README.md

Task Management API

This project provides a set of APIs for a task management app using Django and Django Rest Framework (DRF). The system allows users to create tasks, assign tasks to users, and retrieve tasks assigned to specific users.

Setup Instructions

1. Prerequisites

- Python 3.x installed on your system.
- pip (Python package manager) installed.

2. Clone the Repository

1. ZIP-FILE

- bash: unzip task-management.zip
- bash: cd task-management

2. Clone Repository

- git clone https://github.com/chandanschandu/Task-Management-API.git
- cd Task-Management-API

3. Create a Virtual Environment

- bash: python -m venv venv

Activate the virtual environment:

On Windows : venv\Scripts\activate

On macOS/Linux: source venv/bin/activate

4. Install Dependencies

Install the required Python packages using pip:

- bash: pip install -r requirements.txt

5. Run Migrations

Apply database migrations to set up the necessary tables

- bash: python manage.py migrate

6. Start the Development Server

Run the Django development server

- bash: python manage.py runserver

The API will now be accessible at: http://127.0.0.1:8000/api/

API Endpoints

1. **Create a User**
   Create a new user in the system.

- Endpoint : **POST /api/users/create/**
- Request: bash

```
POST /api/users/create/
Content-Type: application/json

{
  "name": "chandan",
  "email": "chandan@example.com",
  "mobile": "1234567890"
}
```

- Response :json

```
{
  "id": 4,
  "name": "chandan",
  "email": "chandan@example.com",
  "mobile": "1234567890"
}
```

2. **Create a Task**
   Create a new task in the system.
   Allows the creation of new tasks with a name and description.

- Endpoint : **POST /api/tasks/create/**
- Request : bash

```
POST /api/tasks/create/
Content-Type: application/json

{
  "name": "Complete Django Project",
  "description": "Finish the API development task.",
  "task_type": "work"
}
```

- Response : json

```
{
  "id": 1,
  "name": "Complete Django Project",
  "description": "Finish the API development task.",
  "created_at": "2023-10-01T12:00:00Z",
  "completed_at": null,
  "status": "pending",
  "task_type": "work",
  "assigned_to": []
}
```

3. **Assign a Task :** Assign a task to one or multiple users.
   Enables assigning a task to one or multiple users.

- Endpoint : **POST /api/tasks/assign/<int:task_id>/**
- Request : bash

```
POST /api/tasks/assign/1/
Content-Type: application/json

{
  "assigned_to": [1, 2]
}
```

- Response: json

```
{
  "message": "Task assigned successfully"   }
```

4. **Get Tasks for a User**

Fetches all tasks assigned to a particular user.

- Endpoint : **GET /api/tasks/user/<int:user_id>/**
- Request :

```
GET /api/tasks/user/1/
```

- Response :

```
[
  {
    "id": 1,
    "name": "Complete Django Project",
    "description": "Finish the API development task.",
    "created_at": "2023-10-01T12:00:00Z",
    "completed_at": null,
    "status": "pending",
    "task_type": "work",
    "assigned_to": [1, 2]
  }
]
```

Admin Interface : Access the Admin Panel

- URL: http://127.0.0.1:8000/admin/
- Login with the superuser credentials:
  Username: admin
  Password: admin
- Features

  Manage Users : Add, edit, or delete users.

  Manage Tasks : Add, edit, or delete tasks, and assign them to users.

Test Credentials

Users

ID: 4, Name: chandan, Email: chandan@example.com

ID: 5, Name: charan, Email: kanta@gmial.com

Tasks

ID : 1, Name : "Complete Django Project", Task Type : "work"

ID : 3, Name : "Complete frapee Project" ", Task Type : "work"