**Цель приложения:** предоставление всего необходимого для пользователей интернет пространства. (общение, новости, ответы на вопросы, курсы, приятный интерфейс, работа (пока нет))

**Цель приложения:** интернет пространство.

1. **Основные функциональности**
   1. **Регистрация и аутентификация пользователей**

Аутентификация пользователей должна осуществляться через логин/пароль или OAuth 2.0.

* 1. **Профиль пользователя**
     1. **Основная информация:**

- имя фамилия

- логин

- пароль

- аватарка (не совсем обязательная, но в разделе обязательная)

- биография

* + 1. **Доп.информация:**

- пол

- дата рождения

- спец. навыки (теги)

- местоположение (страна)

- соц сети (VK, Telegram, YouTube)

- портфолио (ссылки на работы или проекты)

- опыт работы

- интересы и хобби

* + 1. **Настройки:**

1. **Приватность**

- показывать доп. информацию

- показывать активность на сайте

1. **Уведомления**

- рассылка в telegram

- новые записи

- комментарии под постами

- упоминания

- новый друг

* + 1. **Поиск пользователей**

- Добавить возможность поиска пользователей по логину. Фильтр: компания, сообщество, возраст, страна проживания.

* + 1. **Подписки на пользователей**

- Добавить возможность подписки на пользователей, чтобы пользователи могли следить за обновлениями профиля других пользователей.

* + 1. **Рейтинг пользователей**

- Реализовать возможность оценки пользователей другими пользователями.

- Добавить возможность просмотра рейтинга каждого пользователя на странице его профиля.

* 1. **Статья**

- Возможность создания, редактирования и удаления статей.

- Статьи должны иметь заголовок, описание, теги и содержимое.

- Возможность поиска статей по заголовку, тегам и содержимому.

- Возможность комментирования статей пользователями.

- Возможность ставить оценку.

- Возможность добавить статью в закладки.

* 1. **Новости**

- Не имеет возможности оставлять комменатрий.

- Возможность создания, редактирования и удаления новостей.

- Новости должны иметь заголовок, описание, теги и содержимое.

- Возможность ставить оценку.

- Возможность добавить новость в закладки.

**- !!! новости опубликованные администрацией имеет приоритет над остальными !!!**

* 1. **Сообщества**

- Возможность создания, редактирования и удаления сообществ.

- Сообщества должны иметь название, описание и теги.

- Возможность присоединения к сообществам другими пользователями.

- Возможность поиска сообществ по названию и тегам.

- Возможность создания ролей сообщества.

**1.5.2. Настройка сообщества.**

- Открытое / Закрытое сообщество.

- Редактирование название, описание и теги.

**1.5.2. Настройка ролей (разрешения).**

- Администратор (может все).

- Модератор контента.

- Разрешение на приглашение участников.

- Разрешение на редактирование информации сообщества.

- Разрешение на бан/разбан участников.

- Разрешение на публикацию статей.

- Разрешение на публикацию новостей.

**1.5.2. Роли.**

- Возможность создания, удаления и выдача ролей.

**1.5.1. Статьи и новости.**

- Возможность создания, редактирования и удаления новостей и статей.

- Статьи и новости могут создаваться локально и глобально.

- Локальные статьи или новости появятся только в сообществе и будут доступны к просмотру только их участниками.

**1.5. Чаты**

**1.5.1. Чаты между пользователями.**

- Возможность создания, блокировки и разблокировки чатов.

**1.5.2. Чаты сообществ.**

- Возможность создания, блокировки и разблокировки чатов.

- Возможность блокировки пользователей чата.

- Удаление чатов из списка чатов.

- Чаты должны иметь название и список участников.

- Возможность отправки текстовых сообщений в чате.

**2. Технические требования**

**2.1. Backend**

- Бэкенд должен быть написан на языке программирования Python с использованием фреймворка Django.

- Для базы данных использовать PostgreSQL.

**2.2. Frontend**

Фронтенд должен быть написан на языке программирования JavaScript с использованием фреймворка React.

Для управления состоянием приложения использовать Redux.

**2.3. Аутентификация и авторизация**

Для аутентификации и авторизации пользователей использовать библиотеку Django REST framework.

Использовать JWT-токены для аутентификации и авторизации пользователей.

**2.4. API**

Создать RESTful API для взаимодействия между фронтендом и бэкендом.

Документировать API с помощью Swagger.

**Дизайн**

**3.1. Общий стиль**

Использовать современный дизайн, удобный для пользователей.

Стиль должен быть согласован с темой сообщества онлайн работников.

**3.2. Адаптивный дизайн**

Обеспечить адаптивный дизайн, чтобы сайт корректно отображался на различных устройствах (компьютерах, планшетах, мобильных телефонах).

**3.3. UI элементы**

Использовать стандартные UI элементы, чтобы пользователи быстро могли разобраться в интерфейсе.

Обеспечить хорошую читаемость контента, выбрав соответствующие шрифты и цвета.

**Безопасность**

**4.1. Защита от XSS и CSRF атак**

Обеспечить защиту от атак типа XSS (межсайтовый скриптинг) и CSRF (межсайтовая подделка запроса).

Использовать механизмы защиты, предоставляемые Django и React.

**4.2. Защита паролей**

Хранить пароли пользователей в зашифрованном виде, используя хеширование и соль.

Ограничить число неудачных попыток ввода пароля, чтобы предотвратить атаки перебором.

**Юнит-тестирование**

Написать юнит-тесты для всех моделей, представлений и сериализаторов.

**Развёртывание (PS.kz)**

**6.1. Хостинг**

**6.2. HTTPS**

**6.3 SSL**