

# Assignment: Event Management System - CRUD and Optimized Queries with Tailwind CSS

## Objective

Create a fully functional Event Management System that demonstrates your understanding of Django's ORM, optimized query techniques, and responsive front-end design using Tailwind CSS.

---

## Requirements

### 1. Data Models (15 Marks)

Design the following models:

- **Event:**
    - Fields: `name`, `description`, `date`, `time`, `location`, `category` (ForeignKey)
  - **Participant:**
    - Fields: `name`, `email`, and a `ManyToMany` relationship with `Event`.
  - **Category:**
    - Fields: `name` and `description`.
- 

### 2. CRUD Operations (10 Marks)

- Implement **Create**, **Read**, **Update**, and **Delete** functionality for all models (`Event`, `Participant`, `Category`).
  - Ensure proper **validation** for forms to prevent invalid data submission.
- 

### 3. Optimized Queries (10 Marks)

- Use `select_related` to fetch the category of an event while listing events.
- Use `prefetch_related` to fetch participants for events.
- Add an **aggregate query** to calculate the total number of participants across all events.
- Add a **filter query** to display events based on:
  - Their category.
  - A specific date range.

---

## 4. UI with Tailwind CSS (35 Marks)

### 4.1 Responsive and User-Friendly Interface (15 Marks)

Design an intuitive and responsive interface for:

- Listing all events, including:
  - Their categories and participant counts.
- Viewing detailed information for each event, including all associated participants.
- Adding and updating events, participants, and categories through forms.

### 4.2 Navigation and Styling (5 Marks)

- Ensure consistent styling across all pages.
- Include a navigation bar with links to different sections of the application.

### 4.3 Organizer Dashboard (15 Marks)

- Create an **Organizer Dashboard** that displays the following:
  1. **Stats Grid:**
    - Total number of participants.
    - Total number of events.
    - Number of upcoming events.
    - Number of past events.
  2. **Today's Events Listing:**
    - Below the stats grid, show a list of events scheduled for today.
  3. **Interactive Stats:**
    - Clicking on **Total Events**, **Upcoming Events**, or **Past Events** dynamically updates the data displayed below the stats grid.

---

## 5. Search Features (10 Marks)

- Implement search functionality to find events by **name** or **location**.
- **Hint:**
  - Use `request.GET` to capture the search text.
  - Query the database using `icontains`.

---

## Grading Breakdown

- **Data Models:** 15 Marks
  - **CRUD Operations:** 10 Marks
  - **Optimized Queries:** 10 Marks
  - **UI with Tailwind CSS:** 35 Marks
  - **Search Features:** 10 Marks
- 

## Additional Notes

- Ensure the system is well-structured and follows Django's **MVT architecture**.
- Use Tailwind CSS classes to create a responsive and visually appealing UI.
- Test optimized queries using Django's **Debug Toolbar** to highlight query efficiency.
- Ensure all forms and views are fully functional and user-friendly.