### **Assignment**

### ***"Building a Campus Event Management System with Role-Based Access"***

### **Objective**

Design and implement a **Campus Event Management System** that allows:

1. Role-based access for event management.
2. Registration and tracking of attendees (students, staff, visitors).
3. Event scheduling and capacity management.

This assignment will integrate advanced Python programming skills, including OOP, file handling, and real-world application design.

### **Features and Deliverables**

#### **1. Roles and Access Control**

* **Admin**:
  + Create, update, and delete events.
  + View all events and attendees.
* **Event Organizer**:
  + Manage attendee registration for their events.
  + View event details.
* **Student/Visitor**:
  + Search for events and register.
  + View their registered events.

#### **2. Core Functionalities**

1. **Event Management**:
   * Add, update, and delete events.
   * Track event capacity (e.g., max attendees).
   * Input validation for event names, dates, and capacities.
2. **Attendee Registration**:
   * Register attendees for events with capacity checks.
   * Provide a confirmation message for successful registration.
   * Prevent duplicate registrations.
3. **Calculate**:
   * Total number of attendees across all events.
   * Events with the highest and lowest attendance.
4. **Data Persistence**:
   * Save events and attendee data to files (CSV or JSON).
   * Export statistical reports to files (e.g., CSV) for easy sharing.

#### **3. Submission Requirements**

1. **Code**: A complete Python implementation of the Event Management System.
2. **Documentation**:
   * Provide a flowchart to show the user interactions with the system.
3. **Screenshots**:
   * Demonstrating interactions for each user role.

### **Evaluation Criteria**

1. **Functionality and Coverage** (40%):
   * Includes event management, registration, and reporting.
2. **Code Quality** (20%):
   * Readable and modular code with proper use of OOP principles.
3. **Documentation** (20%):
   * Well-explained clear flowchart and description of classes/methods in the program.
4. **Creativity and Features** (20%):
   * Additional features like automated notifications or search filters.