**Техническое задание на разработку «Веб-платформы для поиска единомышленников-программистов для совместной работы над проектами»**

**СОДЕРЖАНИЕ**

[1. Наименование программы 3](#_Toc7915)

[2. Основание для разработки 3](#_Toc19933)

[3. Назначение разработки 3](#_Toc26977)

[4. Требования к программе 4](#_Toc30903)

[4.1 Функциональные характеристики 4](#_Toc21568)

[Пользовательская часть: 4](#_Toc22392)

[4.2 Требования к надёжности 5](#_Toc13575)

[6](#_Toc11424)

[4.3 Условия эксплуатации 6](#_Toc819)

[4.4 Требования к техническим средствам 6](#_Toc15566)

[Серверная часть: 6](#_Toc14257)

[4.5 Требования к информационной и программной совместимости 7](#_Toc28895)

[Структура базы данных: 7](#_Toc9476)

[5. Стадии и этапы разработки 7](#_Toc4599)

[6.Порядок контроля и приёмки 8](#_Toc30122)

**1. Наименование программы**

Наименование программы — «Веб-платформа для поиска единомышленников-программистов».

**2. Основание для разработки**

Разработка ведётся в рамках преддипломной практики по направлению подготовки 09.02.07 «Информационные системы и программирование» для автоматизации процессов поиска и взаимодействия программистов при реализации совместных проектов.

**3. Назначение разработки**

Цель разработки — создание платформы для объединения программистов с общими интересами и навыками, включая:

- Регистрацию пользователей с указанием технических навыков;

- Создание и публикацию проектов;

- Поиск участников по критериям (языки программирования, фреймворки, уровень опыта);

- Управление заявками на участие в проектах;

- Коммуникацию между участниками (чат, комментарии);

- Администрирование пользователей и проектов.

Целевая аудитория:

- Пользователи — программисты, разработчики, студенты IT-направлений, ищущие команду для совместной работы.

- Администраторы — модераторы, управляющие контентом и пользователями.

Преимущества автоматизации:

- Сокращение времени на поиск команды;

- Упрощение взаимодействия между участниками;

- Централизованное управление проектами и участниками.

**4. Требования к программе**

**4.1 Функциональные характеристики**

**Пользовательская часть:**

- Регистрация и аутентификация через email или OAuth (GitHub, Google);

Создание профиля с указанием:

- Навыков (языки, фреймворки, инструменты);

- Уровня опыта (Junior, Middle, Senior);

- Портфолио (ссылки на GitHub, проекты);

Создание проектов:

- Название, описание, требуемые навыки;

-/Выбор статуса (открыт для поиска, в разработке, завершён);

- Поиск проектов/участников по фильтрам:

- Технологии, уровень опыта, локация (опционально);

- Подача заявок на участие в проектах;

-/Личный кабинет с управлением своими проектами и заявками;

- Система уведомлений (новые заявки, изменения статуса);

- Чат для обсуждения проектов.

Административная часть:

- Управление пользователями:

- Блокировка/разблокировка аккаунтов;

-/Просмотр статистики (активность, количество проектов);

Модерация проектов:

- Удаление несоответствующего контента;

- Изменение статусов проектов;

Аналитика:

- Количество активных пользователей и проектов;

- Популярные технологии;

- География участников.

**4.2 Требования к надёжности**

- Защита от SQL-инъекций и XSS-атак;

- Валидация входных данных (навыки, email, ссылки);

- Резервное копирование базы данных;

-/Обработка ошибок при работе с внешними API (GitHub, OAuth);

- Логирование критических действий (регистрация, удаление проектов);

- Авторизация через JWT-токены.

**4.3 Условия эксплуатации**

- Доступ через веб-браузер (Chrome, Firefox, Safari);

- Наличие интернет-соединения;

- Базовые навыки работы с веб-интерфейсами.

**4.4 Требования к техническим средствам**

**Серверная часть:**

- Язык: Python 3.10+;

- Фреймворк: Django 4.2+;

- СУБД: PostgreSQL 14+;

- Хостинг: VPS с Ubuntu 22.04 LTS;

Минимальные ресурсы:

- CPU: 2 ядра;

- RAM: 2 ГБ;

- Диск: 10 ГБ.

- Клиентская часть:

- Фреймворк: React 18+;

- Библиотеки: Redux, Axios, WebSocket.

**4.5 Требования к информационной и программной совместимости**

**Структура базы данных:**

- users (id, email, password\_hash, name, skills, experience\_level, portfolio\_links);

- projects (id, owner\_id, title, description, required\_skills, status, created\_at);

- applications (id, project\_id, user\_id, status, message);

messages (id, sender\_id, receiver\_id, project\_id, content, timestamp).

Интеграции:

- GitHub API для импорта портфолио;

- Google OAuth для аутентификации.

**5. Стадии и этапы разработки**

- Анализ и постановка задачи:

- Сбор требований, описание пользовательских сценариев.

Проектирование:

- Создание ER-диаграммы БД, проектирование API, UI/UX-дизайн.

Реализация:

- Разработка серверной и клиентской части, интеграция внешних сервисов.

Тестирование:

- Модульное тестирование (pytest, Jest);

- Проверка безопасности (OWASP ZAP).

- Документирование и сдача:

- Составление руководства пользователя и администратора.

**6.Порядок контроля и приёмки**

- Проверка соответствия функционала требованиям раздела 4;

-/Тестирование регистрации, создания проектов, поиска и коммуникации;

- Проверка административных функций: модерация, аналитика;

- Тестирование на уязвимости (SQL-инъекции, XSS);

- Аудит кода на соответствие PEP8 и ESLint;

- Оформление акта сдачи-приёмки с подписью заказчика.