Django Project Documentation

# Event Countdown System

Unit: Internet Application Programming

Unit Code: ICS 2203

**Group Members**

**Name: Emmanuel Mwangangi Reg No: SCT 212-0331/2023**

**Name: Dennis Kirehu Reg No: SCT 212-0045/2023**

**Name: Cecil Kioko Reg No: SCT 212-0047/2023**

**Name:Roy Marigi Reg No:SCT212-0054/2023**

## Project Overview

Name: Event Countdown System

Purpose:  
To allow users to create and manage countdowns for upcoming events. Users can input event details including name, description, and date, and the system displays a live countdown timer to each event.

## Features

- Add upcoming events with name, description, and date.  
- View all added events in a list.  
- Real-time countdown timers using client-side JavaScript.

## Technologies Used

- Backend: Django (Python)  
- Frontend: HTML, CSS, JavaScript  
- Database: SQLite  
- Styling: Tailwind CSS (via CDN)  
- Others: Django Template Engine

## Distinctiveness and Complexity

The project is distinctive due to its integration of dynamic frontend JavaScript timers with Django’s backend logic and database. Unlike standard CRUD apps, this project introduces:

- Real-time JavaScript countdowns rendered on client side using Date objects.  
- Date-time input using HTML5’s datetime-local.  
- Data filtering and templating through Django’s engine.  
- Countdown logic that is reusable across multiple events rendered from the backend.

Complexity:  
- Dynamic frontend that updates using JavaScript and DOM manipulation.  
- Integration of Django forms and model instances with HTML5 datetime widgets.  
- Template-level handling of JavaScript data attributes.  
- Proper formatting and parsing of datetime fields between Python and JavaScript.

## Design Approach

### Model-View-Template (MVT) Architecture

- Models: Define the structure of the Event with fields like name, description, event\_date, and user.  
- Views: Controls the logic for displaying all events and adding new ones.  
- Templates: Renders data dynamically and passes countdown information to JavaScript.

### Frontend Design

- Tailwind CSS was used via CDN for fast styling and responsiveness.  
- Countdown logic’s separated into a JavaScript file loaded through Django{% static %} tag.  
- Each event dynamically attaches countdown functionality based on its HTML data attributes.

## File Structure and Descriptions

### Project Root: event\_countdown

- settings.py: Django settings for the application including app registration and static file configurations.  
- urls.py: URL routing for the entire project(contains admin url)and includes the app urls.

### App Directory: Event

- **models.py**: Contains the Event model with fields for name, description, datetime, and user.

class Event(models.Model):

**name** = models.CharField(max\_length=100)

**description** = models.TextField()

**event\_date** = models.DateTimeField()

**user** = models.ForeignKey(User, on\_delete=models.CASCADE)

- **forms.py**: Defines the EventForm for creating events.

class EventForm(forms.ModelForm):

**class Meta**:

**model** = Event

**fields** = ['name', 'description', 'event\_date']

**widgets** = {

'event\_date': forms.DateTimeInput(attrs={

'type': 'datetime-local',

'class': 'form-control'

})

}

- **views.py**: Displays the events and handles event creation.

- **urls.py**: Maps the URLs to views.

### Templates Directory: templates/Event/

- index.html: Lists all events with countdowns.  
- add\_event.html: Form to add a new event.

### Static Files: static/Event/

- countdown.js: The JavaScript countdown logic.

## How to Run the Application

1. Set up a Virtual Environment (Optional):  
 python -m venv venv  
 source venv/bin/activate   
  
2. Install Dependencies:  
 pip install django  
  
3. Navigate to the Project Directory:  
 cd event\_countdown  
  
4. Run Migrations:  
 python manage.py makemigrations  
 python manage.py migrate  
  
5. (Optional) Create a Superuser:  
 python manage.py createsuperuser  
  
6. Start the Server:  
 python manage.py runserver  
  
7. Access the App:  
 Open http://127.0.0.1:8000 in a browser.

(Has nav links for navigation between the pags Home, New Event , History)

## Additional Information

### Improved User Experience

- It has a clean UI with Tailwind CSS.  
- It has Real-time JavaScript updates.  
- There is smooth form handling using Django forms

### Potential Enhancements

- Add user authentication.  
- Have notifications for a due event  
- Add different event categories/tags.  
- Have a good Calendar view.