Felicity Event Management System

Design & Analysis of Software Systems

Assignment 1

Deadline: *19th Feb 2026* **Submission:** Single ZIP file

Part - 1: Core System Implementation [70 Marks]

1 Introduction

Club Council have a long-standing tradition of somehow managing a large number of events, clubs, and participants, though with a generous dose of chaos. As Felicity draws closer, this chaos tends to take familiar forms: midnight Google Forms, barely comprehensible spreadsheets, and important information vanishing into the depths of WhatsApp groups. The result is predictable , participants are never quite sure whether they are registered, organizers are unsure who is actually attending, and payments exist in a vague limbo of "screenshots for confirmation."

With the upcoming fest approaching and the Felicity Tech Team's patience wearing thin, it has finally been decided that something must change. You have been appointed to save the fest from this organizational mess by designing a system that brings order to the madness: a centralized platform that allows clubs to conduct events smoothly and participants to register, track, and attend them without relying on a heroic combination of Google Forms, spreadsheets, and hope.

2 Technology Stack

The system **must** be implemented using the **MERN stack**:

1. **MongoDB** – Database
2. **Express.js** – Backend framework implementing REST APIs
3. **React / React-based framework** – Frontend
4. **Node.js** – Runtime

3 User Roles

Each user can have **exactly one role**. **Role switching is strictly prohibited.**

3.1 Participant

- IIIT Student

- Non-IIIT Participant

3.2 Organizer

- Clubs / Councils / Fest Teams

3.3 Admin

- System-level administrator

4 Authentication & Security [8 Marks]

4.1 Registration & Login [3 Marks]

4.1.1 Participant Registration

- **IIIT Participants**: Must register using IIIT-issued email ID only (email domain validation mandatory)
- **Non-IIIT Participants**: Must register using email and password

4.1.2 Organizer Authentication

- **No self-registration.** Organizer accounts are provisioned by the Admin.
- Organizers login using credentials provided by the Admin.
- Password resets must be requested and handled by the Admin (see Section 13.2).

4.1.3 Admin Account Provisioning

- Admin is the first user in the system.
- Admin email and password are provisioned and handled by the backend only (no UI registration).
- Admin has exclusive privileges to create/remove clubs/organizers.

4.2 Security Requirements [3 Marks]

- Passwords must be hashed using **bcrypt** (no plaintext storage)
- **JWT-based authentication** is mandatory for all protected routes
- All frontend pages (except login/signup) must be protected with role-based access control

4.3 Session Management [2 Marks]

- Login must redirect the user to their respective Dashboard; sessions must persist across browser restarts unless explicitly logged out; logout must clear all authentication tokens.

5 User Onboarding & Preferences [3 Marks]

After signup, participants (only) may select or skip the following preferences:

1. **Areas of Interest** (multiple selection allowed)

2. **Clubs / Organizers to Follow** (if applicable)

Notes:

- Participants may set these during onboarding or skip and configure later
- Preferences must be stored in the database and editable from the Profile page
- Preferences must influence event ordering and recommendations

6 User Data Models [2 Marks]

You must add additional attributes as required and justify them appropriately in the report.

6.1 Participant Details

Each participant record must store:

- First Name
- Last Name
- Email (unique)
- Participant Type
- College / Org Name
- Contact Number
- Password (hashed)

6.2 Organizer Details

Each organizer record must store:

- Organizer Name
- Category
- Description
- Contact Email

7 Event Types [2 Marks]

All activities are modeled as **Events**. Each event belongs to **exactly one** type:

7.1 Normal Event (Individual)

- Single participant registration Examples: workshops, talks, competitions

7.2 Merchandise Event (Individual)

- Used for selling merchandise (T-shirts, hoodies, kits, etc.) Individual purchase only

8 Event Attributes [2 Marks]

Each event must atleast store:

- Event Name
- Event Description
- Event Type
- Eligibility
- Registration Deadline
- Event Start Date
- Event End Date
- Registration Limit
- Registration Fee
- Organizer ID
- Event Tags

Additional Requirements by Event Type:

- **Normal**: Custom registration form (dynamic form builder)

- **Merchandise Events**: Item details (size, color, variants), stock quantity, configurable purchase limit per participant

9 Participant Features & Navigation [22 Marks]

9.1 Navigation Menu [1 Mark]

Navbar provides: **Dashboard**, **Browse Events**, **Clubs/Organizers**, **Profile**, **Logout**.

9.2 My Events Dashboard [6 Marks]

- **Upcoming Events**: Displays all registered upcoming events with essential details such as event name, type, organizer, and schedule.
- **Participation History**: Maintains a categorized record of user participation using tabs — Normal, Merchandise, Completed, and Cancelled/Rejected.
- **Event Records**: Each record includes event name, event type, organizer, participation status, team name (if applicable), and a clickable ticket ID for reference.

9.3 Browse Events Page [5 Marks]

- **Search**: Partial & Fuzzy matching on Event/Organizer names
- **Features**: Trending (Top 5/24h)
- **Filters**: Event Type | Eligibility | Date Range (works with search) | Followed Clubs | All events

9.4 Event Details Page [2 Marks]

- **Info**: Complete details | Type indicated | Registration/Purchase button (validation)
- **Blocking**: Deadline passed | Registration limit/stock exhausted

9.5 Event Registration Workflows [5 Marks]

- **Normal Event**: Upon successful submission, the ticket should be sent to the participant via email and should also be accessible in their Participation History.
- **Merchandise**: Purchase implies registration; stock is decremented upon purchase; ticket with QR is generated; confirmation email is sent; purchases of out-of-stock items are blocked.
- **Tickets & QR**: Includes event and participant details, and contains a QR code and a unique Ticket ID.

9.6 Profile Page [2 Marks]

- **Editable Fields**: First Name, Last Name, Contact Number, College/Organization Name, Selected Interests, Followed Clubs
- **Non-Editable Fields**: Email Address, Participant Type (IIIT / Non-IIIT)
- **Security Settings**: Provides a password reset or change mechanism with appropriate authentication and validation, allowing flexibility in implementation design.

9.7 Clubs / Organizers Listing Page [1 Mark]

- **List**: All approved organizers (Name, Category, Description)
- **Action**: Follow / Unfollow

9.8 Organizer Detail Page (Participant View) [1 Mark]

- **Info**: Name, Category, Description, Contact Email
- **Events**: Upcoming | Past

10 Organizer Features & Navigation [18 Marks]

10.1 Navigation Menu [1 Mark]

Navbar provides: **Dashboard**, **Create Event**, **Profile**, **Logout**, **Ongoing Events**.

10.2 Organizer Dashboard [3 Marks]

- **Events Carousel**: Displays all events created by the organizer as cards, showing Name, Type, Status (Draft/Published/Ongoing/Closed); allows the organizer to view and manage each event via a link to the detail page.
- **Event Analytics**: registrations/sales/revenue/attendance stats of all the completed events;

10.3 Event Detail Page (Organizer View) [4 Marks]

- **Overview**: Name, Type, Status, Dates, Eligibility, Pricing
- **Analytics**: Registrations/Sales, Attendance, Team completion, Revenue
- **Participants**: List (Name, Email, Reg Date, Payment, Team, Attendance); Search/Filter; Export CSV

10.4 Event Creation & Editing [4 Marks]

- **Flow**: Create (Draft) → Define Required Fields (Section 8) → Publish
- **Editing Rules & Actions**: Draft (free edits, can be published); Published (description update, extend deadline, increase limit, close registrations); Ongoing/Completed (no edits except status change, can be marked completed or closed).
- **Form Builder**: Provide organizers with an option to create custom registration forms for events. Support for various field types (text, dropdown, checkbox, file upload, etc.), marking fields as required/flexible, and reordering fields. Forms are locked after the first registration is received.

10.5 Organizer Profile Page [4 Marks]

- **Editable**: Name, Category, Description, Contact Email/Number (Login email non-editable)
- **Discord Webhook**: Auto-post new events to Discord

11 Admin Features & Navigation [6 Marks]

11.1 Navigation Menu [1 Mark]

Navbar provides: **Dashboard**, **Manage Clubs/Organizers**, **Password Reset Requests**, **Logout**.

11.2 Club/Organizer Management [5 Marks]

- **Add New Club/Organizer**: Admin can create new club/organizer accounts; system auto-generates login email and password; admin receives and shares credentials with the club/organizer; new accounts can immediately log in.
- **Remove Club/Organizer**: Admin can view list of all clubs/organizers; remove or disable accounts (removed clubs cannot log in); option to archive or permanently delete.

12 Deployment [5 Marks]

12.1 Hosting Requirements

- **Frontend:** Deploy to static hosting (e.g., Vercel/Netlify); provide production URL
- **Backend:** Deploy to managed Node hosting (e.g., Render/Railway/Fly/Heroku); provide base API URL
- **Database:** Use MongoDB Atlas; connection via environment variable

12.2 Links for Evaluation

Include root-level `deployment.txt` with: Frontend URL

Part - 2

13 Advanced Features [30 Marks]

Instructions

The following features are organized into **three tiers**. You must implement features from each tier as specified below to achieve exactly **30 marks**. Clearly mention in your `README.md` which features you have implemented from each tier along with justification of your choices.

Implementation Requirements:

- **Tier A**: Choose **2 features** (8 marks)
- **Tier B**: Choose **2 features** (6 marks)
- **Tier C**: Choose **1 feature** (2 marks each)

Total: 16 + 12 + 2 = 30 Marks

13.1 Tier A: Core Advanced Features [Choose 2 — 8 Marks]

Select **any 2** features from this tier. These features demonstrate advanced workflow management and complex business logic.

1. **Hackathon Team Registration [8 Marks]**

Support for team-based event registration where a team leader creates a team, sets the team size, and invites members via a unique code or link. Registration is marked complete only when all invited members accept and the team is fully formed. Include team management dashboard, invite tracking, and automatic ticket generation for all team members upon completion.

2. **Merchandise Payment Approval Workflow [8 Marks]**
   Implement a payment verification system for merchandise purchases. Users must upload a payment proof (image) after placing an order, upon which the order enters a *Pending Approval* state. Organizers can view a separate tab with all orders showing uploaded payment proofs, current status (Pending/Approved/Rejected), and actions to approve or reject payments. On approval, the order is marked *Successful*, stock is decremented, a ticket with QR is generated, and a confirmation email is sent. No QR is generated while the order is in pending or rejected state.

3. **QR Scanner & Attendance Tracking [8 Marks]**
   Implement a built-in QR code scanner for organizers to validate tickets and track attendance during events. Features include: scanning participant QR codes (using device camera or file upload), marking attendance with timestamp, rejecting duplicate scans, live attendance dashboard showing scanned vs. not-yet-scanned participants, export attendance reports as CSV, and manual override option for exceptional cases with audit logging.

13.2 Tier B: Real-time & Communication Features [Choose 2 — 6 Marks]

Select **any 2** feature from this tier. These features demonstrate real-time communication and collaborative functionalities.

1. **Real-Time Discussion Forum [6 Marks]**
   Implement a real-time discussion forum on the Event Details page where registered participants can post messages, ask questions, and interact. Organizers can moderate the forum (delete/pin messages), post announcements, and respond to queries. Include notification system for new messages, message threading support, and ability to react to messages.

2. **Organizer Password Reset Workflow [6 Marks]**
   Implement a complete password reset system for organizers. Organizers can request password reset from the Admin; Admin can view all password reset requests with details (club name, date, reason); Admin can approve or reject requests with comments; upon approval, system auto-generates new password which Admin receives and shares with the organizer; includes request status tracking (Pending/Approved/Rejected) and password reset history.

3. **Team Chat [6 Marks]**
   Implement a real-time chat system for hackathon teams (requires Tier A Feature 1). Team members can communicate within their team chat room after team formation. Features include real-time message delivery, message history, online status indicators, typing indicators, notification for new messages, and ability to share files/links within the team.

13.3 Tier C: Integration & Enhancement Features [Choose 1 — 2 Marks Each]

Select **1** feature from this tier. These features enhance user experience through integrations and additional functionalities.

1. **Anonymous Feedback System [2 Marks]**
   Enable participants to submit anonymous feedback for events they have attended. Feedback

includes a star rating (1–5) and text-based comments. Organizers can view aggregated ratings, filter feedback by rating, view average ratings

2. **Add to Calendar Integration [2 Marks]**
   Allow participants to export their registered events to external calendar applications by providing downloadable .ics files for universal calendar import, along with direct integration links for Google Calendar and Microsoft Outlook.

3. **Bot Protection [2 Marks]**
   Implement CAPTCHA verification on login and registration pages to prevent automated bot attacks. Use services like Google reCAPTCHA v2/v3 or hCaptcha .

14 Deliverables

Submission Guidelines

- Submit a **single ZIP file** with the following directory structure:

```
<roll_no>/
|-- backend/
|-- frontend/
|-- README.md
|-- deployment.txt
```

- Submission of a **corrupt ZIP file** will result in **0 marks**.

15 README Requirements

README Documentation

Your `README.md` must clearly document:

- **All libraries, frameworks, and modules** used in frontend and backend with proper justification for each choice
- **Advanced features implemented** from Tier A, Tier B, and Tier C with:
  - Justification for your feature selections
  - Explanation of design choices and implementation approach
  - Technical decisions made during development
- **Setup and installation instructions** for running the project locally

Important Notes:

- **No restrictions on UI libraries** – You may use any UI framework (Material-UI, Tailwind CSS, Bootstrap, etc.)
- Justify why you chose each library/framework and what problem it solves

16 Instructions

IMPORTANT: Academic Integrity Policy

AI Usage Policy:

- The use of AI tools (ChatGPT, GitHub Copilot, or any other AI-assisted coding tools) is **strictly NOT ALLOWED** for this assignment.
- Any instance of AI usage will result in a score of **0 (zero)** for the assignment.

Plagiarism Policy:

- All submissions will be run through **plagiarism detection software**.
- Any instance of copying code from other students, online sources, or previous submissions will result in a score of **0 (zero)** for the assignment.
- You must write your own original code. Collaboration is not permitted.

Evaluation Process:

- There will be **evaluations (evals)** conducted for this assignment.
- During the evaluation, you will be asked to explain the code you have written.
- **If you cannot explain what you wrote, it will be considered as plagiarism.**
- Inability to explain your code will result in a score of **0 (zero)** for the assignment.