями між компонентами та службами. Це покращує модульність коду та спрощує його тестування, оскільки залежності можуть бути легко замінені під час тестування.

3. Директиви: AngularJS дозволяє створювати користувацькі HTML-теги та атрибути за допомогою директив. Це дозволяє розробникам розширювати можливості HTML, додаючи поведінку та функціональність елементам DOM.

4. Маршрутизація: Фреймворк забезпечує вбудовану підтримку маршрутизації, що полегшує створення багатосторінкових додатків в межах однієї сторінки, забезпечуючи зручну навігацію між різними представленнями [36].

5. Фільтри: Фільтри надають потужні засоби для форматування даних, що відображаються в інтерфейсі користувача. Вони дозволяють трансформувати та обробляти дані перед відображенням.

6. Шаблони: AngularJS використовує шаблони для створення структурованих і модульних представлень. Це дозволяє розробникам легко розділити код на логічні частини та підвищити читаємість і підтримуваність коду [37].

7. Тестування: AngularJS забезпечує вбудовану підтримку тестування, включаючи можливості для юніт-тестування та інтеграційного тестування. Це дозволяє виявляти та виправляти помилки на ранніх етапах розробки [28].

8. Активна спільнота розробників: Завдяки великій та активній спільноті AngularJS, доступно багато ресурсів, документації, прикладів коду та форумів підтримки, що сприяє швидкому вирішенню проблем та розвитку знань.

Недоліки Angular.js

AngularJS, як і будь-який інший фреймворк, має свої недоліки, які можуть впливати на вибір розробників при створенні веб-додатків:

1. Продуктивність: Двостороннє зв'язування даних та відстеження змін в DOM можуть впливати на продуктивність додатку, особливо при роботі з великою кількістю елементів або складною логікою.

2. Підтримка застарілого коду: AngularJS є застарілим фреймворком після випуску Angular 2 та подальших версій. Розробники, які продовжують використовувати AngularJS, можуть зіткнутися з обмеженою підтримкою та оновленнями.

3. Крива навчання: AngularJS може бути складним для початківців через велику кількість функцій, концепцій та специфічний синтаксис. Це може ускладнити процес навчання та впровадження [38].

4. Проблеми сумісності: З часом сумісність з новими стандартами веб-технологій та іншими інструментами може стати проблемою, оскільки AngularJS більше не оновлюється так активно, як інші сучасні фреймворки.

5. Важкість коду: У великих проєктах AngularJS може призвести до великої кількості коду, який може бути складно підтримувати та оптимізувати.

6. Обмеження в динаміці модулів: Динамічне завантаження модулів може бути складним у AngularJS, особливо у порівнянні з сучасними фреймворками, які забезпечують кращу підтримку модульності.

7. Складність виявлення помилок: Виявлення помилок у великому коді AngularJS може бути складним через розповсюдження коду між компонентами та сервісами.

У підсумку, AngularJS був важливим фреймворком у світі веб-розробки, який пропонував потужні інструменти для створення односторінкових додатків і динамічних інтерфейсів. Однак, через свої недоліки, такі як низька продуктивність та обмежена підтримка застарілої версії, його популярність зменшилась з появою більш сучасних фреймворків. Для нових проєктів розробники часто обирають більш актуальні фреймворки, що забезпечують покращену продуктивність, підтримку та модульність. Незважаючи на це, AngularJS все ще використовується для підтримки старих проєктів, але його використання з часом стає все менш актуальним [27].

HTML 5 — це мова розмітки для створення веб-сторінок, яка є стандартом у веб-розробці. Це основа для створення веб-сайтів. При розробці форуму ми будемо використовувати найкращі практики, включаючи валідність, семантику та доступність HTML-коду [24].



Рисунок 2.12 – Логотип HTML 5

CSS 3 (каскадні таблиці стилів) — це спеціалізована мова, що дозволяє визначати стильові властивості блоків і тексту. Окрім зовнішнього вигляду, вона також дає змогу змінювати розміщення цих елементів на сторінці [14].



Рисунок 2.13 – Логотип CSS 3

JavaScript — це динамічна мова програмування, яка відповідає стандарту ECMAScript. Вона найчастіше використовується для створення різних сценаріїв на веб-сторінках, що дозволяє взаємодіяти з користувачами на стороні клієнта та керувати браузером [15].



Рисунок 2.14 – Логотип JavaScript

Для створення макетів сторінок використовувався редактор Figma, який є векторним онлайн-сервісом для розробки інтерфейсів та прототипування. Він надає можливості для спільної роботи та розробляється однойменною компанією. Figma доступна у форматах браузера та десктопної

програми, дозволяючи зберігати онлайн-версії файлів, з якими працював користувач [26]. У цій програмі можна розробляти дизайн сторінок та легко переносити стилі шрифтів або блоків на реальну сторінку сайту за допомогою інструментів програми.

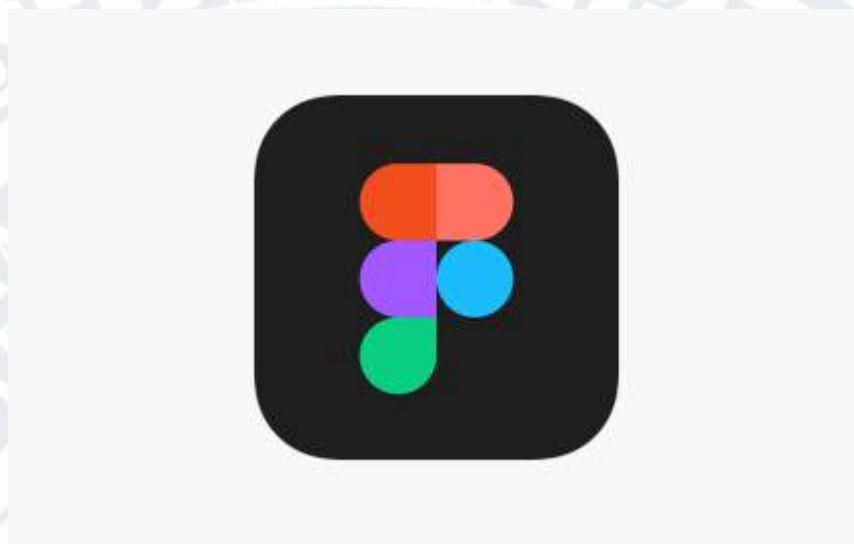


Рисунок 2.15 – Логотип Figma

Крім того, Figma підтримує інтеграцію з різноманітними плагінами, що розширюють її функціонал і дозволяють автоматизувати деякі рутинні завдання. Завдяки своїй гнучкості та широким можливостям для налаштування, цей редактор став незамінним інструментом для багатьох дизайнерів по всьому світу.

Отже, для розробки вебсайту мною було використано сучасні інструменти та технології, які забезпечать ефективне та якісне створення проекту. Перш за все, основною платформою для розробки є Visual Studio Code, яка пропонує зручний редактор коду з безліччю плагінів для розширення функціональності та підвищення продуктивності роботи.

Основний технологічний стек включатиме:

- React.js - попередній JavaScript-фреймворк для створення інтерфейсів користувача. Він забезпечує швидкий і зручний процес розробки за рахунок повторного використання компонентів та ефективного керування станом.

- JavaScript - основна мова програмування, яка забезпечує динамічність та взаємодію користувача з веб-сайтом [39].
- HTML5 та CSS3 - стандартні технології для створення структурованого контенту та стильового оформлення веб-сторінок.
- Figma - інструмент для створення макетів та дизайну веб-сайту. Забезпечує візуальне моделювання інтерфейсу, що допомагає при створенні узгодженого та привабливого дизайну [23].
- MySQL Workbench – інструмент для написання бази даних.

Ці технології є основою для створення сучасного та інноваційного веб-сайту, який буде відповідати вимогам користувачів та забезпечить високу продуктивність і зручність у використанні.

2.3. Проектування бази даних

Розробка бази даних є ключовим етапом у створенні будь-якого веб-додатку, оскільки належне структурування даних гарантує ефективне функціонування системи та спрощує процес обслуговування. Основна мета проектування бази даних полягає у створенні працездатної моделі, яка відображає реальні об'єкти та їх взаємозв'язки, зберігаючи цілісність і забезпечуючи швидкий доступ до інформації.

Першим етапом у створенні бази даних є вибір підходящої системи управління базами даних (СУБД). Для даного проекту було обрано MySQL, одну з найпопулярніших та широко використовуваних СУБД завдяки своїй надійності, продуктивності і використанню відкритого коду. MySQL ідеально підходить для вебзастосунків, оскільки забезпечує швидку обробку запитів та можливості масштабованості [22].



Рисунок 2.16 – Логотип MySQL

Наступним кроком є розробка моделі даних, яка включає створення концепційних, логічних та фізичних моделей бази даних. Концепційна модель зосереджена на високорівневих структурах даних і їх взаємозв'язках, використовуючи сутність-зв'язок діаграми. Логічна модель розгортає концептуальну, додавши атрибути до сутностей та визначивши ключові елементи. Фізична модель враховує унікальні характеристики обраної системи управління базами даних та визначає фактичне масштабування даних.

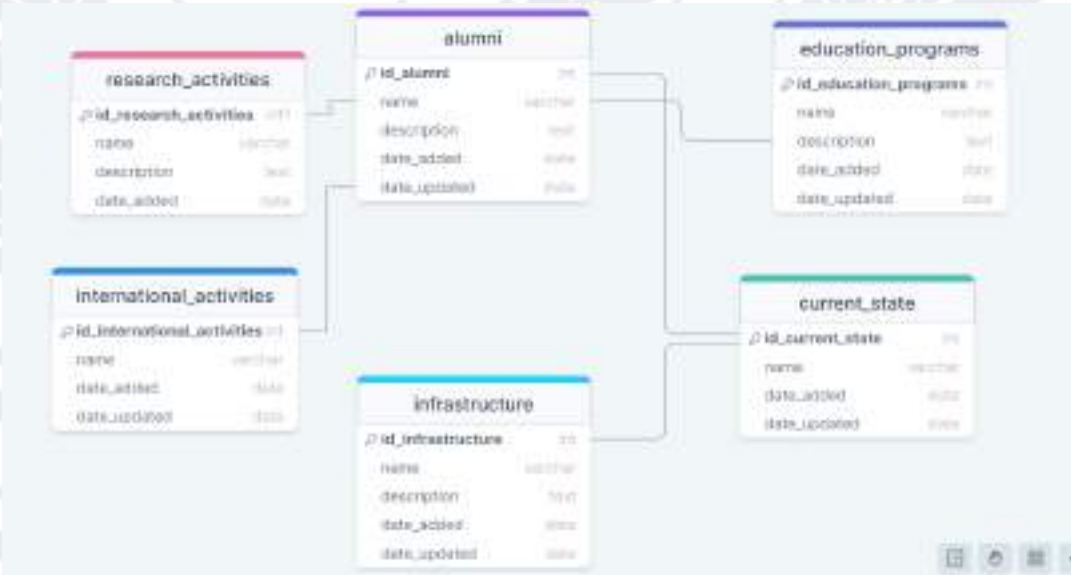


Рисунок 2.17 – Концепційна модель бази даних

Висновки до 2 розділу

У цьому розділі були розглянуті ключові інструменти та технології, що використовуються для розробки вебсайтів. Ми проаналізували сучасні засоби для створення сайтів, включаючи різноманітні редактори коду та інструменти для дизайну, які сприяють ефективній розробці.

Також ми розглянули передові технології для створення веб-сайтів, такі як HTML5, CSS3 та JavaScript, які є основою для побудови сучасних веб-додатків. Крім того, були розглянуті технології розробки, включаючи React.js, Angular.js, Vue.js. та використання Figma для візуального проектування. Окрему увагу було приділено проектуванню бази даних за допомогою MySQL, що забезпечує надійне зберігання та ефективне управління даними вебдодатків.

Завдяки використанню цих інструментів і технологій, можна створити вебсайт, який буде не лише функціональним та ефективним, але й сучасним та привабливим для користувачів, що в свою чергу забезпечить високу якість проекту та сприятиме успіху в його реалізації.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ САЙТУ «ІСТОРІЯ ЗАКЛАДУ ОСВІТИ»

3.1. Дизайн сайту «Історія закладу освіти»

Однією з первинних умов нашого технологічного стеку було створити сучасний та красивий дизайн сайту, щоб користувачам було зручно та приємно ним користуватись. Дизайн було виконано за допомогою Figma.

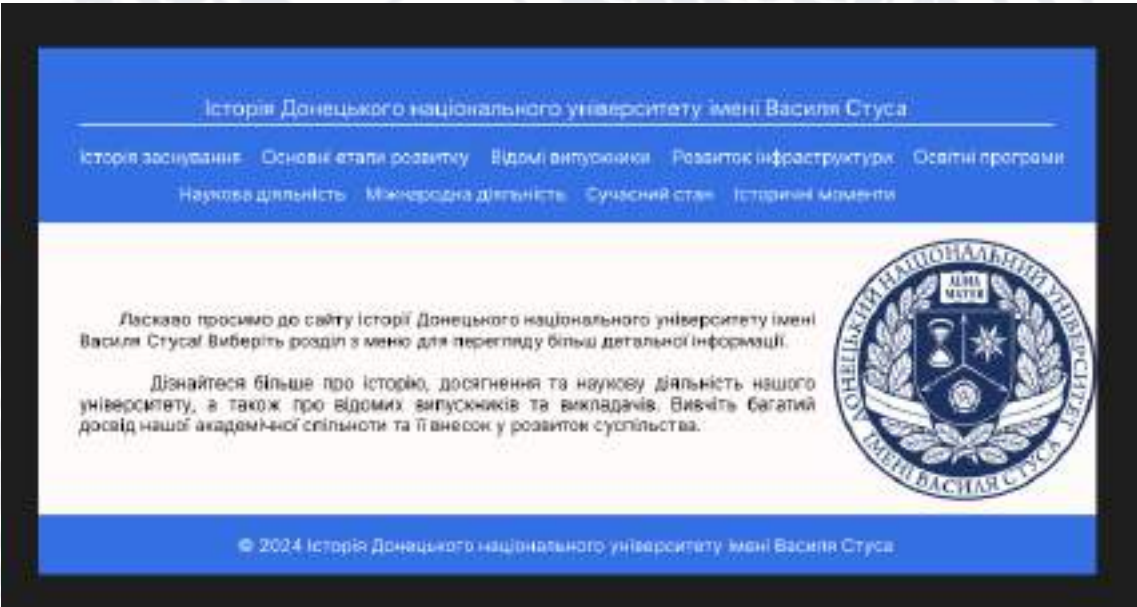


Рисунок 3.1 – Дизайну головної сторінки

Для створення зручного користувацького досвіду на сайті була обрана кольорова палітра, яка встановлює єдиний стиль для всіх сторінок[30]. Крім цього, обрана палітра кольорів сприяє створенню приємної візуальної атмосфери, яка підвищує залученість користувачів і полегшує навігацію.

■	FFFBFB	100%
■	3571E5	100%
■	FFFFFF	100%

Рисунок 3.2 – Кольорова палітра сайту

У сучасному цифровому світі мобільні пристрої відіграють вирішальну роль у доступі користувачів до Інтернету [25]. З кожним роком кількість людей, які переглядають вебсайти зі смартфонів та планшетів, зростає, що підкреслює важливість оптимізації сайтів під ці пристрої. Адаптація вебсайтів для мобільних пристроїв забезпечує зручність використання, швидку навігацію та швидке завантаження контенту, що значно підвищує задоволеність користувачів і збільшує їхню ймовірність залишитися на сайті [40]. Крім того, адаптовані сайти покращують рейтинг у пошукових системах і сприяють зростанню бізнесу, створюючи позитивне враження на мобільних користувачів [11].

Наш сайт не є винятком, він повинен бути функціональним як на великих моніторах, так і на маленьких ноутбуках, телефонах та планшетах. Тому був розроблений мобільний дизайн, що гнучко підлаштовується під будь-який тип пристрою. Платформа повинна бути повністю адаптивною та підтримувати роботу у різних браузерах.



Рисунок 3.3 – Мобільна адаптація головної сторінки сайту

При використанні сайту з мобільного пристрою додана кнопка, що приховує навігаційний блок сайту. При її натисненні відкривається блок з усіма сторінками, на які можна перейти, що в свою чергу, робить користування сайтом більш зручним та виваженим [12].



Рисунок 3.4 – Приклад роботи навігаційного блоку при мобільній адаптації головної сторінки сайту

Підсумовуючи дизайн сайту, варто підкреслити, що він відображає сучасні принципи мінімалізму, які проявляються в простому та чистому вигляді, що забезпечує користувачам інтуїтивно зрозумілий інтерфейс. Цей підхід підкреслює важливість виділення головних елементів на сторінці, зменшуючи візуальний шум та запобігаючи перевантаженню користувачів зайвими деталями.

Дизайн доповнений гармонійною кольоровою палітрою, яка не лише створює приємний візуальний досвід, але й допомагає відвідувачам легко сприймати інформацію та орієнтуватися на сайті. Збалансоване використання кольорів також сприяє підвищенню читабельності та акцентує увагу на важливих елементах сайту.

Загалом, дизайн сайту поєднує в собі естетику, функціональність та зручність, забезпечуючи користувачам позитивний досвід перегляду і взаємодії з сайтом. Такий підхід відповідає сучасним тенденціям у веб-дизайні та сприяє підвищенню залученості користувачів.

3.2 Програмна реалізація

Сайт використовує сучасні технології веброзробки, включаючи HTML, CSS, JavaScript, React.js для побудови інтерактивного інтерфейсу користувача та базу даних MySQL для збереження даних та забезпечення функціональності бекенду.

HTML використовується для створення базової структури вебсторінок, включаючи розмітку заголовків, абзаців, посилань та інших елементів, що забезпечують основний каркас сайту [20].

```
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Історія АДМН</title>  
  <link rel="stylesheet" href="styles.css">  
</head>  
  
<body>  
  <header>  
    <div>Історія Донського національного університету Ім'я Баски Стуса</div>  
    <div>  
      <!-- burger menu -->  
      <button class="burger-menu">☰</button>  
  
      <div class="nav-menu">  
        <ul>  
          <li><a href="foundation.html">Фундація</a></li>  
          <li><a href="development.html">Розвиток плану розвитку</a></li>  
          <li><a href="people.html">Люди</a></li>  
          <li><a href="infrastructure.html">Розвиток інфраструктури</a></li>  
          <li><a href="programs.html">Освітні програми</a></li>  
          <li><a href="research.html">Наукова діяльність</a></li>  
          <li><a href="international.html">Міжнародна співпраця</a></li>  
          <li><a href="current.html">Сучасний стан</a></li>  
          <li><a href="accents.html">Історичні акценти</a></li>  
        </ul>  
      </div>  
    </div>  
  </header>
```

Рисунок 3.5 – HTML код головної сторінки сайту

CSS відповідає за стилізацію сайту, визначаючи зовнішній вигляд елементів, таких як кольори, шрифти, розміри та розташування. Завдяки CSS сайт має єдиний і привабливий дизайн, що відповідає бренду університету.

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #eafaf4;
  line-height: 1.6;
}

header {
  background-color: #4a86e8;
  padding: 20px;
  color: white;
  text-align: center;
}

main {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 100px;
  background-image: url('images/Donetsk_National_University.png');
  background-repeat: no-repeat;

  --background-pos-x: 95%;
  --background-pos-y: 10%;

  background-position: var(--background-pos-x) var(--background-pos-y);
  background-size: 450px;
}

/* Count down menu */
main p {
  max-width: 60%;
}
```

Рисунок 3.6 – CSS код головної сторінки сайту

Для забезпечення зручного використання сайту на мобільних пристроях, у файлі CSS застосовано медіа-запити, які змінюють стилі при ширині екрану менше 768 пікселів. Завдяки цим змінам, головний контент переходить у стовпчик, вирівнюється по центру, і фонове зображення вимикається для більшої читабельності тексту. Також розміри шрифту та відступи підлаштовуються під мобільні пристрої, що забезпечує кращу зручність користування.

На мобільних пристроях бургер-меню стає доступним, дозволяючи користувачам легко взаємодіяти з меню, яке розміщується у вертикальний стовпчик. Крім того, шрифти у footer зменшуються для більш компактного

вигляду. Ці зміни забезпечують оптимальний досвід користування сайтом на мобільних пристроях.

```
@media (max-width: 768px) {  
  
  main {  
    flex-direction: column;  
    align-items: center;  
    padding: 20px;  
    background-image: none;  
  }  
  
  main p {  
    max-width: 100%;  
    text-indent: 25px;  
    margin-bottom: 40px;  
  }  
  
  main img {  
    width: 100%;  
    margin-top: 20px;  
  }  
  
  .burger-menu {  
    display: block;  
  }  
  
  nav ul {  
    flex-direction: column;  
    align-items: center;  
    display: none;  
  }  
}
```

Рисунок 3.7 – Код для мобільної адаптації сайту

JavaScript використовується для додавання динамічності та взаємодії з користувачем на сайті. Скрипти JavaScript також застосовуються для керування бургер-меню, що полегшує навігацію по сайту, особливо на мобільних пристроях. Це меню дозволяє користувачам швидко отримати доступ до розділів сайту та перейти між сторінками. Взаємодія з користувачем за допомогою JavaScript також включає відстеження поведінки користувачів, що може бути корисним для покращення функціональності сайту та більш точного пристосування до потреб аудиторії [18].

```

<script>
  document.addEventListener('DOMContentLoaded', function() {

    const burgerMenu = document.querySelector('.burger-menu');
    const navMenu = document.querySelector('.nav-menu');

    burgerMenu.addEventListener('click', function() {
      navMenu.classList.toggle('open');
    });
  });
</script>

```

Рисунок 3.8 – Реалізація бургер-меню з допомогою JavaScript

React.js також допомагає в оптимізації взаємодії з користувачем, завдяки використанню подій для обробки дій на сторінці, таких як натискання кнопок або вибір пунктів меню. Компоненти React дозволяють розбивати інтерфейс на окремі функціональні частини, що полегшує управління складністю та забезпечує підтримуваність коду.

Завдяки React.js, можна ефективно керувати станом додатку та оновлювати інтерфейс користувача відповідно до змін у базі даних MySQL. Це надає користувачам можливість швидко та зручно переглядати необхідну інформацію про історію та розвиток Донецького національного університету імені Василя Стуса

```

import { Link, LinkProps } from 'react-router-dom';
import { ReactNode, memo } from 'react';
import { classNames } from '@shared/lib/classNames/classNames';
import cls from './AppLink.module.scss';

export type AppLinkTheme = 'clear' | 'apple-link';

interface AppLinkProps extends LinkProps {
  className?: string;
  theme?: AppLinkTheme;
  children?: ReactNode;
}

export const AppLink = memo((props: AppLinkProps) => {
  const {
    to, children, className, theme = 'clear', ...otherProps
  } = props;

  return (
    <Link
      to={to}
      className={classNames(cls.AppLink, {}, [className, cls[theme]])}
      {...otherProps}
    >
      {children}
    </Link>
  );
});

```

Рисунок 3.9 – Використання бібліотеки React.js

На завершення, використання сучасних технологій веброзробки забезпечує ефективне функціонування сайту та високий рівень зручності для користувачів. Завдяки оптимізованому дизайну, динамічній взаємодії з користувачами та легкому доступу до інформації, сайт пропонує швидкий та інтуїтивно зрозумілий досвід перегляду контенту про історію та розвиток Донецького національного університету імені Василя Стуса.

Мобільна адаптація відіграє ключову роль у наданні комфортного досвіду користування на різних пристроях. Завдяки цій адаптації, сайт безперешкодно підлаштовується під різні розміри екранів, зберігаючи при цьому візуальну привабливість. Такий підхід сприяє глибшому залученню користувачів та підвищенню їх задоволення.

У результаті поєднання передових технологій та продуманої архітектури, сайт стає надійним та ефективним вебресурсом. Він здатен задовольняти вимоги сучасної аудиторії, пропонуючи високоякісні послуги та гнучкість у користуванні. Це, своєю чергою, забезпечує довгострокову привабливість сайту, який залишається актуальним і конкурентоспроможним в Інтернеті.

3.3 Функціонал сайту

Перше, що користувачі помітять після переходу на сайт — це головна сторінка, яка слугує своєрідною візитною карткою сайту та першим враженням про його зміст. На цій сторінці розміщений короткий опис основних розділів, що надає користувачам загальне уявлення про доступні категорії інформації та функціональні можливості сайту.

Цей короткий опис розділів допомагає користувачам швидко орієнтуватися в змісті сайту та визначати, на який саме розділ з інформацією їм слід перейти для задоволення своїх інтересів або потреб.

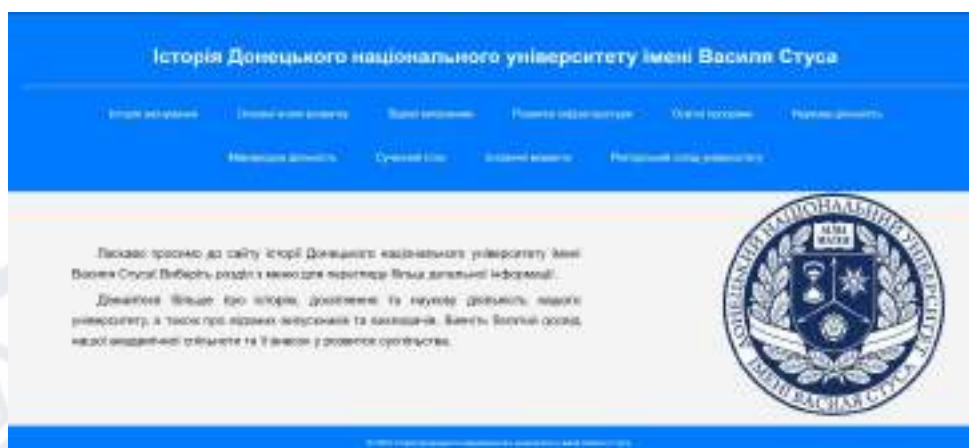


Рисунок 3.10 – Головна сторінка сайту

Навігаційна панель сайту ретельно продумана для забезпечення максимально зручного та оптимально зрозумілого досвіду користування. Користувачі можуть швидко та легко переміщатися між різними розділами сайту завдяки розташуванню кнопок меню та їх логічній організації. Це включає можливість доступу до таких важливих розділів, як:

- **Історія заснування:** Цей розділ розповідає про історичне підґрунтя та передумови створення установи, даючи користувачам змогу дізнатися про основні аспекти її формування та розвитку.
- **Основні етапи розвитку:** Тут висвітлюються ключові аспекти та досягнення у процесі становлення і зростання установи, від перших років існування до сучасності.
- **Відомі випускники:** Цей розділ присвячений випускникам, які досягли успіхів у своїх сферах діяльності та принесли славу своєму закладу освіти.
- **Розвиток інфраструктури:** У цьому розділі можна знайти інформацію про інновації та оновлення інфраструктури, які сприяли покращенню навчального та дослідницького середовища.
- **Освітні програми:** Розділ надає вичерпну інформацію про навчальні програми, спеціалізації та факультети, що пропонуються установою, включаючи деталі щодо кожного курсу або програми.

- **Наукова діяльність:** Цей розділ демонструє наукові досягнення установи, включаючи публікації, проекти та дослідницькі групи, що працюють над розробкою інноваційних рішень.
- **Міжнародна діяльність:** Тут представлено інформацію про міжнародне співробітництво установи, включаючи партнерства з іноземними університетами, участь у міжнародних конференціях та обмінах.
- **Сучасний стан:** Розділ містить актуальні новини та події, що відбуваються в установі, даючи користувачам можливість бути в курсі поточних подій.
- **Історичні моменти:** Цей розділ зосереджений на видатних подіях та досягненнях, які формували історію установи, підкреслюючи її вагомий внесок у розвиток суспільства.
- **Ректорський склад університету:** Даний розділ присвячений ректорам університету, які у різні роки займали свою посаду та покращували якість освіти впроваджуючи всілякі інновації у освітній процес. У розділі висвітлюються не тільки досягнення кожного з ректорів, але й їх внесок у розвиток університетської інфраструктури, наукових досліджень та міжнародної співпраці.

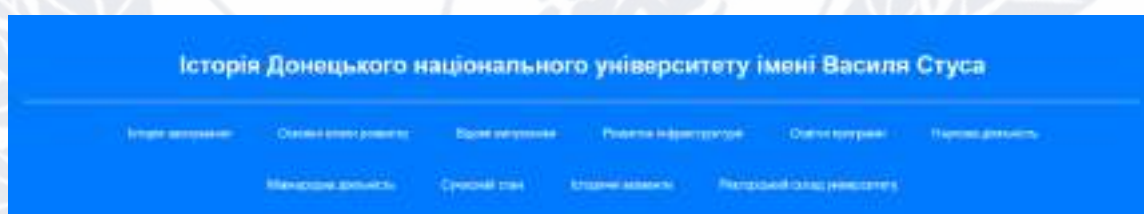


Рисунок 3.11 – Навігаційна панель сайту

На сайті також реалізовано зручну навігацію для мобільних пристроїв, користувачі можуть скористатися кнопкою-бургером для доступу до меню, також враховано адаптацію контенту для різних пристроїв, включаючи шрифти, розташування об'єктів та інші важливі аспекти дизайну.

Використовуються адаптивні шрифти, які коректно відображаються на різних екранах та забезпечують оптимальну читабельність. Такі шрифти гарантують, що текст залишається чітким і буде легко сприйматись

користувачем незалежно від пристрою, що використовується для перегляду сайту. Це дозволяє уникнути проблем, пов'язаних з нечітким текстом або необхідністю масштабування, що може знизити його якість.

Розташування об'єктів на сторінці налаштовано таким чином, щоб воно змінювалося відповідно до розміру екрана пристрою користувача, забезпечуючи максимально комфортний та зручний досвід перегляду.

Це означає, що вміст вебсайту плавно пристосовується до різної ширини екрану, завдяки чому користувачі можуть комфортно переглядати та взаємодіяти з сайтом незалежно від того, чи використовують вони смартфон, планшет або настільний комп'ютер.

Це робить сайт більш доступним та зручним для користувачів незалежно від пристрою, який вони використовують для перегляду. Забезпечення комфортного перегляду та взаємодії з сайтом на різних пристроях сприяє кращому користувацькому досвіду та підвищує загальну задоволеність відвідувачів сайту.



Рисунок 3.12 – Приклад мобільної адаптації сайту

Сайт розроблено з урахуванням потреб різних груп користувачів, таких як студенти, викладачі, випускники та інші зацікавлені особи. Це дозволяє кожному швидко та легко знайти потрібну інформацію. Завдяки інтуїтивно зрозумілій структурі та функціональності, користувачі можуть комфортно взаємодіяти з контентом і скористатися усіма можливостями сайту.

Отже, сайт ефективно поєднує зручну навігацію з адаптивним дизайном, що дозволяє користувачам легко орієнтуватися в розділах і знаходити потрібну інформацію незалежно від пристрою, який вони використовують. Розроблений з урахуванням потреб різних груп користувачів, сайт надає інтуїтивно зрозумілу структуру і функціональність для студентів, викладачів, випускників та інших зацікавлених осіб. Завдяки цьому кожен користувач має можливість швидко та зручно скористатися всіма можливостями сайту та отримати необхідну інформацію.

Висновок до розділу 3

У розділі 3 було детально розглянуто програмну реалізацію сайту «Історія закладу освіти». Цей розділ охоплював широкий спектр аспектів, включаючи дизайн, програмну реалізацію та функціональні можливості сайту.

Дизайн сайту був створений з урахуванням сучасних веб-стандартів, враховуючи зручність користування та естетичні вимоги. У розділі було представлено структуру дизайну, з акцентом на забезпеченні привабливості та функціональності інтерфейсу для відвідувачів. Дизайн передбачав простоту навігації, логічне розташування елементів та гармонійне поєднання візуальних та текстових матеріалів.

Програмна реалізація сайту включала детальний огляд технологій та інструментів, які використовуються для розробки сайту. Особлива увага була приділена вибору відповідної мови програмування, систем керування контентом, баз даних та інших ключових компонентів, необхідних для

ефективного та стабільного функціонування сайту. Описувалися принципи побудови архітектури сайту та оптимізації для швидкого завантаження і безперебійної роботи.

Функціональні можливості сайту були розроблені з урахуванням потреб користувачів та особливостей закладу освіти. У розділі було детально описано різні можливості сайту, включаючи доступ до інформації про історію закладу, а також зручну навігацію по різних розділах, які висвітлюють найважливіші аспекти історії та розвитку установи. Крім того, розглядалися інші можливості сайту, такі як пошукова система та інтерактивні елементи, які підвищують загальний рівень взаємодії з користувачами.

Підсумовуючи, розробка сайту була виконана відповідно до всіх вимог та цілей, встановлених на початку проекту. Розділ продемонстрував, що сайт ефективно реалізував необхідні функціональні можливості, забезпечивши високу якість та зручність використання для користувачів. Завдяки цьому користувачі можуть легко отримати доступ до історії та важливих матеріалів закладу освіти, що сприяє розумінню і сприйняттю його спадщини та досягнень.

ВИСНОВКИ

У результаті виконано поставлені завдання щодо розробки сайту «Історія закладу освіти». Ми дослідили актуальність такої платформи та її конкурентні переваги. Сайт відповідає сучасним вимогам користувачів, зважаючи на зростаючий інтерес до історії освіти та необхідність надання зручного доступу до інформації про навчальні заклади.

Розробка сайту зосереджувалася на наданні якісного контенту про історію закладу освіти, зручного інтерфейсу та надійної програмної реалізації. Особлива увага приділялася адаптивному дизайну, який забезпечує зручність перегляду сайту на мобільних пристроях. Це дозволяє користувачам отримувати доступ до інформації будь-де та будь-коли.

Для розробки клієнтської частини ми застосували сучасні підходи до вебдизайну та дослідили методики проектування архітектури. Було проаналізовано аналогічні сайти, їх переваги та недоліки, щоб забезпечити максимально ефективну розробку нашого проекту.

Головною метою було створення сайту, який забезпечує зручний та функціональний доступ до інформації про історію закладу освіти, і вона була успішно досягнута. Всі функції працюють згідно з описом та принципами UI/UX, а також були проведені тестування для виявлення можливих недоліків.

У процесі розробки були використані такі інструменти та технології: Visual Studio Code, HTML, CSS, JavaScript, фреймворк React.js, Figma та MySQL Workbench.

Загалом, робота показала ефективність вибраних технологій та підходів до розробки сайту, а також підтвердила актуальність та конкурентоспроможність сайту «Історія Донецького національного університету імені Василя Стуса».

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

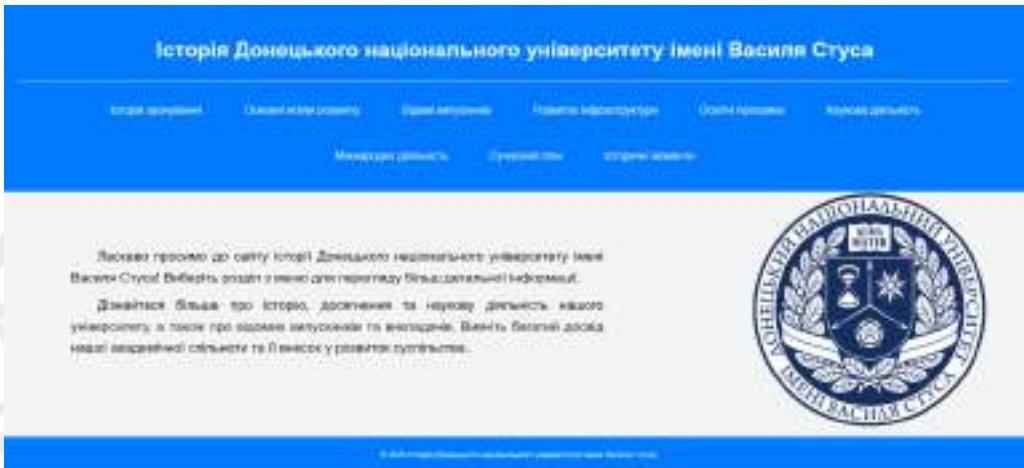
1. Історія школи №51 [Електронний ресурс]: http://school51.edu.kh.ua/pro_shkolu/istoriya/ (Дата звернення 17.05.2024)
2. Історія ліцею №198 [Електронний ресурс]: http://www.lyceum198.edukit.kiev.ua/informaciya_pro_zaklad/istoriya/ (Дата звернення 15.05.2024)
3. Історія Вінницького національного технічного університету [Електронний ресурс]: <https://vntu.edu.ua/uk/about-university/history.html> (Дата звернення 15.05.2024)
4. Інформація про «Visual Studio Code» як інструмент для створення веб-застосунків [Електронний ресурс]: https://uk.wikipedia.org/wiki/Visual_Studio_Code (Дата звернення 18.05.2024)
5. Herron D. Node.js Web Development – 4. Packt Publishing, Birmingham, 2018. 123-145 с.
6. Vue.js Router [Електронний ресурс]: <https://router.vuejs.org/> (Дата звернення 18.05.2024)
7. Chau G. Vue.js 2 Web Development Projects: Learn Vue.js by building 6 web apps. Packt Publishing, Birmingham, 2017. 56-78 с.
8. Resig J. Pro JavaScript Techniques: Second Edition. Apress, New York, 2006. С. 89-112
9. Crockford D. JavaScript: The Good Parts. O'Reilly Media, Sebastopol, 2008. С. 34-58
10. Chinnathambi K. Learning React: A Hands-On Guide to Building Web Applications Using React and Redux. Addison-Wesley, Boston, 2017. 67-69 с.
11. Haverbeke M. Eloquent JavaScript: A Modern Introduction to Programming. No Starch Press, San Francisco, 2018. С.110-113
12. Робсон Е., Фрімен Е. Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages. O'Reilly Media, Sebastopol, 2012. С. 101-102
13. Macrae C., Мейсон П. Vue.js: Up and Running: Building Accessible and Performant Web Apps. O'Reilly Media, Sebastopol, 2018. С. 9-12
14. Verou L. CSS Secrets: Better Solutions to Everyday Web Design Problems. O'Reilly Media, Sebastopol, 2015. С. 23-25
15. Flanagan D. JavaScript: The Definitive Guide: Activate Your Web Pages. O'Reilly Media, Sebastopol, 2020. С. 210-233
16. Wilken J. Angular in Action. Manning Publications, Shelter Island, 2018. С. 345-367
17. Chinnathambi K. Learning React: A Comprehensive Guide to Building Web Applications with React and Redux. Addison-Wesley, Boston, 2018. С. 100-102

18. Crockford D. JavaScript: The Good Parts: The Good Parts. O'Reilly Media, Sebastopol, 2008. 34 с.
19. Schwarzmüller M. Angular 10 Complete Guide: A Comprehensive Guide to Angular 10. Packt Publishing, Birmingham, 2020. 78 с.
20. Дакетт Дж. HTML and CSS: Design and Build Websites. John Wiley & Sons, Indianapolis, 2011. С.123-125
21. Огороднік М. О., Зелінська О.В. Переваги та недоліки реляційних та нереляційних баз даних. Вісник студентських наукових робіт. ДонНУ ім.В. Стуса. Вінниця. 2023. С.68-69
22. Семенюк А. М., Зелінська О.В. Обробка інформації з використанням реляційних баз даних. Вісник студентських наукових робіт. ДонНУ ім.В. Стуса. Вінниця. 2022. С.18-20
23. Зелінська О.В., Колосова К.К. Огляд методів UX-досліджень під час створення ІТ-продуктів. Вісник студентських наукових робіт. ДонНУ ім.В. Стуса. Вінниця. 2023. С. 56-57
24. Зелінська О.В., Нестеренко В. Ю. Web-додаток для управління особистими фінансами. Вісник студентських наукових робіт. ДонНУ ім.В. Стуса. Вінниця. 2022. С. 41-43
25. Зелінська О.В., Якубич К.О. Розробка веб-додатку моніторингу закладів освіти місцевого рівня. Вісник студентських наукових робіт. ДонНУ ім.В. Стуса. Вінниця. 2023. С.12-13
26. Федоренко Є. О., Зелінська О. В. Інструменти розробки онлайн-сервісів. Вісник студентських наукових робіт. ДонНУ ім.В. Стуса. Вінниця. 2022. С.11-12
27. Duckett J. JavaScript and JQuery: Interactive Front-End Web Development. John Wiley & Sons, Indianapolis, 2014. С. 120-124
28. Simpson K. You Don't Know JS: Scope & Closures. O'Reilly Media, Sebastopol, 2014. С. 45-46
29. Zakas N. C. Mastering Modular JavaScript. O'Reilly Media, Sebastopol, 2018. 101 с.
30. Robbins J. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media, Sebastopol, 2018. С. 98-103
31. Stefanov S. JavaScript Patterns: Build Better Applications with Coding and Design Patterns. O'Reilly Media, Sebastopol, 2010. С. 90-101
32. Djirdeh H., Murray N., Lerner A. Fullstack Vue: The Complete Guide to Vue.js. Fullstack.io, San Francisco, 2018. С. 120-124
33. Grigorik I. High Performance Browser Networking. O'Reilly Media, Sebastopol, 2013. 89 с.
34. Zakas N. C. Professional JavaScript for Web Developers. Wrox, Indianapolis, 2012. 45 с.

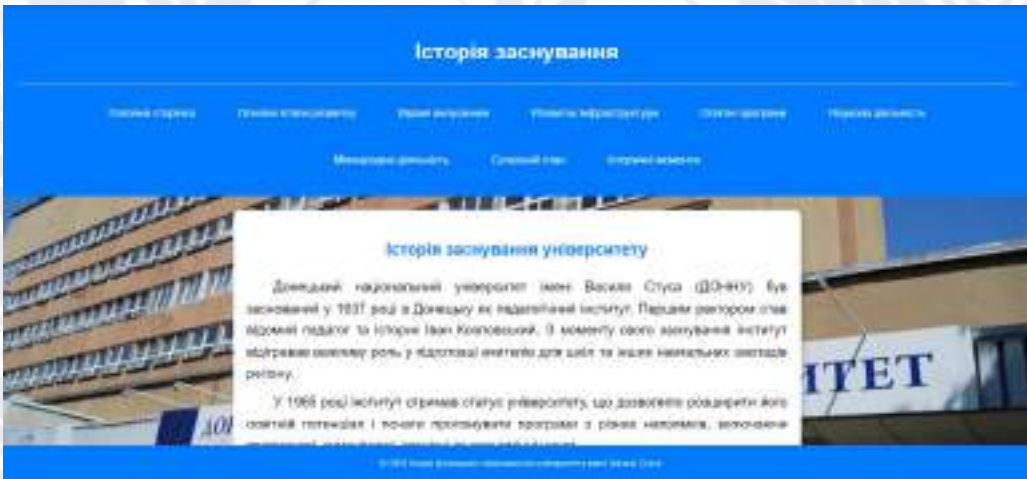
35. Frain B. Responsive Web Design with HTML5 and CSS. Packt Publishing, Birmingham, 2015. C. 103-104
36. Ullman L. Modern JavaScript: Develop and Design. Peachpit Press, San Francisco, 2012. C. 55-67
37. Crowder T. J. JavaScript: The New Toys. Apress, New York, 2020. C. 139-148
38. Resig J., Bibeault B. Secrets of the JavaScript Ninja. Manning Publications, Shelter Island, 2016. C. 99-112
39. Scott A. D. JavaScript Everywhere: Building Cross-Platform Applications with GraphQL, React, React Native, and Electron. O'Reilly Media, Sebastopol, 2020. C. 91-103
40. Aquino C., Gandee T. Front-End Web Development: The Big Nerd Ranch Guide. Big Nerd Ranch Guides, Atlanta, 2016. C.143-145

ДОДАТКИ

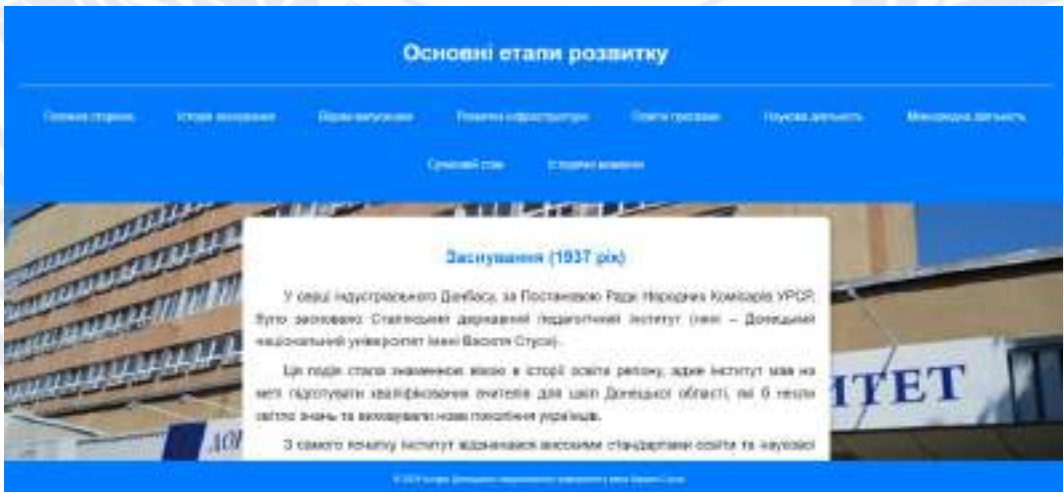
ДОДАТОК А



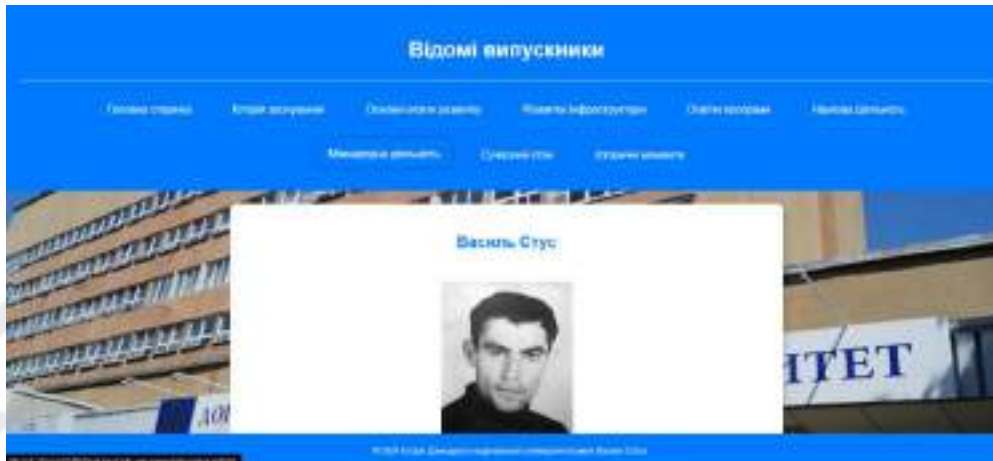
Головна сторінка сайту



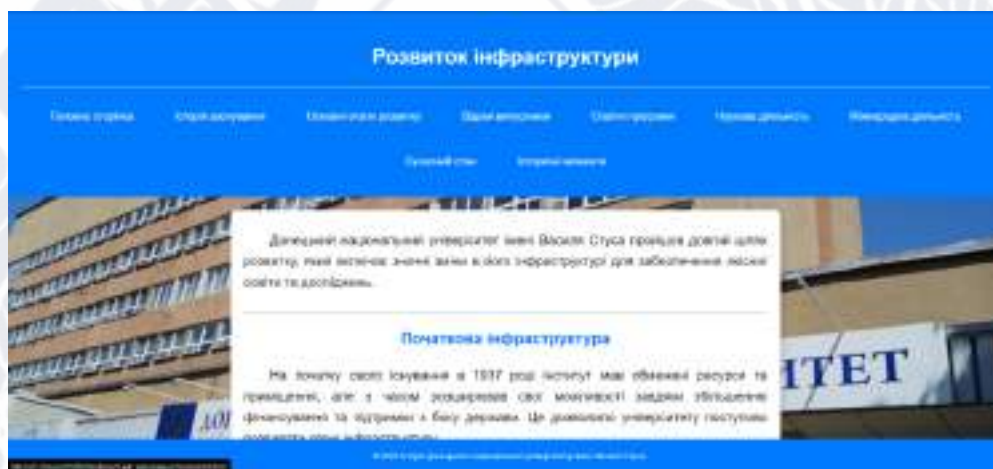
Розділ сайту на тему «Історія заснування»



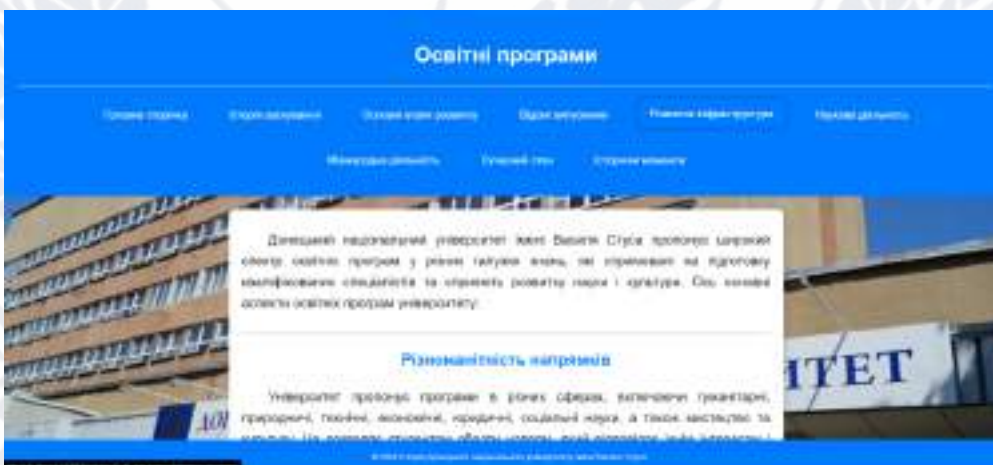
Розділ сайту на тему «Основні етапи розвитку»



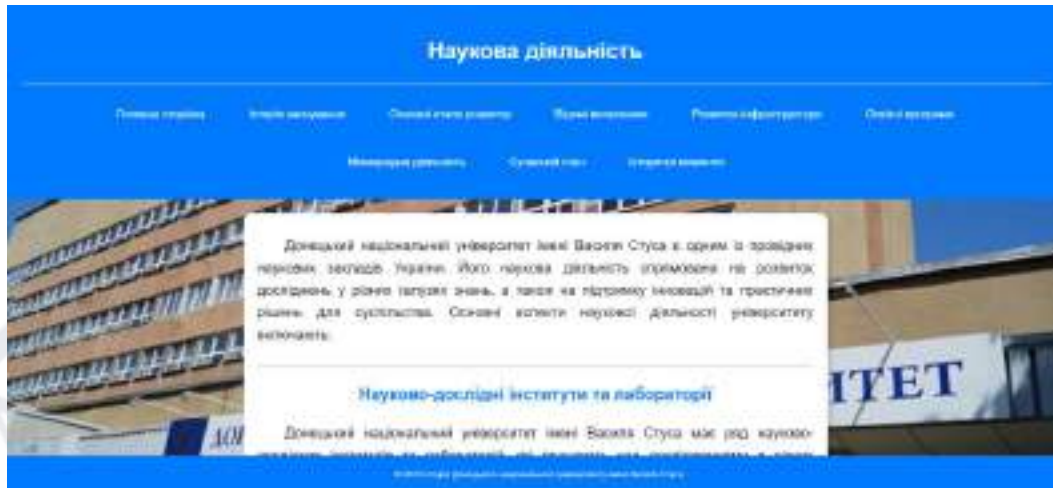
Розділ сайту на тему «Відомі випусники»



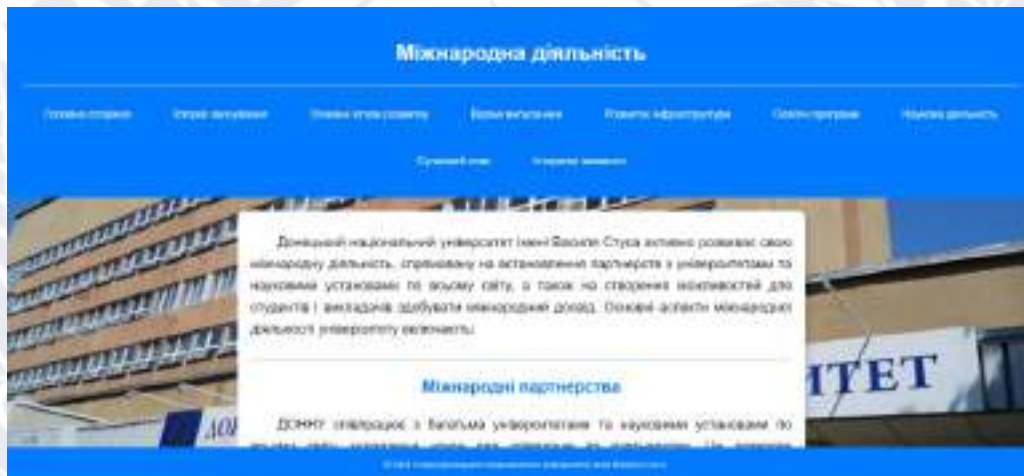
Розділ сайту на тему «Розвиток інфраструктури»



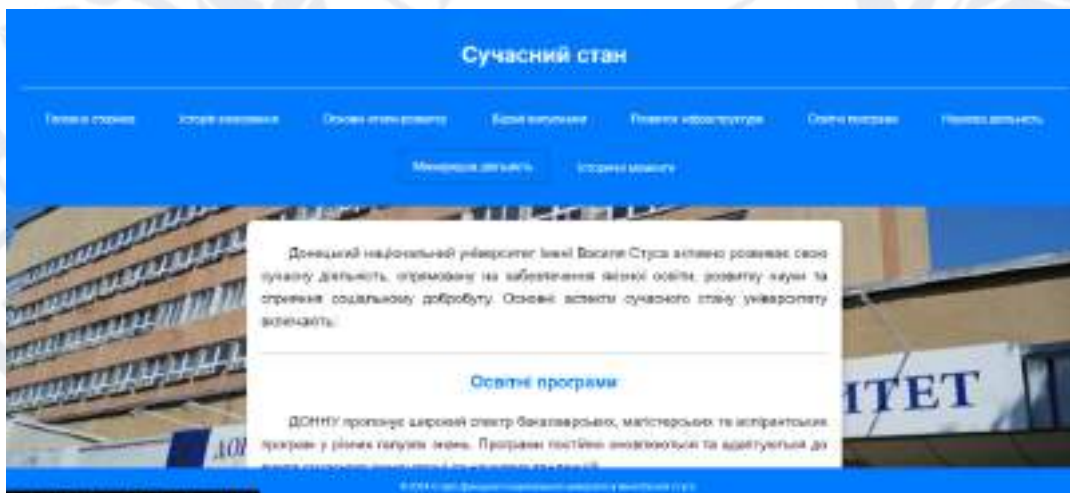
Розділ сайту на тему «Освітні програми»



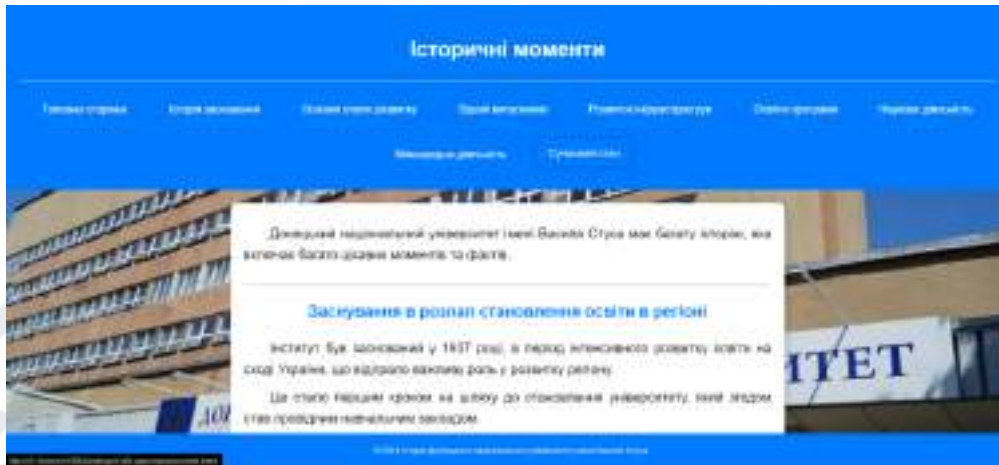
Розділ сайту на тему «Наукова діяльність»



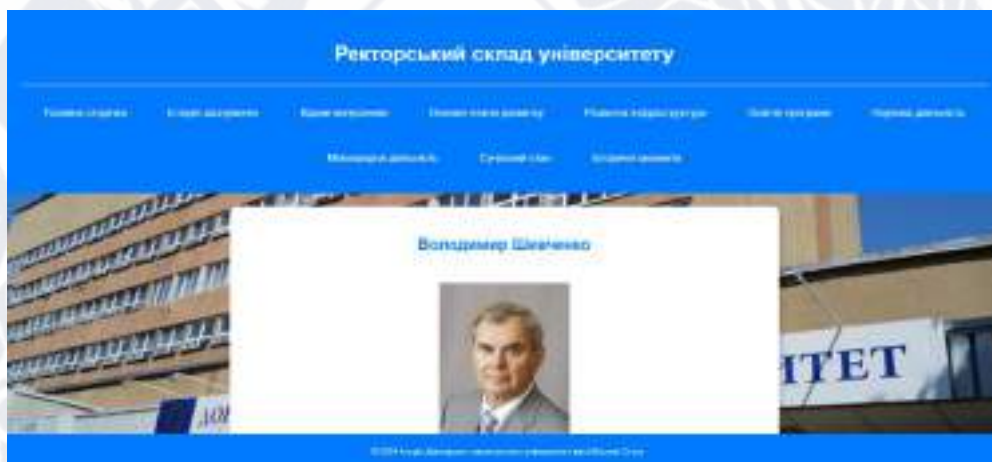
Розділ сайту на тему «Міжнародна діяльність»



Розділ сайту на тему «Сучасний стан»



Розділ сайту на тему «Історичні моменти»



Розділ сайту на тему «Ректорський склад університету»