Assignment-2: Advanced Features for Event Management System

Exam Submission:

1. **Deploy Live Link**
2. **GitHub Repository Link**
3. **Admin panel username and password**

Objective:

Extend the Event Management System built in **Assignment-1** by implementing user authentication, role-based access control (RBAC), email-based account activation, Django signals, and RSVP functionality.

Requirements & Instructions:

1. Authentication (10 Marks)

- Implement **user authentication** features:
  - User signup with username, email and password, first_name, last_name.
  - Login and logout functionality.

2. Role-Based Access Control (RBAC) (30 Marks)

- Define roles for users:
  - **Admin**: Full access to all features. Only the admin can change roles, create groups, delete groups, and delete participants.
  - **Organizer**: Can create, update, and delete **events** and categories
  - **Participant**: Replace Participant model with **User model**. Participants can only view events.
- Use **Django Groups** to assign roles to users.
- Restrict access to views based on roles using **decorators.**

3. RSVP System (25 Marks)

- Participants must be able to **RSVP to events** they are interested in.
- Store RSVP responses in a **ManyToMany relationship** between Users and Events.
- Ensure that a user **cannot RSVP more than once** for the same event.
- Add a **confirmation email** when a user RSVPs for an event.

- Provide a way for users to **view events they have RSVP'd to** in their **Participant Dashboard**.

4. Email Activation (15 Marks)

- Upon signup, send an **activation email** to the user.
- Include a **secure activation link** in the email.
- Prevent unactivated users from logging in.
- Use Django's **default_token_generator** for activation.

5. Django Signals & Media Files(15 Marks)

- Automate processes using **signals**:
    - ○ Send an **email notification** to participants when they RSVP'd an event
    - ○ Send account **activation email** to registered email
- Add ImageField For Event Model. Must give a d**efault value with a default image**

6. Maintain User-Specific Dashboards (5 Marks)

- Redirect users to their respective dashboards after login:
    - ○ **Admin Dashboard**: Manage all events, participants, and categories.
    - ○ **Organizer Dashboard**: Manage only **events and categories**.
    - ○ **Participant Dashboard**: View **events they have RSVP'd to**.

Grading Breakdown:

Section	**Marks**
Authentication	10
Role-Based Access Control (RBAC)	30
RSVP System	25
Email Activation	15
Django Signals	15
User-Specific Dashboards	5
Total	**100 Marks**

Submission Guidelines:

1. **Branch Create:** Create a new git branch named `assignment-2` and commit all changes here
2. **Code Submission**: Push your Django project to **GitHub** and provide the repository assignment-2 branch link.
    a. Command to push specific branch `git push origin assignment-2`
3. **Live Link:** Deploy your project and provide the live link
    a. Checkout to main branch `git checkout main`
    b. Merge to main: `git merge assignment-2`
    c. Push to main branch: `git push origin main`
    d. Then deploy it using **manual deploy** on render