# Construction Management System Documentation

## Project Overview

This Django-based project is an API for managing a construction project. It enables user registration, login, logout, change password  
authentication, and management of resources (such as materials and workers). The project also supports image uploads   
and worker role-based actions, such as supervisors adding new workers.

## 1. User Management

- Allows new users to register with a username, email, password, and role (either manager or supervisor.

### Registration

Path: /register/  
Method: POST  
Description: Allows new users to register with specific roles.

#### Parameters:

- username (string, required): The unique identifier for the new user.  
- email (string, required): User's email address.  
- password (string, required): Secure password for authentication.  
- role (string, required): Specifies user role (manager or supervisor).

#### Sample Request:

{  
 "username": "user",  
 "email": "user@example.com",  
 "password": "securepassword123",  
 "role": "supervisor"  
}

#### Sample Response:

Success (201 Created):  
{  
 "message": "User registered successfully.",  
 "user": {  
 "id": 1,  
 "username": "user",  
 "role": "supervisor"  
 },  
 "token": "abc12345token"  
}  
Error (400 Bad Request):  
{  
 "error": "Invalid data. Please check the fields."  
}

### Login

Path: /login/  
Method: POST  
Description: Authenticates a registered user by verifying credentials and issuing an access token.

#### Parameters:

- username (string, required): Username of the user.  
- password (string, required): User's password.

#### Sample Request:

{  
 "username": "user",  
 "password": "securepassword123"  
}

#### Sample Response:

Success (200 OK):  
{  
 "message": "Login successful.",  
 "token": "abc12345token"  
}  
Error (401 Unauthorized):  
{  
 "error": "Invalid username or password."  
}

### Logout

Path: /logout/  
Method: POST  
Description: Logs out the user by deleting the authentication token.

#### Sample Response:

Success (200 OK):  
{  
 "message": "Logout successful."  
}  
Error (401 Unauthorized):  
{  
 "error": "Authentication credentials were not provided."  
}

### Change Password

Path: /change\_password/  
Method: POST  
Description: Allows users to change their password.

#### Parameters:

- current\_password (string, required): Current password of the user.  
- new\_password (string, required): New password (minimum 6 characters).

#### Sample Request:

{  
 "current\_password": "oldpassword",  
 "new\_password": "newpassword123"  
}

#### Sample Response:

Success (200 OK):  
{  
 "message": "Password changed successfully."  
}  
Error (400 Bad Request):  
{  
 "error": "Current password is incorrect."  
}

### User Detail

Path: /user/  
Method: GET  
Description: Fetches details of the currently authenticated user.

#### Sample Response:

Success (200 OK):  
{  
 "id": 1,  
 "username": "user",  
 "email": "user@example.com",  
 "role": "supervisor"  
}  
Error (401 Unauthorized):  
{  
 "error": "Authentication credentials were not provided."  
}

## 2. Project and Resource Management

Handles project-specific actions, including uploading images and managing construction resources.

### Image Upload

Path: /upload/  
Method: POST  
Description: Uploads project images with detailed project information.

#### Parameters:

- project\_name (string, required): Name of the project.  
- project\_image (file, required): Image file related to the project.  
- project\_location (string, required): Location of the project.  
- project\_start\_date (date, required): Start date of the project.  
- project\_end\_date (date, required): Expected end date.  
- people\_working (integer, optional): Number of workers involved.

#### Sample Request:

{  
 "project\_name": "Project S",  
 "project\_image": "image.jpg",  
 "project\_location": "Manikonda",  
 "project\_start\_date": "2023-01-01",  
 "project\_end\_date": "2024-12-31",  
 "people\_working": 25  
}

#### Sample Response:

Success (201 Created):  
{  
 "message": "Image uploaded successfully.",  
 "image\_id": 1  
}  
Error (400 Bad Request):  
{  
 "error": "Invalid image file."  
}

## 3. Worker Management

Provides endpoints for supervisors to manage workers assigned to them.

### Add Worker

Path: /add\_worker/  
Method: POST  
Description: Allows supervisors to add workers and assign specific job titles. Only supervisors can access this endpoint.

#### Parameters:

- name (string, required): Name of the worker.  
- job\_title (string, required): Job title assigned to the worker.  
- hired\_by (integer, optional): ID of the supervisor (only accessible by supervisors).

#### Sample Request:

{  
 "name": "karthik",  
 "job\_title": "Electrician"  
}

#### Sample Response:

Success (201 Created):  
{  
 "message": "Worker added successfully.",  
 "worker\_id": 1  
}  
Error (403 Forbidden):  
{  
 "error": "Permission denied. Only supervisors can add workers."  
}