

# Django Project-Based Examination

**Total Marks:** 100

**Duration:** 3 hours

**Submission Deadline:** Same Day

---

## Instructions

- Complete all sections of the exam.
  - Document your work clearly, including code, diagrams, and explanations.
  - Save the document as: [Student Name] - Task Management System - [Date] - Exam 05.
- 

## Project Overview

### Project Title: Task Management System

Create a web application that allows users to manage tasks, including user authentication, task management, category organization, and comments. The application should provide both a web interface and a RESTful API.

---

## Project Tasks

### 1. Create a New Django Project (5 marks)

- Create a new Django project named `task_manager`.
- Create a new app within the project called `tasks`.

### 2. User Authentication (20 marks)

- Implement user registration functionality where users can sign up with a username, email, and password (7 marks).
- Enable users to log in and log out using Django's built-in authentication system (7 marks).
- Document the authentication process in your README file (6 marks).

### 3. Define Models (25 marks)

- **Task Model:** Create a `Task` model with the following fields:
  - Title (`CharField`)
  - Description (`TextField`)
  - Status (`BooleanField`)
  - Created At (`DateTimeField`)

- Updated At (DateTimeField)
- Assigned To (ForeignKey to User model) (12 marks)
- **Category Model:** Create a Category model with:
  - Name (CharField)
  - Description (TextField) (6 marks)
- **Comment Model:** Create a Comment model with:
  - Content (TextField)
  - Created At (DateTimeField)
  - ForeignKey to Task model
  - ForeignKey to User model (7 marks)

#### 4. Implement CRUD Operations (30 marks)

- **Task Endpoints:**
  - Create a task (POST) (6 marks)
  - Retrieve all tasks (GET) (5 marks)
  - Retrieve a single task by ID (GET) (5 marks)
  - Update a task (PUT) (6 marks)
  - Delete a task (DELETE) (6 marks)
- **Category Endpoints:**
  - Create a category (POST) (3 marks)
  - Retrieve all categories (GET) (3 marks)
  - Update a category (PUT) (3 marks)
  - Delete a category (DELETE) (3 marks)
- **Comment Endpoints:**
  - Create a comment for a task (POST) (3 marks)
  - Retrieve comments for a specific task (GET) (3 marks)
  - Delete a comment (DELETE) (3 marks)

#### 5. Frontend Integration (10 marks)

- Use Django's template engine to create:
  - A task creation form (2 marks)
  - A page to list all tasks with categories and comments (4 marks)
  - A form for adding comments (2 marks)
  - A task update form (2 marks)

#### 6. Database Schema (5 marks)

- Create an ERD (Entity-Relationship Diagram) for your models and include it in your documentation (5 marks).

#### 7. Postman Collection (5 marks)

- Create and export a Postman collection documenting all your API endpoints (5 marks).

#### 8. Documentation (5 marks)

- Prepare a README file that includes:
  - Project title and introduction

- Installation instructions
- Usage instructions
- API documentation (5 marks).

#### 9. **Presentation Practice** (20 marks)

- Be prepared to explain your project, code, and logic in a short presentation. This includes:
    - Clarity and structure of your explanation (5 marks)
    - Understanding of the project's functionality and code (10 marks)
    - Ability to answer questions and provide insights on your design decisions (5 marks)
- 

## **Submission Requirements**

- Upload all project files to a GitHub repository.
  - Include your README, ERD, and Postman collection in the repository.
  - Be prepared to present your project, explaining the code and its functionality.
- 

## **Evaluation Criteria**

Your project will be evaluated based on functionality, code quality, documentation, and your presentation.

---

**Good luck!**