Розділ №1 АНАЛІЗ

Аналіз предметної області можна розділити на два підпункти: 1.1 - володіння мовами та інструментами фронтенд-розробки і 1.2 - навички роботи з серверними технологіями та базами даних.

1.1 Володіння мовами та інструментами фронтенд-розробки:

HTML: Маємо впевнене володіння мовою розмітки гіпертекстових сторінок HTML.

CSS: Маємо володіння каскадними таблицями стилів CSS. Це означає, що студент повинен знати основні властивості CSS, селектори, можливості роботи з кольорами, шрифтами, розмірами та позиціонуванням елементів на сторінці.

Препроцесори CSS: Маємо досвід роботи з препроцесорами CSS, такими як Sass або Less. Це дозволяє студенту використовувати покращені функції, які надають ці інструменти, такі як змінні, міксини, вкладені стилі та інші, щоб полегшити процес створення та підтримки CSS коду.

JavaScript: Маємо глибокі знання мови програмування JavaScript. Це включає розуміння основних концепцій, таких як змінні, функції, об'єкти, умовні оператори, цикли, робота з подіями та DOM-деревом.

Фреймворки: Маємо досвід роботи з одним з фреймворків React.js, Angular або Vue.js. Це означає, що студент знає основні концепції та можливості цих фреймворків, вміти створювати компоненти, маршрутизацію, управляти станом додатку та інше.

1.2 Навички роботи з серверними технологіями та базами даних:

Серверні технології: Студент має навички роботи з серверними технологіями, такими як Node.. Це дозволяє йому створювати серверну логіку, обробляти запити, взаємодіяти з базою даних та надавати дані для фронтенду.

Бази даних: Маємо досвід роботи з базами даних, наприклад, MySQL або MongoDB. Це означає, що студент повинен знати основні поняття реляційних або нереляційних баз даних, розуміти мову запитів SQL та уміти зберігати та отримувати дані з бази.

1.3: Крім володіння мовами та інструментами фронтенд-розробки, студент також повинен мати розуміння процесу проектування веб-додатків та кращих практик розробки. Деякі з цих аспектів включають:

Дизайн і взаємодія: Студент повинен мати базове розуміння принципів дизайну і взаємодії, таких як вибір кольорів, типографії, композиції та ергономіки. Вміння створювати зручні та інтуїтивно зрозумілі інтерфейси допоможуть забезпечити позитивний досвід користувачів.

Версіонування та керування проектом: Студент повинен мати навички роботи з системами контролю версій, такими як Git, щоб ефективно керувати розробкою коду, працювати над різними гілками, об'єднувати зміни та вирішувати конфлікти. Керування проектом, таке як використання інструментів для стеження за завданнями (наприклад, Jira або Trello), також є важливим аспектом розробки.

Тестування та відладка: Студент повинен бути знайомий з основними підходами до тестування веб-додатків, такими як функціональне та модульне тестування. Розуміння процесу відлагодження та вміння використовувати інструменти для відстеження помилок, такі як консоль розробника, є важливими для забезпечення якості коду та виявлення проблем.

Знання про безпеку: Студент повинен мати базове розуміння принципів безпеки веб-додатків, таких як захист від кросс-сайтового скриптингу (XSS), внедрення SQL-запитів та інших типових атак. Використання засобів, таких як захищені запити та валідація даних, допоможе забезпечити безпеку веб-додатка.

Оптимізація та продуктивність: Студент повинен мати уявлення про оптимізацію швидкості завантаження та продуктивності веб-додатків. Це включає кешування, стиснення ресурсів, мінімізацію запитів до сервера та оптимізацію запитів до бази даних.

Висновок:

У цій предметній області студент повинен продемонструвати впевнене володіння мовами та інструментами фронтенд-розробки, такими як HTML, CSS, JavaScript та фреймворки React.js, Angular або Vue.js. Він також повинен мати навички роботи з серверними технологіями, такими як Node.js або PHP, і базами даних, такими як MySQL або MongoDB.

Результатом виконаної роботи повинен бути повноцінний web-додаток, який включає в себе всі компоненти фронтенду і здатний взаємодіяти з сервером через REST API. Студент повинен також мати навички розгортання додатку на безкоштовному хостингу для його демонстраційного перегляду. Початковий код повинен бути доступний у власному git-репозиторії, що дозволяє іншим особам переглядати, вносити зміни та співпрацювати над проектом.Розуміння процесу проектування веб-додатків, кращих практик розробки, дизайну та взаємодії, тестування та відлагодження, керування проектом та безпеки є також важливими аспектами, які студент повинен мати у своєму арсеналі.

Усі ці знання та навички допоможуть студенту стати компетентним фронтенд-розробником, здатним створювати високоякісні та функціональні веб-додатки, які задовольняють потреби користувачів та вимоги проекту.

Розділ 2 ПРОЕКТУВАННЯ

Проектування архітектури додатку та вибір технологій для його реалізації є важливим етапом у розробці веб-застосунків. Для створення web-застосунку порталу, необхідно розглянути декілька ключових аспектів.

Використання мови розмітки гіпертекстових сторінок HTML та каскадних таблиць стилів CSS: HTML використовується для структурування та визначення змісту сторінок, а CSS - для стилізації та оформлення цих сторінок. Використання CSS-фреймворків, таких як Bootstrap, може спростити процес створення зовнішнього вигляду та респонсивного дизайну.

Мова програмування JavaScript та бібліотеки: JavaScript використовується для динамічної взаємодії зі сторінками та додавання функціональності до веб-застосунку. У даному випадку, для розробки порталу можна розглянути використання бібліотеки React.js, Angular або Vue.js, які забезпечують швидку розробку і високу продуктивність.

Використання додаткових бібліотек та інструментів: Для полегшення роботи зі сторінками та додавання елементів взаємодії, можна використовувати бібліотеку JQuery та її розширення JQueryUI.

У виборі технологій варто враховувати функціональні вимоги до проекту, складність та обсяг розробки, а також власні навички та знання. Кожен зі згаданих фреймворків (React.js, Angular, Vue.js) має свої переваги та особливості, і вибір залежить від конкретних потреб та пріоритетів проекту.

Загалом, правильний вибір архітектури та технологій дозволить створити потужний та функціональний веб-застосунок, який задовольнятиме вимоги проекту та користувачів.

При проектуванні додатку і виборі технологій для його реалізації необхідно розглянути деталі кількох ключових аспектів:

Використання HTML і CSS: HTML використовується для створення структури та змісту сторінок, а CSS - для стилізації та оформлення. У процесі проектування слід уважно обрати, як будуть використовуватись HTML теги для логічної організації додатку, а CSS - для досягнення бажаного зовнішнього вигляду, включаючи використання технологій Flexbox, Grid та медіазапитів для адаптивного дизайну.

Вибір фреймворку: Розгляньте використання фреймворка, такого як React.js, Angular або Vue.js, для полегшення розробки та забезпечення високої продуктивності. Кожен з цих фреймворків має свої переваги і особливості, і вибір залежить від ваших потреб та навичок.

Використання додаткових бібліотек та інструментів: Розгляньте використання бібліотек, таких як JQuery і JQueryUI, для спрощення маніпулювання елементами сторінки та взаємодії з ними. Також варто розглянути використання Flexbox та Grid для гнучкого розташування елементів та медіазапитів для адаптивного дизайну.

Враховуйте функціональні вимоги, обсяг проекту, доступні ресурси та власні навички при виборі архітектури та технологій. Правильний вибір допоможе створити потужний та функціональний веб-додаток, який задовольнятиме вимоги проекту та користувачів.

Висновок:

В цілому, розробка та реалізація функціональних компонентів користувацького інтерфейсу інформаційного порталу "Об'єкт дослідження" на основі web-технологій вимагає визначення вимог, розробки дизайну, реалізації компонентів, їх верстки, інтеграції з бекендом, тестування та підтримки після релізу.

Розділ 3 РЕАЛІЗАЦІЯ

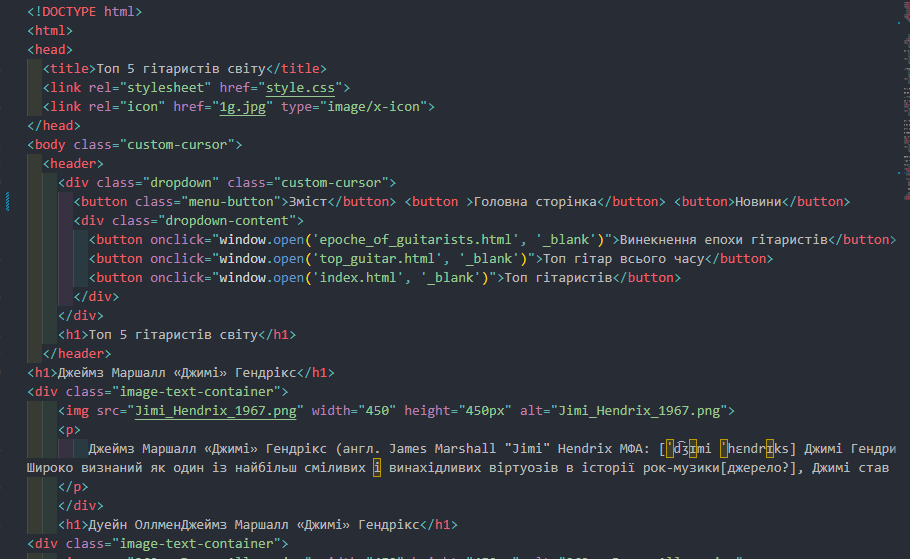
Розробка та реалізація функціональних компонентів користувацького інтерфейсу інформаційного порталу "Об'єкт дослідження" на основі web-технологій включає кілька кроків. Давайте розглянемо їх детальніше:

Вимоги до користувацького інтерфейсу: Спочатку ми маємо уточнити вимоги до користувацького інтерфейсу нашого порталу. Ми проводимо аналіз цільової аудиторії, визначаємо основні функції та можливості порталу, а також розташування елементів на сторінках.

1. Дизайн інтерфейсу: Після цього ми переходимо до розробки дизайну інтерфейсу. Ми створюємо макети сторінок, вибираємо кольорову палітру, шрифти та інші дизайнерські елементи. Наша мета - створити зручний, привабливий та сучасний дизайн, який відповідає бренду та функціональності порталу.
2. Розробка функціональних компонентів: Після завершення дизайну ми переходимо до розробки функціональних компонентів, які забезпечують необхідну функціональність порталу. Це можуть бути компоненти, такі як меню навігації, пошукова рядок, форми для введення даних, списки, таблиці, графіки та інші елементи, які дозволяють користувачам взаємодіяти з інформацією на порталу.
3. Верстка і реалізація: Після розробки функціональних компонентів ми виконуємо верстку, тобто написання HTML-коду для структури сторінок, та CSS-стилізацію, щоб задати вигляд і розміщення компонентів на сторінці.
4. Тестування: Після реалізації компонентів ми проводимо тестування, щоб переконатися, що користувацький інтерфейс працює належним чином. Ми перевіряємо функціональність компонентів, відповідність дизайну та проводимо тестування на різних пристроях та браузерах.
5. Реліз та підтримка: Після успішного завершення тестування ми можемо випустити наш портал в експлуатацію. Ми також забезпечуємо підтримку та оновлення порталу, виправляємо помилки та додаємо нові функції, які можуть знадобитися користувачам.

Це загальний опис процесу розробки та реалізації функціональних компонентів користувацького інтерфейсу веб-порталу "Об'єкт дослідження" на основі web-технологій.

1. На Рис.(1) зображено макети сторінок,визначена кольорова палітра,шрифти та інші дизайнерські елементи



2)На Рис.(2) відображені елементи навігації, а саме бібліотека JQuery



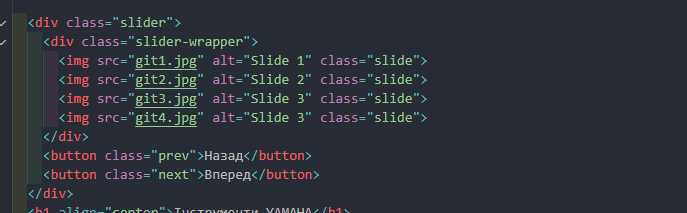
**Рис.(2)**

3)На Рис.(3) зображено вигляд CSS стилізації

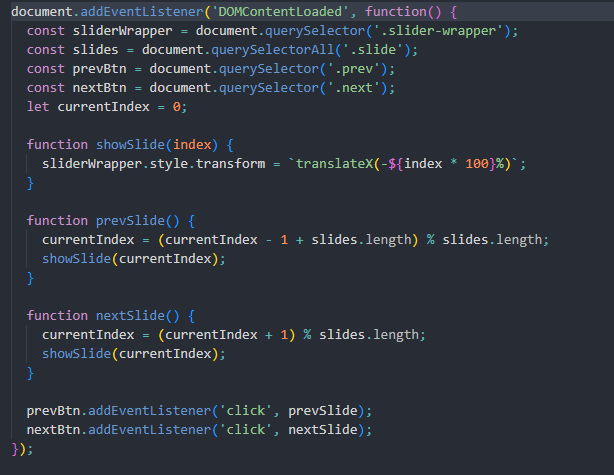


**Рис.(3)**

5. На Рис.(4-5) Зображено код для слайдеру



**Рис.(4)**



**Рис.(5)**

4)На Рис.(6) зображено тестування реалізації компонентів та перевірка їхньої функціональності



Рис.(4)

5)За допомогою безкоштовного хостингу (Netlifly) ми можемо випустити наш портал в експлуатацію.