

Mid Term Exam: Enhancing Event Management System

Objective:

Enhance the Event Management System by refactoring function-based views (FBVs) into class-based views (CBVs), implementing user profile features, and creating a custom user model.

Requirements & Instructions:

1. Convert Function-Based Views to Class-Based Views (25 Marks)

- Convert **at least five (5) function-based views** to class-based views.
- List the modified views in your submission document.
- Ensure the CBVs maintain the same functionality as before.

2. Implement Profile Features (30 Marks)

- **Profile Page:** Users should be able to view their profile.
- **Edit Profile:** Allow users to update their information (first name, last name, profile picture, phone number, etc.).
- **Change Password:** Implement a password change feature within the profile section.
- **Reset Password:** Provide an email-based password reset functionality.

3. Create a Custom User Model (25 Marks)

- Modify the default Django `User` model to include:
 - **Profile Picture** (`ImageField` with a default image)
 - **Phone Number** (`CharField` with validation)
 - Ensure the profile picture is properly stored and served.
 - Update authentication forms to use the custom user model.
-

Grading Breakdown (Total: 80 Marks):

Section	Marks
Convert Function-Based Views to Class-Based Views	25
Implement Profile Features	30
Create a Custom User Model	25
Practice Problem 17.5	20
Total	100

Submission Guidelines: Provide a google docs link with the following information

- Provide a Document with the Following Information:**
 - **GitHub branch link:** `mid-term-exam`
 - **Live deployed link**
 - **Modified views list (FBV → CBV)**
 - **Admin panel credentials and test users for different roles**
 - Code Submission:**
 - Create a new Git branch named `mid-term-exam`
 - Commit and push all changes to `mid-term-exam`
 - Merge `mid-term-exam` into `main` and push to GitHub
 - Deploy the project and provide the live link
 - Provide Credentials for Different User Roles:**
 - Admin
 - Organizer
 - Participant
 - View Conversion Documentation:**
 - Clearly list the names of the views that were converted from function-based to class-based views with full path
 - `<your_app_name>.views.py`
 1. ViewName 1
 2. ViewName 2
-