**System Controle Project**

**Общее описание**

Цель проекта

Создание упрощенной системы управления проектами и задачами для командной работы, с базовым функционалом, но в компактной форме.

**Основные возможности**

* *- Создание/редактирование проектов с указанием сложности и статуса*
* *- Управление задачами внутри проектов*
* *- Система ролей (Администратор/Исполнитель)*
* *- Форум для обсуждений (в будущих обновлениях)*
* *- Управление участниками проектов*
* *- Персональные профили с настройками*
* *- Модерация пользователей (блокировка, удаление уч.записи)*

**Модуль views.py**

**Основные функции**

**home(request)**

1. Назначение:

* Главная страница с отображением проектов

1. Сортировки:

* По алфавиту (*возрастание/убывание*)
* По сложности (*Низкий/Средний/Высокий*)
* По статусу (*Выполнено/В разработке/Ожидае*т)
* По дате создания (*новые/старые*)

**project management**

* -**add\_project** - *Создание* *нового* *проекта*
* - **edit\_project** - *Редактирование* *параметров* *проекта (только для владельца)*
* - **delete\_project** - *Удаление проекта (только для владельца)*
* - **join\_project/unjoin\_project** - *Вход/выход из участников проекта*

**task management**

* - **add\_task** - *Создание задачи в проекте*
* - **edit\_task** - *Редактирование задачи*
* - **delete\_task** - *Удаление задачи (только создатель)*
* **- join\_task/unjoin\_task** - *Взятие/отказ от задачи*
* - **done\_task** - *Отметка задачи как выполненной*

**user management**

* - **ban\_user/unban\_user** - *Блокировка/разблокировка пользователей (админы)*
* - **delete\_profile** - *Удаление профиля (админы)*
* - **profile\_settings** - *Редактирование персональных настроек*

**Forum  
(в последующих обновлениях)**

* - **forum** *- Просмотр всех постов*
* - **create\_post** - *Создание нового обсуждения*
* - (*реализация комментариев требует дополнения*)

**Система безопасности**

* - **@login\_required** - *Требование авторизации*
* - *Проверки прав доступа:*
* - *Только владелец может редактировать/удалять проект*
* - *Только администраторы могут блокировать пользователей*
* - *Запрет на самоблокировку*

**Модуль models.py**

3.1 Основные модели

**Def Project**

class Project(models.Model):

* title = models.CharField(max\_length=100) # *Название*
* description = models.TextField() # *Описание*
* owner = models.ForeignKey(User) # *Владелец*
* participants = models.ManyToManyField(User) # *Участники*
* status = models.CharField(...) # *Статус* *проекта*
* complexity = models.CharField(...) # *Сложность*

**Task**

class Task(models.Model):

* project = models.ForeignKey(Project) # *Связь* *с* *проектом*
* creator = models.ForeignKey(User) # *Создатель*
* assignee = models.ForeignKey(User) # *Исполнитель*
* priority = models.CharField(...) # *Приоритет*
* status = models.CharField(...) # *Статус* *выполнения*
* code = models.TextField() # *Поле* *для* *кода*

**Profile**

class Profile(models.Model):

* puser = models.OneToOneField(User) # *Связь с пользователем*
* role = models.CharField(...) # *Роль в системе*
* status = models.CharField(...) # *Блокировка*
* avatar = models.ImageField(...) # *Аватарка с автоочисткой*
* contacts = models.TextField() # *Контактная информация*

**3.2 Вспомогательные модели**

**Post**

(в последующих обновлениях)

* - Обсуждения на форуме с полями:
* - title (заголовок)
* - text (полное описание)
* - creator (автор)

**Comment**

(в последующих обновлениях)

* - Комментарии к постам (требует реализации в views)

**Status/Language/Role**

* - Справочники для выбора значений в основных моделях

**Особенности реализации**

1. Система файлов:

* *- Автоматическое удаление аватарок при изменении профиля*
* *- Хранение изображений в `/profile\_avatars/`*

2. Безопасность:

* *- Проверка прав доступа перед критическими операциями*
* *- Защита от CSRF через Django-формы*

3. Дополнительные функции:

* *- Система сортировки проектов*
* *- Отправка email (заготовка)*
* *- Менеджер участников проекта*

4. Расширения User:

* *- Кастомный профиль с социальными ссылками*
* *- Система ролей и блокировок*

**5. Примеры использования**

Сценарий 1: Создание проекта

1. *Авторизация*
2. *Переход в "Добавить проект"*
3. *Заполнение формы с выбором языка и сложности*
4. *Автоматическое назначение владельцем*

Сценарий 2: Работа с задачей

1. *Вход в проект*
2. *Создание задачи с описанием*
3. *Назначение исполнителя (или самовыбор)*
4. *Отметка выполнения*

Сценарий 3: Модерация

1. *Просмотр списка пользователей*
2. *Блокировка нарушителей*
3. *Управление ролями*

**6. Перспективы развития**

1. Реализация системы комментариев
2. Добавление истории изменений
3. Уведомления по email
4. API для интеграции
5. Система тегов и поиска
6. Доработка и выпуск мини-форума
7. Выдача предупреждений в профиль нарушителям