#### **1. Введение**

Цель проекта – разработка веб-системы «Создатель резюме», которая позволит пользователям регистрироваться, создавать, редактировать и удалять резюме с возможностью выбора шаблонов. Система будет работать на основе REST API.

#### **2. Структура проекта**

* **Back-end**: Java.
* **Front-end**: HTML5, CSS3, JavaScript (React.js или другой JavaScript-фреймворк).
* **База данных**:
  + Таблица “Пользователи” (хранение учетных данных пользователей).
  + Таблица “Резюме” (хранение данных резюме, связанных с пользователями).

#### **3. Основные функциональные модули**

##### **3.1. Клиентская часть (Front-end)**

* **Главная страница**:
  + Отображение информации о возможностях системы.
  + Кнопки для перехода к регистрации или авторизации.
* **Страница регистрации**:
  + Форма для регистрации новых пользователей (поля: имя пользователя, пароль).
  + Валидация данных на клиенте перед отправкой (например, проверка длины пароля).
* **Страница авторизации**:
  + Форма для авторизации пользователей (логин и пароль).
* **Страница создания резюме**:
  + Форма для ввода данных резюме (опыт работы, образование, навыки и т. д.).
  + Возможность выбора шаблона резюме.
  + Кнопка для сохранения резюме.
* **Страница редактирования резюме**:
  + Возможность редактирования ранее созданного резюме.
  + Внесение изменений в данные или изменение шаблона.
* **Страница с перечнем резюме**:
  + Отображение всех резюме, созданных пользователем.
  + Возможность редактирования или удаления каждого резюме.

#### **4. Архитектура REST API**

##### **4.1. Endpoints для работы с пользователями**

* **Регистрация пользователя** (POST /api/users/register):
  + Входные данные: username, password (хэшируется с использованием SHA-256).
  + Результат: Создание нового пользователя в базе данных и возврат статуса успешной регистрации.
* **Авторизация пользователя** (POST /api/users/login):
  + Входные данные: username, password.
  + Результат: Проверка учетных данных, генерация и возвращение токена авторизации (JWT или другой механизм).
* **Получение данных пользователя** (GET /api/users/{id}):
  + Входные данные: ID пользователя.
  + Результат: Возвращение данных пользователя (только ID, имя пользователя и дата создания).

##### **4.2. Endpoints для работы с резюме**

* **Создание резюме** (POST /api/resumes):
  + Входные данные: content (данные резюме), template (выбранный шаблон).
  + Результат: Создание нового резюме для авторизованного пользователя.
* **Получение списка резюме** (GET /api/resumes):
  + Входные данные: токен авторизации.
  + Результат: Возвращение всех резюме, связанных с текущим пользователем.
* **Получение одного резюме** (GET /api/resumes/{id}):
  + Входные данные: ID резюме.
  + Результат: Возвращение данных конкретного резюме.
* **Редактирование резюме** (PUT /api/resumes/{id}):
  + Входные данные: обновленный content и/или template.
  + Результат: Обновление существующего резюме.
* **Удаление резюме** (DELETE /api/resumes/{id}):
  + Входные данные: ID резюме.
  + Результат: Удаление резюме из базы данных.

#### **5. База данных**

##### **5.1. Таблица пользователей**

* **Пользователи** (users):
  + id (PK, автоинкремент): Идентификатор пользователя.
  + username (VARCHAR): Уникальное имя пользователя.
  + password (VARCHAR): Хэшированный пароль (SHA-256).
  + created\_at (TIMESTAMP): Дата и время создания учетной записи.

##### **5.2. Таблица резюме**

* **Резюме** (resumes):
  + id (PK, автоинкремент): Идентификатор резюме.
  + user\_id (FK): Идентификатор пользователя, которому принадлежит резюме.
  + content (TEXT): Содержимое резюме в формате JSON или текста.
  + template (VARCHAR): Выбранный шаблон резюме.
  + created\_at (TIMESTAMP): Дата создания резюме.

#### **6. Требования к интерфейсу**

##### **6.1. Клиентская часть (Front-end)**

* **Главная страница**:
  + Легкий и понятный дизайн.
  + Интуитивная навигация по сайту.
* **Регистрация и авторизация**:
  + Простые формы с понятной валидацией.
* **Создание и редактирование резюме**:
  + Интерфейс пошагового заполнения данных.
  + Возможность выбора шаблона оформления резюме.

##### **6.2. Верстка и адаптивность**

* **Адаптивный дизайн**: Должен корректно отображаться на компьютерах, планшетах и мобильных устройствах.
* **Совместимость**: Поддержка всех современных браузеров.

#### **7. Технические требования**

##### **7.1. Back-end**

* **Java** для создания REST API.

##### **7.2. Front-end**

* **HTML5**, **CSS3**, **JavaScript**.
* Фреймворки: **React** или другой.

##### **7.3. База данных**

* **MySQL** или **MSSQL**.
* Хранение данных пользователей и резюме в отдельных таблицах с необходимыми связями.

#### **8. Безопасность**

* Хранение паролей в хэшированном виде с использованием SHA-256.