Титульный лист

Содержание

Введение (2–3 стр.)

Подводка: сторонние приложения-ежедневники обладают избыточным функционалом, завязаны на чужие серверы и могут быть подвержены утечкам данных. Необходимо создать такой ежедневник, который можно самостоятельно поднять на собственном сервере (желательно одной командой терминале).

Тема проекта: Разработка веб-приложения “Планер-ежедневник” с помощью языка JAVA.

Цель: Разработать приложение-ежедневник с веб-интерфейсом и PWA. Наработать практические навыки и получить опыт разработки прикладных программ.

Задачи:

1. Изучить литературу, касающуюся темы проекта.
2. Рассмотреть основные этапы разработки веб-приложений.
3. Спроектировать архитектуру приложения.
4. Определить инструменты и технологии, необходимые для разработки.
5. Разработать бэкенд.
6. Разработать фронтенд.
7. Выполнить ручное тестирование веб-приложения.
8. Развернуть веб-приложение.
9. Подготовить инструкцию по самостоятельному развёртыванию.

Инструменты: Postman, валидатор W3C, GitHub, Visual Studio Code, Intellij IDEA, JDK, Docker…

Глава 1. Основы разработки веб-приложений на языке Java (~15 стр)

- Что такое веб-приложение, его особенности (плагиатный пример) https://github.com/MorgunovaAO/Diploma_project/blob/paragraph_1.1/Paragraph_1.1.md
- Возможности языка Java
    - для чего используется
    - как работает (JVM)
    - как установить (JDK)
- Этапы разработки веб-приложения
    - Предпроектная подготовка
        - Определение требований к приложению и разработка ТЗ
    - Проектирование

        - Архитектура приложения
        - Дизайн интерфейса пользователя
    - Разработка веб-приложения
    - Тестирование
    - Запуск и сопровождение

Глава 2. Подготовка к разработке веб-приложения. Выбор технологического стека (~15 стр)

- Описать видение продукта, ТЗ
- Проектирование архитектуры приложения
    - Определить подходящий паттерн проектирования (MVC, MVP…)
        - ***не понимаю как, может оттолкнуться от фреймворка?***
    - UML-диаграмма компонентов и их взаимодействия
    - UML-диаграммы классов для каждого компонента
- Проектирование базы данных: определение структуры таблиц и связей между ними
    - ERD диаграмма связей и таблиц БД
- Определить инструменты, которые будут использованы
    - фреймворк Spring Boot или Javalin или Spark - ***вот тут ваще тёмный лес***
    - Postman для тестирования запросов REST API
    - GitHub для размещения репозитория
    - JDK + Intellij IDEA как основной инструмент создания бэкенда
    - Visual Studio Code как основной инструмент создания фронтенда

Глава 3. Разработка, тестирование и запуск веб-приложения (~20 стр.)

- Реализация бэкенда: создание Java-классов и методов для обработки запросов от фронтенда, взаимодействия с базой данных и выполнения основных функций приложения, таких как создание, редактирование и удаление задач.
- Реализация фронтенда: создание пользовательского интерфейса с использованием HTML, CSS и JavaScript. Для создания PWA можно использовать фреймворк React или Angular.
    - ***не умею, можно привлечь стороннего фронтендера, или студента?***
- Тестирование
    - Тестирование функциональности
    - Тестирование интерфейса
    - Тестирование безопасности
- Подготовка инструкции по развертыванию веб-приложения
- Запуск и подведение итогов.

Заключение (~ 4 стр.)

Ну вот как-то так

Список используемой литературы

букварь

Приложения

один
два
три

Презентация