**1. Общее описание проекта**

Данное веб-приложение представляет собой платформу для обучения программированию, где пользователи могут:

* Регистрироваться и авторизовываться в системе
* Просматривать доступные задания по программированию
* Отправлять свои решения задач
* Получать обратную связь о корректности решений
* Отслеживать свой прогресс обучения
* Сравнивать свои достижения с другими пользователями через таблицу лидеров

**2. Технологический стек**

* **Backend**: Flask
* **База данных**: SQLAlchemy
* **Шаблоны**: Jinja2 (встроен во Flask)
* **Аутентификация**: Сессии Flask с хешированием паролей (werkzeug.security)
* **Тестирование кода**: Наша самописная система тестирования (TestingSystem из модуля tester). Имеет защиту с помощью Docker

**3. Структура базы данных**

Приложение использует 4 основные таблицы:

1. **users** - хранит данные пользователей:
   * Учетные данные (логин, хеш пароля)
   * Электронная почта(если указана)
   * Дата регистрации
   * Количество набранных очков(изначально 0)
2. **tasks** - содержит задания по программированию:
   * Название и описание задачи
   * Уровень сложности
   * Связь с тест-кейсами
3. **test\_cases** - тестовые случаи для проверки решений:
   * Входные данные
   * Ожидаемый результат
   * Связь с конкретной задачей
4. **user\_progress** - отслеживает прогресс пользователей:
   * Отметки о выполнении задач
   * Даты завершения
   * Связь пользователь-задача

**4. Основные функциональные модули**

**4.1. Аутентификация и авторизация**

* Регистрация с валидацией уникальности имени пользователя
* Вход в систему с проверкой учетных данных
* Управление сессиями пользователей
* Защита паролей с помощью хеширования

**4.2. Работа с заданиями**

* Просмотр списка задач с фильтрацией по сложности
* Отображение детальной информации о задаче
* Отправка решений на проверку
* Автоматическая проверка кода пользователя
* Начисление очков за успешное решение

**4.3. Отслеживание прогресса**

* Отображение выполненных и невыполненных задач
* История последних решенных задач
* Статистика по количеству решенных задач
* Таблица лидеров с ранжированием по очкам

**4.4. Профиль пользователя**

* Персональная информация
* Достижения и набранные очки

**5. Безопасность**

* Хранение паролей в виде хешей (generate\_password\_hash/check\_password\_hash)
* Защита сессий с помощью секретного ключа
* Проверка авторизации для защищенных маршрутов
* Ограниченный доступ к функционалу для неавторизованных пользователей

**6. Особенности реализации**

1. **Инициализация базы данных**: При первом запуске создаются таблицы и заполняются тестовыми данными (3 задачи разного уровня сложности для теста, к сдаче их будет больше и разные сложности).
2. **Начисление очков**: Количество начисляемых очков зависит от сложности задачи (Easy=1, Medium=2, Hard=3 и т.д.).
3. **Прогресс пользователя**: Для каждого нового пользователя автоматически создаются записи о прогрессе для всех существующих задач.
4. **Таблица лидеров**: Сортировка по количеству очков (по убыванию) и имени пользователя (по возрастанию).

**7**. **Как работает.**Имеется модуль TestingSystem, который отвечает за работу сайта. Поддерживаются различные языки программирования, такие как Python, C++.   
В тестирующую систему сперва с сайта передаются данные о задаче и формируется тесты из бд. После этого система автоматически определяет расширение загруженного файла и, как следствие, язык. Определив язык, она понимает, как именно запускать код и прогоняет его на всех тестах, **независимо от результата предыдущих.** Далее она смотрит на корректность всех ответов, после чего выносит вердикт. Для усложнения нахождения ошибки сама система говорит лишь вердикт, при этом не упоминая, в чем конкретно была ошибка. Также она не показывает тесты, во избежание любителей условий.   
Данные о задаче, как уже сказано ранее, берутся из бд, а идентификатором служит id, которое можно обнаружить в поисковой строке, находясь на странице сдачи задачи