

# Django Task Manager - Technical Documentation

## 1. Introduction

This document provides an industry-standard technical overview and setup guide for the Django Task Manager application. It outlines installation, configuration, core functionality, and additional features including Celery-based background task scheduling, REST API endpoints, and email notifications.

## 2. System Requirements

- Python 3.12 or higher
- Django 5.x
- Redis (for Celery message broker)
- Celery 5.x
- A working SMTP email server for notifications

## 3. Installation Steps

1. Clone the repository and navigate into the project directory.

2. Create a Python virtual environment:

```
python -m venv env
```

3. Activate the virtual environment:

- Windows: env\Scripts\activate
- Mac/Linux: source env/bin/activate

4. Install dependencies:

```
pip install -r requirements.txt
```

5. Apply migrations:

```
python manage.py migrate
```

6. Create a superuser:

```
python manage.py createsuperuser
```

7. Run the development server:

```
python manage.py runserver
```

## 4. Application Features

- User authentication (Register, Login, Logout)
- Task management (Create, Update, Delete, Mark Completed)
- Dashboard filtered views (All, Completed, Pending)

- Due date validation (no past dates allowed)
- Automated email reminders 1 day before task due date (Celery + Redis)
- Django REST API for CRUD operations

## 5. Celery & Redis Setup

### 1. Install Redis:

- Windows: Use WSL or download from

<https://github.com/microsoftarchive/redis/releases>

- Mac: brew install redis
- Linux: sudo apt install redis-server

### 2. Install Celery:

pip install celery

### 3. Start Redis server:

redis-server

### 4. Start Celery worker:

celery -A task\_manager worker --loglevel=info

### 5. Start Celery beat scheduler:

celery -A task\_manager beat --loglevel=info

## 6. REST API Endpoints

- GET /api/tasks/ – List all tasks
- POST /api/tasks/ – Create a new task
- PUT /api/tasks/<id>/ – Update a task
- DELETE /api/tasks/<id>/ – Delete a task

## 7. Email Notification Setup

Configure SMTP settings in settings.py:

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.example.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'your_email@example.com'
EMAIL_HOST_PASSWORD = 'your_password'
```

## **8. Conclusion**

The Django Task Manager is a scalable and extendable application that integrates core task tracking features with background scheduling and API accessibility. This document provides a reference for both deployment and future enhancements.