**Промежуточный отчет по программному проекту**

**1. Основные планы и этапы проекта**

1.1 Краткое описание проекта:

Веб-приложение, предназначенное для автоматизации процесса обслуживания клиентов в ресторанах и кафе.

Название проекта: “Умный стол”

Цель проекта:

Создание системы для управления заказами в ресторанах с интеграцией через Telegram и удобной административной панелью для персонала

Краткое описание задач:

* Разработка фронтенда для взаимодействия с пользователями (клиентами и персоналом).
* Интеграция с бэкендом для отображения меню, корзины, информации о заказах и пользователях.
* Реализация функционала для работы с сессиями, добавлением и редактированием позиций в заказах, а также взаимодействия с системой через Telegram.

1.2 Планы и этапы выполнения проекта:

|  |  |  |  |
| --- | --- | --- | --- |
| Этап проекта | Описание работ | Ожидаемые результаты | Сроки выполнения |
| Техническое задание | Сбор данных, постановка задач, описание функционала фронтенда, определение требований | Утвержденное техническое задание с описанием всех функций фронтенда | 01.11.24 – 15.11.24 |
| Разработка фронтенда | Создание интерфейса для отображения меню, заказов, сессий, корзины и взаимодействия через Telegram, также разработка админ-панели. | Рабочий фронтенд, интегрированный с бэкендом и API, реализующие основные функции | 16.11.24 – 31.03.25 |
| Разработка документации | Создание документации по API и архитектуре фронтенда, описание методов и функционала | Полная документация, включая описание работы с API и функций интерфейса | 01.04.25 – 15.04.25 |
| Испытание программы | Тестирование, отладка интерфейса и интеграции с бэкендом, исправление ошибок | Протестированный фронтенд, без ошибок, с исправленными багами и корректной интеграцией | 16.04.25 – 30.04.25 |
| Внедрение | Подготовка интерфейса для защиты, финальные доработки | Готовый фронтенд для защиты проекта, успешная защита | 01.05.25 – 15.05.25 |

**2. Используемый технологический стек и его обоснование**

2.1 Перечень используемых технологий:

|  |  |  |
| --- | --- | --- |
| Технология/Инструмент | Описание | Причины выбора |
| React | Библиотека для создания пользовательских интерфейсов | React обеспечивает создание динамичных, компонентных UI и хорошую производительность. |
| VS Code | Редактор кода с поддержкой JavaScript, React и множества плагинов | VS Code выбран за его легкость, поддержку множества расширений и хорошую интеграцию с React и JavaScript. |
| JavaScript (ES6+) | Язык программирования, основанный на JavaScript | JavaScript является универсальным языком для разработки фронтенда и идеально интегрируется с React. |
| HTML/CSS | Технологии для разметки и стилизации страниц | Используются для создания структуры и оформления пользовательского интерфейса. |

2.2 Обоснование выбранного технологического стека:

JavaScript был выбран для разработки фронтенда, так как это основной язык для веб-разработки, поддерживаемый всеми браузерами. Он идеально подходит для асинхронной работы с API, что важно для динамичных веб-приложений. В сочетании с React, JavaScript позволяет эффективно управлять состоянием, обновлять интерфейс без перезагрузки страницы и обеспечивать плавный пользовательский опыт.

React был выбран благодаря своей способности быстро разрабатывать интерактивные и высокоскоростные интерфейсы. React позволяет строить интерфейсы на основе компонентов, что облегчает повторное использование и поддержку кода.

HTML и CSS — стандартные технологии для разметки и стилизации.

VS Code выбран как основной редактор кода благодаря своей гибкости, мощным функциям для разработки с React и возможностям интеграции с различными инструментами для улучшения продуктивности.

**3. Критерии оценивания проекта**

|  |  |
| --- | --- |
| **Критерий** | **Описание** |
| Работа с базой данных | Будут использованы/Не будут использованы |
| Управление доступом | Будут использованы/Не будут использованы |
| Функциональность - Процент выполнения функциональных требований | Выполненные требования в процентах от общего количества |
| Функциональность - Количество реализованных функций | Абсолютное количество функций, которые работают правильно |
| Производительность и эффективность - Время отклика | Среднее время ответа программы на запросы пользователя (в секундах) |
| Качество кода - Средняя цикломатическая сложность | Средняя сложность функций |
| Тестирование - Процент успешных тестов (%) | Процент успешно пройденных тестов из общего количества |
| Соблюдение сроков и плана - Количество дней отклонения от плана | Общее число дней отклонения от плана |
| Использование технологического стека - Процент использования функциональности стека (%) | Процент использования функциональности выбранного стека технологий |
| Оценка командной работы - Среднее время коммуникации (в часах) | Среднее время, потраченное на обсуждение задач и решение вопросов |
| Оценка командной работы - Количество завершенных задач на каждого участника | Общее число задач, выполненных каждым членом команды |

**4. Особые пометки**

* В процессе разработки могут возникнуть трудности при интеграции фронтенда с бэкендом, особенно при работе с API и асинхронными запросами. Для этого потребуется тесное сотрудничество с бэкенд-разработчиками.
* Также, учитывая, что проект использует React, возможно, потребуются дополнительные библиотеки для управления состоянием и взаимодействия с бэкендом.
* Важно обеспечить поддержку адаптивности интерфейса для различных устройств, включая мобильные телефоны, что требует тщательного тестирования интерфейса.