### Техническое задание (ТЗ)

#### 1. Введение

**Цель проекта**: Создание веб-приложения для организации и управления мероприятиями, где пользователи могут создавать мероприятия, приглашать участников, а участники могут регистрироваться на мероприятия и получать напоминания.

**Технологии**:

* Backend: Django
* База данных: PostgreSQL
* Frontend: HTML, CSS, JavaScript
* Вспомогательные библиотеки: Django REST Framework (DRF), jQuery/Ajax

#### 2. Основные функции

##### 2.1 Пользовательский интерфейс

**Регистрация и авторизация пользователей**

* + Регистрация нового пользователя с валидацией данных. #1 валидация
  + Вход в систему с проверкой подлинности.
  + Восстановление пароля через email. ?

**Просмотр доступных мероприятий**

* + Главная страница с перечнем всех предстоящих мероприятий.
  + Фильтрация и поиск мероприятий по названию, дате, месту проведения и категории.

**Регистрация на мероприятия**

* + Возможность зарегистрироваться на выбранное мероприятие.
  + Просмотр статуса регистрации. ?

##### 2.2 Административный интерфейс

**Создание и управление мероприятиями**

* + Создание нового мероприятия с указанием названия, описания, даты, времени, места проведения, организатора и категории.
  + Редактирование и удаление мероприятий.
  + Просмотр списка зарегистрированных участников.

**Управление участниками**

* + Добавление и удаление участников вручную.
  + Отправка приглашений участникам по email. ?
  + Управление статусами участников (подтвержден, в ожидании, отменен). ?

##### 2.3 Дополнительные функции

**Календарь мероприятий**

* + Интерактивный календарь для просмотра мероприятий по датам.

**Уведомления**

* + Уведомления по email о предстоящих мероприятиях и изменениях в расписании.
  + Напоминания о предстоящих мероприятиях за день до начала.

**Отзывы и рейтинги мероприятий**

* + Возможность оставлять отзывы и оценки по завершению мероприятия.

#### 3. Технические требования

##### 3.1 Backend

* **Django**: Создание моделей, представлений и форм для работы с мероприятиями и пользователями.
* **Django REST Framework**: Создание API для взаимодействия с фронтендом.
* **PostgreSQL**: Хранение данных о пользователях, мероприятиях и регистрациях.
* **Django Admin**: Использование встроенной админ-панели для управления данными.

##### 3.2 Frontend

* **HTML/CSS**: Разметка и стилизация страниц.
* **JavaScript/jQuery**: Обработка событий, валидация форм и AJAX-запросы.
* **Bootstrap или другой CSS-фреймворк**: Для упрощения создания адаптивного дизайна.

##### 3.3 API

* **API для мероприятий**: Создание, редактирование, удаление и получение списка мероприятий.
* **API для пользователей**: Регистрация, авторизация, получение профиля пользователя.
* **API для регистрации на мероприятия**: Регистрация и управление статусом регистрации.

#### 4. Пользовательские роли

**Обычные пользователи**

* + Регистрация и авторизация.
  + Просмотр и регистрация на мероприятия.
  + Просмотр статуса регистрации.
  + Оставление отзывов и оценок.

**Организаторы**

* + Все права обычного пользователя.
  + Создание, редактирование и удаление своих мероприятий.
  + Управление регистрациями и участниками.

**Администраторы**

* + Все права организатора.
  + Управление всеми мероприятиями и пользователями.
  + Доступ к административной панели Django.

#### 5. Примерная структура базы данных

1. **User**: (имя, email, пароль, роль)
2. **Event**: (название, описание, дата, время, место проведения, организатор, категория)
3. **Registration**: (пользователь, мероприятие, статус)
4. **Feedback**: (пользователь, мероприятие, отзыв, оценка)

#### 6. Примерные URL маршруты

1. **/register**: Регистрация пользователя.
2. **/login**: Авторизация пользователя.
3. **/events**: Просмотр всех мероприятий.
4. **/events/<id>**: Просмотр информации о конкретном мероприятии.
5. **/events/create**: Создание нового мероприятия (для организаторов).
6. **/events/<id>/edit**: Редактирование мероприятия (для организаторов).
7. **/events/<id>/register**: Регистрация на мероприятие.
8. **/admin**: Админ-панель для администраторов.

#### 7. План работ

**Разработка модели данных**

* + Создание моделей и миграций для базы данных.
  + Настройка связей между моделями.

**Создание API**

* + Настройка Django REST Framework.
  + Создание API для пользователей, мероприятий и регистраций.

**Разработка пользовательского интерфейса**

* + Создание страниц для регистрации, авторизации, просмотра и создания мероприятий.
  + Интеграция с API.

**Реализация административного интерфейса**

* + Настройка Django Admin.
  + Создание страниц для управления мероприятиями и пользователями.

**Тестирование и отладка**

* + Написание тестов для проверки функциональности.
  + Отладка и исправление ошибок.

**Документация**

* + Создание документации по установке и использованию приложения.
  + Написание пользовательской и технической документации.