

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
АВТОМАТИЗОВАНИХ СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА
на тему: «ДОСЛІДЖЕННЯ ТА РОЗРОБКА ІНТЕРАКТИВНИХ ВЕБ-
ДОДАТКІВ З ВИКОРИСТАННЯМ СУЧАСНИХ ФРЕЙМВОРКІВ ТА
БІБЛІОТЕК ДЛЯ ПІДВИЩЕННЯ РІВНЯ КОРИСТУВАЦЬКОГО
ДОСВІДУ»**

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Артем ГАРБЕЦЬКИЙ
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: Гарбецький Артем Михайлович
здобувач вищої освіти
група ІСДМ-62

Керівник: Олег СЕНЬКОВ
науковий ступінь, д.т.н., професор
вчене звання

Рецензент: _____
науковий ступінь, Ім'я, ПРІЗВИЩЕ
вчене звання

Київ 2023

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти Магістр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру ІЗАС

_____ Каміла СТОРЧАК

« _____ » _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Гарбецький Артем Михайлович

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: дослідження та розробка інтерактивних веб-додатків з використанням сучасних фреймворків та бібліотек для підвищення рівня користувацького досвіду.

керівник кваліфікаційної роботи Олег СЕНЬКОВ д.т.н., професор,
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, технічна документація HTML/CSS, JavaScript, вимоги до підвищення користувацького досвіду.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження принципу створення веб-додатків

Аналіз можливості використання веб-додатків для підвищення рівня користувацького досвіду

Розробка веб-додатку

5. Перелік графічного матеріалу: *презентація*

Використання інтерактивних веб-додатків з використанням сучасних фреймворків та бібліотек для підвищення рівня користувацького досвіду. Технічні аспекти розробки веб-додатків додатків для підвищення рівня користувацького досвіду. Бібліотеки та фреймворки для створення веб-додатків. Прототипування мобільного додатку CodeCube

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Вивчення теоретичних основ розробки і застосування веб-додатків для підвищення рівня користувацького досвіду	05.11-12.11.23	
3	Дослідження технічних аспектів реалізації веб-додатків	13.11-18.11.23	
4	Аналіз ефективності веб-додатків для підвищення рівня користувацького досвіду	19.11-23.11.23	
5	Розробка інтерфейсу та програмного забезпечення мобільного додатку CodeCube	24.11-03.12.23	
6	Прототипування веб-додатку CodeCube	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

_____ (підпис)

Артем ГАРБЕЦЬКИЙ

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

_____ (підпис)

Олег СЕНЬКОВ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 71 стор., 39 рис., 31 джерел.

Мета роботи – оцінка можливостей, переваг та обмежень сучасних фреймворків та бібліотек, що використовуються у розробці веб-додатків з акцентом на їхню інтерактивність.

Об'єкт дослідження – сучасні засоби програмування, такі як фреймворки та бібліотеки, що застосовуються для створення інтерактивних веб-додатків.

Предмет дослідження – процеси розробки та використанні інтерактивних веб-додатків, з особливим акцентом на використанні оновлених фреймворків та бібліотек.

Короткий зміст роботи: У роботі проведено дослідження ефективності мобільних додатків для підтримки фізичного здоров'я. сервісів хмарних вимог. Проаналізовано технічні аспекти і етапи розробки таких мобільних додатків. Розроблено прототип мобільного додатка Active Pulse та проведено його тестування на цільових групах користувачів.

КЛЮЧОВІ СЛОВА: ВЕБ-ДОДАТКИ, ФРЕЙМВОРКИ, БІБЛІОТЕКИ, FIGMA.

ABSTRACT

The text part of the qualifying work for obtaining a master's degree: 99 pages, 39 figures, 31 sources.

The purpose of the study is to evaluate the capabilities, advantages and limitations of modern frameworks and libraries used in the development of web applications with a focus on their interactivity.

Object of research - modern programming tools, such as frameworks and libraries, used to create interactive web applications.

The subject of research is the processes of developing and using interactive web applications, with a special emphasis on the use of updated frameworks and libraries.

Summary of the work: The paper investigates the effectiveness of mobile applications for maintaining physical health. services of cloud requirements. The technical aspects and stages of development of such mobile applications are analyzed. A prototype of the mobile application Active Pulse was developed and tested on target user groups.

KEYWORDS: WEB APPLICATIONS, FRAMEWORKS, LIBRARIES, FIGMA.

ЗМІСТ

ВСТУП.....	6
1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНТЕРАКТИВНИХ ВЕБ-ДОДАТКІВ З ВИКОРИСТАННЯМ СУЧАСНИХ ФРЕЙМВОРКІВ ТА БІБЛІОТЕК.....	8
1.1 Огляд сучасних фреймворків та бібліотек для розробки веб-додатків.....	8
1.2 Принципи підвищення користувацького досвіду.....	14
1.3 Теоретичні поняття веб-додатків.....	20
2 АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБКИ ІНТЕРАКТИВНИХ ВЕБ-ДОДАТКІВ З ВИКОРИСТАННЯМ СУЧАСНИХ ФРЕЙМВОРКІВ ТА БІБЛІОТЕК.....	25
2.1 Огляд існуючих інструментів для розробки веб-додатків.....	25
2.2 Метрики ефективності веб-додатків.....	36
2.3 Аналіз ефективності веб-додатків.....	38
3 ПРОТОТИПУВАННЯ ІНТЕРАКТИВНОГО ВЕБ-ДОДАТКУ ДЛЯ ПІДВИЩЕННЯ РІВНЯ КОРИСТУВАЦЬКОГО ДОСВІДУ.....	47
3.1 Визначення вимог інтерактивного веб-додатку для підвищення рівня користувацького досвіду.....	47
3.2 Проектування та дизайн прототипу.....	51
3.3 Програмне забезпечення для створення веб-додатку CodeCube....	56
3.4 Аналіз ефективності додатку CodeCube.....	62

ВСТУП

Актуальність теми: дослідження та розробка інтерактивних веб-додатків за допомогою останніх фреймворків та бібліотек - це не лише реакція на запит користувачів щодо зручності та швидкості, але й метод просування в конкурентному середовищі. Ці інструменти дають змогу створювати захопливі, гнучкі застосунки, які працюють як на комп'ютерах, так і на мобільних пристроях. Все це породжує необхідність постійного удосконалення та використання передових технологій для задоволення потреб користувачів.

Головною метою дослідження є оцінка можливостей, переваг та обмежень сучасних фреймворків та бібліотек, що використовуються у розробці веб-додатків з акцентом на їхню інтерактивність. Основне завдання полягає у проведенні аналізу цих технологій для вибору найефективніших інструментів з метою покращення користувацького досвіду у веб-додатках.

Завдання дослідження включають:

1. Аналіз фреймворків та бібліотек.
2. Оцінка продуктивності.
3. Вивчення впливу на UX.
4. Експерименти та тестування.
5. Вибір оптимальних рішень.

Об'єктом дослідження є сучасні засоби програмування, такі як фреймворки та бібліотеки, що застосовуються для створення інтерактивних веб-додатків. Ці інструменти є ключовими в напрямі розробки, оскільки вони дозволяють створювати веб-застосунки, що відповідають вимогам зручності, швидкості та якості для користувачів.

Предметом дослідження є процеси розробки та використанні інтерактивних веб-додатків, з особливим акцентом на використанні оновлених фреймворків та бібліотек.

Для досягнення мети використовуватимуться наступні *методи дослідження*:

1. Аналіз літератури та джерел.
2. Експерименти та тестування.
3. Опитування та інтерв'ю.
4. Компаративний аналіз.
5. Кодування та практична реалізація.

Апробація результатів магістерської роботи: Гарбецький А. М. «Інтерактивні веб-додатки для підвищення рівня користувацького досвіду». Тези доповіді на Всеукраїнській Науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і Світу». – Київ, 28 листопада 2023 р.

Публікації: Гарбецький А. М. «Сучасні фреймворки як спосіб покращення користувацького досвіду». Стаття у загальногалузевому науково-виробничому журналі «Зв'язок», м.Київ - №1, 2024. – С.220-228.

Науковою новизною є створення інноваційних інтерактивних веб-додатків, що враховують індивідуальні особливості користувачів та забезпечують персоналізовані можливості для підвищення рівня користувацького досвіду. *Практична значущість* полягає в можливості впровадження систем, що сприяють покращенню користувацької взаємодії та надають інноваційні рішення для підвищення якості використання веб-додатків та покращення загального користувацького досвіду.

1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНТЕРАКТИВНИХ ВЕБ-ДОДАТКІВ З ВИКОРИСТАННЯМ СУЧАСНИХ ФРЕЙМВОРКІВ ТА БІБЛІОТЕК

1.1 Огляд сучасних фреймворків та бібліотек для розробки веб-додатків

При розробці програмних продуктів, веб-сайтів та організації даних, завжди виникає необхідність спростити процес розробки. Як можна скористатися існуючим досвідом та досягненнями для більш ефективної роботи? Ця ситуація привела до виникнення поняття фреймворків, що сприяють вирішенню цих завдань. Давайте розглянемо це більш детально.

Фреймворки в мовах програмування – це набори інструментів, які включають компоненти, бібліотеки та інструментарій для вирішення конкретних завдань. Головна мета полягає у спрощенні розробки для програмістів, наданні готових рішень та підвищенні надійності коду.

Іншими словами, фреймворки – це засоби, які містять у собі шаблони, код та інші складові, спрямовані на ефективне створення програмного забезпечення. Вони допомагають розробникам ефективно скорочувати час на написання коду та підвищувати його якість.

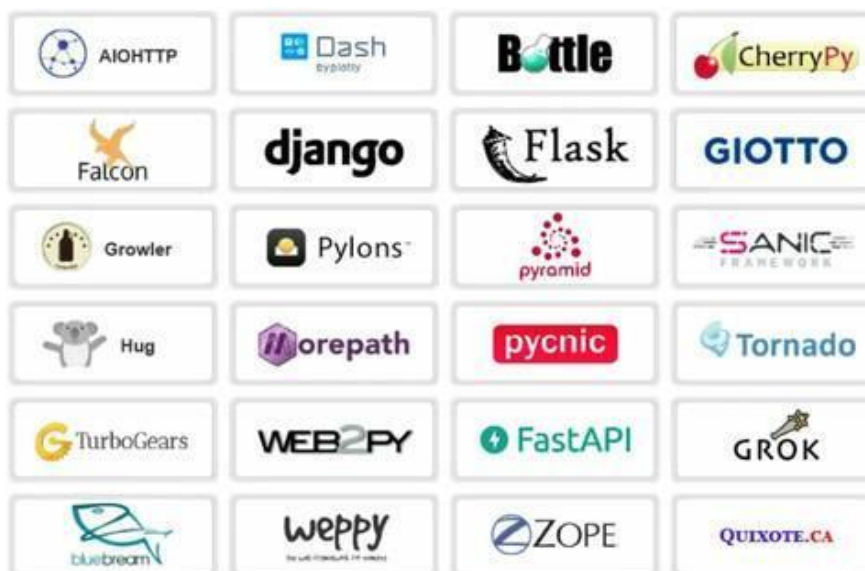


Рис. 1.1 Різновиди фреймворків [1]

Кожен фреймворк спроектований з урахуванням потреб різних розробників та проектів. У кожного з них є свої особливості, архітектура та методи розробки, які можуть бути більш підходящими для певних типів проектів чи стилів програмування. Деякі фреймворки спрямовані на високу продуктивність, інші – на зручність використання та інші аспекти. Це дає розробникам можливість обирати фреймворк, який найкраще відповідає їхнім потребам.

Основна мета фреймворків полягає у спрощенні та прискоренні процесу розробки. Вони пропонують розробникам готові структури, що не потребують повноцінного написання з нуля, дозволяючи застосовувати їх для створення програм та веб-сайтів. Це дозволяє розробникам зосередитися на розв'язанні бізнес-задач та функціональних особливостей програми. Фреймворки сприяють підвищенню підтримки та надійності коду, оскільки вони проходять тестування від їх авторів і проявляють стабільність під час роботи.

Основні переваги використання фреймворків включають:

1. Збільшення продуктивності та прискорення процесу розробки завдяки наявності готових рішень і шаблонів.

2. Покращення якості коду та його обслуговування. Фреймворки приділяють значну увагу цим аспектам.
3. Спрощення процесу тестування та налагодження за рахунок вбудованих інструментів.
4. Підвищення зручності підтримки та швидкість внесення змін завдяки гнучкості та зручності у використанні та модифікації фреймворків.

Фреймворки надають розробникам готовий набір коду, бібліотек, шаблонів та рекомендацій для організації структури програми. Розробникам залишається лише ефективно використовувати ці готові рішення при розробці продукту.

У роботі розробник повинен дотримуватися встановленої структури та правил, використовувати класи та методи відповідно до документації.

Якщо розробник використовує фреймворк вперше, для нього важливо пройти такі кроки:

1. Ознайомлення з документацією: освоєння базових особливостей та можливостей фреймворку через ознайомлення з документацією.
2. Планування проекту: визначення вимог до проекту та його структури відповідно до архітектури фреймворку.
3. Установка: розгортання фреймворку, його компонентів та необхідних інструментів.
4. Конфігурування: налаштування фреймворку з урахуванням вимог проекту.
5. Розробка: написання коду з використанням можливостей фреймворку.
6. Тестування: перевірка функціоналу та виправлення помилок.
7. Розгортання: розміщення готового проекту у робочому середовищі.

Послідовність та конкретні етапи можуть змінюватися в залежності від завдань та пріоритетів розробника у конкретній роботі.

Розглядаємо найбільш відомі фреймворки, створені для мов програмування та розмітки, що широко застосовуються у веб-розробці. Деякі з них є більш популярними, інші менш відомі, але кожен має свої переваги та недоліки. Загалом вони отримали позитивні відгуки від багатьох розробників, тому ми рекомендуємо розглянути найкращі варіанти:

JavaScript володіє найбільшим арсеналом фреймворків у сфері розробки програмного забезпечення. Серед найпопулярніших фреймворків для JavaScript можна відзначити:

- React
- Angular
- Vue.js
- Next.js
- Ember.js



Рис. 1.2 Фреймворки JavaScript [2]

Давайте розглянемо кожен з них більш детально.

React - це набір інструментів для створення веб-інтерфейсів на JavaScript, розроблений компанією Facebook. Він дозволяє створювати компоненти користувацького інтерфейсу, які можна легко поєднувати для створення великих програм. Основою React є концепція Virtual DOM, яка оптимізує продуктивність і прискорює відображення інтерфейсу.

Розробники цінують його гнучкість та потужні функції, а також можливість використання як для веб-, так і для мобільних додатків.

Angular - це фреймворк для створення веб-додатків, розроблений та підтримуваний компанією Google. Angular пропонує широкі можливості для керування станом програми, маршрутизації, відображення та тестування. Він також включає інструменти розробки та підтримки мобільних платформ. Це популярний вибір для багатofункціональних веб-застосунків зі складним функціоналом.



Рис. 1.3 Angular [3]

Vue.js - це популярний JavaScript фреймворк для створення користувацьких інтерфейсів. Він відрізняється легкою архітектурою, зручною системою шаблонів та потужними інструментами для швидкої розробки. Vue.js дозволяє легко створювати динамічні інтерфейси, а його активна спільнота розробників постійно розвиває та покращує функціональність.



Рис. 1.4 Vue.js [4]

Vue.js - це фреймворк для розробки програм на JavaScript, побудований на основі React. Він спеціалізується на серверній обробці та забезпечує ефективне виконання на стороні клієнта. Next.js надає зручні інструменти для розробки та підтримки складних програм і може інтегруватися з іншими технологіями.

Ember.js - це JavaScript-фреймворк для створення веб-застосунків, що використовує концепцію маршрутизації та моделей для спрощення розробки. Він має широку спільноту розробників і пропонує потужні інструменти для управління динамікою, шаблонизацією та тестуванням.

Щодо фреймворків Python, Django - це повноцінний фреймворк для створення веб-додатків, Flask - мінімалістичний і підходить для менших проєктів, Pyramid - гнучкий та підходить для проєктів різних розмірів, а Tornado - спеціалізується на високопродуктивних додатках.



Рис. 1.5 Ember.js [5]

Фреймворк Django є інструментом для швидкої та ефективної розробки веб-додатків на Python. Він ґрунтується на патерні Модель-Шаблон-Представлення (MVT), пропонуючи широкий спектр інструментів, таких як адміністративний інтерфейс, обробка HTTP-запитів та URL-роутинг.

Фреймворк Flask, навпаки, цілеспрямований на простоту використання та легкість. Він забезпечує базові інструменти для створення веб-застосунків, такі як роутинг та робота з HTTP, залишаючи простір для розширення та налаштування за потребою.



Рис. 1.6 Flask [6]

Pyramid пропонує гнучку архітектуру та різноманітність можливостей для розробки веб-додатків. Він підтримує різні бази даних та веб-сервери, має потужні можливості для роботи з URL та підтримує широкий спектр технологій та шаблонів.

Tornado спеціалізується на створенні високопродуктивних веб-додатків та веб-сервісів, використовуючи асинхронні можливості Python. Його асинхронна природа дозволяє обробляти багато одночасних підключень, і це чудовий вибір для реал-тайм додатків.

У свою чергу, фреймворки PHP, такі як Laravel, CodeIgniter, Symfony та інші, є популярними інструментами серед розробників PHP для швидкої розробки веб-додатків.

1.2 Принципи підвищення користувацького досвіду

Більшість компаній мають веб-сайти з метою збільшення прибутку, привернення клієнтів до фізичного магазину, отримання подальших консультацій або для онлайн-продажу товарів. Проте багато таких сайтів не досягають поставлених цілей, особливо у власників невеликих бізнесів, оскільки їм часто не вистачає досвіду для оптимізації користувацького досвіду й досягнення максимального рівня конверсії. Сьогодні я пропоную кілька порад, як підвищити конверсію вашого бренду через поліпшення користувацького досвіду.

Користувацький досвід - це спосіб, яким відвідувачі взаємодіють з вашим веб-сайтом після його відвідування. Якщо користувацький досвід низький, це може призвести до розгубленості клієнтів через труднощі отримання необхідної інформації через такі фактори, як погана структура сторінки, складна навігація, повільне завантаження, неякісні зображення, надмірне використання стокових зображень, неправильне розташування кнопок (особливо на мобільних пристроях) та застарілий контент. Рисунок

1.6 ілюструє відмінність між хорошою та поганою архітектурою веб-сайту, зокрема сайтів електронної комерції.

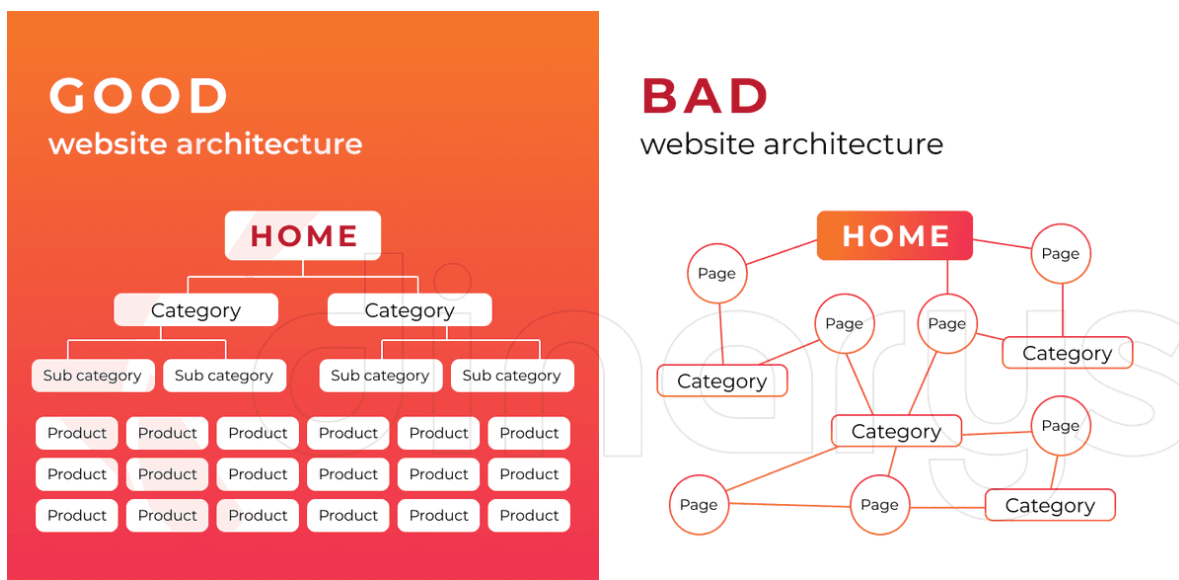


Рис.1.7 Відмінність між хорошою та поганою архітектурою веб-сайту[7]

Наприклад, позитивний користувацький досвід протистоїть цьому, оскільки він включає в себе створення оригінального контенту, що відповідає потребам відвідувачів. Значна увага зосереджується на важливості користувача або відвідувача в сприйнятті досвіду, що буде розглянуто докладніше пізніше. Юзабіліті охоплює всі аспекти користувацької взаємодії з веб-сайтом, включаючи дизайн, навігацію, контент та загальний рівень зручності використання. У сучасному інтернеті це також означає увагу до потреб користувачів з різними можливостями, такими як люди з поганим зором, через, наприклад, використання атрибутів alt для зображень.

Принципи UX-дизайну - це надійний посібник для створення інтуїтивно зрозумілих, доступних і приємних дизайнів і користувацьких інтерфейсів. Вони допомагають дизайнерам ухвалювати рішення щодо організації та виділення візуальних підказок, привернення уваги та задоволення потреб користувачів максимально простими способами. Ці

принципи слугують основою, що спрямовує мислення дизайнерів і яку вони можуть використовувати в разі невизначеності.

- Ієрархія керує досвідом

Структура сайту - це ключовий принцип UX-дизайну. Ось чому их wireframing завжди стає в нагоді. Існує два важливих типи ієрархії, про які потрібно пам'ятати в процесі проектування - інформаційна та візуальна ієрархія.

Інформаційна архітектура, або ієрархія, означає спосіб організації контенту на веб-сайті або в додатку. Коли користувачі дивляться на панель навігації і бачать головне і другорядні меню, вони дивляться на інформаційну ієрархію. Цей тип ієрархії повинен бути передбачуваним і зрозумілим, щоб користувачі, які вперше заходять на ваш сайт, знали, де знайти потрібну їм інформацію, не замислюючись про це двічі.

Візуальна ієрархія - це те, як ви організовуєте елементи на сторінці, і як це впливає на навігацію користувачів. Хороша візуальна ієрархія - це та, в якій важливий контент розташований на видному місці і виділений, відображає потреби користувачів і дозволяє їм легко знаходити часто використовувані елементи. Візуальний дизайн спрямовує сприйняття користувачів від найважливіших елементів до найменш важливих.

- Простий дизайн

Простий дизайн означає зручність для користувачів. Його мета - зменшити зусилля, які люди витрачають на розуміння та використання веб-сторінки. Якщо зменшити балакання елементів і викласти головне чітко, це полегшить користування і зробить його більш зручним. Навпаки, перенасиченість інформацією може збити з пантелику та змусити людей шукати інше місце для зручності.

Ідеальний дизайн - це той, який такий простий, що користувачі майже не помічають, як переміщуються по сайту. Мінімалістичний стиль має відповідати очікуванням користувачів і не заважати їм перегляду сторінки.

Якщо треба показати багато інформації, це робиться поступово і з урахуванням контексту, а необхідні дії користувача робляться частинами, щоб навігація була простою. Результат - користувач легко переміщується по сайту.

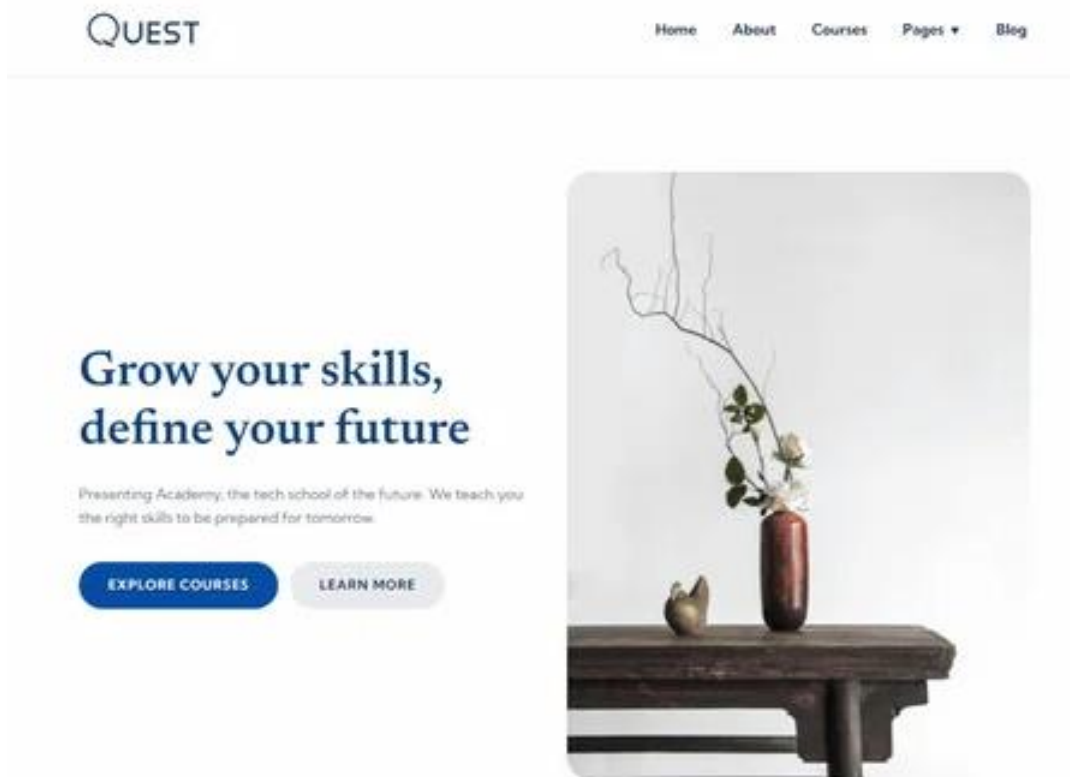


Рис.1.8 Простий дизайн [8]

- Думати про користувача

Ще один важливий принцип UX-дизайну підкреслює постійну пріоритетність користувача. Хоча дизайн може подобатися або здаватися приємним дизайнеру, його остаточна перевірка полягає в тому, як користувачі сприйматимуть, взаємодіятимуть і отримуватимуть від нього користь - по суті, йдеться про розуміння користувацького потоку. Це вимагає зосередження на юзабіліті та привнесення людського дотику в досвід.

Юзабіліті нерозривно пов'язане з розумінням больових точок та вподобань користувачів, формуванням дизайну, що дозволяє оперативно

реагувати на ці аспекти. Це передбачає інтеграцію інтуїтивно зрозумілих елементів керування (кнопок, ярликів, навігації), оптимізацію часу завантаження сторінок, а також спрощення взаємодії з сайтом за допомогою простої мови, інтерактивних функцій і шаблонів дизайну, щоб зменшити візуальний безлад і зайву інформацію. Надання видимості стану системи дозволяє користувачам отримувати оперативний зворотній зв'язок, що дає їм відчуття контролю.

Підхід, орієнтований на користувача, не лише покращує його контроль, але й усуває бар'єри та візуальні перешкоди. Такі прийоми, як використання контрастних кольорів або розташування міток за межами полів для введення тексту, суттєво допомагають користувачам з вадами зору або тим, хто працює в умовах недостатнього освітлення.

З іншого боку, гуманізація досвіду виходить за рамки простої навігації по інтерфейсу. Вона передбачає включення людського фактору в UX-дизайн, запуск емоційних зв'язків і реакцій через мікровзаємодію, гумор і візуальні ефекти, тим самим виходячи за рамки суто функціональної взаємодії.

Гармонізуючи ці аспекти, ми зосереджуємо увагу на користувачеві, створюючи середовище, сприятливе для більшої залученості.

- Оберіть належну типографіку

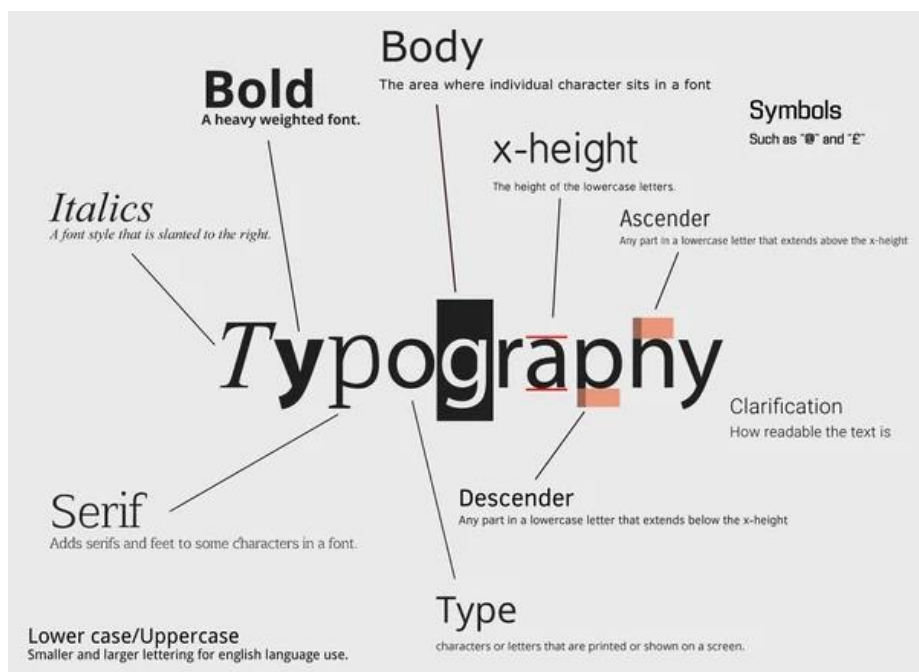


Рис.1.9 Види типографії[9]

Вибір правильної типографії впливає на весь UX-дизайн, здатну як поліпшити його зручність використання, так і зруйнувати. Це не лише справа вибору шрифту, його розміру і стилю, але й про розміщення пробілів, структурування контенту для візуальної ієрархії та взаємодію різних шрифтів. Усе це впливає на зрозумілість, зручність читання та доступність. Під час роботи над типографікою важливо розуміти, чи підкреслює вона ваші ідеї та як саме це відбувається.

- Тестування юзабіліті

Перевірка веб-сайту чи додатку без проведення тестування на зручність — це майже неможливо.

Тести на зручність — це ефективний метод перевірити вашу роботу та оцінити, наскільки ваш дизайн дотримується принципів UX, зокрема, користувацького дизайну. Вони показують, як користувачі взаємодіють з дизайном та де вони мають проблеми. Якщо труднощі накопичуються у певній частині користувацького досвіду, це дає вам чіткі вказівки, що потребує виправлення.

Такі тести важливо проводити на різних етапах UX-дизайну — від початкового проектування і прототипування до фінальної версії і мобільного дизайну, а також після значних змін. Різні методи тестування, такі як A/B-тести, записи сеансів, відстеження руху очей та інтерв'ю, дозволяють отримати різноманітну інформацію про поведінку користувачів. Проведення широкого спектру тестів — це добра ідея, оскільки це допомагає виявити різні аспекти користувацької поведінки.

Why Usability Test?

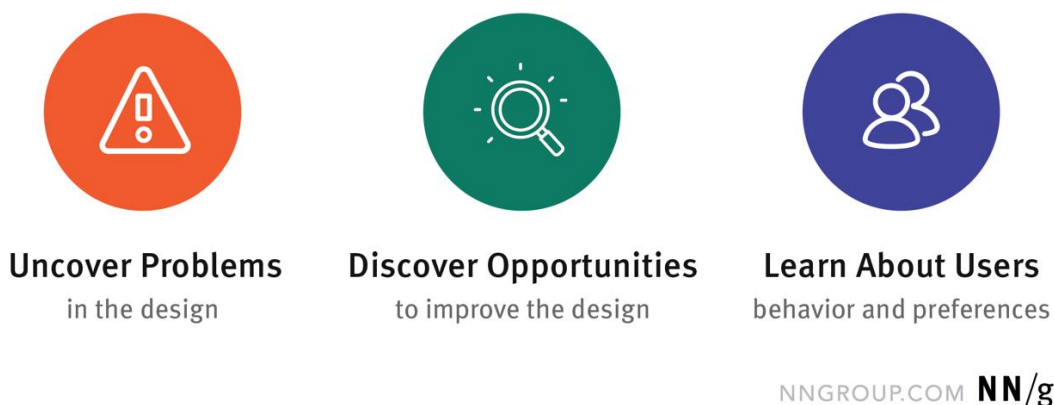


Рис.1.10 Що таке юзабіліті тестування? [10]

1.3 Теоретичні поняття веб-додатків

Веб-додаток - це програмне забезпечення, розміщене на віддаленому сервері, що запускається через веб-браузер користувача. Його перевага полягає в тому, що він сумісний з більшістю стандартних комп'ютерів і операційних систем, оскільки використовує браузери. Крім того, він не потребує значної пам'яті на жорсткому диску комп'ютера і доступний з будь-якого пристрою. Декілька користувачів можуть використовувати його

одночасно, що сприяє спільній роботі. Хоча для його роботи потрібне підключення до мережі, це обмеження стає менш суттєвим зі зростанням доступності Інтернету.

Будь-який сервіс, що доступний через Інтернет, може бути визнаний як веб-додаток. Наприклад, це можуть бути онлайн-форми, корзини для покупок, стрімінгове відео, соціальні мережі, ігри або електронна пошта. Завдяки широкому доступу до мережі, багато веб-додатків розробляються для функцій, які раніше використовували окремі програми поза Інтернетом, такі як обробка тексту, створення електронних таблиць, редагування графіки чи відео.

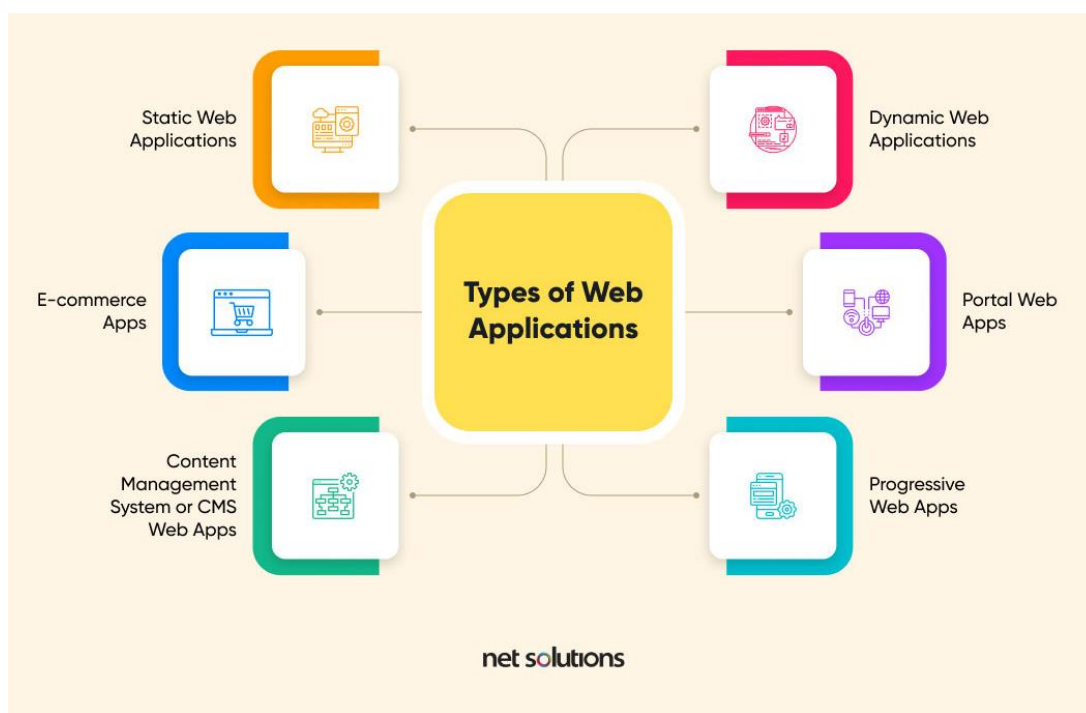


Рис.1.11 Що таке юзабіліті тестування? [11]

Кожного року зростає кількість компаній, які використовують Інтернет як основний інструмент для спілкування з клієнтами та співробітниками. Ця мережа дозволяє обмінюватися інформацією з цільовою аудиторією та проводити безпечні транзакції. Проте для ефективної взаємодії необхідно швидко збирати та обробляти дані про

клієнтів, реагувати на їх запити тощо. Звичайні сайти не завжди впораються з усіма цими вимогами бізнесу, тому на допомогу приходять веб-додатки. Веб-додатки - це інтерактивні програми, що складаються з двох основних частин:



Рис.1.12 Що таке веб-додаток? [12]

- Перша частина завантажується у браузері, коли користувач вводить відповідну команду.
- Друга частина розташована на веб-сервері, де міститься вся інформація, вбудована в додаток. Вона отримує запит від користувача, знаходить інформацію та передає її у браузер.

Для більшості клієнтів веб-додатки ззовні майже не відрізняються від звичайних сайтів. Але вони мають велику кількість вбудованих функцій - від роботи з графікою та таблицями до онлайн оформлення замовлень.

Переваги веб-застосунків:

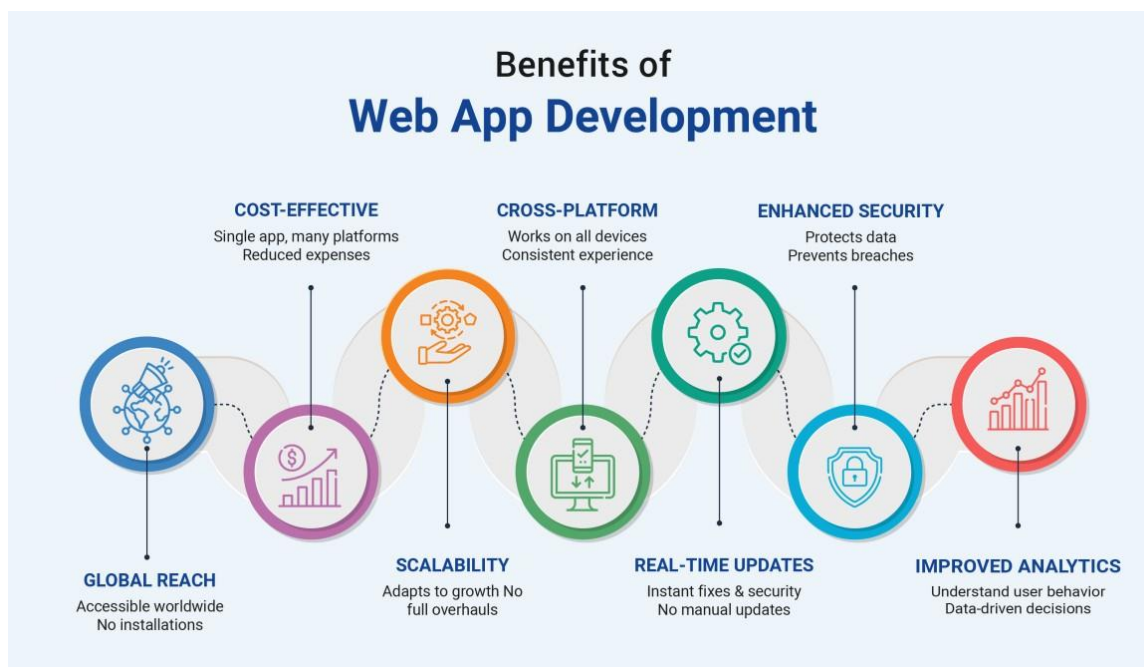


Рис.1.13 Переваги веб-додатків? [13]

- Веб-додатки не потребують установки на пам'ять пристрою.
- Користувачам не потрібно дбати про оновлення програми, оскільки вони автоматичні та централізовані.
- Усі користувачі мають доступ до однієї версії веб-застосунку, що уникає непорозумінь.
- Розробка веб-додатків відбувається з урахуванням можливості відкриття з будь-якого браузера чи операційної системи.
- Веб-додатки зменшують витрати для бізнесу та кінцевих користувачів, оскільки вимагають менше ресурсів для роботи та обслуговування.
- Застосунки є простими у використанні і не потребують спеціальних знань або навичок, що дозволяє працювати з різною аудиторією.
- Веб-застосунки економічні, оскільки вони дешевші за класичні програмні рішення, що дозволяє використовувати їх компаніям з обмеженим бюджетом.
- Веб-додаток можна легко адаптувати під потреби будь-якого бізнесу.

Веб-додатки розробляються з різних цілей - для взаємодії з клієнтами та для оптимізації бізнес-процесів всередині компанії. Можливості цих інструментів майже необмежені.

Вони дозволяють:

- обмінюватись інформацією між співробітниками та клієнтами;
- здійснювати транзакції;
- здійснювати дистанційну купівлю та продаж товарів/послуг;
- працювати з нотатками, списками завдань та керуванням файлами у хмарі;
- конвертувати валюту та одиниці вимірювання (наприклад, сантиметри в метри);
- здійснювати переклад тексти тощо.

Переваги веб-додатків полягають у тому, що вони дають можливість працювати з онлайн-документами (текстові файли, електронні таблиці, презентації), надавати доступ до них обмеженій групі осіб і спільно працювати над проектами навіть на відстані. Зберігання всієї інформації в одному місці допомагає керівникам зрозуміти стан бізнесу, швидко виявляти й виправляти проблеми.

2 АНАЛІЗ ЕФЕКТИВНОСТІ РОЗРОБКИ ІНТЕРАКТИВНИХ ВЕБ-ДОДАТКІВ З ВИКОРИСТАННЯМ СУЧАСНИХ ФРЕЙМВОРКІВ ТА БІБЛІОТЕК

2.1 Огляд існуючих інструментів для розробки веб-додатків

При огляді існуючих інструментів, треба приділити увагу інструментам для управління базами даних (СКБД) є важливою складовою при розробці веб-додатків, які операціонують з великим обсягом даних. Визначення успіху впровадження та подальшої ефективності використання бази даних часто залежить від уважного аналізу конкретних аспектів цих інструментів. Роздивимось детальніше існуючі рішення:

MySQL:



Рис. 2.1 MySQL [14]

MySQL є однією з найвідоміших та популярних відкритих систем управління реляційними базами даних (СКБД). Розроблена компанією Oracle, вона користується широкою популярністю завдяки своїй надійності та продуктивності. MySQL є ідеальним вибором для веб-проектів, де важлива швидкодія та надійність в обробці структурованих даних.

MongoDB:

Рис. 2.2 MongoDB [15]

MongoDB, це нереляційна база даних, яка визнана як одна з провідних систем управління базами даних в екосистемі NoSQL, відзначається своєю унікальною документо-орієнтованою моделлю даних. Розроблена компанією MongoDB Inc., ця база даних здобула широкий застосунок завдяки своїм особливостям, які роблять її ідеальною для веб-проектів та додатків, де вимагається гнучкість, швидкість та простота масштабування.

PostgreSQL:

Рис. 2.3 PostgreSQL [16]

PostgreSQL, визнана як одна з переважаючих систем управління реляційними базами даних (СКБД), виділяється своєю потужною та гнучкою архітектурою. Створена спільнотою розробників і активно

розвивається, PostgreSQL широко використовується завдяки своїм високим стандартам та підтримці різноманітних функцій. Ця реляційна СКБД ідеально підходить для проектів, де потрібна складна структура даних та велика ефективність обробки запитів.

PostgreSQL визначається своєю здатністю до роботи зі складними зв'язками та динамічними структурами. Ця СКБД приділяє особливу увагу високій надійності та розширюваності, зроблюючи її оптимальним рішенням для великих та вимогливих проектів.

PostgreSQL не просто надійна система для зберігання даних; вона також відкрита для розширень та підтримує розширення стандарту SQL, надаючи розробникам великий простір для творчості та адаптації до конкретних потреб їхніх проектів. Це робить PostgreSQL потужним інструментом для широкого спектру застосувань, включаючи великі корпоративні системи, веб-проекти та додатки зі складною структурою даних.

Розглядаючи системи керування базами даних, важливо також звернутися до інших ключових інструментів для розробки веб-додатків, зокрема інструментів контролю версій. Ці інструменти грають вирішальну роль у забезпеченні ефективного та спільного робочого процесу команди розробників.

Git:



Рис. 2.4 Git [17]

Git – це система контролю версій, розроблена для ефективного керування змінами у програмному коді проектів. Її висока популярність обумовлена не лише потужністю та гнучкістю, але й розширеним спектром інструментів, які вона пропонує розробникам для колективної роботи та управління кодовою базою.

Ключовою функцією Git є можливість створення та управління гілками. Гілки дозволяють розробникам одночасно працювати над різними функціональностями чи виправленнями, зберігаючи при цьому стабільність та чистоту основного коду. Об'єднання гілок (merge) або їх перебудова (rebase) дає можливість комбінувати зміни та інтегрувати їх у головну гілку.

Git інтегрований з різними хмаровими платформами, такими як GitHub, Bitbucket та GitLab, що дозволяє розробникам зберігати свій код у хмарі та мати доступ до нього з будь-якого місця. Крім того, ці платформи розширюють можливості Git, надаючи функції для відстеження змін, проведення код-рев'ю, створення завдань та обговорення коду, сприяючи ефективній комунікації та співпраці в команді розробників.

Git також дозволяє докладно вести журнал змін, що допомагає аналізувати історію проекту та відстежувати внесок кожного розробника. Це істотно для виявлення помилок, відкату змін або відновлення попередніх версій коду.

Підсумовуючи, Git є необхідним інструментом для розробників, надаючи їм контроль над розробкою програмного забезпечення. Його інтеграція з хмаровими платформами спрощує колективну роботу команди, забезпечуючи зручність та ефективність у процесі спільної роботи.

GitHub, GitLab, Bitbucket:



Рис. 2.4 GitHub, GitLab, Bitbucket [18]

GitHub:

GitHub виконує роль не лише системи для зберігання та управління кодовими проектами, але і стає центральним пунктом для спільної роботи розробників у режимі реального часу. Платформа надає розширений набір інструментів для обговорень, ведення списків завдань (issues), використання вбудованих CI/CD-сервісів для тестування коду, аналізу внеску кожного учасника та інших зручних можливостей співпраці.

Однією з ключових переваг GitHub є можливість створення пул-реквестів (pull requests), які дозволяють розробникам пропонувати зміни в основний код проекту. Це ефективний механізм для обговорення та

рецензування внесків, а також для автоматизованого тестування перед їх об'єднанням з основною гілкою. Такий підхід забезпечує контроль якості коду та дозволяє здійснювати розподілену розробку, навіть у великих командах.

Узагальнюючи, GitHub, спільно з Git, утворює потужний інструментарій для розробників, який сприяє не лише контролю версій, але й високому рівню співпраці, комунікації та управління проектами на всіх етапах їх розробки.

GitLab:

GitLab є інтегрованою платформою для DevOps, що включає в себе не лише інтеграцію з Git, але й розширений функціонал для автоматизації розробки та випуску продуктів. Аналогічно до Git, вона є системою контролю версій, спроектованою для ефективного керування змінами у програмному коді проектів. Її популярність пояснюється не лише великою потужністю та гнучкістю, але й розширеним набором інструментів для колективної роботи та управління кодовою базою.

Ключовою можливістю GitLab, як і Git, є створення та управління гілками. Це дозволяє розробникам одночасно працювати над різними функціональностями чи виправленнями, забезпечуючи при цьому стабільність та чистоту основного коду. Можливість об'єднання гілок (merge) або їх перебудова (rebase) дає можливість комбінувати зміни та інтегрувати їх у головну гілку проекту.

GitLab інтегрований з різними хмаровими платформами, що дозволяє розробникам зберігати свій код у хмарі та отримувати до нього доступ з будь-якого місця. Крім того, платформа розширює базові можливості Git, надаючи інструменти для відстеження змін, проведення код-рев'ю, створення завдань та обговорення коду, що сприяє ефективній комунікації та співпраці в команді розробників.

Зазначимо, що GitLab дозволяє вести детальний журнал змін, що допомагає аналізувати історію проекту та відстежувати внесок кожного розробника. Це є ключовим для виявлення помилок, відкату змін або відновлення попередніх версій коду.

Підсумовуючи, GitLab представляє собою необхідний інструмент для розробників та DevOps, надаючи контроль над розробкою програмного забезпечення та автоматизуючи процеси розробки та випуску продуктів. Його інтеграція з хмаровими платформами полегшує колективну роботу команди, забезпечуючи зручність та ефективність у процесі спільної роботи.

Bitbucket:

Bitbucket, подібно GitLab, є інтегрованою DevOps-платформою, яка об'єднує інтеграцію з Git та розширений функціонал для автоматизації розробки та випуску продуктів. Як і Git, це система контролю версій, створена для ефективного керування змінами у коді проектів. Bitbucket володіє великою потужністю та гнучкістю, надаючи розширений набір інструментів для колективної роботи та управління кодовою базою.

Однією з ключових можливостей Bitbucket, подібно GitLab, є можливість створення та управління гілками, що дозволяє розробникам одночасно працювати над різними функціональностями та зберігати стабільність основного коду. Можливість об'єднання гілок (merge) або їх перебудова (rebase) дозволяє комбінувати та інтегрувати зміни у головну гілку проекту.

Bitbucket інтегрований з хмаровими платформами, дозволяючи розробникам зберігати код у хмарі та отримувати доступ з будь-якого місця. Платформа розширює базові можливості Git, надаючи інструменти для відстеження змін, проведення код-рев'ю, створення завдань та обговорення коду, сприяючи ефективній комунікації та співпраці в команді розробників.

Bitbucket дозволяє вести детальний журнал змін, що полегшує аналіз історії проекту та внесок кожного розробника. Це важливо для виявлення помилок, відкату змін або відновлення попередніх версій коду. Узагальнюючи, Bitbucket є необхідним інструментом для розробників та DevOps, забезпечуючи контроль над розробкою програмного забезпечення та автоматизуючи процеси випуску продуктів.

Хмарні технології широко застосовуються для розробки веб-додатків. Розглянемо деякі з найпопулярніших хмарних сервісів, які використовуються для цієї мети.

AWS:



Рис. 2.4 AWS [19]

Amazon Web Services (AWS) є провідним лідером у наданні хмарних послуг, пропонуючи розширений асортимент сервісів у сферах сховищення даних, обчислень, мереж та інших стратегічних галузей. AWS відкриває можливості для клієнтів ефективно та безпечно зберігати, обробляти та масштабувати свої дані та додатки, використовуючи передову інфраструктуру в хмарі. Платформа надає широкий спектр інструментів, що сприяють спрощенню розгортання та ефективному управлінню

інфраструктурою, а також гарантують високий рівень доступності та безпеки. AWS постійно оновлюється та розширює свій функціонал, щоб відповідати різноманітним вимогам користувачів та корпоративних клієнтів у сфері хмарних технологій

Azure:



Рис. 2.5 Azure [20]

Azure, розроблений Microsoft, визначається як висококласний провайдер хмарних послуг, що пропонує розширений набір сервісів для сховищення даних, обчислень, мереж і інших стратегічних областей. Платформа Azure дозволяє клієнтам забезпечити безпечне зберігання, обробку та масштабування своїх даних та додатків, використовуючи вражаючу хмарну інфраструктуру.

Azure забезпечує доступ до різноманітних інструментів, спрямованих на полегшення розгортання та ефективного керування інфраструктурою. Це дозволяє користувачам насолоджуватися високим рівнем доступності та безпеки їхніх ресурсів. Платформа постійно розвивається, додаючи нові можливості та інновації, щоб відповідати зростаючим вимогам користувачів і бізнес-спільноти в області хмарних технологій. Як і AWS, Azure виступає сильним інструментом для створення та впровадження передових хмарних рішень.

Lighthouse:



Рис 2.6 Lighthouse [21]

Lighthouse є інструментом у веб-розробці та оптимізації веб-сайтів, який розробила компанія Google для покращення продуктивності та користувацького досвіду. Цей інструмент дозволяє веб-розробникам та власникам веб-сайтів отримувати детальний аналіз продуктивності своїх сторінок та впроваджувати оптимізації для поліпшення завантаження та взаємодії з контентом.

Lighthouse визначається як високотехнологічний інструмент для автоматизованого тестування веб-сторінок та забезпечення їх ефективності. Він має розширений набір інструментів для аналізу різних аспектів веб-сайту, таких як продуктивність, доступність, багатомовність та інші ключові параметри.

Lighthouse дозволяє розробникам виявляти проблеми, які впливають на продуктивність веб-сайту, і надає рекомендації з виправлення цих проблем. Також генерує звіти, які вказують на конкретні області для оптимізації та розвитку.

Цей інструмент є невід'ємною частиною стратегії веб-розробки та оптимізації, допомагаючи забезпечити користувачам швидке завантаження

веб-сайтів і поліпшити їхню зручність та задоволеність взаємодією. Lighthouse постійно оновлюється, додаючи нові функції та інструменти, щоб відповідати сучасним вимогам веб-індустрії та гарантувати максимальний рівень продуктивності веб-сайтів.

WebPageTest:



Рис. 2.7 WebPageTest [22]

WebPageTest - це інструмент для вивчення та вдосконалення продуктивності веб-сайтів. Як відкритий інструмент, він дозволяє веб-розробникам аналізувати швидкість завантаження сторінок та впроваджувати оптимізації для поліпшення враження користувачів.

Також він є високотехнологічним інструментом для автоматизованого вивчення продуктивності веб-сайтів. Він виявляє та рекомендує оптимізації для прискорення завантаження сторінок і генерує звіти з конкретними областями для вдосконалення.

Він грає ключову роль у стратегії розробки та вдосконалення веб-сайтів, забезпечуючи ефективне завантаження та поліпшення взаємодії з користувачем. Інструмент постійно оновлюється, щоб відповідати останнім вимогам та гарантувати максимальну продуктивність веб-сайтів.

Ми детально розглянули та вивчили наявні інструменти, що забезпечують можливість розробки веб-додатків високої якості та їх подальшу оптимізацію. Цей аналіз охоплює етапи як процесу розробки, так

і після випуску версії, спрямовуючись на покращення ефективності та якості продукту.

2.2 Метрики ефективності веб-додатків

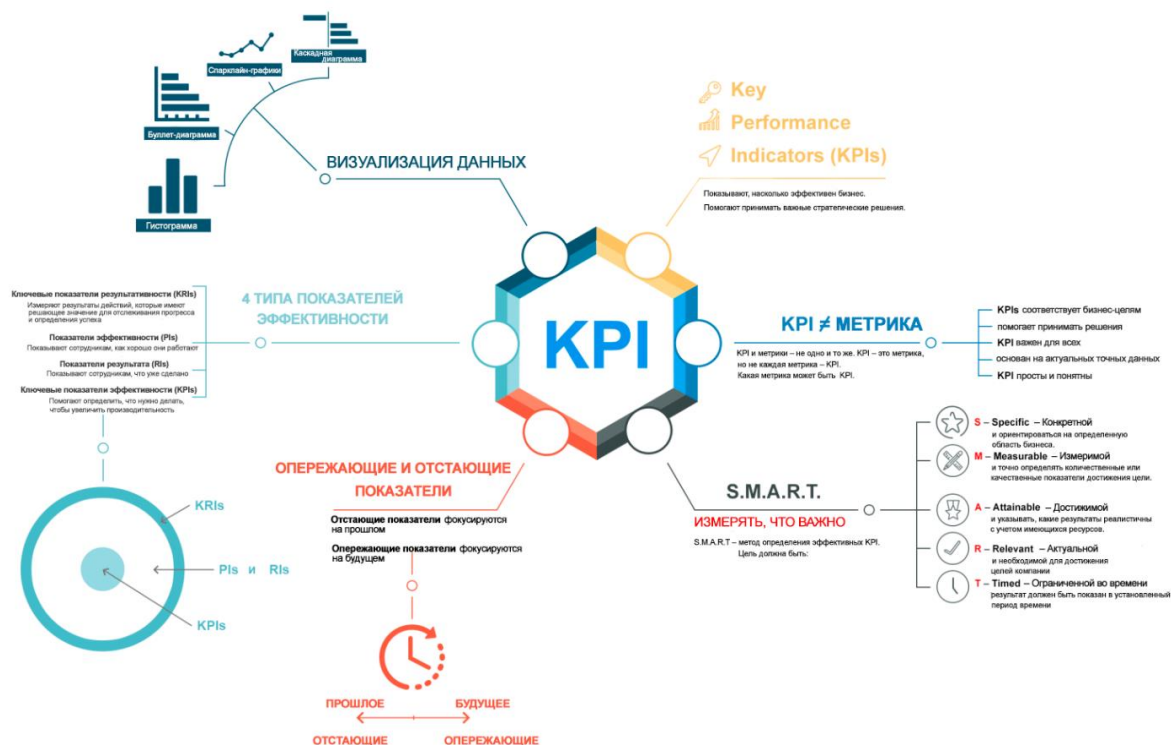


Рис 2.8 KPI [23]

Метрики ефективності веб-додатків служать для комплексної оцінки їх продуктивності, спрямованої на швидкість та взаємодію з користувачами. Розглянемо докладніше кожен ключову метрику:

Час завантаження сторінки (Page Load Time): Ця метрика вимірює час, необхідний для завантаження вмісту сторінки. Важлива для першого враження користувача та зменшення відсотку відмов.

Time to First Byte (TTFB): Час, який проймає від запиту користувача до отримання першого байта від сервера. Низький TTFB свідчить про ефективність серверної відповіді.

Кількість запитів (Number of Requests): Визначає кількість ресурсів, які браузер повинен запитати для завантаження сторінки. Мінімізація кількості запитів поліпшує швидкість завантаження.

Час відгуку сервера (Server Response Time): Міра часу, який сервер витрачає на обробку запиту. Впливає на загальну ефективність веб-додатка.

Продуктивність JavaScript (JavaScript Performance): Визначає, наскільки ефективно виконується JavaScript-код, впливаючи на взаємодію та реакцію додатка.

Час відображення першого контенту (First Contentful Paint): Визначає момент, коли перший контент стає видимим для користувача, важливий для враження швидкості завантаження.

Час інтерактивності (Time to Interactive): Час, коли веб-додаток стає повністю взаємодійним для користувача, визначаючи готовність до взаємодії.

Відсоток відмов (Bounce Rate): Показник, що вказує на відсоток випадків, коли користувачі покидають сторінку без взаємодії, що може сигналізувати про проблеми з ефективністю чи контентом.

За допомогою цих метрик можна оцінити загальний стан веб-додатка та визначити області для оптимізації. Далі розглянемо додаткові метрики:

Кешування (Caching): Метрика визначає ефективність використання кешу для зберігання ресурсів, що сприяє швидкому завантаженню для повторних відвідувачів.

Обсяг переданих даних (Data Transfer Size): Кількість даних, що передаються між сервером та клієнтом, впливає на час завантаження, особливо на мобільних пристроях.

Стабільність сторінки (Page Stability): Метрика вимірює стабільність сторінки під час завантаження, виключаючи стрибки та зміщення елементів.

Використання ресурсів (Resource Utilization): Оцінює ефективність використання ресурсів, таких як CPU, пам'ять та мережа, що впливає на загальну продуктивність.

Кількість помилок (Error Rate): Визначає відсоток запитів, які викликали помилку, свідчать про стабільність та надійність веб-додатка.

Мобільна оптимізація (Mobile Optimization): Метрика визначає, наскільки добре додаток оптимізований для мобільних пристроїв, враховуючи швидкість та зручність використання на різних екранах.

Всі ці метрики разом надають повнісінький образ ефективності веб-додатка, що дозволяє розробникам та адміністраторам здійснювати інформовані рішення для оптимізації та поліпшення користувацького досвіду.

2.3 Аналіз ефективності веб-додатків

Аналіз ефективності веб-додатків – це комплексне завдання, яке передбачає систематичне дослідження та аналіз різноманітних показників для визначення рівня продуктивності, швидкості реакції та загального користувацького досвіду, який забезпечує веб-додаток. Цей процес включає кілька ключових етапів, які детально розглядають різні аспекти функціонування додатка.

Визначення Ключових Метрик:

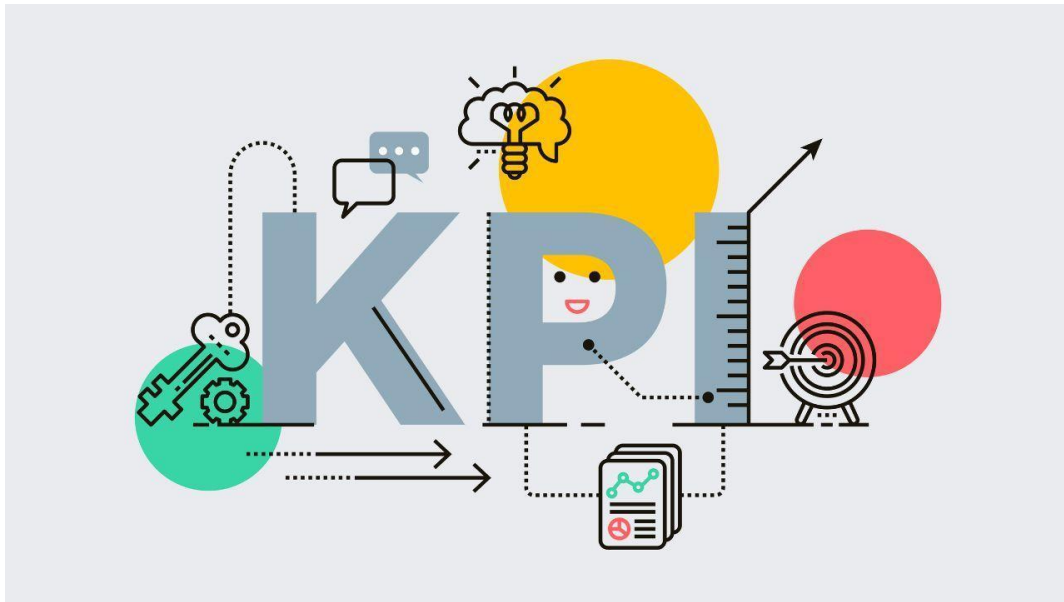


Рис 2.10 Визначення Ключових Метрик [25]

Time to First Byte (TTFB) та кількість запитів до сервера є важливими показниками ефективності серверної відповіді. TTFB визначає період часу від подачі користувачем запиту до отримання першого байта від сервера, вказуючи на швидкість обробки запитів. Кількість запитів свідчить про обсяг ресурсів, які потрібно завантажити для відображення сторінки, і мінімізація цього параметра може сприяти покращенню загальної швидкості завантаження.

Час відгуку сервера визначає тривалість обробки запиту, що має прямий вплив на продуктивність веб-додатка. Ефективне управління цими метриками дозволяє розробникам підтримувати оптимальний баланс між часом завантаження та ресурсами, що, в свою чергу, формує задоволення та зручність взаємодії для кожного користувача. Ураховуючи ці показники, аналітики та розробники здатні точно визначити області для оптимізації, забезпечуючи максимальну ефективність веб-додатка та високий ступінь задоволення його користувачів.

Вимірювання Продуктивності JavaScript:



Рис. 2.11 Вимірювання Продуктивності JavaScript [26]

Ретельний аналіз продуктивності виконання JavaScript-коду та його вплив на динамічні елементи сторінки є ключовим етапом у вивченні ефективності веб-додатка. JavaScript відіграє важливу роль у розробці інтерактивних та динамічних веб-додатків, тому забезпечення його оптимальної продуктивності має вирішальне значення.

Під поняттям "продуктивність JavaScript" мається на увазі ефективність виконання скриптів, яка безпосередньо впливає на швидкість завантаження та взаємодії з користувачем. Під час аналізу роботи JavaScript-коду розробники можуть виявити та вирішити проблеми, пов'язані з його ефективністю, забезпечуючи плавну та приємну роботу додатка.

Динамічні елементи сторінки, такі як анімації, взаємодійні форми та оновлення контенту в реальному часі, визначаються якісним виконанням JavaScript. Відповідальність за оптимізацію та ефективність коду покладається на розробників, оскільки це має прямий вплив на користувацький досвід.

Ефективність JavaScript не лише гарантує високу продуктивність додатка, але й дозволяє зберігати низькі витрати ресурсів пристрою

користувача. Оптимізація коду сприяє прискоренню завантаження сторінок та зниженню часу відгуку, що є ключовим для забезпечення задоволення від використання веб-додатка в умовах постійно змінюючогося інтернет-середовища.

Визначення Метрик Користувацького Досвіду:



Рис. 2.12 Визначення Метрик Користувацького Досвіду [27]

На етапі визначення метрик користувацького досвіду проводиться оцінка двох основних факторів: часу відображення першого контенту та часу інтерактивності, які визначають ефективність та оперативність взаємодії користувача з веб-додатком.

Час відображення першого контенту (First Contentful Paint) є критичним для формування першого враження від веб-додатка, визначаючи, скільки часу потрібно користувачу для перегляду першого вмісту сторінки. Це є важливим фактором у задоволенні користувача та створенні позитивного враження від взаємодії.

Час інтерактивності (Time to Interactive) визначає момент, коли веб-додаток готовий для повноцінної взаємодії з користувачем. Це критичний показник, який визначає готовність додатка відповідати на команди користувача та взаємодіяти з ними.

Справедливе врахування цих метрик не обмежується лише вимірюванням часу, але й враховує конкретику додатка та його завдань. Наприклад, для новинного веб-дodatка швидке відображення актуальних новин може бути ключовим, тоді як для інтерактивної гри важливою буде максимальна оперативність відповіді на дії гравця.

Такий аналіз дозволяє розробникам налаштовувати параметри відображення та взаємодії для досягнення найкращого користувацького досвіду. Врахування цих метрик стає важливим кроком у вдосконаленні та оптимізації функціоналу веб-дodatка для задоволення потреб користувачів.

Оцінка Стійкості та Надійності:

Оцінка стабільності та надійності веб-дodatка включає в себе аналіз показника відсотка відмов, відомого як Bounce Rate. Цей показник вказує на те, який відсоток користувачів покидає сторінку без взаємодії або перегляду інших сторінок. Високий Bounce Rate може вказувати на проблеми, які потрібно вирішити для поліпшення користувацького досвіду та утримання відвідувачів на сайті.

Розуміння цього показника потребує врахування контексту конкретного додатка. Наприклад, для новинарського ресурсу високий Bounce Rate може бути природним, оскільки користувачі швидко переглядають заголовки та переходять до цікавих новин. У той час як для інтернет-магазину чи блогу важливо забезпечити те, щоб відвідувачі залишалися та взаємодіяли з більшою кількістю сторінок.

Чинники, що можуть впливати на високий Bounce Rate, варіюються від якості контенту до зручності дизайну чи швидкості завантаження сторінок. Такий аналіз показника дозволяє виявляти проблеми та вживати заходів для їх виправлення.

Зменшення Bounce Rate досягається оптимізацією веб-дodatка, поліпшенням контенту, забезпеченням зручної навігації та урахуванням

потреб аудиторії. Це сприяє збереженню та привертанню відвідувачів, зміцнюючи стабільність та надійність веб-додатка в онлайн-середовищі.

Дослідження Додаткових Параметрів:

Дослідження аспектів, таких як ефективність кешування та обсяг передачі даних між сервером та клієнтом, є важливою складовою процесу оцінки веб-додатка. Оцінка ефективності кешування визначає, наскільки добре додаток використовує збережені копії ресурсів для прискорення завантаження сторінок, особливо при повторних відвідуваннях.

Ефективне кешування зменшує час завантаження, оскільки браузер може використовувати локально збережені дані замість повторного запиту сервера. Це стає ключовим аспектом для поліпшення користувацького досвіду та оптимізації роботи сервера, зменшуючи його навантаження.

Обсяг передачі даних визначає кількість інформації, яку сервер передає користувачеві. Проведення аналізу цього параметра є важливим, особливо в умовах обмеженого інтернет-трафіку або при використанні мобільних пристроїв. Оптимізація обсягу передачі даних сприяє зниженню часу завантаження сторінок та покращенню продуктивності веб-додатка, особливо для користувачів з різними умовами використання інтернету.

Врахування цих параметрів дозволяє розробникам налагоджувати стратегії кешування та оптимізації передачі даних, забезпечуючи високу ефективність та оптимальну роботу веб-додатка в різних сценаріях використання.

Оцінка Використання Ресурсів:

Аналіз того, як веб-додаток використовує ресурси системи, включаючи центральний процесор (CPU), оперативну пам'ять та мережеві ресурси, є критично важливим етапом у визначенні його продуктивності. Розглядання, як програма взаємодіє з цими ресурсами, дозволяє отримати глибоке розуміння того, як ефективно вона функціонує.

Оцінка використання CPU стає ключовою для визначення того, наскільки добре додаток використовує обчислювальні ресурси. Надмірне навантаження CPU може мати негативний вплив на стабільність системи та призвести до зменшення загальної продуктивності.

Розглядання використання оперативної пам'яті дозволяє визначити, як ефективно програма управляє ресурсами пам'яті. Великий обсяг використовуваної пам'яті може свідчити про можливі проблеми з оптимізацією та впливати на продуктивність.

Аналіз мережевого використання вказує на те, наскільки добре програма обмінюється даними з сервером та іншими ресурсами. Ефективне використання мережі важливо для забезпечення швидкого завантаження сторінок та покращення загальної продуктивності додатка.

Систематичне вивчення всіх цих аспектів використання ресурсів дозволяє розробникам вдосконалювати код та здійснювати оптимізації для досягнення найвищого рівня продуктивності та задоволення від використання веб-додатка.

Вивчення Стабільності та Оптимізації для Мобільних Пристроїв:

Оцінка стійкості веб-сторінки та оптимізація для мобільних пристроїв є важливою складовою аналізу ефективності веб-додатка. Стабільність сторінки визначається надійністю та плавністю завантаження, що дозволяє уникнути негативних вражень, таких як стрибки чи зміщення. Це гарантує користувачеві позитивний та стабільний досвід взаємодії з додатком.

Мобільна оптимізація оцінює, наскільки ефективно веб-додаток пристосований до різних типів мобільних пристроїв та розмірів їхніх екранів. Це включає в себе адаптивний дизайн, зручність навігації на сенсорних екранах та загальну легкість використання на портативних пристроях.

Ефективна мобільна оптимізація гарантує, що всі функції додатка доступні користувачам без будь-яких труднощів, незалежно від типу

мобільного пристрою, чи це смартфон, планшет чи інший пристрій. Це стає ключовим для задоволення користувачів та створення комфортного досвіду використання додатка на різних пристроях та в різних умовах.

З урахуванням постійного зростання популярності мобільних пристроїв, стабільність сторінки та мобільна оптимізація стають стратегічно важливими аспектами для успішного функціонування та конкурентоспроможності веб-дodatка. Розробники повинні постійно вдосконалювати ці параметри, покращуючи якість відтворення на різних пристроях та забезпечуючи стабільну роботу під час мобільного використання.

Завершальний етап аналізу:

Завершальний етап аналізу веб-дodatка – це завдання, що належно розглядається з точки зору максимальної оптимізації та підвищення ефективності його функціонування. Цей етап є ключовим, оскільки визначає подальший успіх додатка, його відповідність високим стандартам та здатність забезпечувати задоволення користувачів.

На початку цього етапу проводиться глибокий аналіз всіх показників продуктивності, розглядаючи ключові метрики, такі як час завантаження сторінки, вимірювання продуктивності JavaScript та параметри користувацького досвіду. Визначення слабких та сильних сторін дозволяє розробникам точно зорієнтуватися на напрямки оптимізацій.

Особлива увага приділяється вимірюванню користувацького досвіду, який є основним критерієм задоволення користувачів. Аналіз часу відображення першого контенту та часу інтерактивності дає можливість не лише виявити проблемні моменти, а й розробити стратегії для покращення цих параметрів.

Далі, на основі отриманих даних, розробляється стратегія оптимізації, включаючи технічні покращення та корекції. Оптимізація коду, управління

ресурсами та інші технічні заходи допомагають досягти високого рівня продуктивності.

Контроль та моніторинг після впровадження змін є не менш важливим етапом. Розробники вивчають реакцію системи на нововведення, аналізуючи продуктивність та вчасно виявляючи можливі проблеми. Це гарантує стабільну та надійну роботу веб-додатка в реальних умовах використання.

У підсумку, завершальний етап аналізу не тільки вирішує конкретні технічні завдання з оптимізації, але і створює стратегію для подальшого розвитку та вдосконалення. Його успішність полягає в тому, як добре розробники адаптуються до виявлених викликів та впроваджують покращення для забезпечення високої якості та задоволення від використання веб-додатка.

3 ПРОТОТИПУВАННЯ ІНТЕРАКТИВНОГО ВЕБ-ДОДАТКУ ДЛЯ ПІДВИЩЕННЯ РІВНЯ КОРИСТУВАЦЬКОГО ДОСВІДУ

3.1 Визначення вимог інтерактивного веб-додатку для підвищення рівня користувацького досвіду

UX, або користувацький досвід, представляє собою комплексний підхід до створення продуктів, систем або послуг, який спрямований на задоволення потреб і очікувань користувачів через їх ефективне сприйняття, використання та взаємодію з ними. UX включає в себе ряд аспектів, які сприяють формуванню позитивного сприйняття користувачем продукту або послуги.

Основні складові UX включають:

1. Дослідження користувачів: Аналіз потреб, звичок, вподобань та характеристик цільової аудиторії для розуміння їхніх потреб та очікувань.
2. Інформаційна архітектура: Організація контенту та структури продукту для зручного та логічного сприйняття користувачами.
3. Дизайн інтерфейсу: Розробка інтерактивних елементів, які дозволяють користувачам легко взаємодіяти з продуктом або послугою.
4. Тестування користувацького досвіду: Проведення випробувань, щоб забезпечити, що продукт або послуга відповідають потребам користувачів і є інтуїтивно зрозумілими.
5. Відгуки користувачів: Збір та аналіз відгуків користувачів для подальшого вдосконалення продукту.

Існує декілька підходів та методик для покращення UX, таких як Design Thinking, Agile, Lean UX тощо. Важливою є постійна адаптація та вдосконалення UX через врахування змін у потребах користувачів та нових технологій.

Наприклад, якщо йдеться про веб-сайти, ефективний UX означає не лише звернення уваги на дизайн та естетику, але й на оптимізацію швидкості завантаження, легкість навігації, наявність корисного контенту,

адаптивний дизайн для різних пристроїв та інші фактори, що сприяють комфортному та ефективному використанню.

Загальний успіх бізнесу часто залежить від того, наскільки добре він розуміє та враховує потреби своїх користувачів, що робить UX ключовим аспектом в сучасному світі бізнесу та технологій.



Рис.3.1 UI/UX дизайн [28]

UX дизайн можна розділити на такі категорії:

1. Дизайн взаємодії (IxD) - це складова UX-дизайну, яка фокусується на способах, якими люди взаємодіють з продуктом або сервісом. Його мета полягає в створенні оптимального, зручного та приємного досвіду для користувачів під час їх взаємодії з продуктом. Це охоплює різні аспекти, такі як навігація, анімація, реакція на взаємодію користувача, розташування елементів на екрані, графічний дизайн елементів та інші фактори.

В дизайні взаємодії велика увага приділяється спрощенню процесу взаємодії користувача з продуктом, зробленню його легким у використанні та зрозумілим для широкого кола користувачів. Це може включати створення зручних інтерфейсів, які допомагають користувачам швидко

здійснювати дії, а також розробку систем навігації, що полегшують переміщення користувачів по продукту.

На практичному рівні дизайн взаємодії може означати створення інтерактивних прототипів, їх тестування з реальними користувачами, аналіз отриманих відгуків та постійне вдосконалення взаємодії на основі цих даних. Головна мета полягає в тому, щоб зробити користування продуктом чи сервісом максимально зручним, приємним та ефективним для кожного користувача.

2. Візуальний дизайн - це область, де творці створюють естетично приємні, функціональні та зрозумілі образи, використовуючи різноманітні елементи, такі як ілюстрації, фотографії, типографіка, простір, макети та колір. Основна мета візуального дизайну - це поліпшення користувацького досвіду, подання інформації зрозумілою та привабливою для сприйняття.



shutterstock.com · 1770464108

Рис.3.2 Візуальний дизайн [29]

Принципи художнього дизайну, такі як баланс, простір і контраст, відіграють вирішальну роль у візуальному дизайні. Баланс дозволяє розташувати елементи так, щоб вони створювали відчуття стабільності і

гармонії. Простір визначає відношення між об'єктами, створюючи візуальну організацію. Контраст додає цікавості та виокремлює елементи один від одного.

Колір, форма, розмір і інші елементи також є ключовими складовими візуального дизайну. Кольори можуть впливати на емоційний стан користувача і керувати його увагою. Форма та розмір визначають вигляд та відчуття об'єктів на макеті або сторінці. Усі ці елементи спільно використовуються для створення привабливого, функціонального та зрозумілого дизайну, який сприяє кращому сприйняттю інформації користувачем.

3. Дослідження користувачів - це процес збору, аналізу та розуміння інформації про людей, які використовують ваш продукт або послугу. Це ключовий етап у розробці UX (користувацького досвіду), оскільки дозволяє зрозуміти потреби, уподобання, проблеми та поведінку цільової аудиторії. Це може включати різні методи дослідження, такі як опитування, спостереження, інтерв'ю, аналіз даних тощо.

What tools do you use to catalog and share your findings?

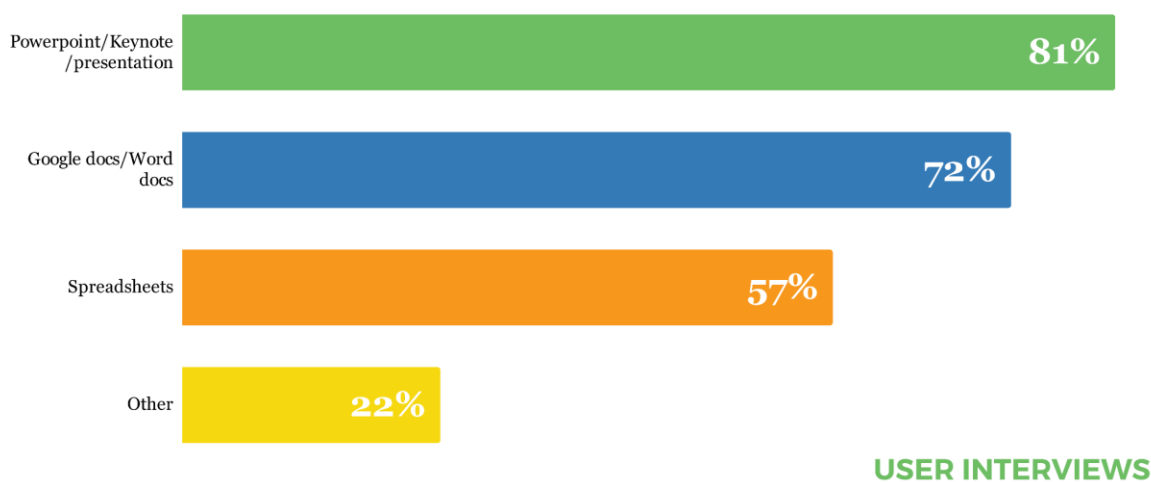


Рис.3.3 Приклад дослідження користувачів [30]

Під час дослідження користувачів важливо збирати різноманітну інформацію: від вимог до функціональності продукту до тих відчуттів, які

він викликає у користувачів. Це дає змогу дизайнерам та розробникам краще зрозуміти, як покращити продукт так, щоб він був зручним, корисним і приємним для своїх користувачів.

Без дослідження користувачів, розробка продукту або веб-сайту може базуватися на припущеннях, що часто призводить до невдалих рішень та незадоволення користувачів. Ретельне дослідження дозволяє зменшити ризики помилок та забезпечити більш точне співвідношення між тим, що пропонується і тим, що потрібно.

4. Дизайнери застосовують концепцію інформаційної архітектури для систематизації та організації вмісту з метою спрощення пошуку інформації для користувачів. Цей підхід використовується на веб-сайтах, у мобільних додатках, на смартфонах і навіть у фізичних місцях, які ми відвідуємо. Простота використання та доступність вважаються ключовими аспектами інформаційної архітектури, що робить її важливим елементом UX-дизайну.

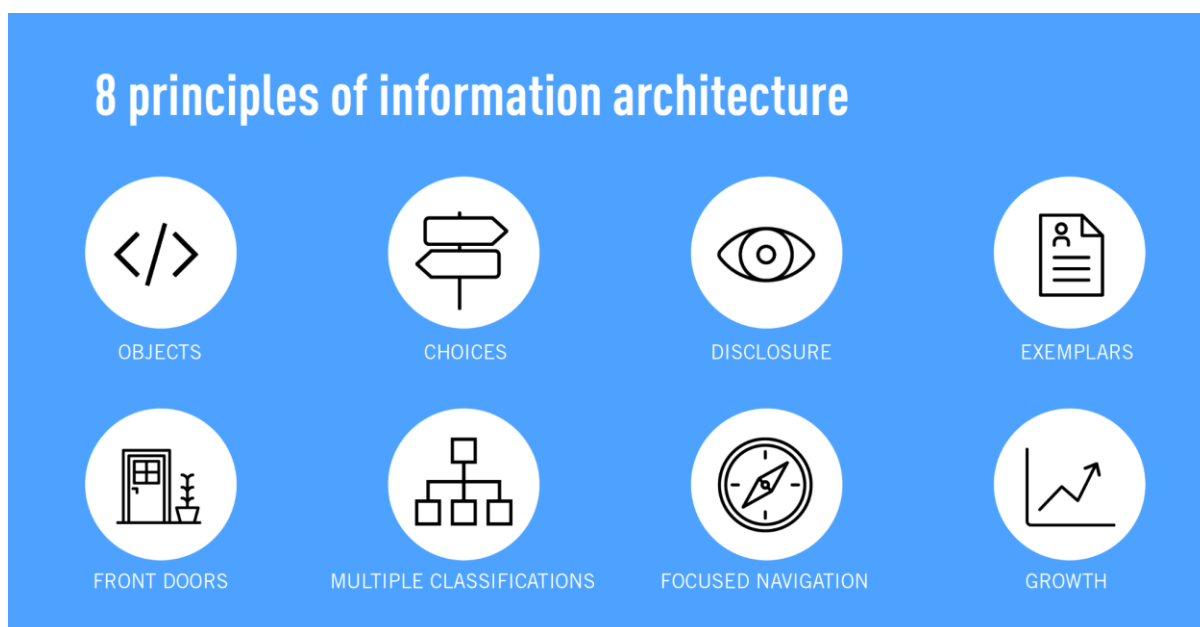


Рис.3.4 Принципи інформаційної архітектури [31]

3.2 Проектування та дизайн прототипу

Для розробки прототипу було використано застосунок Figma (рис. 3.1). Figma - ідеальний інструмент для веб-дизайнерів, що надає зручність та можливість спільної роботи над графічними проектами. Унікальність полягає в тому, що всі проекти можна зберігати в хмарі, що полегшує колективну роботу, адже весь процес можна вести через зручне посилання, без необхідності завантаження файлів.

Основною перевагою Figma є можливість візуальної спільної роботи в реальному часі. Кожен учасник може внести зміни, спостерігаючи за роботою інших членів команди, що полегшує взаємодію та спільне творчий процес.

Зазначене також, що робота в хмарі дозволяє уникнути зайвих етапів завантаження або обміну файлами. Просто надішліть посилання на проект, і ваша команда може приступити до спільної роботи, миттєво внісши зміни та удосконалюючи дизайн.

Figma - це не просто інструмент, але й інноваційний крок у сфері веб-дизайну. Його можливості забезпечують швидкість, простоту та ефективну спільну роботу, роблячи його незамінним для творців, які цінують якість та результативність своєї роботи.

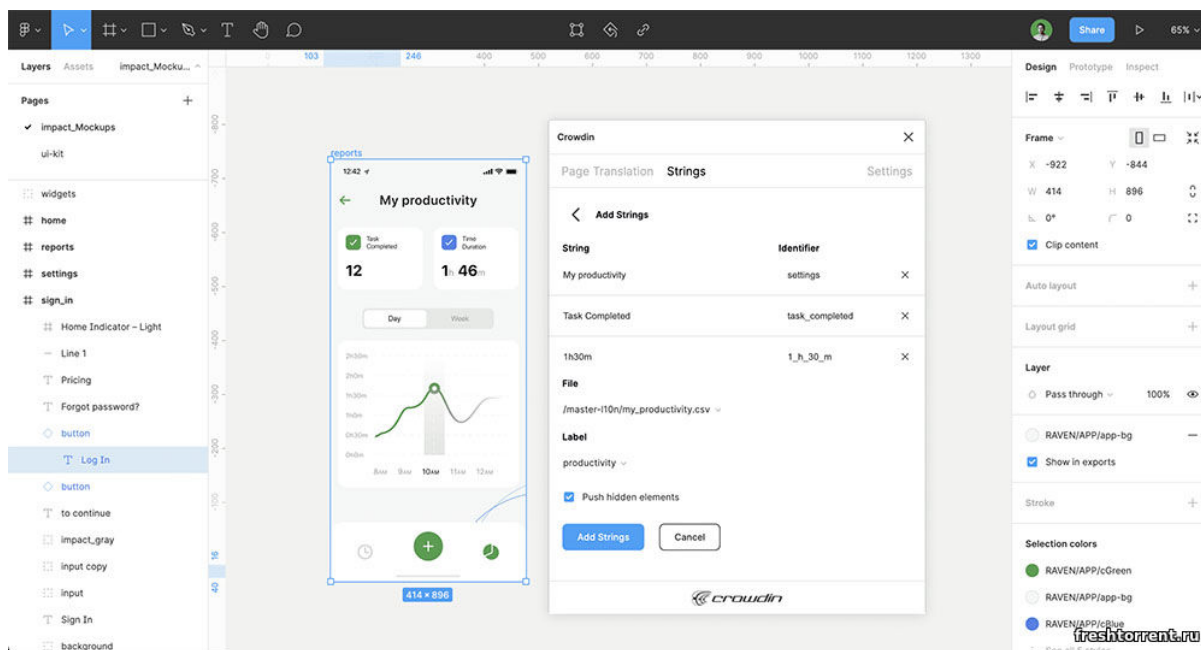


Рис.3.5 Інтерфейс додатку Figma [32]

Можна користуватися Figma через браузер або завантажити програму на свій комп'ютер (доступно для Windows і Mac). Якщо обрано другий варіант, то матимете можливість редагувати проекти навіть офлайн, а підключившись до мережі, всі зміни автоматично синхронізуються.

Цей редактор дозволяє створювати:

- векторні ілюстрації;
- іконки, меню, вікна, кнопки та інші елементи інтерфейсу;
- інтерактивні прототипи для мобільних додатків та веб-сайтів.

CodeCube - це платформа для онлайн навчання програмування, яка пропонує унікальний та ефективний досвід вивчення.

Вона надає:

1. Різноманітність курсів: CodeCube пропонує широкий вибір курсів з різних мов програмування, таких як Python, JavaScript, Java, C++, а також з фреймворків та інструментів розробки.
2. Персоналізований підхід: Платформа індивідуалізує навчання, враховуючи потреби та рівень знань кожного студента. Користувачі можуть вибирати швидкість та складність курсів.
3. Інтерактивність: CodeCube дозволяє студентам практикувати навички безпосередньо на платформі через інтерактивні завдання, виклики та проекти.
4. Спільнота: Студенти мають доступ до форумів для обговорень, обміну порадами та взаємодопомоги, що сприяє створенню спільноти для спільного навчання.

5. Відстеження прогресу: Платформа надає інструменти для контролю над власним прогресом - від завершених курсів до досягнень та результатів тестів.

6. Адаптивний дизайн: CodeCube легко користується на будь-яких пристроях, включаючи комп'ютери, планшети та смартфони, завдяки своєму адаптивному інтерфейсу.

Ця платформа створена для активного навчання програмування, де студенти не лише засвоюють теорію, а й відразу застосовують отримані знання у практичних завданнях.

Для веб-додатку CodeCube було створено зручний та зрозумілий інтерфейс. Розглянемо спершу головний екран додатку:

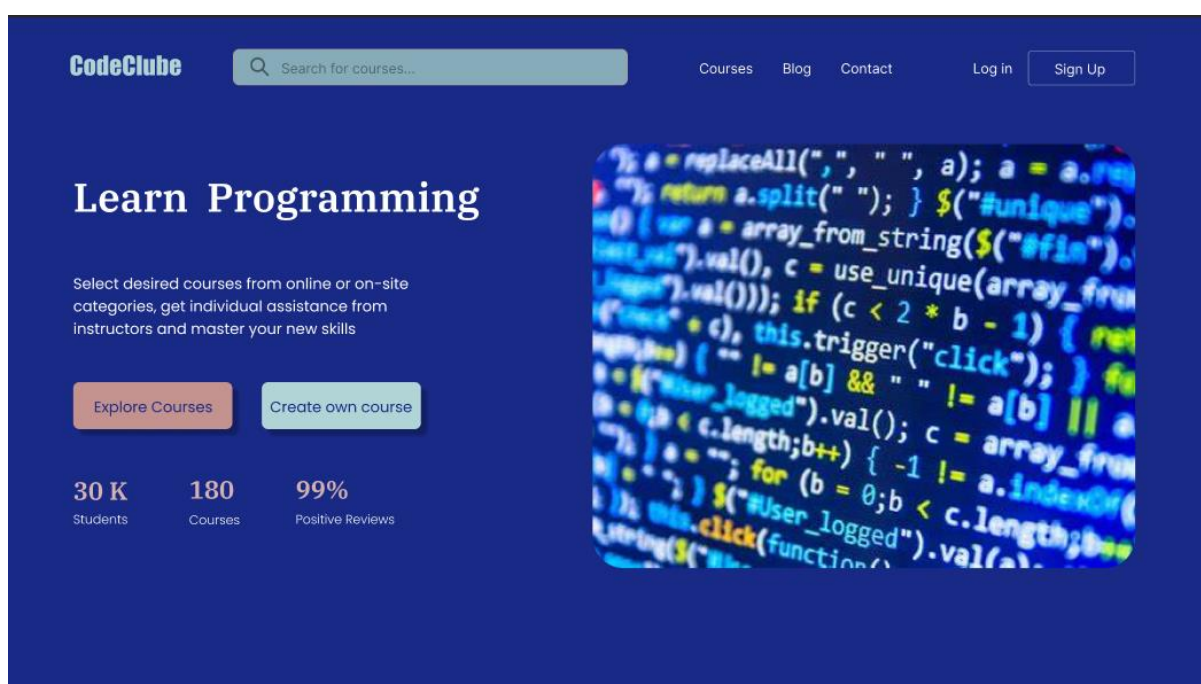


Рис.3.6 Головний екран веб-додатку CodeCube

На головному екрані є меню, де можна відкрити вкладки курсів, блогу, контактів або зареєструватися/увійти в акаунт.

Ця сторінка (рис.3.1) - обличчя веб-додатку, отже задля підвищення користувацького досвіду та щоб запобігти перевантаженню користувача, ця сторінка містить мінімум тексту та тільки корисну та необхідну інформацію.

Також є дві кнопки - Дізнатися про курси та створити власний курс, отже, розглянемо їх детальніше:

- Кнопка Дізнатися про курси.

Кнопка, що автоматично переносить на вкладку з доступними онлайн курсами та тренінгами.

- Кнопка Створити власний курс.

Ця кнопка (рис. 3.1) надає можливість користувачу створити та завантажити особисто створений курс, але він буде допущений на платформу тільки після перевірки модератором. Також створення власного курсу та подовження терміну розміщення на платформі буде платним, у вигляді підписки.

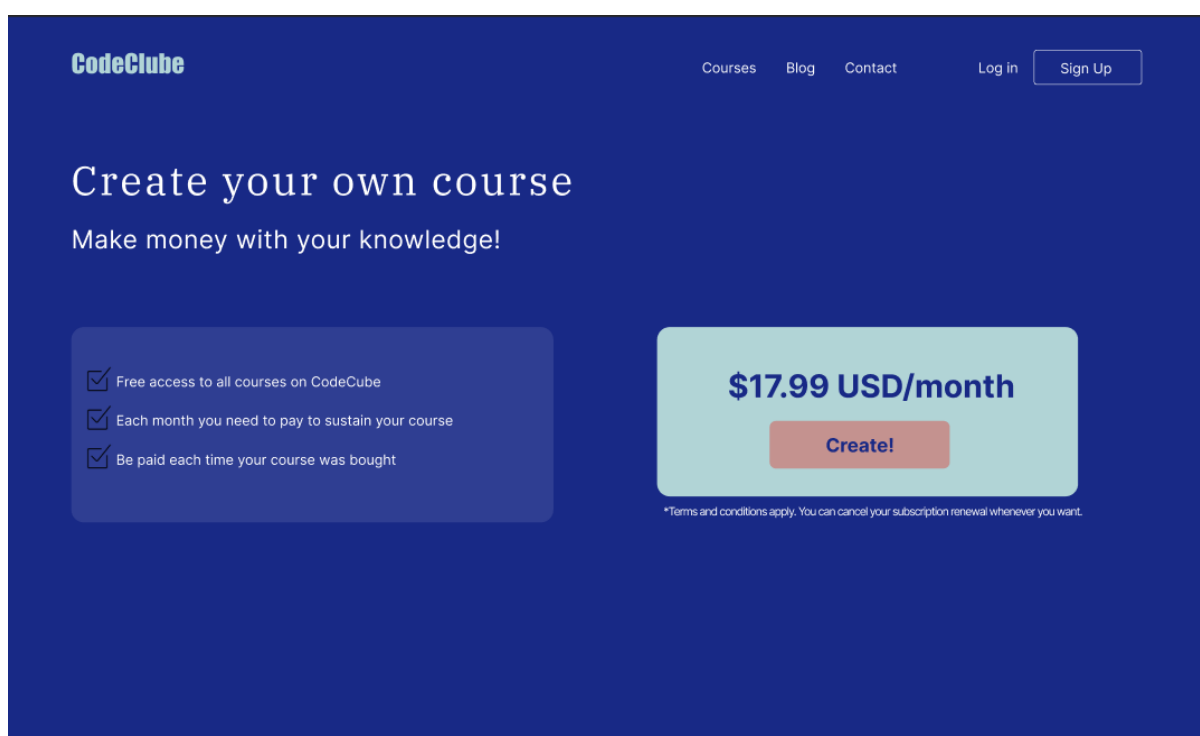


Рис.3.7 Сторінка створення особистого курсу

Щодо вигляду сторінки Courses (Курси) (рис. 3.1) - це сторінка з усіма курсами, де можна побачити короткий опис курсу, його назву, кількість людей, яким сподобався курс та кількість покупок курсу.

CodeCube [Courses](#) [Blog](#) [Contact](#) [Log in](#) [Sign Up](#)

Courses

- All Courses
- Online Courses**
- On-Site Courses
- Entrepreneurship Courses
- New Courses
- Top Rated Courses


Categories

- SQL
- PYTHON
- C++
- C#
- JavaScript
- Html

Filters

- Duration
- Type
- Provider
- Price

Online Programming courses Sort by: **Popular** ▼




MongoDB for total nubies

A course by Paul Wills

Learn one of the most popular frameworks for working with SQL and databases

👤 300 🌟 98%

Enroll for \$19.99




JavaScript Mastery: Unleashing Dynamic Web Development

This course provides a comprehensive journey through JavaScript, from its fundamental syntax to advanced concepts, empowering you to create dynamic and interactive web applications.

👤 540 🌟 95%

Enroll for \$17.99




Python Mastery Series: From Basics to Brilliance

Unlock the power of Python with our comprehensive course series designed for all levels. From foundational concepts to advanced techniques, master the language that drives innovation across industries.

👤 384 🌟 100%

Enroll for \$17.99




C++ Mastery: Unleashing Power Through Code

This course takes you on a journey through C++ programming, from the foundational concepts to advanced techniques.

👤 325 🌟 98%

Enroll for \$17.99




SQL Fundamentals: Mastering Database Queries

Dive into the essential principles and commands of SQL, gaining proficiency in querying databases efficiently and effectively.

👤 875 🌟 95%

Enroll for \$17.99



Sharp Skills: Mastering C#

Dive into the world of C# with our comprehensive courses designed to take you from beginner to advanced levels.

👤 675 🌟 100%

Enroll for \$17.99

Scroll Down to see more courses
↓

Рис.3.8 Сторінка курсів

3.3 Програмне забезпечення для створення веб-додатку CodeCube

Прототип веб-додатку CodeCube був створений з використанням HTML та CSS мов програмування, що відповідають за створення візуальної складової коду.

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CodeCube</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }
  </style>

```

Рис.3.9 Створення базової структури документу

Цей фрагмент коду (рис. 3.9) встановлює базову структуру HTML-документа та визначає його основні елементи та налаштування.

1. `<!DOCTYPE html>`: Це вказівка для браузера про тип документа, конкретно - HTML5. Розміщується спочатку файлу і допомагає браузеру правильно обробляти код.
2. `<html lang="uk">`: Це початок HTML-документа. Елемент `<html>` є кореневим і містить усі інші HTML-елементи. Атрибут `lang="uk"` вказує на українську мову контенту.
3. `<head>`: Це розділ документа, що містить метадані, такі як метатеги, стилі, посилання на скрипти та заголовок документа.
4. `<meta charset="UTF-8">`: Цей метатег показує, що символи у документі кодуються за стандартом UTF-8, що дозволяє коректне відображення символів різних мов.
5. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Цей метатег налаштовує відображення на мобільних пристроях, вказуючи на ширину сторінки, яка відповідає ширині пристрою, та масштаб 1:1.
6. `<title>CodeClube</title>`: Це заголовок документа, що відображається у вкладці браузера чи на вікні сторінки.
7. `<style>`: Цей тег починає розділ стилів CSS, які впливають на вміст документа CSS код можна побачити на рисунку 3.1.

Таким чином, цей HTML-код формує початкову структуру документа, визначає метадані та заголовок сторінки, а також містить початок розділу стилів, де можуть бути визначені додаткові стилі для відображення елементів на сторінці.

```

header {
  background-color: #333;
  color: white;
  padding: 10px 20px;
  text-align: center;
}

nav {
  margin: 10px 0;
}

nav ul {
  list-style-type: none;
}

nav ul li {
  display: inline;
  margin-right: 10px;
}
</style>
</head>
<body>
<header>
  CodeClube
</header>
<nav>
  <ul>
    <li>Курси</li>
    <li>Блог</li>
    <li>Контакти</li>
  </ul>
</nav>

```

Рис.3.10 Створення хедеру веб-додатку

Цей код HTML (рис. 3.10) описує основну структуру сторінки веб-сайту, присвяченої навчанню програмуванню.

Давай розглянемо детальніше його складові елементи та їхнє призначення:

1. `<main>`: Визначає основний зміст сторінки.
2. `<h1>`: Заголовок першого рівня, який акцентує увагу на головній темі сторінки - "Вивчайте програмування".
3. `<p>`: Абзаци тексту, що описують переваги платформи навчання. Заохочують вибір курсів, отримання підтримки від викладачів та отримання нових навичок.
4. `<button>`: Дві кнопки, які пропонують вибір курсів або створення власних. Спрямовані на стимулювання активності користувачів на сайті.
5. `<p>`: Додатковий абзац, що містить короткий огляд статистики платформи: кількість студентів, курсів і відсоток позитивних відгуків.

6. ````: Елемент для вставки зображення коду. Має атрибут ``src`` з посиланням на файл із зображенням та ``alt`` для альтернативного тексту у випадку недоступності зображення.

```

<main>

<h1>Вивчайте програмування</h1>
<p>Виберіть бажані курси з онлайн або офлайн категорій,
отримуйте індивідуальну допомогу від викладачів і освоюйте
нові навички.</p>
<button>Дослідити курси</button><button>Створити власний курс</button>
<p>30К Студентів | 180 Курсів | 99% Позитивних відгуків </p>


</main>

```

Рис.3.11 Створення основної частини

Даний HTML-код описує складові основної частини (`<main>`) веб-сторінки, де розміщений різноманітний контент:

1. ``<h1>`` - це заголовок першого рівня з текстом "Вивчайте програмування". Використовується для основного заголовку сторінки.
2. ``<p>`` - абзац тексту з описом, що стимулює вибір курсів онлайн або офлайн, надає можливість індивідуальної допомоги від викладачів і розвитку нових навичок.
3. ``<button>`` - кнопки для переходу на інші сторінки чи виконання дій: Дослідити курси і Створити власний курс.
4. ``<p>``, що містить коротку статистику про кількість студентів, курсів та відсоток позитивних відгуків.
5. ```` - тэг для вставки зображення. ``src`` містить шлях до файлу зображення, а ``alt`` - альтернативний текст для зображення, який відображається при неможливості завантаження чи відображення зображення.

```

body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
}

header {
  background-color: #333;
  color: white;
  padding: 10px 20px;
  text-align: center;
}

nav {
  margin: 10px 0;
}

nav ul {
  list-style-type: none;
}

nav ul li {
  display: inline;
  margin-right: 10px;
  border: 1px solid #333;
  padding: 5px 10px;
}

nav ul li:hover {
  background-color: #333;
  color: white;
}

main {
  padding: 20px;
}

h1 {
  font-size: 36px;
  font-weight: bold;
  color: #333;
}

p {
  font-size: 18px;
  line-height: 1.5;
}

button {
  font-size: 18px;
  background-color: #333;
  color: white;
  border: none;
  padding: 10px 20px;
  margin: 10px;
  cursor: pointer;
}

```

```

button:hover {
  background-color: #555;
}

img {
  width: 300px;
  height: 200px;
  border: 1px solid #333;
  margin: 10px;
}

body {
  font-family: Arial, sans-serif;
}

header {
  background-color: #333;
  color: white;
  padding: 10px 20px;
  text-align: center;
}

nav {
  margin: 10px 0;
}

nav ul {
  list-style-type: none;
}

nav ul li {
  display: inline;
  margin-right: 10px;
}

</style>
</head>
<body>

```

Рис. 3.12 Код CSS

Цей набір CSS правил визначає, як будуть виглядати різні елементи HTML на сторінці. Він задає шрифти, кольори, відступи та інші властивості, щоб створити привабливий вигляд та забезпечити зручність користування.

1. `body`: налаштовує шрифт на Arial або альтернативний sans-serif, встановлює світло-сірий фон для всієї сторінки.
2. `header`: стилізує заголовок, встановлює темний фон, білий текст, вирівнює його по центру.
3. `nav`: задає зовнішні відступи для блоку навігації.
4. `nav ul`: прибирає стандартні маркери списку.
5. `nav ul li`: оформляє кожен пункт навігаційного списку, вирівнює їх в ряд, додає чорну рамку та встановлює внутрішні відступи.
6. `nav ul li:hover`: визначає зміни, які відбуваються при наведенні на пункт меню - змінює колір фону та тексту.
7. `main`: задає внутрішні відступи для основного контенту сторінки.
8. `h1`: стилізує заголовок першого рівня - встановлює великий розмір, жирний шрифт та чорний колір тексту.
9. `p`: визначає стилі для абзаців - встановлює середній розмір шрифту, інтервали між рядками та чорний колір тексту.
10. `button`: оформляє кнопки на сторінці - встановлює розмір шрифту, кольори, відступи та змінює курсор при наведенні.
11. `button:hover`: змінює колір фону при наведенні на кнопку.
12. `img`: встановлює розміри зображень, рамку та відступи навколо них.

Ці стилі призначені для створення привабливого та зручного вигляду веб-сторінки, з урахуванням навігації, розміщення тексту та зображень, а також функціональних кнопок.

3.4 Аналіз ефективності додатку CodeCube

Задля аналізу функціональності додатку ми створили чотири неупереджені групи людей по дванадцять людей у групі різних вікових категорій. Опитуванні використовували прототип веб-сайту протягом трьох тижнів та у першій та останній тижні користування відповідали на запити щодо функціональності веб-додатку. На рисунку 3.1 отримані результати дослідження.

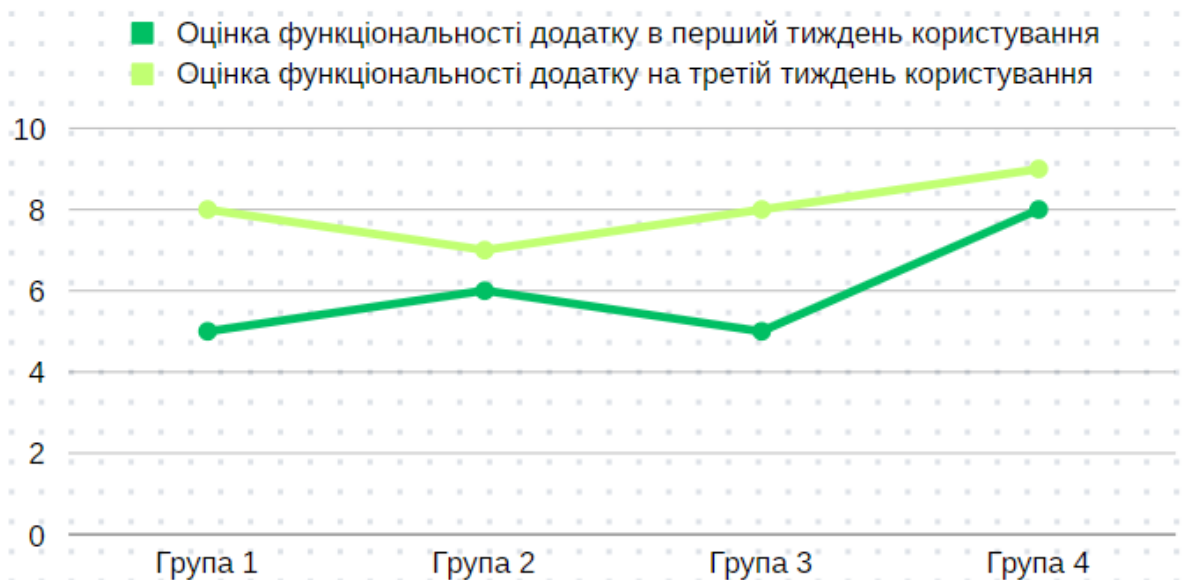


Рис. 3.13 Оцінка функціональності додатку

- Група 1: Найбільший приріст задоволеності функціональністю додатка за три тижні, зросла на 62,5%. Середній рейтинг за цей період становив 5,25.
- Група 2: Також значно піднялася відчуття задоволення, підвищивши свій рейтинг на 60% протягом трьох тижнів. Середній бал групи 2 за цей період становив 6,5.
- Група 3: Не виявила великих змін у рівні задоволеності, залишаючись на рівні трохи вище 6 протягом трьох тижнів. Середній бал групи 3 за цей період склав 6,125.
- Група 4: Мала найвищий початковий рейтинг задоволеності, трохи більше 8, і зберегла його протягом трьох тижнів. Середній бал групи 4 за цей період становив 8,125.

Ці дані підкреслюють, як певні групи відреагували на зміни протягом трьох тижнів та їхню відносну стабільність чи зміни у рівні задоволеності.

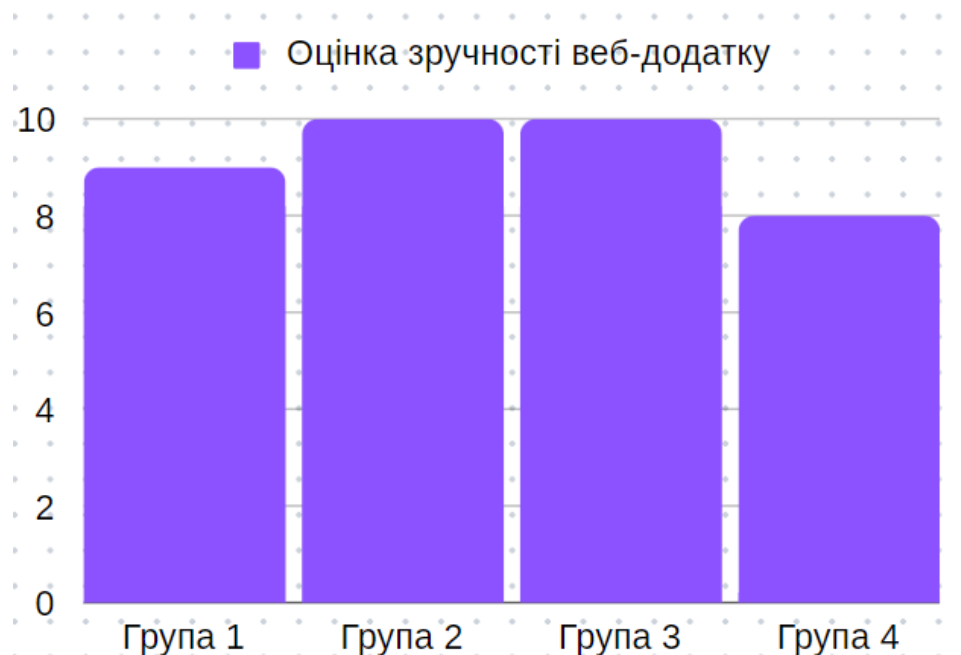


Рис.3.14 Оцінка зручності веб-додатку

Статистика на рисунку 3.1 була зібрана в тих самих трьох груп, максимальна оцінка 10, мінімальна оцінка 1. Групи відповідали на питання щодо зручності у користуванні додатком і було підраховано середні оцінки за всіма питаннями.

- Група 1: високе оцінювання зручності веб-додатка, проте нижче - близько 9 з 10. Це може вказувати на те, що ця група стикалася з певними труднощами у використанні додатка або очікувала більшого рівня зручності.
- Група 2: отримала найвищі оцінки зручності веб-додатка - 10 з 10. Це свідчить про повне задоволення цієї групи додатком і відсутність будь-яких проблем у його користуванні.
- Група 3: також отримала максимальну оцінку зручності веб-додатка - 10 з 10. Це може свідчити про те, що ця група також була дуже задоволена додатком і вважала його легким у використанні.

- Група 4: оцінка зручності веб-додатка майже на рівні групи 1, близько 8 з 10. Це може вказувати на те, що ця група також зазнала певних труднощів у використанні додатка або мала високі вимоги до його функціоналу.

Підсумовуючи, дані статистики показують, що більшість груп були вкрай задоволені рівнем зручності веб-додатка для онлайн-курсів з програмування. Проте деякі з них, можливо, мали б кращий досвід, якби додаток був більш пристосований до їхніх потреб.

ВИСНОВКИ

Висновок дослідження підтверджує великий потенціал сучасних інструментів веб-розробки, зокрема фреймворків та бібліотек, у покращенні враження користувачів. Проведені експерименти свідчать, що певні рішення у цій сфері спроможні створювати веб-додатки високої якості з

інтуїтивним інтерфейсом, що знаходять позитивний відгук серед користувачів.

Дослідження поглядів фахівців і відгуків користувачів підтверджує великий вплив обраних інструментів на поліпшення користувацького досвіду. Це підкреслює важливість правильного вибору інструментів у процесі веб-розробки для досягнення бажаного результату.

Аналіз різних веб-інструментів у реальних проектах розробки дозволив виявити їхні переваги та недоліки. Це допомогло знайти оптимальні і оптимізовані рішення для створення веб-застосунків, які найкращим чином відповідають потребам користувачів.

Усе це підкреслює важливість проведення досліджень та правильного вибору інструментів для забезпечення високої якості користувацького досвіду у веб-розробці. Обраний набір технологій відіграє важливу роль у створенні веб-додатків, які відповідають потребам та очікуванням користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Фреймворк: що таке, навіщо потрібен, типи *framework* - як вибрати фреймворк для веб-розробки сайту і як ним користуватися.
Фреймворк: що таке, навіщо потрібен, типи framework - як вибрати фреймворк для веб-розробки сайту і як ним користуватися.

- [Електронний ресурс] - Режим доступу до ресурсу:
<https://practicum.yandex/blog/chto-takoe-framework/>.
- 2) Javascript фреймворки. Що це? Навіщо? Чому?. *Створення сайту в Києві, розробка сайтів під ключ веб-студія Рекламаре*. [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.reclamare.ua/blog/javascript-frejmvorki/>.
- 3) Main Page. *Desta LTD*. URL: <https://desta.software/>.
- 4) 7 найпопулярніших фреймворків JavaScript. *Timeweb* - *7 найпопулярніших фреймворків JavaScript*. [Електронний ресурс] - Режим доступу до ресурсу:
<https://timeweb.com/ru/community/articles/7-samyh-populyarnyh-freymvorkov-javascript-v-2017-godu-1>.
- 5) Топ 18 фреймворков и библиотек JavaScript в 2021 году – Plerdy. *Plerdy – tools to improve conversions, usability analysis and SEO*. URL:
<https://www.plerdy.com/ru/blog/top-javascript-frameworks-and-libraries/>.
- 6) PEREIRA L. D. S. Python + Flask - Part 2 - HTTP Methods. *DEV Community*. [Електронний ресурс] - Режим доступу до ресурсу:
<https://dev.to/lucianopereira86/python-flask-part-2-http-methods-15hm>.
- 7) *Ranktracker: The all-in-one platform for effective SEO*. [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.ranktracker.com/uk/blog/user-experience-and-conversion-rate-optimization-key-principles-for-success/>.
- 8) Juviler J. 20 Simple Website Templates You Can Try Today. *HubSpot Blog / Marketing, Sales, Agency, and Customer Success Content*. [Електронний ресурс] - Режим доступу до ресурсу:
<https://blog.hubspot.com/website/simple-website-templates>.

- 9) Is this typography correct?. *Reddit*. [Электронный ресурс] - Режим доступа до ресурсу: https://www.reddit.com/r/typography/comments/3835xf/is_this_typographically_correct/?rdt=50640.
- 10) Usability Testing 101. *Nielsen Norman Group*. [Электронный ресурс] - Режим доступа до ресурсу: <https://www.nngroup.com/articles/usability-testing-101/>.
- 11) Guide to Web Application Development in 2024. *Insights - Web and Mobile Development Services and Solutions*. [Электронный ресурс] - Режим доступа до ресурсу: <https://www.netsolutions.com/insights/web-application-development-guide/>.
- 12) Gottfried J. A complete guide to modern web applications. *Medium*. [Электронный ресурс] - Режим доступа до ресурсу: <https://medium.com/jeremy-gottfrieds-tech-blog/a-complete-guide-to-modern-web-applications-793ae71b57ad>.
- 13) Zaigo Infotech Software Solutions Pvt Ltd. Top 7 benefits of Web Application Development for Business?. *LinkedIn: Log In or Sign Up*. [Электронный ресурс] - Режим доступа до ресурсу: <https://www.linkedin.com/pulse/top-7-benefits-web-application-development/>.
- 14) MySQL logo and symbol, meaning, history, PNG. *1000 Logos - The Famous logos and Popular company logos in the World*. [Электронный ресурс] - Режим доступа до ресурсу: <https://1000logos.net/mysql-logo/>.
- 15) MongoDB: The Developer Data Platform. *MongoDB*. [Электронный ресурс] - Режим доступа до ресурсу: <https://www.mongodb.com/>.
- 16) PostgreSQL logo and symbol, meaning, history, PNG. *1000 Logos - The Famous logos and Popular company logos in the World*.

- [Електронний ресурс] - Режим доступу до ресурсу:
<https://1000logos.net/postgresql-logo/>.
- 17) Git Storage: Comparing GitHub, GitLab, and Bitbucket. *Sparkbox - Web Design & Development Partner and IT Consultants*. [Електронний ресурс] - Режим доступу до ресурсу:
https://sparkbox.com/foundry/github_vs_gitlab_vs_bitbucket.
- 18) Git Storage: Comparing GitHub, GitLab, and Bitbucket. *Sparkbox - Web Design & Development Partner and IT Consultants*. [Електронний ресурс] - Режим доступу до ресурсу:
https://sparkbox.com/foundry/github_vs_gitlab_vs_bitbucket.
- 19) AWS logo 2. *JDRF*. [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.jdrf.org/blog/2020/12/09/jdrf-amazon-web-services-imagine-grant/aws-logo-2/>.
- 20) Download Microsoft Azure (Windows Azure) Logo in SVG Vector or PNG File Format. *Brand Logos Free Download in SVG Vector & PNG File Format - Logo.wine*. [Електронний ресурс] - Режим доступу до ресурсу:
https://www.logo.wine/logo/Microsoft_Azure.
- 21) Google Lighthouse. *Award Winning Web Design Newport - Digital Marketing*. [Електронний ресурс] - Режим доступу до ресурсу:
<https://www.icwebdesign.co.uk/google-lighthouse>.
- 22) Webpagetest Status. Webpagetest Status. [Електронний ресурс] - Режим доступу до ресурсу:
<https://status.webpagetest.org/>.
- 23) Показники марнославства: метрики, які заважають оцінити ефективність бізнесу - блог Webpromoeexperts. *Навчання в школі інтернет маркетингу WebPromoExperts | Курси для маркетолога*. [Електронний ресурс] - Режим доступу до ресурсу:
<https://webpromoeexperts.net/ua/blog/pokazateli-tseslavia-metriki-kotorye-mesaut-ocenit-effektivnost-biznesa/>.

- 24) 40 корисних інструментів для роботи з веб-аналітикою - TutHost.ua. *TutHost.ua*. [Електронний ресурс] - Режим доступу до ресурсу: <https://tuthost.ua/uk/blog/40-korisnih-instrumentiv-dlya-roboti-z-veb-analitikoju/>.
- 25) Махум Z. Метрики та ключові показники ефективності (Key Performance Indicators - KPIs). *Махум Zosum*. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.maxzosim.com/key-performance-indicators-kpis/>.
- 26) *Tonyline*. [Електронний ресурс] - Режим доступу до ресурсу: <https://tonyline.com.ua/blog/site-speed-optimization/>.
- 27) Базові метрики моніторингу соціальних медіа. *YouScan*. [Електронний ресурс] - Режим доступу до ресурсу: <https://youscan.io/ua/blog/top-metrics-in-social-media-monitoring/>.
- 28) *Hire Software Developers | Hire Software Programmers in India*. [Електронний ресурс] - Режим доступу до ресурсу: <https://www.appsdevpro.com/blog/wp-content/uploads/2022/06/Ui-ux-cover-ime.jpg>
- 29) 279 рез. за запитом "Eisenhower matrix" - зображення, стокові фотографії, тривимірні об'єкти і векторна графіка | Shutterstock [Електронний ресурс] - Режим доступу до ресурсу: https://assets-global.website-files.com/59b1667dd2e65000019d07be/5e5539b0b845e4ff652f1010_State%20of%20User%20Research%20Report%202020-%20Q52%20-%20What%20tools%20do%20you%20use%20to%20catalog%20and%20share%20your%20findings_.png
- 30) A Beginner's Guide To Information Architecture in UX [2023]. *CareerFoundry*. [Електронний ресурс] - Режим доступу до ресурсу: <https://careerfoundry.com/en/blog/ux-design/a-beginners-guide-to-information-architecture/>.

- 31) Figma скачать торрент бесплатно на русском Фигма (Windows).
Скачать бесплатно новинки софта, программы, игры - Главная страница.[Электронный ресурс] - Режим доступа до ресурсу:
<https://freshtorrent.ru/load/soft/sapr/figma/42-1-0-25224>.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

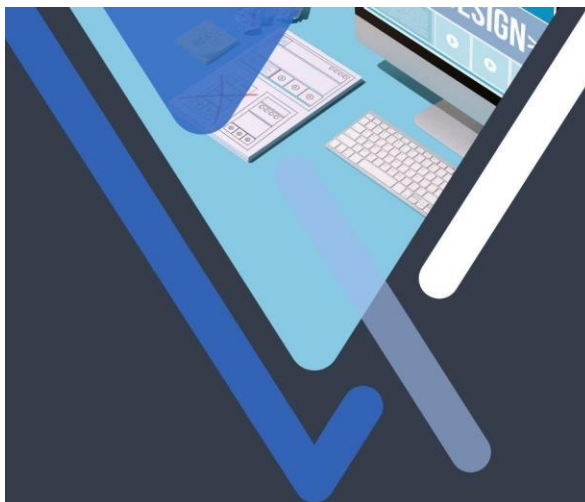
**КВАЛІФІКАЦІЙНА РОБОТА
НА ТЕМУ:
"ДОСЛІДЖЕННЯ ТА РОЗРОБКА
ІНТЕРАКТИВНИХ ВЕБ-ДОДАТКІВ З
ВИКОРИСТАННЯМ СУЧАСНИХ ФРЕЙМВОРКІВ
ТА БІБЛІОТЕК ДЛЯ ПІДВИЩЕННЯ РІВНЯ
КОРИСТУВАЦЬКОГО ДОСВІДУ"**

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи
та технології
освітньо-професійної програми
Інформаційні системи та технології

Виконав: здобувач вищої освіти гр. ІСДМ-62
Артем ГАРБЕЦЬКИЙ
Керівник: доктор філософії, доцент кафедри ІПЗАС
Олег СЕНЬКОВ



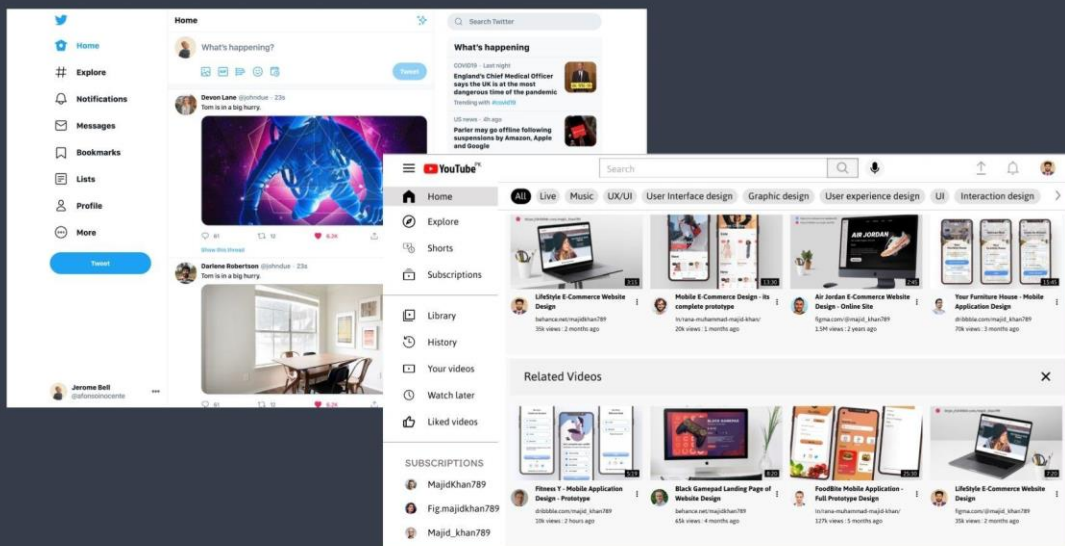
АКТУАЛЬНІСТЬ	Створення інтерактивних веб-додатків з новітніми технологіями - ключ до задоволення потреб користувачів та конкурентоспроможності.
ОБ'ЄКТ	Сучасні засоби програмування, такі як фреймворки та бібліотеки, що застосовуються для створення інтерактивних веб-додатків.
ПРЕДМЕТ	процеси розробки та використанні інтерактивних веб-додатків, з особливим акцентом на використанні оновлених фреймворків та бібліотек.
МЕТА	Оцінка можливостей, переваг та обмежень сучасних фреймворків та бібліотек, що використовуються у розробці веб-додатків з акцентом на їхню інтерактивність.
ЗАВДАННЯ	Аналіз фреймворків та бібліотек, оцінку продуктивності, вивчення впливу на UX, проведення експериментів та тестувань, а також вибір оптимальних рішень.



Веб-додаток

Веб-додаток представляє собою програмне забезпечення, доступне через веб-браузер, яке користувач може використовувати для взаємодії з інтернет-сервером.

- Зручний доступ: Дозволяє користувачам взаємодіяти з функціональністю через веб-браузер, що робить його доступним з будь-якого пристрою з Інтернет-підключенням.
- Розширення можливостей: Надає можливість виконувати різноманітні завдання, від редагування документів до покупок онлайн.
- Оптимізація робочих процесів: Спрощує взаємодію та оптимізує робочі процеси для покращення ефективності користувачів у віртуальному середовищі.



Найкращі фреймворки

React JS



Розроблений Facebook, React є бібліотекою для створення інтерфейсів користувача. Використовується для розробки односторінкових додатків, забезпечуючи ефективну віртуалізацію та компонентний підхід.

Angular



Розроблений Google, Angular - це повноцінний фреймворк для створення веб-додатків. Використовує TypeScript та пропонує широкі можливості для розробки складних додатків.

Vue.js



Легкий та прогресивний, Vue.js визначається простотою використання та інтеграцією. Ідеальний для початківців, але здатний вирішувати завдання будь-якої складності.

01

Задоволення користувачів

Зниження витрат на підтримку

02

03

Конкурентна перевага

Покращення конверсії

04

05

Відповідність бізнес-цільям

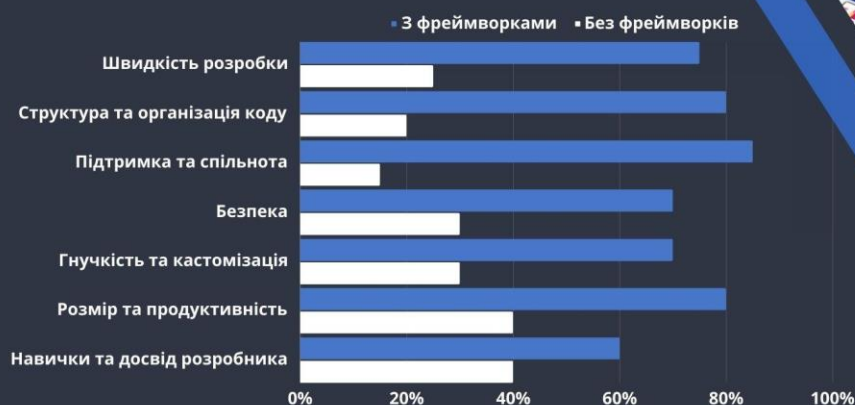
UX User Experience

Визначає, як користувачі сприймають та взаємодіють з продуктом чи послугою під час їх використання.

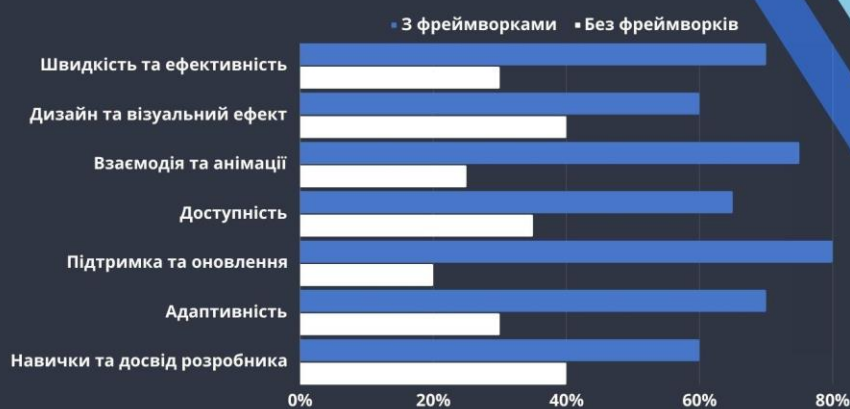
Покращення користувацького досвіду (UX) вимагає комплексного підходу та уваги до різних аспектів взаємодії користувача з продуктом чи послугою.


Дуже важливою для підвищення рівня користувацького досвіду є технічна сторона веб-додатка.

Оцінка ефективності фреймворків для розробки веб-додатків



Оцінка ефективності фреймворків для користувацького досвіду





CodeCube Courses Blog Contact Log In Sign Up


Create your own course
Make money with your knowledge!

- Free access to all courses on CodeCube
- Each month you need to pay to sustain your course
- By paid each time your course was bought

\$17.99 USD/month

Create!

CodeCube



CodeCube Search for courses... Courses Blog Contact Log In Sign Up

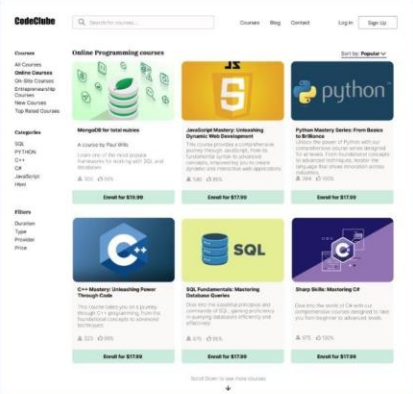
Learn Programming

Select desired courses from online or on-site programs, get individual assistance from instructors and master your new skills.

Explore Courses Check Availability

30 K 180 99%

Reviews Courses People Reviews



CodeCube Search for courses... Courses Blog Contact Log In Sign Up

Courses

Online Programming courses

JS python

Categories: JavaScript for Total noobs, JavaScript Mastery: Understanding Dynamic Web Development, Python Mastery Series: From Basics to AI/ML

Filters: Duration, Type, Availability, Price

Course: C++ Mastery: Unleashing Power Through Code, SQL: Fundamentals: Mastering Database Queries, Deep Dive: Mastering C#



```
// src/App.js
import React, { useState } from 'react';
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';
import CourseList from './components/courseList';
import CourseDetail from './components/courseDetail';

const App = () => {
  const [courses, setCourses] = useState([
    { id: 1, title: 'Introduction to JavaScript', description: 'Learn the basics of JavaScript programming.', duration: '4 weeks' },
    { id: 2, title: 'React Fundamentals', description: 'Explore the fundamentals of React JS.', duration: '6 weeks' },
    // Add more courses as needed
  ]);

  return (
    <Router>
      <div>
        <Route path="/" exact>
          <CourseList courses={courses} />
        </Route>
        <Route path="/course/:id">
          <CourseDetail course={courses[0]} /> { /* Pass the selected course based on the route parameter */ }
        </Route>
      </div>
    </Router>
  );
};

export default App;
```

CodeCube



```
components/courseList.js
import React from 'react';

const CourseList = ({ courses }) => {
  return (
    <div>
      <h2>CodeCube Courses</h2>
      <ul>
        {courses.map(course => (
          <li key={course.id}>{course.title}</li>
        ))}
      </ul>
    </div>
  );
};

export default CourseList;
```

ВИСНОВКИ

- 1 Сучасні фреймворки та бібліотеки підтверджують свій великий потенціал у покращенні користувацького досвіду та створенні високоякісних веб-додатків.
- 2 Дослідження поглядів фахівців та відгуків користувачів підтверджує значний вплив обраних інструментів на поліпшення користувацького досвіду веб-додатків.
- 3 Аналіз веб-інструментів у реальних проектах виявив їхні переваги та недоліки, що допомогло знайти оптимальні та оптимізовані рішення для створення веб-додатків.
- 4 Результат підкреслює важливість проведення досліджень та правильного вибору інструментів у веб-розробці для досягнення високої якості користувацького досвіду.

Апробація результатів дослідження

Конференція

Гарбецький А. М. «Інтерактивні веб-додатки для підвищення рівня користувацького досвіду». Тези доповіді на Всеукраїнській Науково-технічній конференції «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і Світу». – Київ, 28 листопада 2023 р.

Стаття в журналі

Гарбецький А. М. «Сучасні фреймворки як спосіб покращення користувацького досвіду». Стаття у загальногалузевому науково-виробничому журналі «Зв'язок», м. Київ - №1, 2024. – С.220-228.



ДЯКУЮ ЗА УВАГУ