# **Django To-Do List with Popup Reminders**

## **Overview**

A simple Django-based to-do list application with email and popup reminders for tasks due within 10 minutes. Built with Django, Bootstrap, and cron jobs for automated reminders.

## **Features**

* **Task Management**: Add, edit, view, and delete tasks with due dates.
* **Email Reminders**: Sends email reminders for tasks due within 10 minutes (logged to console with console.EmailBackend).
* **Popup Reminders**: Displays a popup on the /tasks/ page for tasks due within 10 minutes.
* **Cron Jobs**: Automates reminders using send\_reminders and mark\_popup\_reminders management commands.

## **Project Structure**

* **Base App (base/)**:
  + models.py: Defines Task and CustomUser models.
  + views.py: Handles task CRUD operations.
  + templates/base/task\_list.html: Renders the task list with a popup reminder.
  + management/commands/send\_reminders.py: Sends email reminders.
  + management/commands/mark\_popup\_reminders.py: Marks tasks for popup reminders.
* **Settings (todo\_list/settings.py)**:
  + Configures Django settings, including EMAIL\_BACKEND = 'django.core.mail.backends.console.EmailBackend'.

## **Setup Instructions**

### **Prerequisites**

* Python 3.9+
* Virtual environment (myenv)
* Cron (for scheduling reminders)

### **Installation**

1. **Clone the Repository**:  
    text  
   CollapseWrapCopy

git clone <repository-url>

1. cd todo\_list
2. **Set Up Virtual Environment**:  
    text  
   CollapseWrapCopy

python3 -m venv myenv

1. source myenv/bin/activate
2. **Install Dependencies**:  
    text  
   CollapseWrapCopy
3. pip install -r requirements.txt
4. **Run Migrations**:  
    text  
   CollapseWrapCopy

python3 manage.py makemigrations

1. python3 manage.py migrate
2. **Create a Superuser**:  
    text  
   CollapseWrapCopy
3. python3 manage.py createsuperuser
4. **Run the Development Server**:  
    text  
   CollapseWrapCopy
5. python3 manage.py runserver

### **Configure Cron Jobs**

The app uses cron jobs to send email reminders and mark tasks for popup reminders every 10 minutes.

1. **Edit Crontab**:  
    text  
   CollapseWrapCopy
2. crontab -e
3. Add the following lines:  
    text  
   CollapseWrapCopy

# Run Django management commands every 10 minutes

\*/10 \* \* \* \* /bin/bash -c "export DJANGO\_SETTINGS\_MODULE=todo\_list.settings && cd /home/utkarsh/todo\_list && /home/utkarsh/todo\_list/myenv/bin/python3 /home/utkarsh/todo\_list/manage.py send\_reminders >> /home/utkarsh/todo\_list/cron.log 2>&1"

1. \*/10 \* \* \* \* /bin/bash -c "export DJANGO\_SETTINGS\_MODULE=todo\_list.settings && cd /home/utkarsh/todo\_list && /home/utkarsh/todo\_list/myenv/bin/python3 /home/utkarsh/todo\_list/manage.py mark\_popup\_reminders >> /home/utkarsh/todo\_list/cron.log 2>&1"
2. **Ensure Cron Is Running**:  
    text  
   CollapseWrapCopy

sudo systemctl status cron

sudo systemctl start cron

1. sudo systemctl enable cron

## **Usage**

1. **Access the App**:
   * Go to http://localhost:8000/tasks/ and log in (e.g., as testuser2).
2. **Create a Task**:
   * Add a task with a due date (e.g., 5 minutes from now):  
      text  
     CollapseWrapCopy

python3 manage.py shell

from base.models import Task, CustomUser

from django.utils import timezone

user = CustomUser.objects.get(username='testuser2')

Task.objects.create(

user=user,

title="Popup Test for Testuser2",

due\_date=timezone.now() + timezone.timedelta(minutes=5)

)

* + exit()

1. **Wait for Reminders**:
   * **Email Reminder**: Sent when the task is due within 10 minutes (logged to console).
   * **Popup Reminder**: Appears on the /tasks/ page when the task is due within 10 minutes.

## **Popup Reminder Feature**

* **Appearance**: A red popup in the top-right corner of the /tasks/ page.
* **Message**: Task "Popup Test for Testuser2" is due in 4 minutes!.
* **Dismiss Button**: Click "Dismiss" to hide the popup.

## **Troubleshooting**

* **Cron Jobs Not Running**:
  + Check the cron log: cat /home/utkarsh/todo\_list/cron.log.
  + Verify cron service: sudo systemctl status cron.
  + Ensure paths in crontab match your setup.
* **Popup Not Appearing**:
  + Ensure mark\_popup\_reminders has run.
  + Check browser console (F12) for JavaScript errors.
* **Email Not Sending**:
  + Verify EMAIL\_BACKEND in settings.py.
  + Check console for email logs.