

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

МОТРУК (МІНТЕНКО) Галина Юріївна

**Графічний інтерфейс веб - сайту студентського гуртка
"DevOps технології" / Graphic interface of the "DevOps
Technologies" student group website**

спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-41
МОТРУК (МІНТЕНКО) Галина Юріївна

Науковий керівник
к.т.н. Піцун О.Й.

Кваліфікаційну роботу допущено
до захисту:

"__" _____ 20__ р.

Завідувач кафедри
_____ О.Л. Дубчак

ТЕРНОПІЛЬ - 2024

ЗМІСТ

Зміст.....	8
Вступ.....	9
1 Аналіз веб-сайтів студентських наукових груп.....	11
1.1 Аналіз веб-технологій розробки сайтів.....	11
1.2 Аналіз сайтів студентських гуртків.....	18
1.3 Порівняльний аналіз веб-сайтів.....	22
1.4 Постановка завдань.....	24
2 Алгоритми роботи веб-сайту студентського гуртка.....	25
2.1 Діаграма функціональних обов'язків користувачів.....	25
2.2 Алгоритми формування блоків веб сторінки.....	28
2.3 Структура та вміст веб-сайту наукового гуртка.....	32
3 Програмна реалізація вебсайту студентського гуртка.....	35
3.1 Серверна структура веб-сайту студентського гуртка.....	35
3.2 Графічний інтерфейс головної сторінки сайту.....	36
3.3 Тестування роботи сторінок сайту.....	44
Висновки.....	49
Список використаних джерел.....	50
Додаток А Техніко-економічне обґрунтування розробки проєкту.....	54
Додаток Б Світлокопія публікації.....	77

					КР.КІ. 07125/20. 00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка графічного інтерфейсу веб-сайту студентського гуртка «Технології DevOps»	Літ.	Арк.	Акрушів
Розроб.	Мотрук Г.Ю.					н	8	79
Перевір.	Піцун О.Й.					ЗУНУ.ФКІТ.КІ-41		
Консуль.	Савка Н.Я							
Н. Контр.	Дубчак Л.О.							
Затверд.	Дубчак Л.О.							

ВСТУП

Актуальність дослідження. Роль веб-сайтів студентських наукових гуртків у формуванні академічного середовища та розвитку наукової спільноти у сучасному світі виявляється ще більш значущою, враховуючи швидкі зміни в освітній та науковій сферах. Сучасні студенти, прагнуть не тільки отримувати традиційну освіту, але й активно залучатися до досліджень та наукових проєктів, віддають перевагу веб-сайтам студентських наукових гуртків. Ці платформи стають центром, де об'єднуються інтереси, амбіції та творчий потенціал студентів та науковців.

Важливо визначити роль веб-сайтів у наданні доступу до актуальних досліджень та проєктів, які здійснюються в університетських наукових гуртках. Це дозволяє студентам не лише отримувати теоретичні знання, але й бути в курсі сучасних наукових тенденцій та відкриттів. Вони можуть вивчати реальні проєкти, робити високоякісні літературні огляди та співпрацювати над власними дослідженнями.

Веб-сайти студентських наукових гуртків стають ефективною платформою для спілкування та співпраці. Ці платформи не лише дозволяють обмінюватися ідеями та дослідницькими результатами, але й розширюють можливості для взаємодії між студентами різних спеціальностей. Важливим аспектом є розвиток навичок створення та управління власним веб-сайтом для студентського наукового гуртка. Створення такого веб-сайту вимагає технічних знань, дизайнерських навичок та стратегічного підходу до вивчення потреб користувачів. Цей процес сам по собі стає частиною навчання студентів, допомагаючи їм розвивати навички в інформаційних технологіях та креативному дизайні.

Об'єкт дослідження – засоби проектування UX/UI дизайну для веб-сайту студентського гуртка.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Предмет дослідження – шаблон веб-сайту студентського наукового гуртка.

Мета розробки веб-сайту полягає в створенні інтерактивної платформи для студентів, яка б забезпечувала їм можливість реєстрації на курси та отримання доступу до актуальних новин у галузі DevOps програмування. Для досягнення мети такі завдання:

- проаналізувати існуючі підходи до розробки веб-сайтів студентських наукових груп;
- проаналізувати сучасні засоби розробки веб-сайтів;
- розробити діаграму функціональних обов'язків користувачів;
- розробити алгоритми аналізу маршрутних таблиць;
- розробити алгоритм формування блоків веб сторінки;
- розробити графічний інтерфейс головної сторінки сайту;
- здійснити тестування програмного засобу.

За результатами роботи опубліковано тези доповіді на ІХ науково-практичній конференції «Інтелектуальні комп'ютерні системи та мережі» [1]. Копії публікації наведено у додатку В.

У першому розділі було проаналізовано сучасні підходи до розробки веб-сайтів студентських наукових гуртків.

У другому розділі було розроблено алгоритми формування блоків на сторінці студентського гуртка.

У третьому розділі реалізовано веб-сайт студентського наукового гуртка.

Отже, веб-сайти студентських наукових гуртків продовжують відігравати важливу роль у формуванні наукового простору та академічної спільноти, забезпечуючи платформу для обміну знаннями, співпраці та інновацій. Їхня актуальність та важливість зростає в контексті сучасних викликів у галузі освіти та досліджень, роблячи їх невід'ємною частиною академічного процесу.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ВЕБ-САЙТІВ СТУДЕНТСЬКИХ НАУКОВИХ ГРУП

1.1 Аналіз веб-технологій розробки сайтів

В сучасному цифровому світі веб-сайти стали необхідною та невід'ємною частиною інтернет-простору, що знаходить застосування в різних галузях діяльності. Сучасний інтернет став невід'ємною частиною життя і змінив спосіб спілкування, навчання та проведення досліджень. Веб-сайти стали платформою для обміну інформацією, аналізу наукових результатів та співпраці у галузі науки та досліджень. Однак для розуміння та вдосконалення веб-сайтів необхідно звернути увагу на веб-технології, що лежать в основі їхньої розробки.

Великий крок у сучасній інформаційній епісі досягнуто завдяки створенню веб-сайтів, які стали важливим інструментом для передачі інформації та взаємодії з користувачами через інтернет.

В зв'язку з цим, важливо ретельно розглянути веб-технології, які лежать в основі функціонування веб-сайтів. Аналіз цих технологій є ключовим етапом для розуміння принципів роботи веб-сайтів та можливостей їх оптимізації для різноманітних завдань та потреб. Такий аналіз стає особливо актуальним в рамках дипломної роботи, сприяючи розширенню наших знань у галузі веб-розробки та інформаційних технологій.

Історія веб-дизайну веде свій початок зі створення перших веб-сайтів у 1990-х роках. У той час сайти були простими текстовими сторінками з мінімальними зображеннями. Однак з розвитком технологій та доступу до Інтернету, веб-дизайн став більш складним та естетичним. Розвиток CSS (Cascading Style Sheets) дозволив відокремити вміст від представлення, що полегшило створення зовнішнього вигляду сайту [5, 9].

Середина 1990-х років це початок веб-дизайну. Перші веб-сайти були дуже простими та обмеженими з точки зору дизайну. Вони склалися

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

головним чином з тексту, базових зображень і гіперпосилань. Основними мовами були HTML і CSS (Cascading Style Sheets), хоча CSS був ще в розробці.

У середині 1990-х - початку 2000-х з'являється графічний веб-дизайн. Виникли перші програми для графічного редагування, такі як Adobe Photoshop, які дозволили дизайнерам створювати більш складні та креативні веб-елементи [14].

На початку 2000-х зростає популярність Flash. Технологія Adobe Flash дала можливість створювати складні анімації та інтерактивні елементи на веб-сайтах.

Середина 2000-х – сучасність це розквіт веб-стандартів і мобільного дизайну. В 2000-х роках було введено стандарти для веб-розробки, такі як XHTML і CSS2, що сприяло більшій стандартизації та розширенню можливостей для дизайнерів. З появою смартфонів в середині 2000-х, веб-дизайнерам довелося думати про мобільну адаптацію і створювати адаптивні веб-сайти [10].

В останні роки стостерігаємо тренд на мінімалізм та матеріальний дизайн. Сучасні тренди включають в себе мінімалістичний дизайн, велику увагу до типографіки, використання яскравих кольорів і анімації.

Сьогодні і майбутнє це час, коли веб-дизайн продовжує розвиватися разом з технологіями. До сучасних трендів відносять анімацію, використання штучного інтелекту для персоналізації веб-сайтів, мікроінтеракції та роботу зі шрифтами. Майбутнє веб-дизайну обіцяє більше експериментів та використання нових технологій, таких як віртуальна реальність і розширена реальність [25-27].

Кожна сторінка може мати свій власний контент і структуру. Найпоширеніші види сторінок включають домашню сторінку, сторінки з інформацією про продукти або послуги, сторінки з контактами, блог, сторінки "Про нас" і багато інших.

Навігаційне меню, яке зазвичай розташовується вгорі або по боках веб-сторінок, допомагає користувачам переміщатися між різними сторінками

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

сайту. Ефективна навігація сприяє зручному використанню сайту. Зображення, відео та аудіофайли додають візуальну і аудіальну привабливість до сайту. Вони можуть бути використані для представлення продуктів, послуг, ілюстрації концепцій тощо.

Текст є важливим компонентом будь-якого веб-сайту. Він використовується для передачі інформації, пояснень, створення контенту для блогу або новин тощо. Форми дозволяють користувачам взаємодіяти з сайтом. Наприклад, форма зворотнього зв'язку дозволяє користувачам надсилати повідомлення адміністратору сайту. Поля введення використовуються для збору інформації від користувачів, такої як імені, адреси електронної пошти або номеру телефону. JavaScript і інші скриптові мови використовуються для створення інтерактивних функцій, таких як анімація, перевірка форм, слайдшоу, ігри тощо [23].

Бази даних (БД) є критичним елементом сучасних інформаційних систем та веб-додатків, забезпечуючи зручний та ефективний доступ до даних. Давайте розглянемо деякі технічні аспекти баз даних. Реляційні Бази Даних (RDBMS) використовуються для зберігання та управління даними у вигляді табличних зв'язків. Приклади: MySQL, PostgreSQL, Oracle, Microsoft SQL Server [1].

NoSQL Бази Даних спрямовані на роботу з неструктурованими чи півструктурованими даними. Включають MongoDB (документ-орієнтована), Cassandra (колоночно-орієнтована), Redis (ключ-значення), Neo4j (графова) та інші.

Вбудовані Бази Даних розташовані безпосередньо в програмі та використовуються для локального зберігання невеликих обсягів даних, наприклад, SQLite.

SQL (Structured Query Language) в основному використовується для взаємодії з реляційними базами даних. SQL дозволяє виконувати операції вставки, вибору, оновлення та видалення даних [3, 4].

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

NoSQL Query Languages кожна NoSQL база даних має власну мову запитів, наприклад, MongoDB використовує BSON-мову запитів для роботи з документами.

Індекси це структури даних, що підвищують швидкість пошуку та фільтрації. Реляційні БД використовують В-дерева, а NoSQL може використовувати різні види індексів в залежності від свого типу.

Оптимізатор запитів автоматично вибирає оптимальний план виконання запиту для підвищення продуктивності. Реляційні БД використовують оптимізатор на основі статистики, а NoSQL може вимагати ручного оптимізатора для конкретного типу даних [8].

Регулярні Резервні Копії важливі для запобігання втрати даних в разі випадкового видалення чи збою. Реляційні і NoSQL бази даних надають інструменти для автоматизованого резервного копіювання.

Журнали Змін деякі системи зберігають журнали змін для відновлення бази даних до конкретного стану на певний момент.

Аутентифікація та авторизація контролюють доступ користувачів до бази даних. Включають в себе ролі, дозволи та механізми шифрування.

Шифрування даних забезпечує захист конфіденційності даних, особливо під час передачі через мережу.

Горизонтальне та вертикальне масштабування:

Горизонтальне масштабування додає додаткові сервери для розподілення навантаження. Часто використовується в NoSQL базах даних.

Вертикальне масштабування це збільшення потужності сервера для обробки більших обсягів даних. Типово для реляційних баз даних [10, 11].

Технічні аспекти баз даних включають в себе широкий спектр функціональностей, які забезпечують надійність, продуктивність та безпеку обробки даних в різних середовищах.

Безпека і аутентифікація це важливий компонент для захисту сайту від несанкціонованого доступу і злому. Включає в себе методи аутентифікації, шифрування і валідацію даних.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Для відстеження відвідуваності та поведінки користувачів може використовуватися аналітика. Це допомагає адміністраторам сайту зрозуміти, як користувачі взаємодіють з сайтом і як його можна поліпшити.

Веб-сервер є ключовим компонентом архітектури веб-додатків, відповідальним за обробку та відправлення запитів від клієнтів. Давайте розглянемо деякі технічні аспекти веб-серверів.

HTTP (Hypertext Transfer Protocol) основний протокол для обміну даними в World Wide Web. HTTP визначає, як повинні бути сформовані та передані дані між веб-сервером і клієнтом. HTTPS (HTTP Secure) захищена версія HTTP, яка використовує шифрування за допомогою протоколу SSL/TLS для захисту конфіденційності даних [32, 33].

Apache HTTP Server один з найпоширеніших та старих веб-серверів. Підтримує багато модулів та конфігурується через файл конфігурації. Nginx легкий та ефективний веб-сервер, призначений для обробки багатокористувацьких запитів. Часто використовується як проксі-сервер. Microsoft Internet Information Services (IIS) веб-сервер від Microsoft для операційних систем Windows, підтримує ASP.NET та інші технології Microsoft. Apache Tomcat веб-сервер та контейнер сервлетів для платформи Java, використовується для розгортання та виконання Java-веб-додатків.

Apache може бути розширений різними модулями для підтримки віртуальних хостів, безпеки, обробки мов програмування та ін. Nginx підтримує динамічні модулі, які можна підключити для розширення його функціональності, такі як балансування навантаження, кешування та інші [28].

Балансування навантаження:

Hardware Load Balancers це фізичні пристрої, що розподіляють запити між серверами для забезпечення стабільності та масштабованості. Software Load Balancers програмні рішення, такі як Nginx або HAProxy, які забезпечують балансування навантаження на рівні програмного забезпечення.

Reverse Proxy Caching використовується для зберігання копій ресурсів на веб-сервері, щоб зменшити час завантаження та збільшити продуктивність.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Content Delivery Network (CDN) мережа серверів розташованих у різних географічних областях, щоб забезпечити швидкий доступ до контенту для користувачів у різних регіонах [16].

SSL/TLS використовується для шифрування комунікації між клієнтом та сервером, щоб забезпечити конфіденційність та цілісність даних. Web Application Firewalls (WAF) захищають веб-додатки від різних атак, таких як SQL-ін'єкції чи кросс-сайтові атаки. Доступ до файлів та директорій налаштування прав доступу для запобігання несанкціонованому доступу до файлів та директорій [20].

Log Files записують події та запити для аналізу та виявлення проблем. Моніторинг продуктивності використовується для визначення швидкості відповіді, завантаження сервера та інших параметрів продуктивності. Автоматизовані засоби використовуються для автоматичного виявлення та вирішення проблем в реальному часі. Веб-сервери є важливою складовою для будь-якого веб-додатка, і їхні технічні характеристики визначають ефективність та безпеку веб-сайту чи додатку.

Інструменти розробки відіграють важливу роль у створенні програмного забезпечення, включаючи веб-сайти та веб-додатки. Давайте розглянемо деякі технічні аспекти інструментів розробки.

Visual Studio Code (VSCode) безкоштовний та легкий редактор коду від Microsoft, підтримує багато мов програмування, має велику кількість розширень для роботи з різними технологіями. Sublime Text швидкий та легкий редактор з великою кількістю плагінів та можливістю кастомізації. Atom редактор, розроблений GitHub, який є відкритим та розширюваним. PyCharm спеціалізований редактор для роботи з мовами програмування Python [17-19].

Eclipse це популярне IDE для роботи з Java, але підтримує багато інших мов. IntelliJ IDEA надає можливості розширеної підтримки для Java та інших мов програмування. Visual Studio від Microsoft, в основному використовується

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

для розробки програм на .NET технології. Xcode розроблене для роботи з платформами Apple, такими як iOS та macOS.

JavaScript/TypeScript використовується для розробки фронтенду веб-додатків, а також серверної частини за допомогою Node.js. Python мова з простим синтаксисом, використовується для розробки веб-додатків, наукових досліджень та багатьох інших областей. Java застосовується для розробки масштабних корпоративних додатків. C# мова програмування для розробки додатків під платформу .NET.

Git розповсюджений інструмент для керування версіями коду. Використовується для спільної роботи над проектами та відстеження змін. GitHub сервіс, який надає хостинг для проектів, що використовують Git. Забезпечує спільну роботу, відгалуження та відстеження змін. Bitbucket інший сервіс для хостингу проектів з Git та Mercurial підтримкою.

React, Angular, Vue.js це фреймворки для розробки фронтенду веб-додатків. Django, Flask (Python) фреймворки для швидкої розробки веб-додатків на Python. Express.js (JavaScript/Node.js) мінімалістичний та гнучкий фреймворк для розробки серверних додатків на Node.js. Ruby on Rails (Ruby) фреймворк для розробки веб-додатків на мові Ruby [12, 13].

JUnit, NUnit, Pytest бібліотеки для написання та виконання тестів у Java, .NET, Python відповідно. Selenium інструмент для автоматизованого тестування веб-додатків. Debugger в IDE вбудовані інструменти для відлагодження коду під час розробки.

Це лише частина інструментів та технологій, які використовуються в розробці програмного забезпечення. Вибір конкретних інструментів залежить від специфіки проекту та особистих вподобань розробника.

Ці інструменти відіграють важливу роль у розробці веб-сайтів і веб-додатків, і вони можуть значно полегшити роботу розробників та дизайнерів.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз сайтів студентських гуртків

Веб-сайти сьогодні стали важливою складовою сучасного інтернет-простору і знаходять широке застосування у різних сферах, включаючи освіту і науку. В рамках даної дипломної роботи проводиться аналіз веб-сайтів, створених студентськими науковими групами. Основна мета аналізу полягає в вивченні сучасних веб-технологій, які використовуються для розробки таких веб-ресурсів.

Сьогодні студентські наукові гуртки активно використовують веб-сайти для просування своїх досліджень, обміну інформацією та спілкування зі спільнотою. Вони створюють ці ресурси з метою забезпечення доступу до актуальних наукових матеріалів, розповсюдження результатів своїх досліджень та сприяння співпраці у галузі науки.

Аналіз таких веб-сайтів є важливим етапом для розуміння їхньої функціональності та ефективності. Ця дипломна робота націлена на глибокий аналіз веб-технологій, які використовуються у розробці сайтів студентськими науковими групами, з метою вдосконалення їхньої діяльності та розширення можливостей в області наукових досліджень. Дослідження цієї теми є актуальним у контексті сучасного наукового співтовариства та розвитку веб-комунікацій. Розглянемо 3 аналоги веб-сайтів студентських наукових гуртків.

Веб-сайт студентського наукового гуртка факультету електроніки та комп'ютерних технологій страждає від кількох суттєвих проблем. Перш за все, його зовнішній вигляд та дизайн залишають бажати кращого, і вони виглядають застарілими та мало привабливо. З іншого боку, меню навігації допомагає знаходити потрібну інформацію, розташовану логічно. Щодо інформаційного наповнення, сторінки сайту містять докладну інформацію про гурток і його діяльність. Проте текстовий контент може бути поданий більш цікаво для користувачів.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

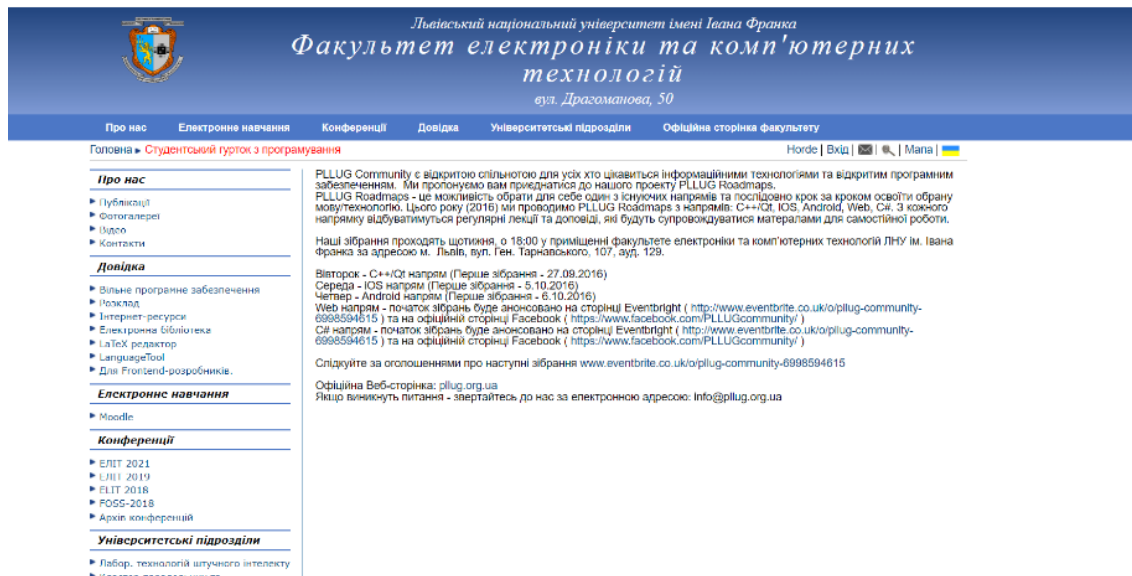


Рисунок 1.1 – Головна сторінка факультету електроніки та комп’ютерних технологій Львівського національного університету імені Івана Франка

Ще однією проблемою є відсутність деяких функцій, які зазвичай очікують від сайту студентського гуртка. Наприклад, сайт не надає можливості реєстрації для учасників гуртка, не має календаря подій для планування та не оновлюється регулярно. Останні записи на сайті датуються 2016 роком, що може відлякувати відвідувачів та зменшити їхню зацікавленість.

Навіть при наявності можливості перегляду сайту на мобільних пристроях, поганий дизайн та вигляд інтерфейсу можуть негативно позначитися на загальному враженні користувачів.

Тому сайт студентського наукового гуртка потребує серйозного оновлення та поліпшень для того, щоб привабити більше відвідувачів та надати їм зручний і актуальний контент.

									Арк.
									19
Змн.	Арк.	№ докум.	Підпис	Дата					



"Студентський науковий гурток"

Додаткова інформація: [Кафедра інженерії програмного забезпечення](#)

СТУДЕНТСЬКИЙ ГУРТОК НАУКОВОГО СПРЯМУВАННЯ «СУЧАСНІ ІНФОРМАЦІЙНО-ПРОГРАМНІ ІННОВАЦІЇ»

Керівник гуртка: доцент кафедри інженерії програмного забезпечення, кандидат технічних наук Вовк Роман Богданович, e-mail: roman.vovk@nung.edu.ua.

Наказ про створення гуртка: (завантажити / [Наказ_100.pdf](#))

Для того, щоб записатися в гурток необхідно заповнити заяву і відправити її на електронну пошту roman.vovk@nung.edu.ua.

(завантажити / [Заява_прийняття_в_гурток.doc](#))

Мета та завдання гуртка – залучення студентів до проведення науково-дослідної та Research and Development діяльності, участі у науково-практичних конференціях, бізнес-проектах, інноваційних заходах та створення умов для творчого зростання у процесі навчання і оволодіння фахом для майбутньої професійної діяльності та кар'єрного росту як спеціалістів в IT галузі.

Основні тематики наукових досліджень

- Інтелектуальні інформаційні та інформаційно-аналітичні технології;
- Наукові засади побудови систем баз даних та знань;

Рисунок 1.2 – Головна сторінка студентського гуртка наукового спрямування “Сучасні інформаційно програмні інновації” Івано-Франківського національного університету нафти і газу

Веб-сайт студентського гуртка наукового спрямування “Сучасні інформаційно-програмні інновації” залишає бажати кращого на багатьох рівнях. Дизайн цього сайту студентського гуртка залишає бажати кращого. Він виглядає застарілим та непривабливим, а використані кольори та шрифти роблять його важким для читання. Навігація практично відсутня, що ускладнює пошук інформації, робить перебування на сайті малоприємним та заплутаним.

На сайті можна помітити велику кількість текстового контенту, що може відлякувати відвідувачів та ускладнити їм розуміння інформації. Потрібно акцентувати увагу на лаконічному та зрозумілому поданні інформації, щоб залучити більше користувачів.

Сайт гуртка обмежений у функціональності. Користувачі не мають можливості спілкуватися, залишати відгуки або долучатися до подій через відсутність інтерактивних інструментів. Додавання цих функцій може сприяти більш активному взаємодії веб-сайту з відвідувачами.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

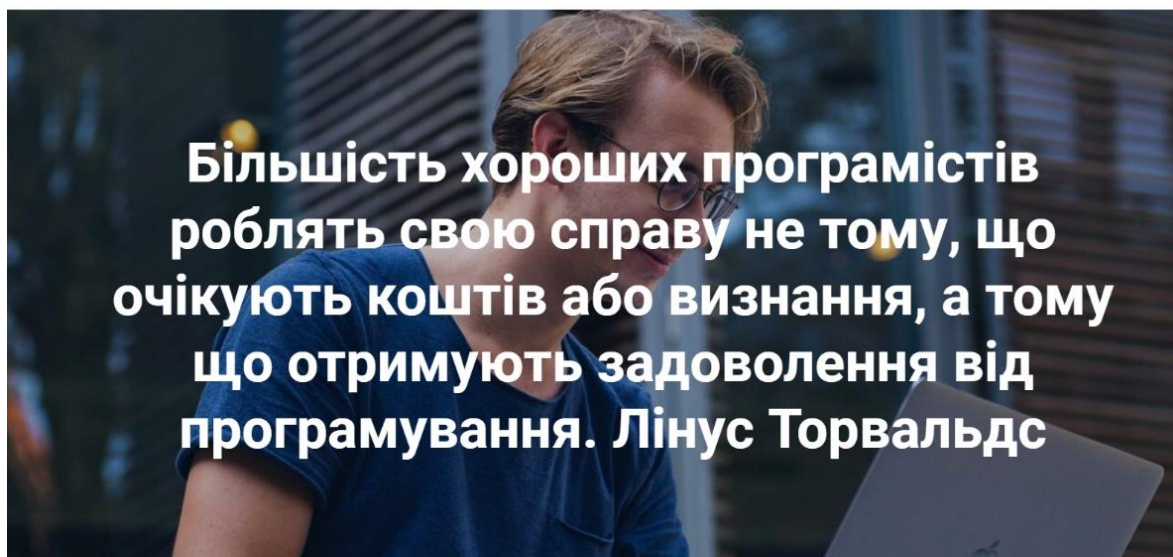


Рисунок 1.3 – Головна сторінка студентського наукового гуртка “ІТ” відокремленого структурного підрозділу «Боярський фаховий коледж НУБіП України»

Сайт студентського наукового гуртка “ІТ” вирізняється привабливим та стильним дизайном, вражає своєю простотою та елегантністю. Він показує, що навіть прості рішення можуть виглядати дуже привабливо, якщо вони створені гармонічно та зі смаком. Кожен елемент дизайну має своє місце та доповнює загальний вигляд сайту. Гурток має свій логотип, хоча він досить простий та не надто запам'ятовується. Можливо, в майбутньому варто розглянути можливість оновлення логотипу для створення більшої впізнаваності та бренду.

Сайт має зручне меню навігації, що спрощує процес пошуку потрібної інформації. Це робить взаємодію з сайтом приємною та ефективною, допомагаючи легко знаходити різні розділи та сторінки. Текстовий контент на сайті викладений дуже професійно. Він є досить інформативним та цікавим для читачів, а його лаконічність та структурованість сприяють легкому сприйняттю інформації.

Серед переваг сайту слід відзначити наявність календаря подій та розділу новин, що робить його більш функціональним. Однак відсутність

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

можливості реєстрації на сайті може обмежувати можливості організації подій та залучення нових учасників.

Сайт постійно оновлюється та має актуальну інформацію, що свідчить про активну діяльність гуртка та прагнення надавати корисну і актуальну інформацію своїм відвідувачам.

1.3 Порівняльний аналіз веб-сайтів

У світлі сучасних тенденцій у вищій освіті, де особлива увага приділяється науковій та освітній діяльності викладачів університетів, важливою стає інформаційна відкритість та доступність. Веб-сайти студентських наукових гуртків стають не тільки засобом комунікації та інформування, але й майданчиком для взаємодії та обміну знаннями.

У процесі створення веб-сайту студентського наукового гуртка слід звернути увагу на декілька ключових аспектів. Перш за все, це структура та організація інформації. Важливо зрозуміти, які дані та розділи будуть представлені на веб-сайті, і як вони будуть взаємозв'язані. Навігація повинна бути легкою та зрозумілою для користувачів, щоб вони могли швидко знаходити потрібну інформацію.

Дизайн грає важливу роль у привабленні відвідувачів. Необхідно звертати увагу на візуальний вигляд вебсайту, використовувати яскраві фотографії та елементи графіки, щоб зробити сторінку привабливою та сучасною [7].

Меню навігації повинно бути зручним і легким для користувачів. Потрібно розмістити його так, щоб воно було доступним та видимим на кожній сторінці веб-сайту.

Текстовий зміст веб-сайту повинен бути інформативним та цікавим. Хорошим рішенням буде додавання блогу або розділу новин, де можна

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

ділитися актуальними подіями та досягненнями гуртка. Спілкування з аудиторією через форум чи коментарі створить сприятливу атмосферу для обміну думками та ідеями.

Функціональність веб-сайту також важлива. Це, наприклад, можливість введення реєстрації для учасників гуртка, створення календаря подій та можливості завантаження матеріалів для учасників. Допоможе зберігати інтерес аудиторії та привертати нових учасників актуальність інформації на веб-сайті та регулярне оновлення його вмісту.

Загалом, створення веб-сайту студентського наукового гуртка - це можливість не лише інформувати, але і взаємодіяти з аудиторією, підтримувати активність та сприяти розвитку наукової спільноти [22].

Після проведення аналізу студентських наукових гуртків інформаційно-програмного спрямування, є можливість застосування інноваційних рішень на веб-сайті студентського гуртка "Технології DevOps".

Отримані результати допомагають покращити користувацький досвід, забезпечити зручну навігацію та зробити цей веб-сайт більш привабливим та ефективним для учасників і відвідувачів.

Загальний аналіз аналогів веб-сайтів студентських гуртків представлено у таблиці 1.1.

Таблиця 1.1 – Загальний аналіз аналогів веб-сайтів студентських гуртків

Критерії	Веб-сайт студентського наукового гуртка факультету електроніки та ком'ютерних технологій	Вебсайт студентського гуртка наукового спрямування "Сучасні інформаційно-програмні інновації"	Сайт студентського наукового гуртка "IT" боярського фахового коледжу національного університету біоресурсів і природокористування України
Дизайн	1	2	4

Продовження таблиці 1.1

Навігаційне меню	+	-	+
Інтерактивність	-	-	+/-
Пошукова система	+	+	+
Якість вмісту	3	2	4
Можливість зворотнього зв'язку	-	+	+
Соціальні медіа	+	-	+
Адаптивність	+	+	+
Логотип	0	0	2
Зручність користування	2	1	4

Вищенаведений аналіз відкриває нові можливості для підвищення якості веб-присутності та забезпечення відкритості для інформаційного обміну та співпраці в галузі інформаційних технологій.

1.4 Постановка завдань

На основі проведеного аналізу були поставлені такі завдання:

- проаналізувати існуючі підходи до розробки веб-сайтів студентських наукових груп;
 - проаналізувати сучасні засоби розробки веб-сайтів;
 - розробити діаграму функціональних обов'язків користувачів;
 - розробити алгоритми аналізу маршрутних таблиць;
 - розробити алгоритм формування блоків веб сторінки;
 - розробити графічний інтерфейс головної сторінки сайту;
- здійснити тестування програмного засобу.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

2 АЛГОРИТМИ РОБОТИ ВЕБ-САЙТУ СТУДЕНТСЬКОГО ГУРТКА

2.1 Діаграма функціональних обов'язків користувачів

В сучасному академічному середовищі акцент на розвиток студентських гуртків, присвячених інноваційним технологіям, має значущий вплив на освітню та наукову сферу. У цьому контексті, веб-платформи для студентів набувають великого значення, створюючи можливості для ефективної взаємодії та обміну інформацією, спрямованою на розвиток їхніх навичок та знань.

User Case діаграма, що досліджує взаємодії на веб-сайті гуртка DevOps програмування, стає важливим інструментом аналізу та розуміння функціональних можливостей платформи [34]. Акцентуючи увагу на різноманітних категоріях користувачів та їхніх специфічних потребах, діаграма визначає ключові сценарії використання та функціональні аспекти веб-сайту.

Аналіз User Case діаграми розкриває глибину взаємодій, що відбуваються на веб-сайті. Детальне розглядання сценаріїв використання для гостей, зареєстрованих студентів та адміністраторів дозволяє зрозуміти, як кожна категорія користувачів взаємодіє із платформою в контексті DevOps програмування.

Цей аспект аналізу відіграє не лише роль визначення функціональних вимог та особливостей веб-сайту гуртка, а й має стратегічне значення для покращення користувацького досвіду. Підходячи до цього розділу з наукового погляду, це дозволяє висвітлити не лише внутрішній механізм взаємодій, але й розкрити можливі перспективи розвитку та удосконалення веб-середовища для студентів гуртка DevOps програмування. На рисунку 2.1 наведено діаграму функціональних обов'язків користувачів сайту.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.1 – User case діаграма

У контексті впровадження веб-інтерфейсу для студентського гуртка, присвяченого проблематиці DevOps програмування, надзвичайно вагомим етапом виявилася розробка User Case діаграми, яка детально структурує взаємодії різних категорій користувачів з системою. Вищезазначені сценарії використання визначають ключові можливості, які надаються користувачам, і враховують основні аспекти функціональності веб-платформи для гуртка.

Гість, як основний користувач системи, має можливість докладно ознайомитися з інформацією щодо гуртка. Це включає в себе доступ до розкладу занять та загальної програми. Також гість може ознайомитися з методикою викладання та отримати детальні відомості про викладача. Пошукові можливості спрощують взаємодію гостя з матеріалами гуртка. Для взаємодії на більш особистому рівні гість може використовувати функціонал реєстрації на сайті та подати заявку для участі у курсі.

Зареєстрований студент, на відміну від гостя, має розширений функціонал. Увійшовши в особистий кабінет, студент може керувати своїми особистими даними та здійснювати редагування інформації. Окрім того, студент може переглядати розклад та програму занять та здійснювати перегляд відеоуроків. Також студент може залишати заявки на нові курси, розширюючи свої можливості для навчання.

Роль адміністратора надає повний контроль над функціоналом веб-платформи. Адміністратор має право доступу до управління користувачами та їхніми персональними даними. Також він може додавати та редагувати відеоуроки, а також вносити зміни в інформацію, що стосується гуртка. Адміністратор відповідає за управління заявками студентів на участь у курсах та регулювання функціоналу сайту.

Ця User Case діаграма становить підґрунтя для аналізу і визначення функціональних вимог до системи. Вона послужить основою для подальшої розробки веб-сайту та реалізації важливих функціональних можливостей для різних категорій користувачів.

В таблиці 2.1 представлено загальний аналіз User Case діаграми.

Таблиця 2.1 – Загальний аналіз User Case діаграми

Функціонал	Адміністратор	Гість	Зареєстрований студент
Керування контентом	+	-	-
Перегляд інформації про гурток	+	+	+
Додавання/редагування інформації	+	-	-
Залишити заявку на курс	-	-	+

Продовження таблиці 2.1

Отримання сповіщень та оновлень	-	-	+
Пошук матеріалів	+	+	+
Перевірка заявок	+	-	-

Адміністратор відіграє ключову роль у керуванні системою та її контентом. Однією з його основних функцій є керування контентом, зокрема інформаційними блоками, що включає у себе додавання нового контенту, редагування існуючого та видалення застарілої інформації. Адміністратор має повноваження здійснювати зміни у всіх аспектах системи для забезпечення актуальності та релевантності інформації.

Гість системи, незареєстрований користувач, може використовувати лише обмежений функціонал. Він має можливість переглядати інформацію про гурток та матеріали, але не має можливості вносити зміни чи залишати заявки. Головною метою гостя є ознайомлення з основною інформацією та змога прийняття рішення щодо реєстрації в системі.

Зареєстрований студент має розширений функціонал порівняно з гостем. Окрім можливості перегляду інформації, зареєстрований користувач може залишати заявки на участь у курсах. Він також має доступ до функціоналу пошуку матеріалів та перевірки статусу своїх заявок. При цьому він не має повноважень щодо редагування інформації чи отримання сповіщень та оновлень, що залишаються відведеними для адміністратора.

2.2 Алгоритми формування блоків веб сторінки

На початковому етапі взаємодії з веб-сайтом студентського гуртка DevOps, користувач вводить URL, і в цей момент розпочинається складний та

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

продуманий алгоритм формування веб-сторінки. Використання HTML та CSS створює базову структуру сторінки, при цьому кожен елемент призначений для важливої ролі у забезпеченні не лише зручного виведення інформації, але й вдумливого користувацького досвіду [6].

Особлива увага приділяється стилізації та оформленню блоків за допомогою CSS, де кожен кольоровий акцент та елемент дизайну враховуються для забезпечення не лише функціональності, але і естетичності веб-сайту [15, 21]. Стиль враховує сучасні тенденції та вимоги щодо зручності взаємодії із сторінкою.

Щоб створити динамічний та вражаючий веб-сайт, використовується JavaScript. Елементи взаємодії, такі як інтерактивні меню чи новини, вбудовані таким чином, щоб не лише надати користувачам можливість швидкого доступу до інформації, але й поглибити їхнє зацікавлення у сфері DevOps.

Окремий блок відведено для розробки меню та навігації, створення якого передбачає створення інтерактивних елементів для швидкого та легкого доступу до всіх розділів гуртка. Це дозволяє студентам ефективно взаємодіяти з важливою інформацією.

DevOps проекти та можливості спільної роботи представлені в окремому блоку, де використовується JavaScript для аналізу та візуалізації даних. Це надає можливість студентам активно вивчати та взаємодіяти з реальними проектами, поглиблюючи свої знання та навички в області DevOps.

Механізм виявлення та виведення помилок реалізований для забезпечення коректної обробки невдалих завантажень скриптів чи ресурсів, що підвищує надійність та стабільність роботи веб-сайту в умовах різних сценаріїв.

Для оптимізації швидкодії використовуються різноманітні методи, такі як кешування та оптимізація завантаження ресурсів [36, 37]. Це впливає на продуктивність користувача та покращує загальний досвід взаємодії.

Алгоритм завершується готовим до використання, надаючи студентам гуртка не лише зручний та ефективний інтерфейс для взаємодії, але і

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

інтегровану систему для отримання актуальної та необхідної інформації в сфері DevOps програмування.

1. Початок алгоритму.
2. Користувач вводить URL веб-сторінки.
3. Програма відправляє запит на сервер для отримання відповіді.
4. Програма очікує відповіді від сервера.
5. Після отримання відповіді від сервера, програма аналізує її, витягує необхідні дані та метадані.
6. Вивід блоку `<head>` сторінки, що містить мета-теги, заголовок і посилання на зовнішні ресурси, такі як CSS та JavaScript файли.
7. Завантаження CSS файлів для оформлення сторінки.
8. Завантаження JavaScript-скриптів для надання функціональності сторінки.
9. Завантаження блоку основного вмісту сторінки.
10. Вивід секції меню, яка, ймовірно, містить посилання на різні частини сторінки або на інші сторінки в цьому ж веб-домені.
11. Запуск циклу для кожного значення i від 0 до n , де n - кількість блоків новин.
12. Виведення блоку новини для кожного значення i .
13. Вивід слайдера.
14. Перевірка успішності завантаження слайдера.
15. У разі, якщо слайдер не завантажився успішно, програма виводить повідомлення про помилку в консолі браузера.
16. Якщо слайдер завантажився успішно, програма продовжує виведення додаткових матеріалів.
17. Кінець алгоритму.

Розглядаючи алгоритм, який відображає послідовність дій під час завантаження веб-сторінки студентського гуртка з DevOps програмування, визначається ключовий етап - завантаження та оптимізація важливих

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

компонентів для забезпечення зручності та ефективності взаємодії користувачів.

Хедер, логотип та блок навігації, на початку процесу користувач витрачає мінімальний час на завантаження хедера, логотипу та блоку навігації. Ці елементи не лише формують основний елемент інтерфейсу, але й визначають структуру веб-сайту, надаючи зручний інструмент для навігації.

Завантаження форми та валідація, важливим кроком є завантаження форми для студентського гуртка, де кожен елемент ретельно розробляється з метою надання можливості введення необхідної інформації. Цей етап додатково включає в себе важливий процес валідації введених даних, щоб забезпечити їх правильність та уникнути можливих помилок. Валідатори форми гарантують, що користувачі вводять коректну інформацію, що підвищує надійність обробки даних.

Відправлення даних та повідомлення, у разі успішної валідації, дані відправляються на сервер, а користувач отримує відповідне повідомлення, що відображається на веб-сторінці. Цей етап дозволяє ефективно взаємодіяти з системою та отримувати повідомлення про стан обробки інформації.

Футер та оптимізація завантаження ресурсів, завершальним етапом є завантаження футера, який додає завершений вигляд сторінці. Оптимізація ресурсів, така як кешування та компресія, використовується для покращення продуктивності та реактивності веб-сайту.

Сучасні фреймворки та питання безпеки, студентський гурток DevOps програмування також враховує використання сучасних фреймворків для створення динамічних та інтерактивних елементів. Заходи безпеки, такі як HTTPS, шифрування даних та застосування сучасних методів забезпечують конфіденційність та надійність обробки інформації.

Загальний підхід до розробки, унікальний підхід до розробки веб-сторінки студентського гуртка DevOps програмування поєднує технічні інновації, принципи дизайну та враховує оптимізацію та безпеку. Це

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечує не тільки ефективне використання системи, але і робить взаємодію з веб-сайтом максимально зручною та продуктивною для студентів.

2.3 Структура та вміст веб-сайту наукового гуртка

У даному розділі приділена особлива увага структурі та організації веб-сайту студентського гуртка, присвяченого області DevOps програмування. Мета розробки сайту полягає в досягненні оптимальної подачі інформації та наданні зручності для користувачів, щоб забезпечити їм якісний та зручний взаємозв'язок із ресурсом.

Зазначений веб-сайт відрізняється своєю пристосованістю до потреб користувачів та враховує найсучасніші стандарти веб-розробки. Здійснено спробу логічно розподілити функціональність сайту з метою надання зрозумілої та ефективної навігації.

Оглядаючи загальний розподіл сторінок та розділів, можна визначити, що велика увага приділена якісній структуризації контенту. Це сприяє не лише легкому знаходженню необхідної інформації, але й покращує взаємодію користувачів із сайтом у цілому.

Детальний розгляд окремих елементів та їхніх функцій дозволяє побачити високий рівень уваги до деталей та потреб користувачів. Такий підхід робить сайт більш привабливим для аудиторії та допомагає в ефективному використанні його можливостей.

Загалом, структура веб-сайту гуртка DevOps програмування відповідає сучасним вимогам та визначається спрямованістю на зручність використання, що робить його цікавим та привабливим для активної аудиторії.

Структура сайту складається з ключових розділів, які логічно організовані для максимальної зручності користувачів:

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Головна сторінка вітає відвідувачів та надає коротку інформацію про гурток. Забезпечує огляд доступного контенту, включаючи ключові функції гуртка. Застосована кнопка “Реєстрація” спрямовує відвідувачів на сторінку реєстрації для участі в гуртку.

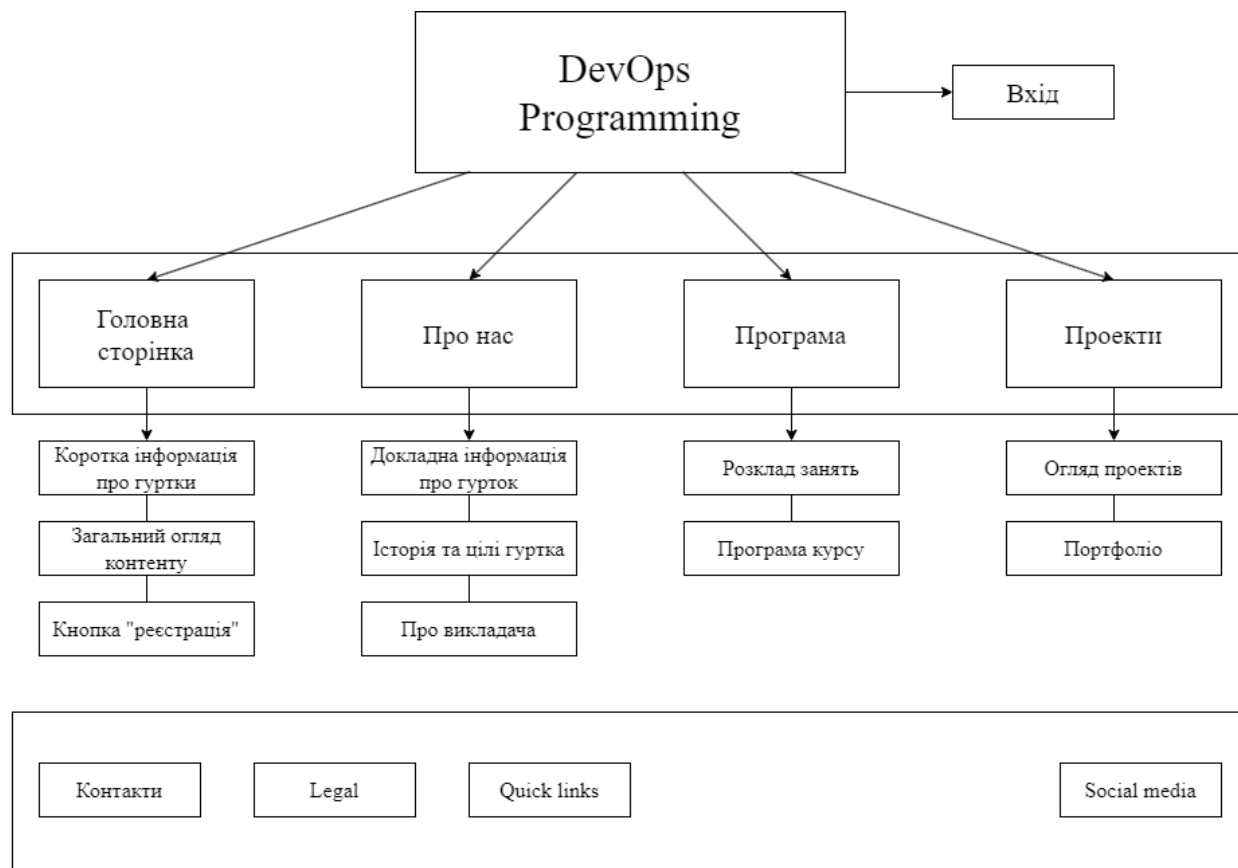


Рисунок 2.2 – Content Mapping

Сторінка “Про нас” містить докладну інформацію про гурток, його історію, цілі та склад команди, включаючи інформацію про викладачів. Детальний опис допомагає розуміти місію та спрямування гуртка.

На сторінці “Програма” розташовані розклад занять та лекцій, а також докладна програма курсів. Забезпечує студентам можливість ознайомитися з розкладом та змістом представлених курсів.

На сторінці “Проекти” студенти можуть вивчати огляд проектів, розроблених членами гуртка, а також ознайомитися з портфоліо успішних

завершених проектів. Це допомагає підвищити зацікавленість учасників та створити сприятливу атмосферу для творчості.

На сторінці “Новини” розміщуються актуальні новини та оголошення. Події та досягнення гуртка привертають увагу студентів та сприяють активній участі в житті спільноти.

“Контакти” це сторінка для зв'язку із адміністрацією гуртка. Форма зворотного зв'язку надає зручний спосіб висловлювання питань чи надання фідбеку.

Функціонал “Пошук” для швидкого пошуку інформації на сайті, з можливістю фільтрації результатів за різними категоріями, такими як курси, відео, новини та інше.

На сторінці “Реєстрація” розташована форма для реєстрації нових студентів, а також інформація про переваги реєстрації, включаючи доступ до закритого контенту та можливість участі в проектах.

Цей загальний огляд структури сторінок надає фундаментальний зміст для подальшого детального розгляду та аналізу в рамках даного дипломного дослідження.

У даному розділі на основі алгоритмічного підходу запропоновано діаграму функціональних обов'язків користувачів, яка включає базові функціональні блоки користувачів. На основі алгоритмічного підходу наведено алгоритм формування блоків веб-сторінки, що включає в себе як технічні, так і функціональні аспекти веб-сторінки студентського гуртка.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБСАЙТУ СТУДЕНТСЬКОГО ГУРТКА

3.1 Серверна структура веб-сайту студентського гуртка

Значну роль у функціонуванні веб-сайту відіграє його серверна частина, аде саме на сервері відбувається обробка інформації, формування відповіді до браузера клієнта та ін [35]. Наразі хорошою практикою вважається використання існуючих фреймворків та систем контролю вмістом, а не написання стандартного функціоналу «з нуля». Базовими підходами в такому випадку є використання фреймворків Laravel та систем контролю вмістом по типу Wordpress [29-31].

Для реалізації поставленого завдання обрано системи контролю вмістом Grav, оскільки вона дозволяла максимально реалізувати власний підхід до дизайну, а не бути обмеженим функціоналом і вимогами наприклад Wordpress. Мініміальні вимоги до серверної складової такі:

- Apache;
- PHP 7.3.6 або вище.

Структуру директорій проекту наведено на рисунку 3.1.

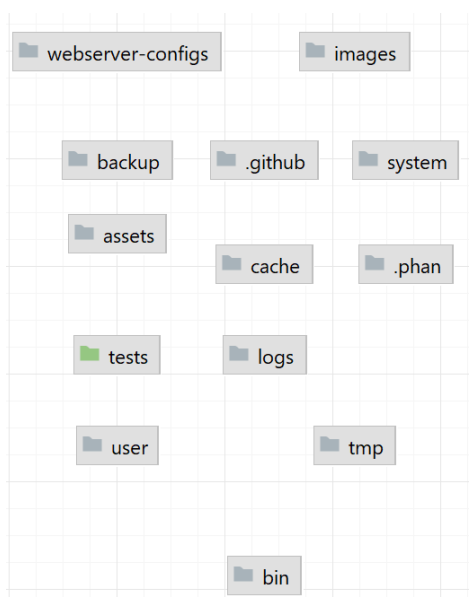


Рисунок 3.1 - Структуру директорій проекту

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Структуру директорії «system» наведено на рисунку 3.2.

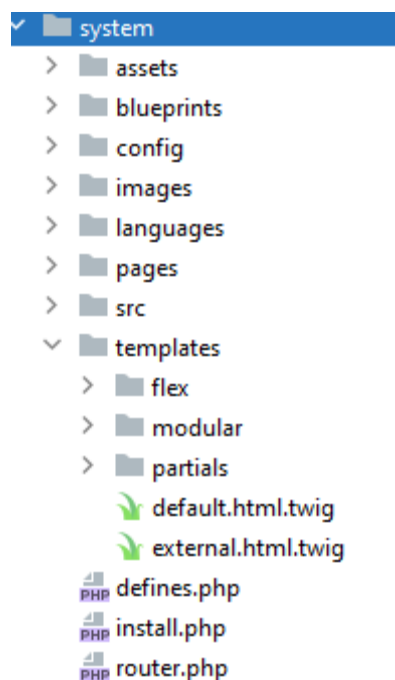


Рисунок 3.2 - Структуру директорії «system»

Наведена структура директорій передбачає окремі модулі для зберігання шаблонів сторінок, що дозволяє реалізувати власні шаблони без обмежень накладених самою системою контролю вмістом.

Наведена структура дозволяє розподілити логіку розподілу файлів-стилів, файлів –шаблонів, бекенд файлів, файлів зображень тощо.

3.2 Графічний інтерфейс головної сторінки сайту

Розробка стилю веб-сайту для студентського гуртка з DevOps програмування вимагає уважного підходу до оформлення, враховуючи особливості цільової аудиторії та його функціональність [24]. Розглянемо ключові аспекти оформлення та стилю веб-сайту для забезпечення оптимального користувацького досвіду.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Графічні елементи мають відповідати тематиці DevOps, використовуючи іконки та зображення, які стимулюють інтерес та відображають ключові аспекти гуртка.

Забезпечення зручної навігації - важливий елемент. Головне навігаційне меню слід розташовувати в легкодоступному місці сторінки, надаючи швидкий доступ до ключових розділів та інформації [9].

Враховуючи вищезазначені аспекти, розроблено графічний інтерфейс веб-сайту студентського гуртка з DevOps програмування, що відображений на рисунку 3.3. Цей дизайн має відображати сучасність, інноваційність та технологічність гуртка, забезпечуючи зручний та привабливий користувацький інтерфейс.

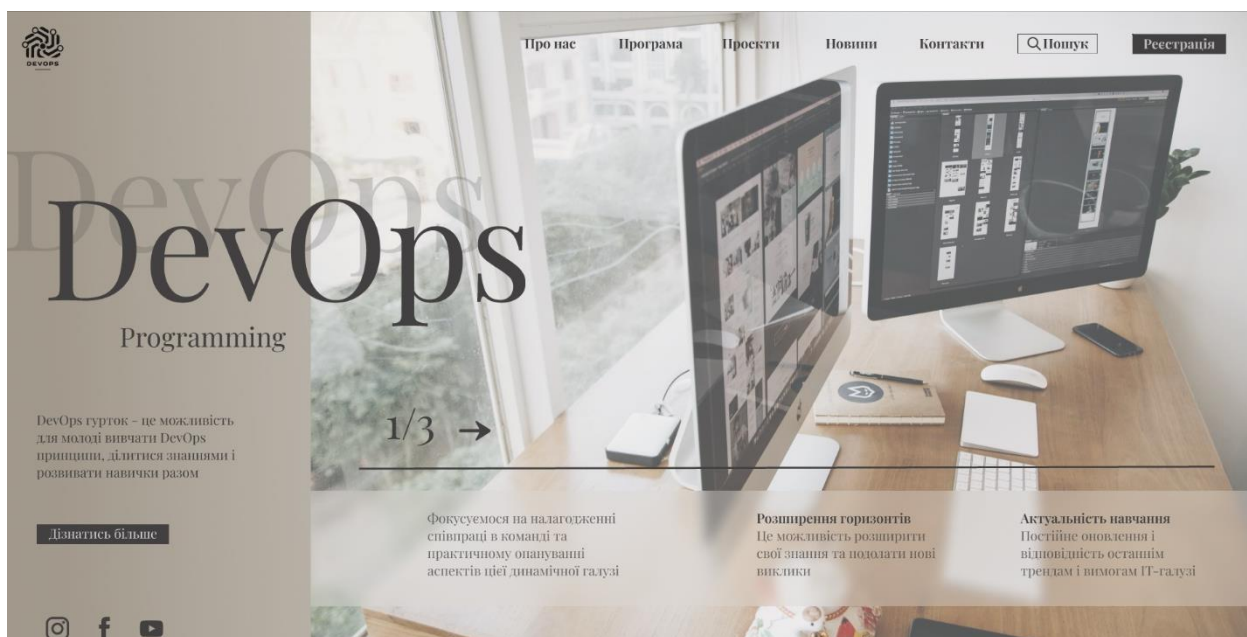


Рисунок 3.3 – Інтерфейс веб-сайту студентського гуртка з DevOps програмування

Нижче наведено приклад htmlкоду для виводу сторінки “Про нас” (рис. 3.4):

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

27     <div class="aboutus">
31         <div class="frame-20">
32             <div class="unsplashb18trxc8upq-wrapper">
38                 </div>
39                 <div class="div172">
40                     <div class="group-33">
41                         <div class="div173">
42                             <div class="div174">Про нас</div>
43                         </div>
44                         <div class="div175">
45                             <div class="div174">
46                                 DevOps: Інновації через спільність та код
47                             </div>
48                         </div>
49                     </div>
50                 </div>
51             </div>
52             <div class="frame-22">
53                 
54
55                 <div class="frame-devops">
56                     
61
62                     

```

Рисунок 3.4 – Приклад html-коду для виводу сторінки “Про нас”

У контексті наданого HTML-коду можна розглядати використання тегу ``<div>`` та ```` для структурування та візуального оформлення вмісту веб-сторінки.

Тег ``<div>`` є блочним елементом, який використовується для групування та стилізації інших елементів на веб-сторінці. Це важливий інструмент для створення структури та розташування різних компонентів веб-сайту. В прикладі коду, що наданий, ``<div>`` використовується для організації різних блоків та розділів сторінки.

Отже, подібно до наданого HTML-коду, використання тегу ``<div>`` для групування та ```` для відображення зображень є потужним засобом організації вмісту веб-сторінки та поліпшення користувацького досвіду.

Приклад htmlрозмітки для реалізації пагінації наведено на рисунку 3.2.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		


```

<html>
<head> ... </head> == $0
  <body>
    <div class="aboutus">
      <div class="group-32"> ... </div> flex
      <div class="frame-20"> ... </div>
      <div class="frame-22"> ... </div>
      <div class="frame-23"> ... </div> flex
      <div class="frame-24"> ... </div>
      <div class="frame-25"> ... </div>
      <div class="frame-26"> ... </div>
      <a href="index.html"> ... </a>
      <div class="div192"> ... </div> flex
      <div class="frame-27"> ... </div>
    </div>
  </body>
</html>

```

Рисунок 3.5 – Приклад html-розмітки для реалізації пагінації

Пагінація в програмуванні - це процес розділення великої кількості даних або вмісту на окремі сторінки з метою полегшення навігації та оптимізації завантаження сторінок [15]. Це часто використовується в веб-розробці та інших програмних застосунках, де велика кількість інформації може стати проблемою для користувача або може призвести до зниження продуктивності.

На деяких сторінках пагінації можуть бути доступні інструменти фільтрації та сортування для полегшення знаходження конкретної інформації. Пагінація допомагає забезпечити кращий користувацький досвід та оптимізує використання ресурсів, що є особливо важливим для великих обсягів даних.

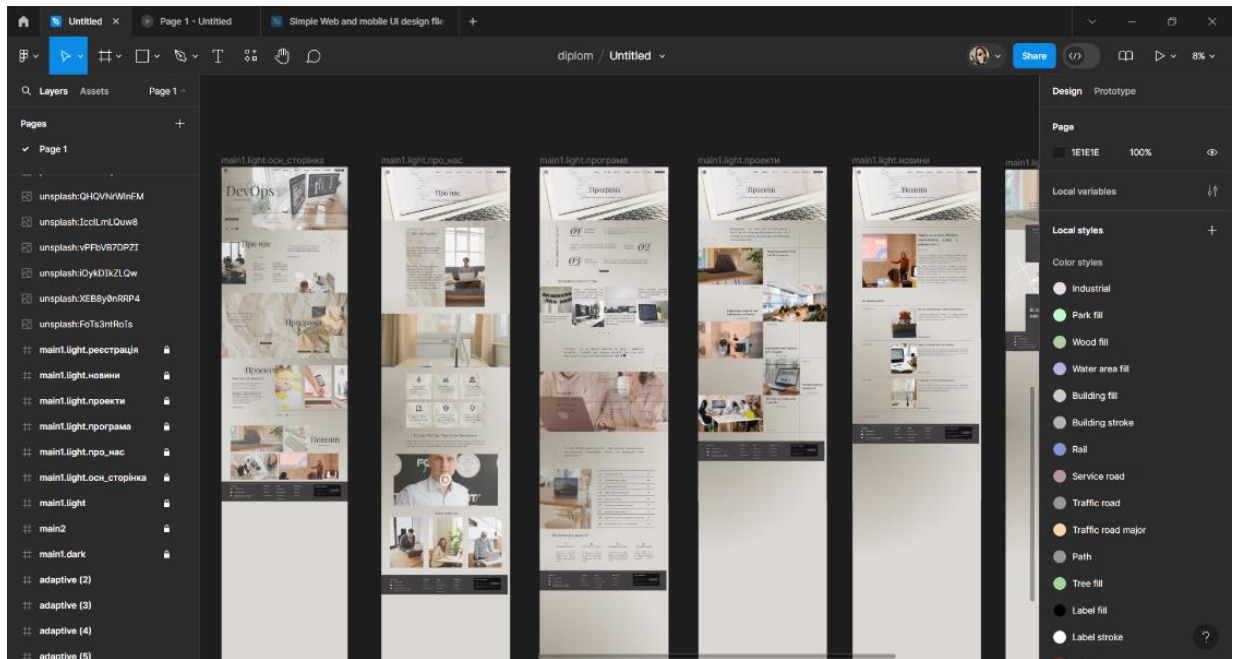


Рисунок 3.6 – Скриншот з Figma розробки дизайну сайту

На скриншоті наведено робочу область розробленого веб-сайту для студентського гуртка з DevOps програмування. Робоча область відображає інтерфейс, який був розроблений з урахуванням основних принципів пагінації для полегшення навігації та зручного доступу до контенту.

На першому плані видно розділення вмісту на окремі сторінки, що дозволяє користувачам легко переглядати та переміщатися між різними частинами веб-сайту. Також наявні навігаційні елементи, такі як кнопки "Про нас" та "Програма" та інші, які спрощують навігацію між сторінками.

Додатково, відображення обмеженої кількості даних на кожній сторінці допомагає уникнути перевантаження користувача та покращує швидкість завантаження. Оптимізація продуктивності та доступність інструментів фільтрації та сортування також були враховані при розробці інтерфейсу.

Цей дизайн робочої області сприяє зручній та ефективній навігації для користувачів, що відвідують веб-сайт гуртка з DevOps програмування.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

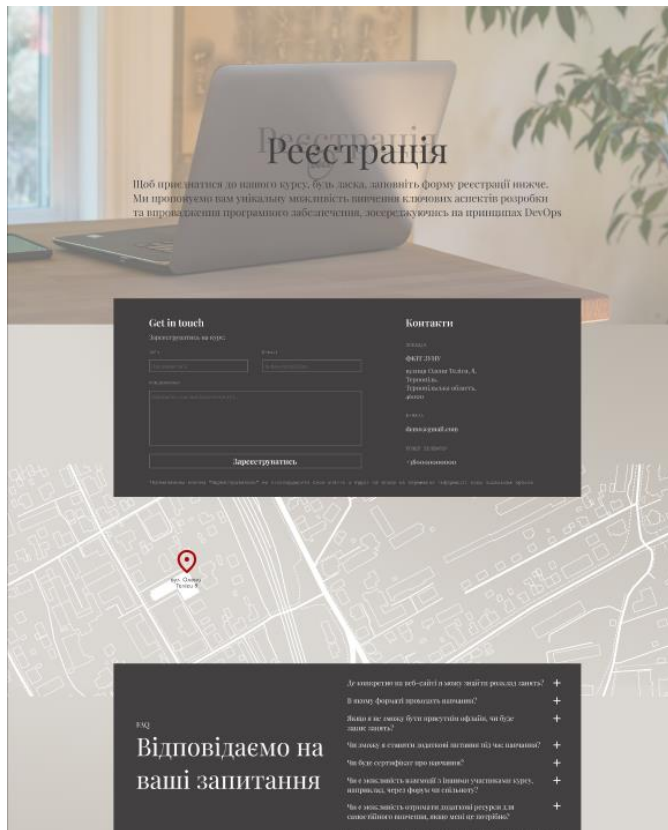


Рисунок 3.7 – Сторінка реєстрації

На зазначеному скриншоті надано зразок дизайну форми, розробленої з метою сприяти зручності та ефективності взаємодії користувачів із веб-сайтом.

Дизайн форми характеризується сучасним стилем та естетичною привабливістю, що підкреслюється використанням гармонійних кольорів та збалансованим розміщенням елементів. Кожен компонент форми, такий як поля для введення даних, кнопки та посилання, відповідає загальному візуальному зразку та забезпечує послідовність у взаємодії користувача з інтерфейсом.

З функціональної точки зору, форма реєстрації/входу надає користувачам можливість легко та зручно виконати потрібні дії. Вона включає в себе відповідні поля для введення особистої інформації, а також кнопки для подальших кроків. Це дозволяє забезпечити користувачам максимальний комфорт та ефективність при використанні форми.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

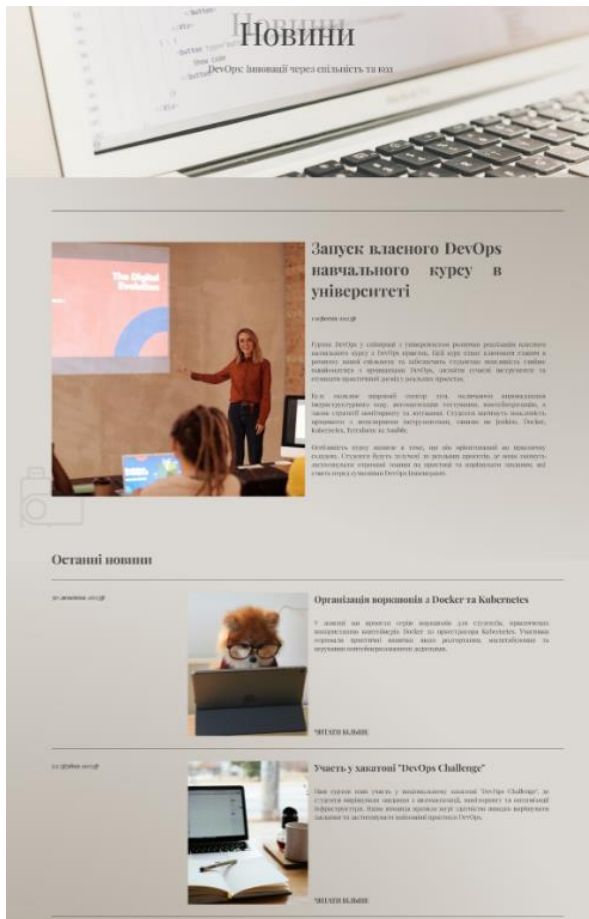


Рисунок 3.8 – Сторінка новини

Сторінка “Новини” є важливою складовою веб-сайту та представляє собою простір для інформування користувачів про останні події, оновлення та цікаві новини. На даному скріншоті видно відображення списку новин, який включає заголовки, короткі анотації та зображення, які привертають увагу користувача.

Дизайн сторінки “Новини” спроектований з урахуванням зручності та естетичного вигляду. Кожен пост має чітку структуру, що включає заголовок для визначення теми новини, анотацію для короткого огляду та зображення для візуального підсилення вмісту. Такий підхід до представлення інформації полегшує сприйняття користувачем та збільшує зацікавленість у відвідуванні сторінки “Новини”.

Крім того, можливість перегляду повної версії новини або переходу за посиланням для отримання додаткової інформації забезпечує гнучкість та

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

комфорт користувача. Такий підхід дозволяє задовольняти потреби різноманітної аудиторії та забезпечує ефективну комунікацію з відвідувачами веб-сайту.

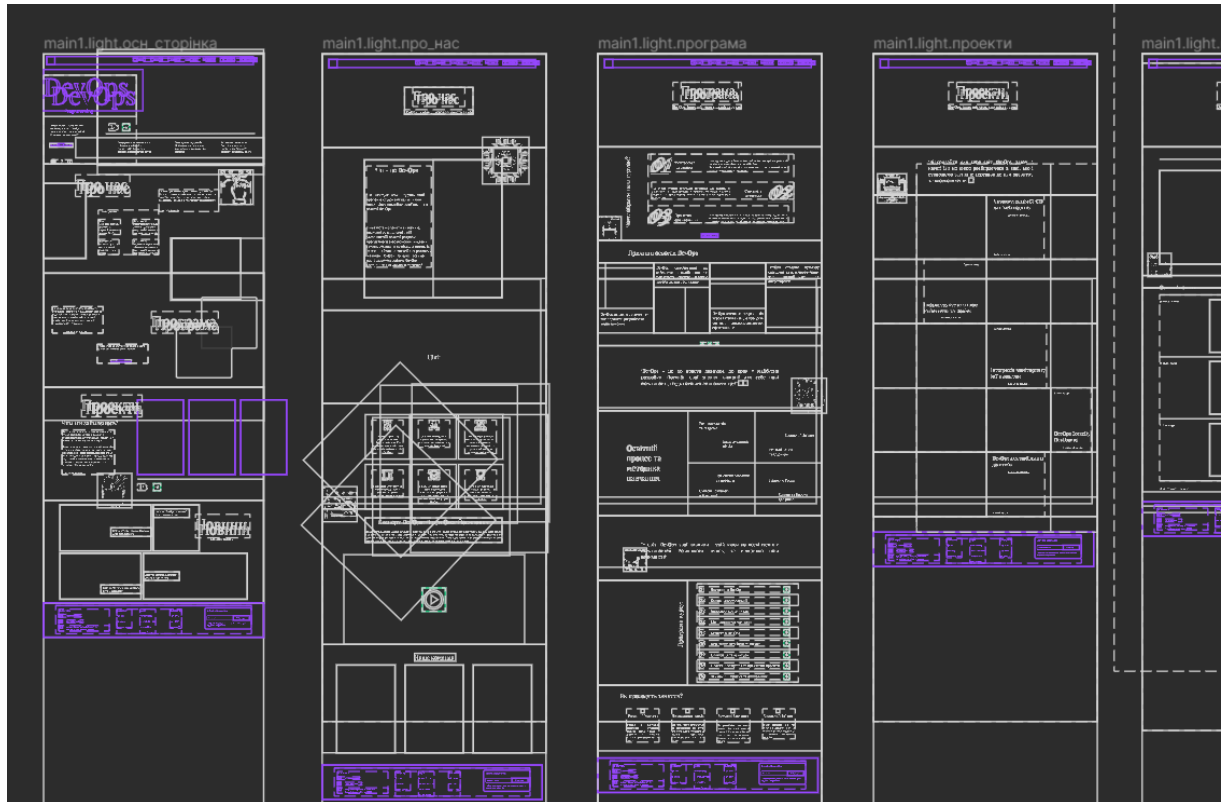


Рисунок 3.9 – Дизайн у режимі контурів

На зазначеному скріншоті, зображеному у режимі контурів у програмі Фігма, простежується детальна структура та розміщення елементів дизайну. Кожен контур та лінія ретельно розташовані, що свідчить про систематичний підхід до компоновання веб-інтерфейсу.

Варто відзначити, що у режимі контурів видно чітку увагу до деталей та розміщення елементів. Це сприяє поліпшенню візуальної зрозумілості та зручності взаємодії для користувача.

Нарешті, аналіз скріншота в режимі контурів дозволяє виявити потенційні області для вдосконалення та оптимізації дизайну, що може сприяти поліпшенню користувацького досвіду.

3.3 Тестування роботи сторінок сайту

Під час тестування функціональності веб-сайту важливо ретельно аналізувати якість графічного інтерфейсу та ефективність роботи. Один із прикладів аналізу можна взяти за основною сторінкою (рисунок 3.10) на веб-сайті. Для проведення тестування був використаний сервіс PageSpeedInsights, який широко використовується для оцінки продуктивності веб-сайтів.

PageSpeedInsights надає інформацію про продуктивність сторінок як на мобільних, так і на десктопних пристроях, а також пропонує рекомендації для оптимізації. Цей сервіс надає як лабораторні дані, отримані зі сторінок під час тестування, так і дані, пов'язані із реальним використанням сторінок користувачами.

Окрім відображення поточного стану сторінки та часу завантаження всіх її елементів, PageSpeedInsights також надає корисні рекомендації щодо покращення продуктивності сторінки. Багато з цих рекомендацій спрямовані на оптимізацію веб-сайту з метою поліпшення його швидкодії та відповідності основним вимогам. Детальний аналіз PageSpeedInsights дозволяє не лише оцінити поточний стан веб-сайту, але й прийняти конкретні заходи для підвищення якості та продуктивності.

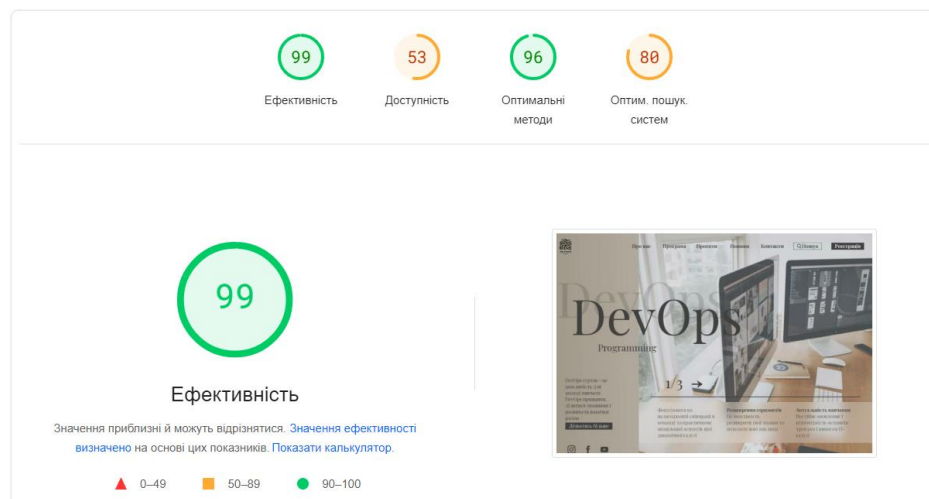


Рисунок 3.10 – Продуктивність сайту на основній сторінці

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

В результаті проведеного дослідження можна зробити висновок, що ефективність функціонування веб-сайту досягає 99%, що представляє собою високий показник і позначається як відмінний результат, знаходячись у “зеленій” зоні.

Під час проведення діагностики були виявлені конкретні аспекти, серед яких тільки три вважаються критичними для оптимізації даного веб-сайту (див. рисунок 3.11).

Мобільний **Комп'ютер**

ДІАГНОСТИКА

▲ Вилучіть ресурси, які блокують відображення — Потенційне заощадження – 370 мс

Ресурси блокують першу візуалізацію сторінки. Надсилайте спершу важливі фрагменти JavaScript або таблиці CSS і відкладайте всі некритичні елементи. [Дізнайтесь, як вилучити ресурси, які блокують відображення.](#) [FCP](#) [LCP](#)

URL-адреса	Розмір передавання	Потенційне заощадження
Google Fonts Cdn	3,7 KiB	830 мс
/css2?family=... (fonts.googleapis.com)	1,3 KiB	210 мс
/css2?family=Courier%20New:wght@400&display=swap (fonts.googleapis.com)	0,0 KiB	210 мс
/css2?family=Poppins:wght@400;600&display=swap (fonts.googleapis.com)	1,1 KiB	210 мс
/css2?family=Nunito:wght@400;500&display=swap (fonts.googleapis.com)	1,2 KiB	210 мс

▲ Правильно виберіть розмір зображень — Можна зменшити на 2 534 KiB

▲ Покажіть зображення в нових форматах — Можна зменшити на 2 235 KiB

Формати зображень WebP і AVIF часто стискаються краще, ніж PNG чи JPEG. Тому вони швидше завантажуються й використовують менше даних. [Докладніше про сучасні формати зображень.](#)

URL-адреса	Розмір ресурсу	Потенційне заощадження

Рисунок 3.11 – Результати діагностики сторінки веб-сайту

Важливим елементом успішної реалізації проекту є дотримання всіх принципів розробки сторінки, а результати вивчення представлені на рисунку 3.12.

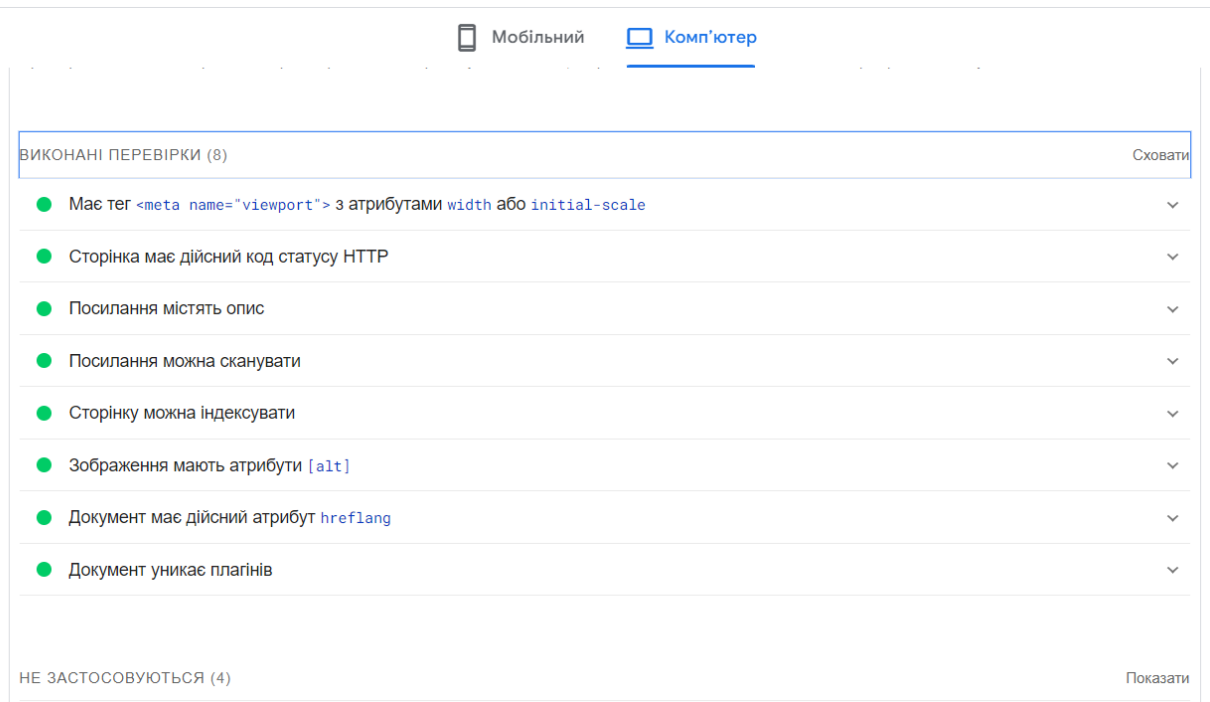


Рис. 3.12 – Результат діагностики сторінки на наявність необхідних тегів

Існуючий і дійсний файл robots.txt важливий для того, щоб вказати веб-сканерам, як правильно індексувати або сканувати ваш веб-сайт. Наявність дійсного атрибуту rel=canonical свідчить про правильне вказання основного URL для відображення в результатах пошуку, що сприяє уникненню проблем з дублюванням контенту. Важливо мати читабельні розміри шрифтів, особливо на мобільних пристроях. Забезпечення шрифтів розміром понад 12px на понад 60% сторінки сприяє комфортному читанню. Великі інтерактивні елементи, такі як кнопки та посилання, важливі для зручності користувачів. Забезпечення їх великими розмірами чи достатньою площею навколо сприяє легкому натисканню без зачіпання інших елементів.

Ці позитивні аспекти говорять про те, що сайт студентського гуртка з DevOps програмування вже має деякі оптимізовані елементи, які сприяють його коректній роботі та покращують взаємодію з користувачами та пошуковими системами.

Таблиця 3.1 – Порівняння сайту студентського гуртка з DevOps програмування з аналогами

Критерії	Студентський гурток з DevOps програмування	Факультет Електроніки та Комп'ютерних Технологій ЛНУ ім. Івана Франка	Студентський гурток "Сучасні Інформаційно-програми інновації" ІФНУНГ	Студентський науковий гурток "ІТ" БФК НУБіП України
Дизайн	Сучасний, естетичний	Стандартний, нейтральний	Простий, нефункціональний	Сучасний, стилізований
Наявність інформації про гурток	Зрозуміла, доступна	Присутня, але неповна	Обмежена, базова	Чітка, але неповна
Навігація	Інтуїтивна, зрозуміла	Відсутня	Проста, але функціональна	Повна, але може бути заплутаною
Контент	Різноманітний, цікавий	Стандартний, досить загальний	Орієнтований на освітній процес	Зручний, з доступом до різних розділів
Швидкодія	Швидкий, оптимізований	Повільний, можливі завантаження	Задовільний, залежить від навантаження	
Якість зображень та мультимедійного вмісту	Висока	Залежить від ресурсів сайту	Орієнтований на текстовий контент	Прийнятна, але можливі випадки низької якості
Інтерактивність	Висока	Мінімальна	Відсутня	Прийнятна, але може бути покращена
Функціональність	Розширений функціонал	Базовий	Базовий, функціональності обмежені	Базовий

Порівнюючи веб-сайт студентського гуртка з DevOps програмування з аналогами, можна відзначити його переваги, зокрема швидкість завантаження сторінок, доступність ключових функцій та зручність навігації. Високий рівень інтерактивності дозволяє користувачам активно взаємодіяти з контентом, зокрема через відеоуроки.

Загалом, веб-сайт студентського гуртка з DevOps програмування представляє собою ідеальне поєднання якості, зручності та інновацій, що робить його відмінним вибором для всіх, хто цінує якість та ефективність у вивченні DevOps програмування.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

На основі аналітичного підходу проведено порівняльний аналіз засобів на інструментів розробки веб – сайтів, що дозволило виділити необхідні підходи до розробки дизайну, макету та реалізувати веб –сайт студентського гуртка.

1. На основі аналітичного підходу проведено порівняльний аналіз наявних веб-сайтів студентських гуртків, що дозволило виділити елементи, необхідні для роботи даного типу сайтів.

2. На основі алгоритмічного підходу розроблено алгоритм формування блоків на сторінці, що дозволяє здійснювати завантаження сторінки з максимальною швидкістю та у зручному для користувача форматі.

3. З допомогою підходу до структурування елементів веб-сторінок, розроблено структуру веб-сторінок для сайту наукового гуртка, що дозволило оптимізувати процес завантаження елементів сторінки.

4. На основі підходів до тестування веб-інтерфейсів ,проведено тестування на різних типах пристроїв та розширень екрану, що дозволило переконатись у працездатності розробленої системи.

5. Порівняльний аналіз розробленого сайту продемонстрував перевагу над аналогами, враховуючи унікальність і нетиповість тематики студентського гуртка.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Батько, Ю. М. Розробка інформаційного web-ресурсу кафедри комп'ютерної інженерії [Текст] / Ю. М. Батько, О. Й. Піцун // Сучасні комп'ютерні інформаційні технології : матеріали V Всеукр. шк.-семінару молодих вчен. і студ. АСІТ'2015 [м. Тернопіль, 22-23 трав. 2015р.] / відп. за вип. М. П. Дивак. - Тернопіль : ТНЕУ, 2015. С. 83-85.
2. Ляцинський, П. Б. Модуль авторизації та аутентифікації користувачів системи автоматизованої мікроскопії [Текст] / П. Б. Ляцинський, П. Б. Ляцинський, О. Й. Піцун // Сучасні комп'ютерні інформаційні технології : матеріали VI Всеукр. шк.-семінару молодих вчен. і студ. АСІТ'2016 [м. Тернопіль, 20-21 трав. 2016 р.] / відп. за вип. М. П. Дивак. Тернопіль : ТНЕУ, 2016. - С. 129-130.
3. Програмування [Текст] : навч. посіб. / О. М. Березький, Ю. М. Батько, М. М. Касянчук [та ін.]. Тернопіль : ТНЕУ, 2018. 256 с.
4. Mozilla Developer Network (MDN Web Docs). (n.d.). *HTML: Hypertext Markup Language*. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTML>
5. Smashing Magazine. (n.d.). *Understanding CSS Grid: Creating A Grid Container*. Retrieved from <https://www.smashingmagazine.com/understanding-css-grid/>
6. World Wide Web Consortium (W3C). (n.d.). *HTML 5.2 Specification*. Retrieved from <https://www.w3.org/TR/html52/>
7. Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional.
8. Заяць В. М. Функційне програмування : Навч. посіб. для студ. вищ. навч. закл., що навч. за спец. "Програмне забезп. автоматиз. систем". Львів : Бескид Біт, 2003. 160 с.
9. Реклама і дизайн XXI сторіччя: освіта, культура, економіка : Зб. наук. пр. Київ, 2001. Т. 2. 272 с.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

10. Angara J., Prasad S., Sridevi G. Towards benchmarking user stories estimation with COSMIC function points-a case example of participant observatio. International journal of electrical and computer engineering (IJECE). 2018. T. 8, № 5. C. 3076. URL: <https://doi.org/10.11591/ijece.v8i5.pp3076-3083>
11. Asian Development Asian Development Bank. 2018 development effectiveness review. Asian Development Bank Institute, 2019.
12. Bello C. Genital ulcer diseases - at a glance. Nigerian medical practitioner. 2005. T. 47, № 3. URL: <https://doi.org/10.4314/nmp.v47i3.28747>
13. Bouazza A., Van Impe W. F. Liner design for waste disposal sites. Environmental geology. 1998. T. 35, № 1. C. 41–54. URL: <https://doi.org/10.1007/s002540050291>
14. Coss P. C. Lectionary stories for preaching and teaching, cycle a - lent / easter edition. CSS Publishing, 2014. 68 c.
15. Forsgren N. DevOps delivers. Communications of the ACM. 2018. T. 61, № 4. C. 32–33. URL: <https://doi.org/10.1145/3174799>
16. Halstenberg J., Pfitzinger B., Jestädt T. DevOps. Wiesbaden : Springer Fachmedien Wiesbaden, 2020. URL: <https://doi.org/10.1007/978-3-658-31405-7>
17. Galup S., Dattero R., Quan J. What do agile, lean, and ITIL mean to DevOps?. Communications of the ACM. 2020. T. 63, № 10. C. 48–53. URL: <https://doi.org/10.1145/3372114>
18. Kaur G. Css: what is CSS?. Independently Published, 2022.
19. Lopuck L. Web design. Wiley & Sons, Incorporated, John, 2012.
20. MacDonald M. Creating web sites. O'Reilly Media, Incorporated, 2005.
21. Materials T. C. 1001 great web sites for teachers (web sites). KCP Publications Ltd. 48 c.
22. P P J. Exploring devops: challenges and benefits. Journal of information technology and digital world. 2019. T. 01, № 01. C. 27–37. URL: <https://doi.org/10.36548/jitdw.2019.1.004>

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

23. Service Provider DevOps / W. John та ін. IEEE communications magazine. 2017. Т. 55, № 1. С. 204–211. URL: <https://doi.org/10.1109/mcom.2017.1500803cm>
24. Starr J. Design considerations for multilingual web sites. Information technology and libraries. 2005. Т. 24, № 3. С. 107. URL: <https://doi.org/10.6017/ital.v24i3.3370>
25. Veglis A. Learn to design attractive web sites. IEEE distributed systems online. 2007. Т. 8, № 11. С. 6. URL: <https://doi.org/10.1109/mdso.2007.62>
26. Fitzgerald, D. (2018). *Learning DevOps: Continuously Deliver Better Software*. Packt Publishing.
27. Lodato, M. (2020). *UX/UI Design 2020: The New Trend For Creating Stunning Websites and Apps*. Independently Published.
28. Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Wiley.
29. Forsgren, N., Humble, J., Kim, G., & Kersten, N. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press.
30. Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional.
31. Riungu-Kalliosaari, L., Mäkinen, S., & Tyrväinen, P. (2016). "DevOps adoption benefits and challenges in practice: A case study". Information Systems Research Seminar in Scandinavia.
32. A List Apart. (n.d.). Articles on Web Design and Development. Retrieved from <https://alistapart.com/articles/>
33. CSS Tricks. (n.d.). A Complete Guide to Flexbox. Retrieved from <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
34. SitePoint. (n.d.). Web Development and Design Tutorials, Tips and Reviews. Retrieved from <https://www.sitepoint.com/>
35. Jenkins Documentation. (n.d.). Jenkins: The Leading Open Source Automation Server. Retrieved from <https://www.jenkins.io/doc/>

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

36. Kubernetes Documentation. (n.d.). Production-Grade Container Orchestration. Retrieved from <https://kubernetes.io/docs/>
37. React Documentation. (n.d.). A JavaScript library for building user interfaces. Retrieved from <https://reactjs.org/docs/getting-started.html>
38. Мотрук Г. Ю. Веб-інтерфейс студентського гуртка «Devops». ІХ Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 21 травня 2024 р. Тернопіль. Україна. 7 с.
39. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп'ютерних систем»/ Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.
40. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А

Техніко-економічне обґрунтування розробки проекту

А.1 Розрахунок витрат на розробку проекту

У процесі розробки будь-якого проекту невід'ємною складовою є розрахунок витрат, що визначає економічну ефективність та фінансову прийнятність задуманої ініціативи. В даному розділі вивчаються ключові аспекти техніко-економічного обґрунтування розробки проекту, де кожен крок аналізу витрат є необхідним етапом для забезпечення найбільш оптимального та ефективного використання ресурсів.

Розроблений веб-інтерфейс модуля обліку абітурієнтів кафедри комп'ютерної інженерії спрямований на ефективне управління процесом вступу. Алгоритм розробки не вимагає складних обчислень, що забезпечує високу швидкість реалізації. Це дозволяє отримувати необхідні дані із мінімальними затратами часу, підвищуючи продуктивність системи. Розроблений веб-інтерфейс забезпечує зручний та інтуїтивно зрозумілий доступ до інформації для всіх учасників системи вступу.

Отже, першим етапом є розрахунок витрат на розробку на розробку і впровадження програмних засобів, які включають:

$$K = K_1 + K_2, \tag{A.1}$$

де K_1 – витрати на розробку програмних засобів, грн.;

K_2 – витрати на відлагодження і дослідну експлуатацію програми вирішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці;
- відрахування у спеціальні державні фонди;
- матеріальні витрати;

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

- витрати на електроенергію;
- амортизація основних фондів;
- транспортні витрати;
- накладні витрати;
- інші витрати.

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{\text{оп}} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (\text{A.2})$$

де n_{ij} – чисельність спеціалістів i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту спеціалістом i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн., яка визначається за формулою:

$$C_{ij} = \frac{c_{ij}^0 (1 + h)}{PЧ_i}, \quad (\text{A.3})$$

де h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$ – місячний фонд робочого часу спеціаліста i -ої спеціальності j -го тарифного розряду, год. (за даними відділу бухгалтерії).

Отже:

$$C_{ij} = \frac{1980}{168} = 11,78 \text{ грн}$$

$$B_{\text{оп}} = 1 \cdot 160 \cdot 11,78 = 1\,884,8 \text{ грн}$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

Результати розрахунку записуємо до таблиці А.1.

Таблиця А.1 – Витрати на оплату праці

№	Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
1	Керівник ДП, доцент	16	650	10 400
2	Консультант техніко-економічного розділу, доцент	2	650	1 300
3	Студент	144	11,78	1696,32
Разом				13 396,32

Відрахування до спеціальних державних фондів визначаються у відсотковому відношенні до суми основної та додаткової заробітних плат. Згідно із чинною нормативною базою, обсяг відрахувань до спеціальних державних фондів становить 20,5% від суми заробітної плати:

$$V_{\Phi} = k \cdot V_{\text{оп}}, \quad (\text{A.4})$$

і буде дорівнювати:

$$V_{\Phi} = \frac{20,5}{100} \cdot 11\,700 = 2\,398,5 \text{ грн}$$

Далі потрібно розрахувати витрати на матеріали та комплектуючі, всі витрати буде зазначено в таблиці А.2.

Таблиця А.2 – Розрахунок витрат на матеріали та комплектуючі

№	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Папір (формат А4)	Уп	250	2	500	50	550
2	Картридж	Шт	379	1	379	37,9	416,9
3	Олівець	1 шт	11	1	11	1,1	12,1
4	Формат паперу А3 для графічної частини	2 шт	10	2	20	2,0	22,00
5	Ручка	1 шт	44	1	44	4,4	48,4
Разом							1049,4

Під час розробки використовувалось програмне забезпечення/сервіси, які також необхідно включити до розрахунків та зазначити у таблиці.

Таблиця А.3 – Придбання ліцензії

№	Програмне забезпечення/сервіс	Ліцензія
1	Figma	12 US\$/міс.
2	Adobe Creative Cloud	35,99 US\$/міс.
3	Shutterstock	9,99 US\$/міс.
4	Pixabay	Безкоштовно
Разом		2203,24

Далі, важливо розглянути накладні витрати проектних організацій, які включають три групи видатків: витрати на управління, загальногосподарські витрати та невиробничі витрати. Розрахунок цих витрат проводиться відсотково від витрат на оплату праці, приймаючи за середньостатистичний відсоток накладних витрат значення 0,7 від заробітної плати, отже:

$$H = 0,7 \cdot 13\,396,32 = 9\,377,42 \text{ грн.}$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Додатково важливо врахувати інші витрати, які не були узяті до уваги в попередніх статтях. Ці витрати складають 10% від загальної суми заробітної плати:

$$I = 0,1 \cdot 13\,396,32 = 1\,339,63 \text{ грн.}$$

Отже, витрати на розробку програмного забезпечення складають:

$$K_1 = V_{\text{ОП}} + V_{\text{Ф}} + V_{\text{КВ}} + I, \quad (\text{A.5})$$

$$K_1 = 1\,884,8 + 2\,398,5 + 1\,049,4 + 1\,339,63 = 6\,672,33 \text{ грн.}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{\text{м.г.}} \cdot t_{\text{від}}, \quad (\text{A.6})$$

де $S_{\text{м.г.}}$ – вартість однієї машино-години роботи ПК, грн./год (приймаємо б);

$t_{\text{від}}$ – час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Сумарна кількість робочих днів на комп'ютері складає 25 днів, середня тривалість робочого часу на комп'ютері – 1 година, і вартість години роботи комп'ютера становить 6 грн. Звідси можна розрахувати витрати на відлагодження та експлуатацію:

$$K_2 = 6 \cdot 25 = 150 \text{ грн.}$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки розробка проекту включає застосування засобів обчислювальної техніки, то розрахунок витрат на електроенергію буде занесено у таблицю А.4.

Таблиця А.4 – Витрати на електроенергію

Назва устаткування	Паспортна потужність, кВт	Коефіцієнт використання потужності	Час роботи обладнання, год	Ціна електроенергії, Грн/ кВт*год	Сума, грн
Ноутбук	0,51	0,8	144	2,64	155,11
Разом витрати на електроенергію					155,11

Загальна сума витрат на електроенергію розраховується за формулою:

$$V_E = \sum_{i=1}^n P_i \cdot k_i \cdot T_i \cdot C, \quad (A.7)$$

де P_i - паспортна потужність і-го електрообладнання, кВт;

k_i - коефіцієнт використання потужності і-го електрообладнання (приймається 0.7...0.9);

T_i - час роботи і-го устаткування за весь період розробки, год;

C - ціна електроенергії, грн / кВт*год;

i - тип електрообладнання;

n - кількість електрообладнання.

Отже:

$$V_E = 0,51 \cdot 0,8 \cdot 144 \cdot 2,64 = 155,11$$

Необхідно також провести розрахунок амортизації. В рамках статті “Амортизація основних фондів” враховується сума амортизаційних відрахувань від вартості обладнання та приладів, які використовуються під час

розробки КС. Розрахунок амортизаційних відрахувань здійснюється за визначеною формулою:

$$B_{\text{АФ}} = \sum_{i=1}^n \frac{B_i \cdot H_i \cdot T_i}{100 \cdot T_{\text{ЕФІ}}}, \quad (\text{А.8})$$

де B_i – вартість i -го устаткування, грн.;

H_i – річна норма амортизації i -го устаткування, %;

T_i – час роботи i -го устаткування за весь період розробки, год.;

$T_{\text{ЕФІ}}$ – ефективний фонд часу роботи i -го устаткування за рік, год / рік;

i – тип устаткування;

n – кількість устаткування.

$$B_{\text{АФ}} = \frac{33000 \cdot 30 \cdot 144}{100 \cdot 2000} = 712,8$$

Усі розрахунки занесено у таблицю А.5.

Таблиця А.5 – Амортизація основних фондів

Найменування устаткування	Вартість устаткування, грн.	Річна норма амортизації, %	Ефективний фонд часу роботи обладнання, год / рік	Час роботи обладнання для розробки КС, год	Сума, грн.
Ноутбук	33000	30	2000	144	712,8
Разом					712,8

На наступному етапі важливо визначити транспортні витрати. Прогнозується, що транспортні витрати становитимуть від 8% до 12% від загальної суми матеріальних витрат:

$$B_{\text{Т}} = (0,08 \dots 0,12) \cdot B_{\text{М}}, \quad (\text{А.9})$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

В даному випадку:

$$B_T = 0,09 \cdot (1\,049,4 + 2\,203,24) = 292,74$$

На основі представлених даних можна сформулювати кошторис витрат на розробку програмного забезпечення записавши всі дані у таблицю А.6.

Таблиця А.6 – Кошторис витрат на розробку програмного забезпечення

№	Найменування витрат	Сума витрат, грн
1	Витрати на оплату праці	13 396,32
2	Витрати на електроенергію	155,11
3	Відрахування у спеціальні державні фонди	2 398,5
4	Матеріальні витрати	10 049,4
5	Амортизаційні витрати	712,8
6	Транспортні витрати	292,74
7	Накладні витрати	2 203,24
8	Інші витрати	1 339,63
Разом		30 547,74

Результати аналізу витрат на розробку програмного забезпечення свідчать про його економічну доцільність та перспективність. Обґрунтованість проекту підтверджується докладним розглядом різних складових витрат, таких як оплата праці, відрахування у державні фонди, матеріальні ресурси та інші, що узагальнено складають 30 547,74 грн. З позиції користувачів, цей продукт готовий вирішити їхні потреби, надати зручності та стати корисним завдяки своїй функціональності.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

А.2 Розрахунок можливої ціни проекту

У цьому розділі розглядаються аспекти розрахунку можливої ціни проекту. Досліджуються ключові фактори та складові, які впливають на формування вартості, а саме:

- прогнозована ціна КС;
- економічна ефективність розробки проекту;
- термін окупності.

Величина прогнозованої (договірної) ціни КС повинна враховувати ефективність, якість та терміни виконання на рівні, що відповідає економічним інтересам як замовника (споживача), так і виконавця (підрядника). Договірна ціна для прикладних КС обчислюється за визначеною формулою:

$$Ц_{д} = В_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (A.10)$$

де $В_{КС}$ - витрати на розробку КС, грн.;

p - середній рівень рентабельності КС, % (приймається в розмірі 20-30%). Розраховану можливу ціну КС варто порівняти із цінами на аналогічні розробки, що існують на ринку ІТ продуктів.

Зважаючи на вищеописане, можна розрахувати ціну програмного засобу:

$$Ц_{д} = 30\,547,74 \cdot \left(1 + \frac{30}{100}\right) = 39\,712,01$$

Для побудови таблиці показників економічної ефективності розробки програмного продукту потрібно виконати розрахунки необхідних показників.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Розрахунок економічного ефекту у галузі проектування проектного рішення проводиться за визначеною формулою:

$$E_P = \frac{\Pi}{B_{КС}}, \quad (A.11)$$

$$E_P = \frac{22\,000}{30\,547,74} = 0,72$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень:

$$T_P = \frac{1}{E_P}, \quad (A.12)$$

$$T_P = \frac{1}{0,72} = 1,3$$

Отримані економічні показники з розрахунків проекту буде занесено в таблицю А.7.

Таблиця А.7 – Економічні показники розробки проекту

Показник	Значення
Собівартість, грн.	30 547,74
Плановий прибуток, грн.	22 000
Ціна, грн.	39 712,01
Економічна ефективність	0,72
Термін окупності, рік	1,3

Аналіз техніко-економічних показників, пов'язаних із розробкою програмного забезпечення, дозволяє зробити висновок про його економічну

обґрунтованість та потенційну вигоду для учасників проекту. Нижче наведено ключові аргументи, які підкреслюють позитивні аспекти проекту:

Ціна продукту: Вартість продукту, що становить 39 712,01 грн, перевищує витрати на його створення (собівартість). Це свідчить про готовність ринку сплачувати за цей продукт і можливість генерування прибутку з його реалізації.

Економічна ефективність: Значення економічної ефективності становить 0,72, що вказує на співвідношення прибутку до витрат. Це позитивний сигнал, оскільки проект приносить прибуток, який перевищує затрати на його реалізацію.

Термін окупності: Термін окупності проекту складає 1,3 років, що свідчить про те, що інвестиції в проект повернуться протягом короткого періоду, конкретно на другий рік його реалізації.

З урахуванням вищезазначених факторів можна зробити висновок про економічну обґрунтованість та вигідність розробки програмного забезпечення. Цей проект має потенціал приносити прибуток і стати корисним як для студента, який його розробляє, так і для потенційних користувачів, для яких продукт може вирішити їх потреби та стати корисним завдяки його функціональності. Отже, з позиції економічного аналізу, проект є обґрунтованим та перспективним.

А.3 Розрахунок показників економічної ефективності

З метою створення таблиці, яка відображатиме показники економічної ефективності розробки програмного продукту, буде проведено розрахунки необхідних показників.

Перший етап – розрахунок трудомісткості розробки ПЗ, яку розраховують за формулою:

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_d, \quad (A.13)$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50);

t_u – витрати праці на дослідження алгоритму рішення задачі;

t_a – витрати праці на розробку блок-схеми алгоритму;

t_n – витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ – витрати праці на налагодження програми на ПК;

t_d – витрати праці на підготовку документації.

Складові витрат праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється. Умовне число операторів (підпрограм) розраховують за формулою:

$$Q = q \cdot C \cdot (1 + p), \quad (A.14)$$

де q – передбачуване число операторів;

C – коефіцієнт складності програми;

p – коефіцієнт кореляції програми в ході її розробки.

$$Q = 1000 \cdot 1,1 \cdot (1 + 0,05) = 1\,155$$

Витрати праці на вивчення опису задачі визначаються з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \quad (A.15)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

$$t_u = \frac{1\,155 \cdot 15}{80 \cdot 0,8} = 270,7$$

Витрати праці на розробку алгоритму рішення задачі розраховують за формулою:

$$t_a = \frac{q}{(20 \dots 25) \cdot k}, \quad (\text{A.16})$$

$$t_a = \frac{1\,155}{20 \cdot 0,8} = 72,19$$

Витрати на складання програми по готовій блок-схемі розраховують за формулою:

$$t_n = \frac{q}{(20 \dots 25) \cdot k}, \quad (\text{A.17})$$

$$t_n = \frac{1\,155}{25 \cdot 0,8} = 57,75$$

Витрати праці на налагодження програми на ПК розраховують:
- за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{q}{(4 \dots 5) \cdot k}, \quad (\text{A.18})$$

$$t_{\text{отл}} = \frac{1\,155}{4 \cdot 0,8} = 360,94$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

- за умови комплексного налагодження завдання:

$$t_{\text{отл}}^k = 1.5 \cdot t_{\text{отл}}, \quad (\text{A.19})$$

$$t_{\text{отл}}^k = 1.5 \cdot 360,94 = 541,41$$

Витрати праці на підготовку документації розраховують за формулою:

$$t_{\text{д}} = t_{\text{др}} + t_{\text{до}}, \quad (\text{A.20})$$

де $t_{\text{др}}$ – трудомісткість підготовки матеріалів і рукопису, що розраховують за формулою:

$$t_{\text{др}} = \frac{q}{(15..20) \cdot k}, \quad (\text{A.21})$$

$$t_{\text{др}} = \frac{1\ 155}{20 \cdot 0,8} = 72,19$$

$t_{\text{до}}$ – трудомісткість редагування, друкування й оформлення документації:

$$t_{\text{до}} = 0.75 \cdot t_{\text{др}}, \quad (\text{A.22})$$

$$t_{\text{до}} = 0,75 \cdot 72,19 = 54,14$$

$$t_{\text{д}} = 72,19 + 54,14 = 126,33$$

Після проведених розрахунків тепер можна розрахувати трудомісткість:

$$t = 50 + 270,7 + 72,19 + 57,75 + 360,94 + 126,33 = 937,91$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступний етап – розрахунок середньогодинної оплати програміста. Для цього перш за все необхідно визначити його річний фонд грошового забезпечення, враховуючи місячне грошове винагородження, яке становить приблизно 1980,00 гривень. Отже, річний фонд грошового забезпечення складає 23 760 гривень. Далі розраховуємо кількість робочих годин у році за відповідною формулою:

$$N_p = (N - N_{\Pi} - N_B) \cdot 8, \quad (\text{A.23})$$

де N – загальна кількість днів у році,

N_{Π} – кількість святкових днів у році,

N_B – кількість вихідних днів у році.

Приймається, що кількість святкових днів у році – 14, а вихідних – 104.

$$N_p = (365 - 14 - 104) \cdot 8 = 1\,976$$

Середньогодинна оплата праці програміста обчислюється за наступною формулою:

$$C_{\Pi} = \frac{\Phi_p}{N_p}, \quad (\text{A.24})$$

де Φ_p – річний фонд грошового забезпечення.

$$C_{\Pi} = \frac{23\,760}{1\,976} = 12.02$$

Витрати на оплату праці розробників програми становлять:

$$B_{\text{ОП}} = C_{\Pi} \cdot T_3, \quad (\text{A.25})$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

де T_3 – загальна кількість годин роботи програміста над проектом.

$$V_{0П} = 12,02 \cdot 144 = 1\,730,88$$

Витрати, пов'язані з розробкою програми на ПК розраховуються:

$$V_{ПК} = T_{ПК} \cdot C_{ПК}, \quad (A.26)$$

де $T_{ПК}$ – час використання ПК для розробки програми,

$C_{ПК}$ – собівартість машинного часу обчислювальної техніки (розраховує бухгалтерія підприємства).

Собівартість однієї години роботи ПК дорівнює:

$$C_{ПК} = \frac{V_{ПК}}{\Phi_{ПК}}, \quad (A.27)$$

де $V_{ПК}$ - річні поточні витрати на експлуатацію ПК,

$\Phi_{ПК}$ річний фонд часу корисної роботи ПК.

$$C_{ПК} = \frac{610,43}{1\,868} = 0,38$$

$$V_{ПК} = 144 \cdot 0,38 = 54,72$$

Визначивши фактичний річний обсяг роботи ПК у годинах, буде отримано можливість оцінити собівартість годин машинного часу. Фактичний річний фонд часу ПК становить:

$$\Phi_{ПК} = N_p - (\Phi_m + \Phi_{річ}), \quad (A.28)$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

Φ_M - місячний фонд часу на профілактику і ремонт ПК (час профілактики щомісячно – 5 годин),

$\Phi_{річ}$ - річний фонд часу на профілактику і ремонт ПК (час профілактики щорічно – 6 діб)

$$\Phi_{ПК} = 1\,976 - (60 + 48) = 1\,868$$

Річні поточні витрати на експлуатацію програмного забезпечення розраховуються за допомогою формули:

$$V_{ПКр} = V_{Ер} + V_{Ар} + V_{РЕМр} + V_{ДКр} + V_{Ір}, \quad (A.29)$$

де $V_{Ар}$ – річні відрахування на амортизацію,

$V_{Ер}$ – річні витрати на електроенергію для ПК,

$V_{РЕМр}$ – річні витрати на ремонт ПК,

$V_{ДКр}$ – річні витрати на додаткові комплектуючі ПК,

$V_{Ір}$ – інші витрати.

Суму річних амортизаційних відрахувань визначають за такою формулою:

$$V_{Ар} = C_{ПК} \cdot N_A, \quad (A.30)$$

де $C_{ПК}$ – балансова вартість ПК,

N_A – норма амортизаційних відрахувань (дорівнює 15% у квартал).

Для розрахунку балансової вартості ПК використовується наступна формула:

$$C_{ПК} = C_p \cdot (1 + K_{УН}), \quad (A.31)$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

де C_p – ринкова вартість ПК,

K_{yH} – коефіцієнт, що враховує витрати на установку й налагодження ПК (приймається рівним 12%).

Витрати на електроенергію, що споживає ПК, розраховуються відповідно до формули:

$$B_{Ep} = P_{ПК} \cdot \Phi_{ПК} \cdot C_E \cdot K_{IB}, \quad (A.32)$$

де $P_{ПК}$ – паспортна потужність ПК,

$\Phi_{ПК}$ – річний фонд корисного часу роботи ПК,

C_E – вартість 1 кВт/год електроенергії,

K_{IB} – коефіцієнт інтенсивного використання ПК (0,7 - 1).

Таким чином, розрахункове значення витрат на електроенергію, що споживає ПК, складає:

– витрати на поточний і профілактичний ремонт (приймаються рівними 6% від вартості ПК):

$$B_{PEM_p} = C_{ПК} \cdot 0,06, \quad (A.33)$$

- витрати на додаткові комплектуючі – витрати необхідні для забезпечення експлуатації ПК (приймаються рівними 2% від вартості ПК):

$$B_{DK_p} = C_{ПК} \cdot 0,02, \quad (A.34)$$

- інші витрати, тобто непрямі витрати пов'язані з експлуатацією ПК (приймаються рівними 5-10% від вартості ПК):

$$B_{Ip} = C_{ПК} \cdot 0,05, \quad (A.35)$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

Отже:

$$B_{A_p} = 36\,960 \cdot 0,15 = 5\,544$$

$$C_{ПК} = 33\,000 \cdot (1 + 0,12) = 36\,960$$

Витрати на електроенергію розраховувались раніше, тому $B_{E_p} = 155,11$,

$$B_{РЕМ_p} = 33\,000 \cdot 0,06 = 1\,980$$

$$B_{ДК_p} = 33\,000 \cdot 0,02 = 660$$

$$B_{I_p} = 33\,000 \cdot 0,05 = 1\,650$$

$$B_{ПК_p} = 155,11 + 5\,544 + 1\,980 + 660 + 1\,650 = 9\,989,11$$

Під час розробки програмного комплексу на ПК використовуються наступні етапи програмування:

- написання програми за готовою схемою алгоритму;
- налагодження програми на ПК;
- підготовка документації по задачі.

Таким чином, витрати машинного часу склали ($t_{маш}$):

$$t_{маш} = t_n + t_{отл}^k t_d, \quad (A.36)$$

$$t_{маш} = 57,75 + 360,94 + 126,33 = 545,02$$

Витрати на оплату машинного часу розраховуємо за формулою:

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

$$B_{\text{маш}} = t_{\text{маш}} \cdot C_{\text{ПК}}, \quad (\text{A.37})$$

$$B_{\text{маш}} = 545,02 \cdot 0,38 = 207,11$$

Загальні витрати на розробку програмного комплексу складають:

$$B_{\text{заг}} = B_{\text{оп}} + B_{\text{маш}}, \quad (\text{A.38})$$

$$B_{\text{заг}} = 2\,187,64 + 207,11 = 2\,394,75$$

Показник інвестиційних вкладень з урахуванням інфляційних процесів обчислюється за формулою:

$$K_i = \varphi_i \cdot R_i, \quad (\text{A.39})$$

де φ_i - коефіцієнт інфляції на поточний період;

R_i - інвестиційні платежі в i -му періоді (капітальні вкладення).

$$K_i = 100,8\% \cdot 30\,547,74 = 30\,792,12$$

Дохід від розробки ПЗ у i -му періоді розраховуємо за формулою:

$$D_i = J_i \cdot (B_i - C_i), \quad (\text{A.40})$$

де B_i - ціна продажу програмного продукту в i -му періоді;

C_i - собівартість програмного продукту (фактично дорівнює сумі витрат на розробку ПЗ);

J_i - кількість ПЗ.

$$D_i = 1 \cdot (39\,712,01 - 30\,547,74) = 9\,164,33$$

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

Вартість продажу розробленого продукту розраховують за формулою:

$$B_i = B_{\text{зар}} \cdot \left(1 + \frac{p}{100}\right), \quad (\text{A.41})$$

де p - середній рівень рентабельності на поточний період.

$$B_i = 2\,394,75 \cdot \left(1 + \frac{0,151}{100}\right) = 2\,398,37$$

Економічна ефективність полягає у відношенні результату від розробленого програмного продукту до затрачених ресурсів:

$$E = \frac{D_i}{B_{\text{зар}}}, \quad (\text{A.42})$$

$$E = \frac{9\,164,33}{2\,394,75} = 3,83$$

Тоді термін окупності можна розрахувати за такою формулою:

$$T = \frac{1}{E}, \quad (\text{A.43})$$

$$T = \frac{1}{3,83} = 0,26$$

Таблиця А.8 – Показники економічної ефективності проектного рішення

№	Найменування	Значення показників	
		Аналог	Новий варіант
1	Капітальні вкладення	-	30 792,12

Продовження таблиці А.8

2	Ціна придбання	38 000	39 712,01
3	Річні експлуатаційні витрати	1 0000	9989,11
4	Дохід від розробки		22 000
5	Термін окупності проекту	1,3	

Розглядаючи порівняльну таблицю економічної ефективності, можна відзначити, що витрати на капітальні вкладення у новий проект становлять 30 792,12 гривень, що свідчить про потребу в значних інвестиціях порівняно з подібними проектами. Тим не менше, вартість придбання нового продукту складає 39 712,01 гривень, перевищуючи вартість аналогічного продукту на ринку (38 000 гривень).

Річні експлуатаційні витрати для нового продукту оцінюються на рівні 9 989,11 гривень, що менше, ніж аналогічні витрати для вже існуючих рішень (10 000 гривень). Це може свідчити про більш ефективну експлуатацію нового продукту. На жаль, дохід від розробки нового продукту залишається конфіденційним, але різниця в ціні придбання дозволяє припускати, що новий продукт має потенціал для генерації вищого доходу, який може компенсувати витрати.

Термін окупності проекту для нового продукту становить 1,3 роки, свідчаючи про швидке відновлення інвестицій та можливу рентабельність протягом короткого періоду. Узагальнюючи, новий проект має вагомні технічні та економічні переваги, хоча ефективність його капітальних вкладень варто розглядати в контексті доходів та конкретних обставин впровадження.

Розрахунки показують, що розробка програмного забезпечення передбачає витрати трудових ресурсів у розмірі 937,91 годин. Умовне число операторів становить 1,155, вказуючи на середню складність програми. Середньогодинна оплата праці програміста складає 12,02 гривень, а загальні витрати на оплату праці розробників складають 1 730,88 гривень. Витрати,

пов'язані з розробкою програми на ПК, оцінюються на рівні 54,72 годин, враховуючи час використання ПК та собівартість машинного часу.

Отримана інформація надає можливість провести докладний аналіз економічних аспектів розробки програмного продукту. Річні витрати на експлуатацію програмного забезпечення оцінюються в 9 989,11 гривень, включаючи витрати на амортизацію, електроенергію, ремонт, комплектуючі та інші витрати.

Амортизаційні відрахування становлять 5 544 гривень, розраховані на основі балансової вартості ПК та норми амортизаційних відрахувань. Загальні витрати на розробку програми на ПК, включаючи оплату машинного часу, складають 2 394,75 години.

При розрахунку показників економічної ефективності враховано інфляційні процеси. Інвестиційні вкладення становлять 30 792,12 гривень. Дохід від розробки програмного продукту в період складає 9 164,33 гривень, а вартість продажу програми – 2 398,37 гривень. Показник економічної ефективності становить 3,83, що свідчить про позитивну рентабельність проекту. Термін окупності розрахований як 0,26 року, вказуючи на швидку віддачу інвестицій у розробку програмного продукту. З урахуванням усіх визначених показників, можна вважати розробку програми на ПК економічно вигідною та перспективною інвестицією.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК Б
Світлокопія публікації

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ



ІХ НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ
МОЛОДИХ ВЧЕНИХ І СТУДЕНТІВ
«ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА
МЕРЕЖІ»

21 ТРАВНЯ 2024



KI.WUNU.EDU.UA/CONFERENCE/

ТЕРНОПІЛЬ

2024



					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

<i>Мотрук Г.Ю.</i> Вебінтерфейс студентського гуртка «DevOps»	7
<i>Матіяш Ю.Р.</i> Розробка DevOps пайплайну для програми обробки зображень	8
<i>Березький М. О.</i> Аналіз метрик порівняння зображень	9
<i>Рутецький Ю.О.</i> Інформаційна система оптимізації витрат енергоносіїв розумного будинку	10
<i>Медвідь А. Я.</i> Маніпуляція об'єктами з використанням роботуки в реальному середовищі на прикладі відкриття дверей вбиралень	11
<i>Теслюк С. В.</i> Моделі та засоби ІТ-рекрутингу з використанням методів багатокритеріальної оптимізації	13
<i>Боровий О.О., Лечко Р.І.</i> Засоби забезпечення автономності БПЛА з використанням методів комп'ютерного зору	14
<i>Лечко Р.І., Боровий О.О.</i> Методи квантування нейромереж для платформ БПЛА.....	15
<i>Варвянський А.О</i> Онлайн платформа для потокового відео на основі технології React	16
<i>Чирська Т. В.</i> Алгоритми порівняння скелетонів	17
<i>Сич Т. А.</i> Аналіз програмних засобів для генерації тексту в текст	18
<i>Бадзь В. М.</i> Розроблення моделі визначення авторства англійських текстів з використанням згорткових нейронних мереж.....	19
<i>Каштальян О.В.</i> Побудова інформаційної системи віртуальної примірки прикрас	20
<i>Рудик В.В.</i> Вебінтерфейс модуля обліку абітурієнтів кафедри комп'ютерної інженерії	21
<i>Копія А.В.</i> Аналіз методів і алгоритмів тестування та діагностики комп'ютерної техніки	22
<i>Мацюк М. І.</i> Модель управління крокуючими платформами на базі штучних нейронних мереж.....	23
<i>Коваль А. В.</i> Методи і засоби оптимізації ресурсів у хмарних системах з мікросервісною архітектурою	24
<i>Глод С. І.</i> Інтегрування можливостей Edge AI через використання Edge-enabled безпілотників у військові операції	25
<i>Андрухі Б., Плекан В.</i> Моделі та технології штучних нейронних мереж для програмних додатків з елементами штучного інтелекту	27

Інтелектуальні комп'ютерні системи та мережі, Тернопіль, 21 травня 2024 р.

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк. 78
Змн.	Арк.	№ докум.	Підпис	Дата		

ВЕБІНТЕРФЕЙС СТУДЕНТСЬКОГО ГУРТКА «DEVOPS»

Вступ. Сучасні тенденції у розвитку технологій вимагають від нас постійного оновлення підходів у навчанні та освоєнні нових знань. Однією з ключових галузей, яка відчуває вплив цього технологічного прогресу, є освіта. Розробка веб-сайту для студентського гуртка з DevOps програмування стає важливим кроком у напрямку інтеграції новітніх технологій у навчальний процес та підтримки розвитку ІТ-індустрії. Приклади розробки графічних інтерфейсів складних систем наведено у роботах [1,2].

Постановка задачі. Об'єкт дослідження – засоби проектування UX/UI дизайну для веб-сайту студентського гуртка. Предмет дослідження – шаблон веб-сайту студентського наукового гуртка. Мета розробки веб-сайту полягає в створенні інтерактивної платформи для студентів, яка б забезпечувала їм можливість реєстрації на курси та отримання доступу до актуальних новин у галузі DevOps програмування.

Основний матеріал. Розробка веб-сайту для студентського гуртка з DevOps програмування передбачає створення платформи, яка надасть студентам можливість реєстрації на курси та отримання оновлень і новин. Веб-сайт буде служити інформаційним центром, де викладач зможе ділитися досягненнями студентів, описувати проекти, в яких вони беруть участь, а також спільними ініціативами та досягненнями гуртка. Такий підхід сприятиме залученню більшої кількості студентів у гурток і сприятиме їхньому активному участю у ньому.

На початковому етапі взаємодії з веб-сайтом студентського гуртка DevOps, користувач вводить URL, і в цей момент розпочинається складний та продуманий алгоритм формування веб-сторінки. Особлива увага приділяється стилізації та оформленню блоків за допомогою CSS, де кожен кольоровий акцент та елемент дизайну враховуються для забезпечення не лише функціональності, але і естетичності веб-сайту. Щоб створити динамічний та вражаючий веб-сайт, використовується JavaScript. Елементи взаємодії, такі як інтерактивні меню чи новини, вбудовані таким чином, щоб не лише надати користувачам можливість швидкого доступу до інформації, але й поглибити їхнє зацікавлення у сфері DevOps.

Хедер, логотип та блок навігації, на початку процесу користувач витрачає мінімальний час на завантаження хедера, логотипу та блоку навігації. Ці елементи не лише формують основний елемент інтерфейсу, але й визначають структуру веб-сайту, надаючи зручний інструмент для навігації.

Окремий блок відведено для розробки меню та навігації, створення якого передбачає створення інтерактивних елементів для швидкого та легкого доступу до всіх розділів гуртка.

Висновки. Розробка веб-сайту для студентського гуртка з DevOps програмування є важливим кроком у напрямку створення інноваційного середовища для навчання. Цей проект сприятиме розвитку навичок та знань студентів у сфері ІТ, а також підготує їх до викликів сучасного ринку праці. Він дозволить молодому поколінню вільніше маневрувати у світі технологій та забезпечить їм конкурентну перевагу в майбутньому.

Список літератури

1. Піцун О.Й. Мікросервісна архітектура системи опрацювання імуногістохімічних зображень / О.Й. Піцун, К.М. Пришляк // Вісник Хмельницького національного університету Серія: «Технічні науки» №3, 2023 . 321(3) . С. 166-174
2. O. Pitsun, G. Melnyk and Y. Batko, "Graphical interface of hybrid intelligent systems for biomedical imaging analysis," 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 2016, pp. 121-124,

					КР. КІ. 07125/20. 00.00.000 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		