План разработки Frontend-части

1. Составление плана по разработке Frontend-части

- Определение целей и требований проекта.

- Анализ существующих решений и библиотек, которые могут быть использованы.

- Выбор технологий (например, React, Vue, Angular) и инструментов (Webpack, Babel, ESLint).

- Составление структуры проекта и определение основных компонентов.

- Разработка прототипа или макета для демонстрации основных функциональностей.

2. Разработка UI/UX дизайна

- Создание wireframes (черновых эскизов) для всех ключевых страниц сайта.

- Разработка концепции дизайна, включая цветовую схему, шрифты, иконки и другие элементы.

- Создание интерактивных прототипов (можно использовать инструменты, такие как Figma, Sketch, Adobe XD).

- Проведение тестирования пользовательского опыта (UX) и сбор обратной связи.

- Внесение необходимых изменений на основе полученной обратной связи.

3. Создание файла с начальной веб-разметкой

- Написание HTML-разметки для основных страниц сайта.

- Применение базовых стилей с использованием CSS (или препроцессоров, таких как SASS, LESS).

- Добавление минимального JavaScript для взаимодействия с пользователем (если необходимо).

- Валидация HTML-кода на соответствие стандартам и доступность (например, с помощью WAVE или Lighthouse).

4. Создание сайта-витрины

- Разработка компонентов для каждой страницы (например, Header, Footer, ProductCard, CategoryList).

- Интеграция стилей и анимаций для улучшения визуального восприятия.

- Реализация навигации и маршрутизации (например, с использованием React Router, Vue Router).

- Добавление форм, кнопок, модальных окон и других интерактивных элементов.

- Тестирование сайта на различных устройствах и браузерах для обеспечения кросс-браузерной совместимости и адаптивности.

5. Объединение сайта с backend-частью

- Настройка API-эндпоинтов для взаимодействия с backend (например, REST, GraphQL).

- Реализация запросов к серверу (например, с использованием Axios, Fetch API).

- Обработка ответов от сервера и отображение данных на странице.

- Реализация аутентификации и авторизации (например, JWT, OAuth).

- Настройка обработки ошибок и логирования.

6. Проверка работоспособности

- Проведение функционального тестирования всех страниц и компонентов.

- Проверка корректности работы всех форм, кнопок и других интерактивных элементов.

- Тестирование интеграции с backend (например, проверка загрузки данных, отправки форм, аутентификации).

- Проверка производительности и оптимизация скорости загрузки (например, сжатие изображений, минификация CSS и JS).

- Исправление выявленных ошибок и недочетов.

- Финальное тестирование на различных устройствах и браузерах.

- Подготовка документации по использованию и поддержке проекта.