UE21CS341A – Software Engineering

PROJECT REPORT

on

EVENTRA

Submitted by : Team # : T01

PES1UG21CS140	Bhuvan Vijay Kumar	PES1UG21CS169	Deeksha Krishnappa
PES1UG21CS149	Chandana S M	PES1UG21CS918	Dhiruram B

Class of Prof. Raghu B. A.

5th Sem. C Sec.

Table of Contents

Sl. No.	Topic	Page No.
1.	Project Proposal / Synopsis	3
2.	Software Requirements Specification [SRS] with RTM (Initial ver)	5
3.	Project Plan with Gantt Chart (Baseline)	21
4.	Architecture & Design Choices and Diagrams	23
5.	Development - Code Files [Git link]	25
6.	Test Plans	26
7.	Test Cases and Test Results Matrix – including Screenshots of Inputs and Resulting Outputs of Execution of Test Cases	28
8.	Final Gantt Chart (Baseline and Final Timelines)	39
9.	Conclusions	40
10.	Appendix A: Glossary of Abbreviations and Acronyms	41
11.	Appendix B: RTM (Final version)	43
12.	Appendix C: Technology stack and References [Books, Links to web pages/portals, tools]	44

1. **Project Proposal / Synopsis:**

Project Description:

The ***EVENTRA*** project aims to develop a **comprehensive event registration platform** tailored for **college clubs** and **organisations**. The platform will **simplify** the process of **managing and participating** in various events held by clubs within a university campus. It will provide a **user-friendly** interface for both event **organisers** and **participants**, enhancing the **overall event management experience**.

OBJECTIVES:

- Create a centralised platform for event registration and management that is accessible to all clubs within the university.
- Streamline the event registration process for club organisers, reducing administrative workload.
- Offer a user-friendly interface for students to discover and sign up for events easily.
- Provide essential features for event planning, such as event creation, scheduling, ticketing, etc.

SOFTWARE REQUIREMENTS SPECIFICATIONS:

➔ **Functional Requirements:**
- User Registration and Profiles
- Event Creation and Management
- Approval from faculty mentor of club
- Event Discovery
- Registration

➔ **Non - Functional Requirements:**
- Performance
- Usability
- Scalability

WORK BREAKDOWN STRUCTURE (WBS):

Project Phases:

1. **Initiation Phase:**
   - Define project scope, objectives, and requirements.
   - Identify key stakeholders and establish communication channels.

2. **Planning Phase:**
    - Create a detailed project plan, including tasks, timelines, and dependencies.
    - Allocate resources and assign roles and responsibilities.
    - Develop a project schedule.
    - Define technical architecture and choose development technologies.
3. **Design Phase:**
    - Develop wireframes and prototypes for the user interface.
    - Design the platform's database schema.
    - Create a visual design that aligns with the college's branding.
4. **Development Phase:**
    - Build the frontend of the platform using HTML5, CSS3, Streamlit..
    - Implement user registration, authentication, and profile management.
    - Create features for event creation, management, and registration.
    - Implement event discovery and search functionality.
    - Implement a feedback and review system.
    - Conduct continuous testing and bug fixing throughout development.
5. **Testing and Quality Assurance:**
    - Conduct unit testing for code components.
    - Perform integration testing to ensure all features work together seamlessly.
    - Gather feedback and address issues identified during testing.
6. **Post-Launch Maintenance and Updates:**
    - Monitor platform usage and gather user feedback for continuous improvement.
    - Address any bugs, issues, or feature requests promptly.
    - Regularly update the platform to add new features and enhancements.

2. Software Requirements Specification [SRS] with RTM (Initial ver)

SOFTWARE REQUIREMENTS SPECIFICATION

EVENTRA - College Events Management Platform

1. Introduction

1.1 Purpose

The purpose of EVENTRA - a college event management platform, is to facilitate the planning, organisation, promotion, and execution of events within a college or university campus. Such a platform serves several important functions to enhance the overall college experience and campus community.

1.2 Intended Audience

The project documentation is intended for the various stakeholders such as:

- **Developers**: This group needs detailed technical information to understand the software's architecture, features, and functionality.
- **Project Managers**: They require a comprehensive overview of the project's scope, objectives, and requirements to effectively plan and oversee the development process.
- **Marketing staff**: Marketing staff should understand the software's features and benefits to develop marketing strategies and materials that effectively promote EVENTRA.
- **Users**: Users, including college students and staff, need to grasp the system's functionalities and how it will improve their event-related experiences on campus
- **Testers**: Testers must have a clear understanding of the software's requirements to create test cases and perform comprehensive testing.
- **Documentation writers**: Writers need to know the software's functionality and features in-depth to produce user manuals, help guides, and other documentation.

1.3 Product Scope

The College Event Management System encompasses a comprehensive set of features and functionalities to meet the specific needs of colleges and universities in planning, organising, and managing events. The product scope includes:

1. **Efficient Event Planning**
2. **Event Creation and Scheduling**
3. **Simplified Registration**
4. **Communication and Promotion**
5. **Resource Management**
6. **Ticketing and Payment Processing**
7. **Data Analytics and Reporting**
8. **User Roles and Permissions**
9. **Mobile Accessibility**
10. **Integration and Compatibility**

1.4 References

There are no specific external documents or web references referenced in this SRS. The document itself serves as a standalone reference for the specified software requirements of the sports app. However, any relevant internal documents or guidelines from the development team or organisation should be considered during the development process.

2. Overall Description

2.1 Product Perspective

The product specified is "EVENTRA," a college event management platform. Colleges and universities require a tailored event management platform due to the dynamic nature of events they host, spanning academic conferences, sports, cultural festivals, and workshops. Efficient event management is vital for student and staff engagement and a vibrant campus community.

EVENTRA is a new, self-contained product.It is not a follow-on member of a product family, nor is it intended as a replacement for existing systems. Instead, it is developed to address the specific challenges and requirements associated with event management in an academic setting. This platform serves as a comprehensive solution to create a more engaging and inclusive campus community.

2.2 Product Functions

- **User Registration and Profiles**
- **Event Registration and Management**
- **Event Discovery**
- **Participant Registration and Ticketing**
- **Event Approvals**
- **Feedback and Reviews**

2.3 User Classes and Characteristics

- **Club Heads:**
  - **Frequency of Use:** Frequent users, as they are responsible for planning and managing events.
  - **Subset of Functions:** They require access to the full suite of event planning and management features.
  - **Privilege Levels:** Typically have administrative privileges.
- **Participants:**
  - **Frequency of Use:** Occasional users, as they interact with the system primarily to register for events.
  - **Subset of Functions:** Primarily involved in event registration, payment, and access to event-specific information
  - **Privilege Levels:** Limited access to event-related features.
- **Club Mentors:**
  - **Frequency of Use:** Occasional users, as they interact with the system primarily to approve and overlook events.
  - **Subset of Functions:** Primarily involved in event approvals, feedback, and access to event-specific information
  - **Privilege Levels:** Administrative privileges

2.4 Operating Environment

OE1 - Eventra will operate on Windows 10 and above on Google Chrome version 117.0.5938.149

OE2 - Eventra will operate on Ubuntu 23.04 and above on Firefox version 118.0.1.

OE3 - Eventra will operate on macOS 11 (BigSur) and above on Safari version 17.0

2.5 Design and Implementation Constraints

Interoperability: It should be capable of integrating with existing college systems, such as student information systems or academic calendars. Ensuring compatibility and data exchange with these systems might pose technical challenges.

Hardware Limitations: The system should consider potential hardware limitations, such as server capacity, network bandwidth, and memory requirements, to ensure optimal performance during peak event periods.

User Authentication and Authorization: Strict user authentication and authorization mechanisms are required to ensure that only authorised personnel have access to specific features and data, especially for sensitive information.

Scalability: The system needs to be designed to accommodate a potentially large number of events, users, and data over time. Scalability constraints may arise when the system experiences rapid growth in terms of usage.

2.6 Assumptions and Dependencies

Assumptions:

1. Third-Party Services: The project assumes the availability and reliability of third-party services such as email service providers for notifications regarding latest events in college and payment gateways for financial transactions.
2. Stability of APIs: Assumption that the APIs used for integration with external systems (e.g., student databases, calendar applications) will remain stable.
3. Data Accuracy: Assumption that event data provided by organisers is accurate. Inaccurate data could impact event planning.
4. User Connectivity: The project assumes users will have internet access and suitable devices. Network issues for users could affect system usability.

Dependencies:

1. Data Input: Timely and accurate data input by organisers, administrators, and users is a project dependency.
2. Third-Party Services: Dependency on services like email and payment gateways for various functionalities.
3. API Integrations: Integration depends on external API availability and reliability.
4. Server Infrastructure: The project depends on procuring and setting up server infrastructure, possibly involving coordination with hosting providers.

3. External Interface Requirements

3.1 User Interfaces

An overview of the logical characteristics for some common user interfaces in such a platform:

1. **User Registration and Login:** A typical login screen with fields for credentials and buttons for registration and login.
2. **Dashboard:** A dashboard providing an overview of upcoming events, user information, and options to navigate to various parts of the platform.
3. **Event Creation and Management:** A form for creating and managing events, including options to edit, delete, or publish.
4. **Event Listings:** A page displaying event listings in a grid or list format, allowing users to browse and access more event details.
5. **Event Registration:** A page where users can register for events, choose the number of tickets, and complete the registration process.
6. **Event Approvals:** A page which displays the approval status from the Faculty Advisor and the Dean.
7. **Search and Filters:** A search and filtering interface to help users find specific events.

3.2 Software Interfaces

3.2.1 System Connections:

- **Database**: Utilises MySQL as the relational database management system for storing and retrieving event details, user profiles, registrations, and feedback.
- **Operating Systems**: The platform is designed to be cross-platform and is compatible with various operating systems, including Windows (Windows 7 and later), macOS, Linux distributions, Android, and iOS.
- **Third-Party Tools and Libraries**: The software integrates with third-party services for specific functionalities, such as email notifications, payment processing, and authentication services.

3.2.2 Data Flow:

3.2.2.1 Incoming Data and Messages:

- **User Input**: Users provide data input, including event details, registration information, feedback, and profile updates.
- **Third-Party Integrations**: Data flows from third-party services, such as payment gateways and email providers, for payment processing and communication.
- **Event Organisers**: Event organisers input event details and updates through administrative interfaces.
- **Approvals**: Club mentors and dean set the status of approvals.
- **User Registrations**: Users submit registration data for events, including attendee information and ticket purchases.

3.2.2.1 Outgoing Data and Messages:

- **Event Updates**: Messages about event details, registration confirmations, and event changes are sent to users and administrators. These updates keep users informed of event-related changes.
- **Payment Confirmations:** Messages about payment processing, transaction confirmations, and invoice receipts are sent to users and administrators.
- **Approvals**: Messages about approval status from club mentor and dean.
- **Feedback**: Messages about post-event feedback and surveys are sent to attendees and event organisers.

3.2.3 Services and Nature of Communication:

User Registration and Authentication Service:

a. Allows users to register, log in, and manage their profiles.

b. Communication includes user data, login credentials, and profile updates.

Event Management Service:

a. Enables event organisers to create, update, and manage events.

b. Communication includes event details, registrations, and updates.

Payment Processing Service:

a. Handles financial transactions and payment confirmations.

b. Communication includes payment details and transaction status.

Database Service (MySQL):
a. Used to store event data, user profiles, and transaction records.
b. Communication is bidirectional, involving CRUD (Create, Read, Update, Delete) operations over the SQL protocol.

3.3 Communications Interfaces

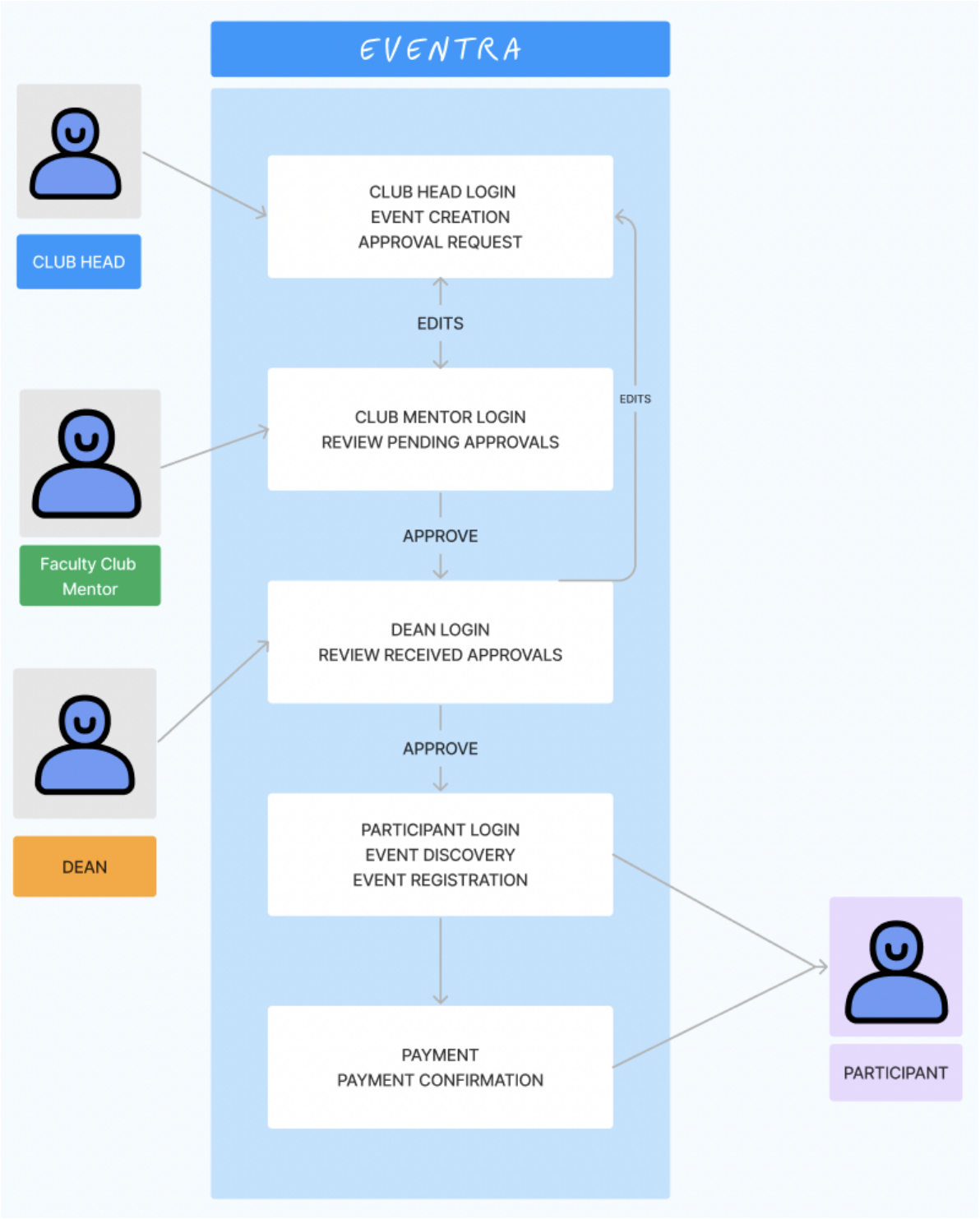
The EVENTRA college event management system relies on various communication interfaces to facilitate its functionality, including but not limited to:

1. **Web Browsing:** EVENTRA operates through standard web browsers (e.g., Chrome, Firefox, Safari) for user access. The system's user interface must be compatible with the latest versions of these browsers.
2. **HTTP/HTTPS:** Communication between the user's browser and the EVENTRA server is conducted over HTTP and secured via HTTPS to protect data transmission.
3. **Email:** The system uses email for various communication purposes, including user registration confirmation, event notifications, and password reset requests. Event-related notifications may contain event details, including name, date, time, and location.
4. **APIs:** EVENTRA may integrate with third-party services or external systems via Application Programming Interfaces (APIs). These APIs could be used for purposes such as event calendar synchronisation, user authentication, or payment processing.
5. **Database Connectivity:** The system communicates with the database server using standard database communication protocols, such as ODBC or JDBC, to perform data storage and retrieval operations.
6. **Internal Messaging:** EVENTRA may include an internal messaging system for communication between event organisers, faculty club representatives, and students. Messages should be formatted to include sender, receiver, content, and timestamp.
7. **Notification Services:** The system may use push notification services for real-time event updates. These services should follow appropriate messaging standards and ensure the security of notifications.

8. **Authentication Services:** To verify user identities, the system may communicate with the college's existing authentication service using standard protocols, such as OAuth 2.0 or SAML.
9. **User Forms:** The system may employ electronic forms for user registration and event creation, which should follow web form standards, including data validation and security measures.
10. **Web Services:** For data exchange with external systems, EVENTRA may use web services, with appropriate message formats (e.g., JSON or XML).
11. **Communication Security:** All communication should be secured with industry-standard encryption protocols (e.g., TLS) to protect data during transmission. Passwords and sensitive data should be hashed and salted for storage.
12. **Data Transfer Rates:** The system should be optimised for efficient data transfer, striving for minimal latency in data retrieval and transmission.
13. **Synchronisation Mechanisms:** In cases where data synchronisation is necessary (e.g., with the academic calendar), appropriate synchronisation mechanisms should be employed to ensure data consistency and accuracy.

4. Analysis Models

Use Case Diagram:



5. System Features

System Feature 1 : Club Registration

5.1.1 Description and Priority

Description: Allow students to login to their EVENTRA account.

Priority: High

5.1.2 Stimulus/Response Sequences

- User navigates to the registration page.
- Users enter their personal information.
- User submits the registration form.
- System validates the information and creates a new student account.
- System sends a confirmation email to the student.

5.1.3 Functional Requirements

REQ-1: The system must provide a user-friendly registration form that includes fields for the student's name, email, password, and other required information.

REQ-2: Upon submission, the system must validate the email format and ensure that the email is not already associated with an existing account.

REQ-3: The system must securely store student account information, including hashed passwords.

REQ-4: After successful registration, the system must send a confirmation email to the registered student's email address.

REQ-5: In case of a registration error (e.g., duplicate email or invalid password), the system must provide appropriate error messages to the user.

System Feature 2 : Event Creation

5.2.1 Description and Priority

Description: Allow event organisers to create new events.

Priority: High

5.2.2 Stimulus/Response Sequences:

- Event organiser logs in and selects the "Create Event" option.
- The Organiser provides event details such as name, date, location, and description.
- System validates and saves the event information.
- The Organizer receives a confirmation message.

5.2.3 Functional Requirements:

REQ-1: The system must offer an event creation form with fields for event name, date, location, description, and other relevant details.

REQ-2: The system must validate event details to ensure that required information is provided and that the date is in the future.

REQ-3: After successful event creation, the system must associate the event with the logged-in organiser.

REQ-4: The system must send a confirmation message to the event organiser upon successful event creation.

REQ-5: If there are issues with event creation (e.g., missing information or conflicting events), the system must provide appropriate error feedback.

System Feature 3 : Approval from faculty mentor of club

5.3.1 Description and Priority:

Description: Allow faculty member club representatives to approve or reject event creation requests from students.

Priority: High

5.3.2 Stimulus/Response Sequences:

- Faculty club representative logs in and navigates to the "Event Approval" section.
- Representative views a list of pending event creation requests.
- Representative selects a request and reviews event details.
- Representative approves or rejects the event creation request.
- System updates the event status and notifies the student organiser.

5.3.3 Functional Requirements:

REQ-1: The system must provide a dedicated "Event Approval" section for faculty club representatives with appropriate access rights.

REQ-2: Faculty club representatives should be able to view a list of pending event creation requests, including event details and organiser information.

REQ-3: Representatives should have the ability to approve or reject event creation requests.

REQ-4: Upon approval or rejection, the system should update the event status accordingly and notify the student organiser of the decision.
REQ-5: If a representative tries to approve or reject a request that is already processed or no longer exists, the system should provide an appropriate error message.

System Feature 4: Event Discovery

5.4.1 Description and Priority:

Description: Enable students to discover and explore events available in the system.
Priority: High

5.4.2 Stimulus/Response Sequences:

- Student logs in and navigates to the "Event Discovery" section.
- Students can search for events based on filters (e.g., date, category, location).
- Students view event details, including name, date, location, and description.
- Students can register for events directly from the discovery interface.

5.4.3 Functional Requirements:

REQ-1: The system must provide an intuitive and user-friendly "Event Discovery" section where students can explore and search for events.
REQ-2: Students should be able to search for events based on various filters, such as event date, category, location, and keywords.
REQ-3: Event details, including the event name, date, location, description, and organiser information, should be accessible to students.
REQ-4: Students should have the ability to register for events directly from the event discovery interface.
REQ-5: In case of event registration errors (e.g., event already full), the system should provide clear error messages to the user.

System Feature 5: Event Registration and Ticketing

5.5.1 Description and Priority:

Description: Allow students to register for events, generate tickets, and manage their event registrations.
Priority: High

5.5.2 Stimulus/Response Sequences:

- Students login and navigate to the "Event Registration" section.
- Students search for events they wish to attend.
- Student selects an event and registers.
- The system generates an electronic ticket for the registered event.
- Students can view, edit, or cancel their event registrations.

5.5.3 Functional Requirements:

REQ-1: The system must offer a user-friendly "Event Registration" section where students can search for and register for events.
REQ-2: Registered students should be provided with an electronic ticket for each event they register for.
REQ-3: The electronic ticket should contain essential event details, including the event name, date, time, location, and a QR code for event check-in.
REQ-4: Students should be able to view, edit, or cancel their event registrations.
REQ-5: When a student cancels an event registration, the system should release the event slot for other potential attendees.
REQ-6: The system should have a seamless event check-in process that allows organisers to scan electronic tickets at the event venue.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

Response Time: The system must respond to user interactions within 2 seconds, ensuring a smooth user experience.

Scalability: The system should be able to handle an increasing number of events and users without a significant decrease in performance. It should support at least 10,000 simultaneous users during peak usage periods.

Database Performance: Database queries and transactions should have a response time of no more than 1 second under standard load.

6.2 Safety Requirements

Data Security: The system must encrypt all sensitive user data (e.g., passwords) using industry-standard encryption methods to prevent unauthorised access.

User Privacy: EVENTRA should comply with all relevant data protection and privacy regulations to safeguard user privacy.

Event Information: Users' personal information, such as email addresses, should not be shared with event organisers without explicit user consent.

6.3 Security Requirements

Authentication: Users should be required to authenticate their identity through secure means, such as email verification, to access their accounts.

Authorization: The system should employ role-based access control, ensuring that users only have access to features and data relevant to their roles.

Secure Data Transmission: Data transmitted between the user and the system should be encrypted using HTTPS to prevent eavesdropping.

Security Auditing: The system should log and monitor security-related events and provide audit trails for user activities.

6.4 Software Quality Attributes

Usability: The system should have a user-friendly interface with clear navigation and well-organised information.

Maintainability: Code should be well-documented and follow coding best practices to facilitate future maintenance and updates.

Availability: The system should aim for 99.9% availability, with planned maintenance communicated to users in advance.
Interoperability: The system should support common web browsers (e.g., Chrome, Firefox, Safari) and mobile platforms (iOS and Android).
Robustness: The system should handle unexpected errors gracefully and provide informative error messages to users.

6.5 Business Rules

Event Approval: Only authorised administrators can approve event creations before they are visible to students.
User Account Roles: Event organisers have different privileges compared to regular students. For example, they can create events, while students can only register for events.
These nonfunctional requirements and domain requirements are essential for ensuring the performance, safety, security, and quality of Eventra, while also aligning with University and regulatory standards.

6.6 Domain Requirements:

Academic Calendar Integration: EVENTRA should be synchronised with the college's academic calendar to avoid scheduling conflicts.
Event Categories: The system should have categories for different types of events (e.g., academic, social, sports).
User Authentication: Use the college's existing authentication system for student verification when registering for EVENTRA.

---

7. Other Requirements:

7.1 Database Requirements:

- **Database Management System:** The system should use a relational database management system (e.g., MySQL, PostgreSQL) for data storage.
- **Data Backup:** The system should automatically back up the database daily, and backups should be stored securely for at least 30 days.
- **Data Retention:** Event data should be retained for at least 2 years, while user data should be retained as per relevant privacy regulations.

7.2 User Support and Training:

- **User Support Channels:** Define the channels (e.g., email, chat, support tickets) through which users can seek assistance.
- **Training Materials:** If applicable, specify the availability of training materials or resources for users.

7.3 Reuse Objectives:

- **Code Reusability:** Code components should be organised and documented in a way that encourages code reusability in future projects or system enhancements.
- **Third-Party Libraries:** Utilise open-source libraries and frameworks that have appropriate licensing for easy integration and future development.

7.4 Documentation Requirements:

- **User Manual:** Provide a comprehensive user manual that explains how to use the system effectively, including user guides for students, event organisers, and administrators.
- **Code Documentation:** All code should be documented, with clear comments and explanations to aid in maintenance and future development.

3. Project Plan with Gantt Chart (Baseline):

Life-cycle followed

The model used in our project ***Eventra - College Event Management Platform*** *is* **Agile Scrum Software Development Lifecycle model.**

Tools Used for this Project

Planning tool - Jira
Design tool - Figma
Version Control - Git via Github
Development tools:

- Visual Studio Code
- Frameworks and Libraries like Streamlit, HTML, CSS and MySQL.

Bug tracking - Jira
Testing tool - Selenium

Deliverables classified as reuse/build components

Reusable / Build Components:

1. **User Authentication Module**
2. **Event listing and Registration Module**
3. **Payment Gateway Integration**
4. **Approval Module**
5. **User Profile Management**

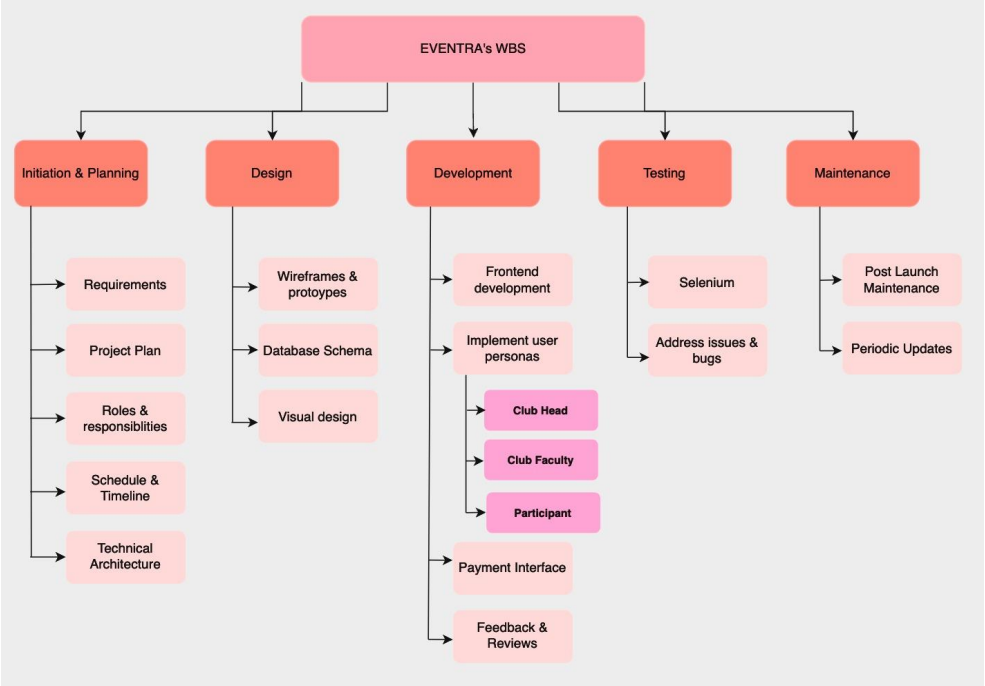
These modules and deliverables are classified as reusable/build components as they are generic and can be used in other similar projects or applications as well. They provide basic core functionalities like user authentication, user profile management, payment interface, etc which are essential in various projects.

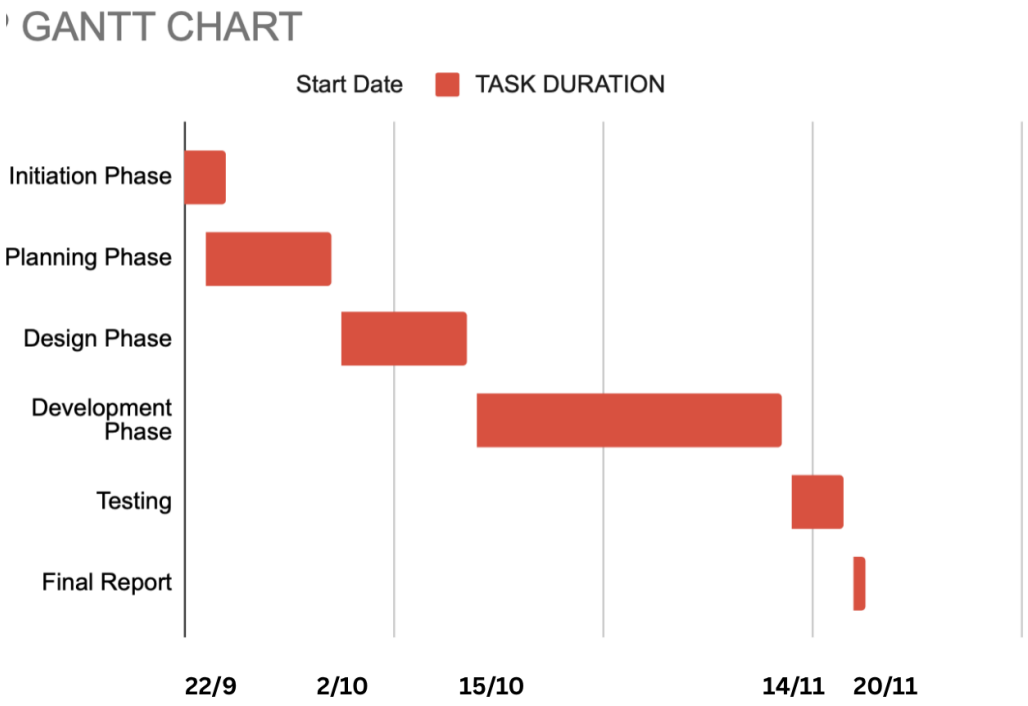
Project Specific Components:

1. **College Event Management UI/UX Design**
2. **Custom Event Registration Forms**
3. **Custom Participant Registration Forms**
4. **Event Database and Database Schema**

These components or deliverables are classified as Project Specific components because they particularly serve the purpose of this project and cannot be used in other projects. They provide specific frameworks like UI/UX designs, Custom registration forms and Databases that are tailor-made to meet the requirements of this specific project.

Work Breakdown Structure:





Effort Estimation (in person-months)

Initiation Phase: 0.184 person-months equivalent to 4 full working days per person.

Planning Phase: 0.323 person-months equivalent to 7 full working days per person.

Design Phase: 0.55 person-months equivalent to 12 full working days per person.

Development Phase: 0.923 person-months equivalent to 20 full working days per person.

Testing and Quality Assurance: 0.46 person-months equivalent to 10 full working days per person.

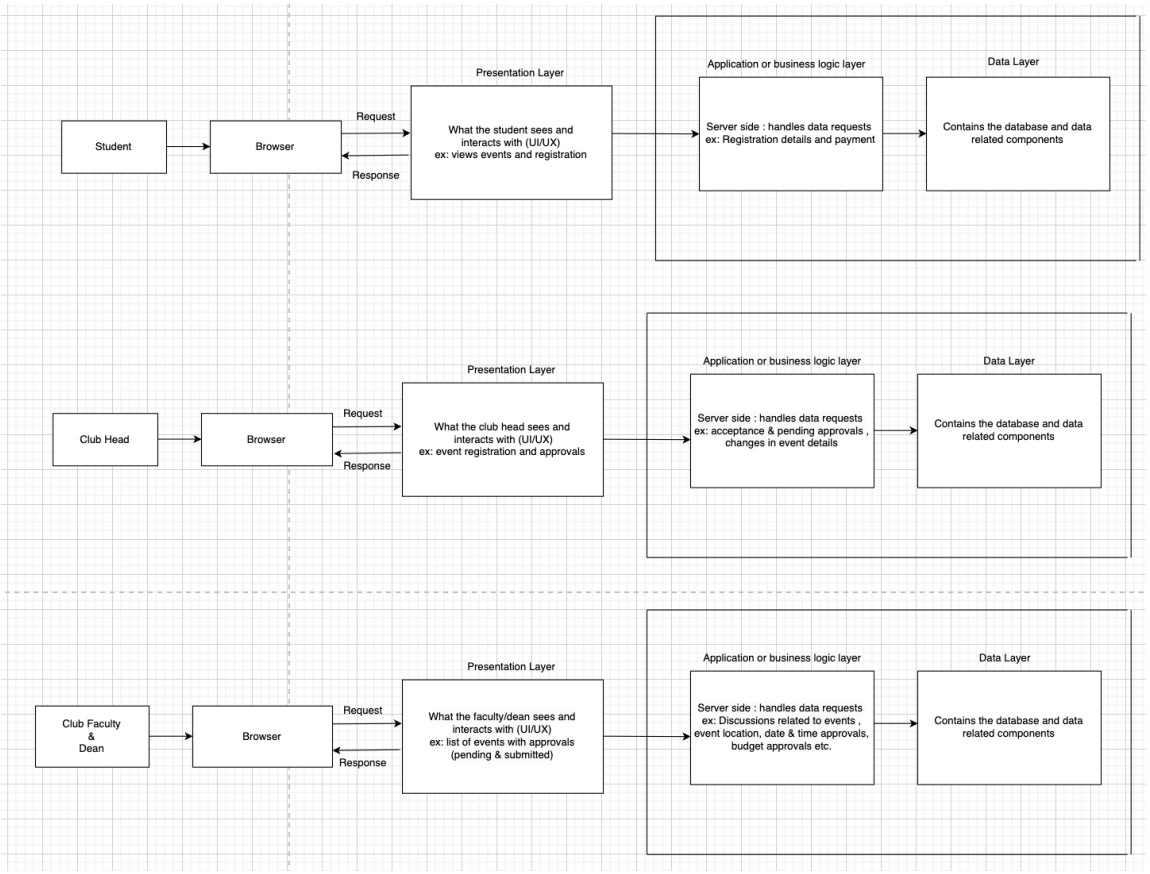
Post-Launch Maintenance and Updates: 0.0923 person-months equivalent to 2 full working days per person.

These estimates are based on the assumption that 1 person-month is roughly equivalent to 21.66 working days.

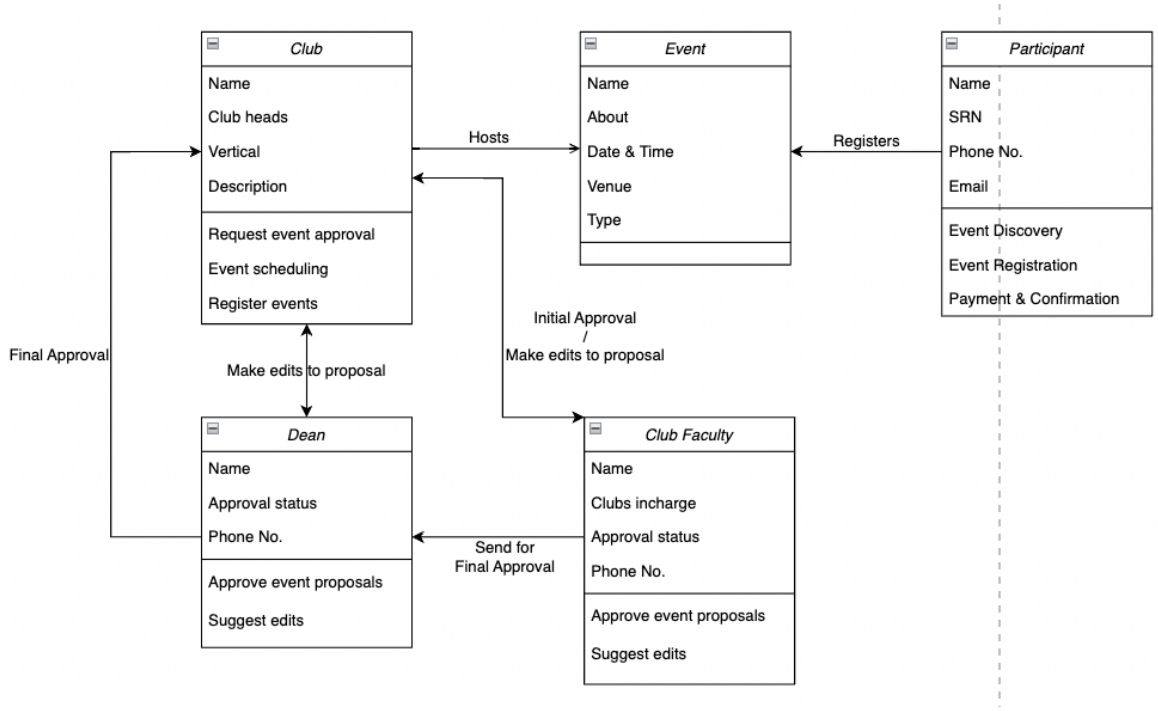
---

4. Architecture & Design Choices and Diagrams:

Architectural Design [Eventra]:

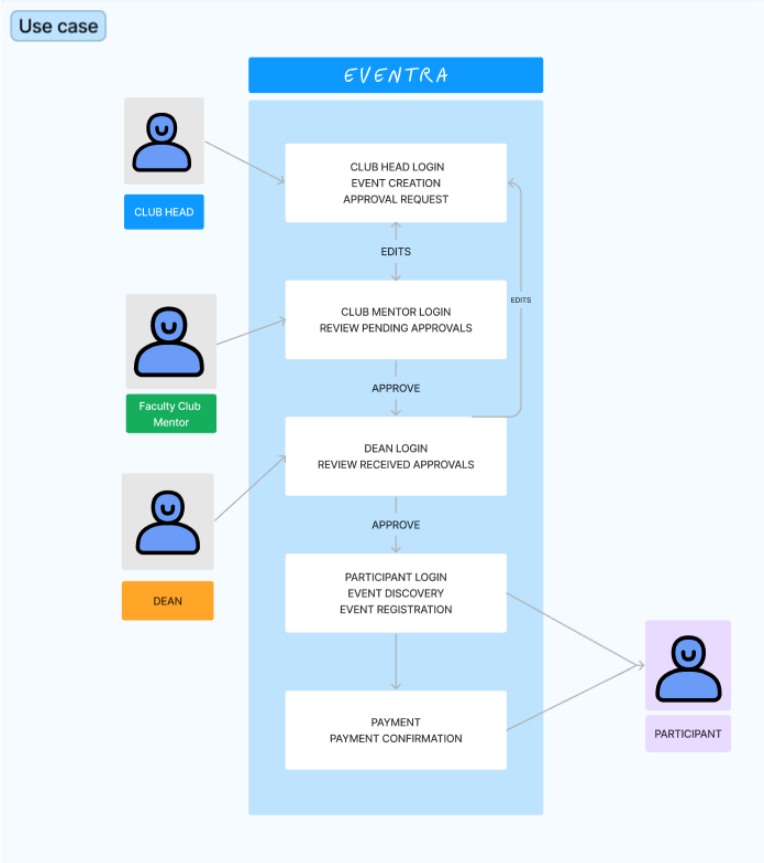


Class Diagram:



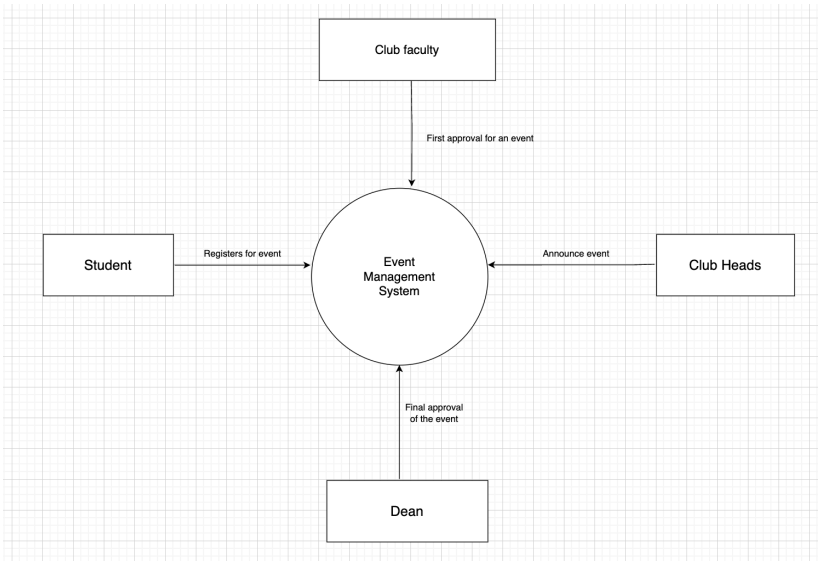
UML

Use Case Diagram:

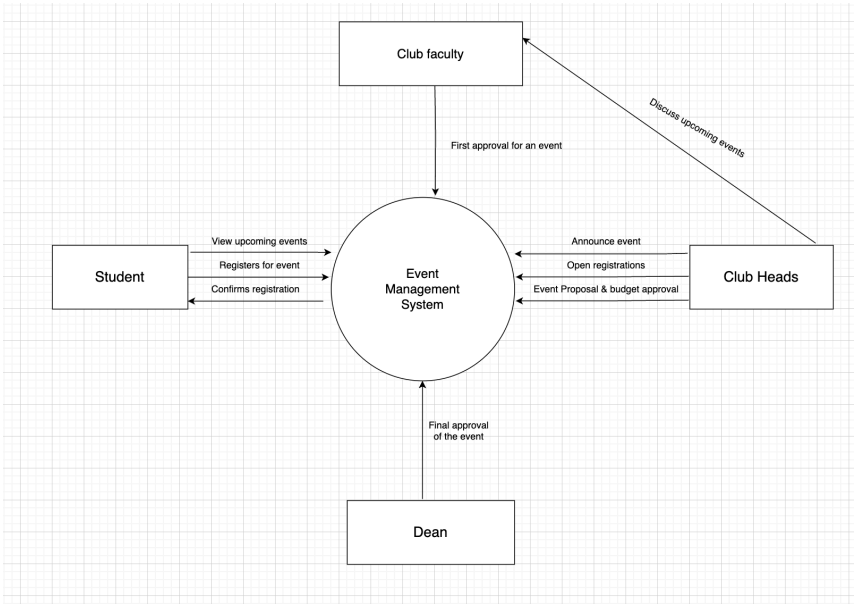


Design Diagrams:

Level 0 DFD:



Level 1 DFD:



5. Development - Code Files [Git link]:

LINK: https://github.com/dhiru037/Eventra/tree/main

6. Test Plans :

The purpose of this test plan is to outline the testing strategy and approach for EVENTRA. This document will identify the test cases to be executed, the resources required, and the schedule for testing activities.

Our main objectives of testing is to:

- Verify that the system meets all functional requirements
- Identify and document any defects in the system
- Ensure that the system is user-friendly and easy to use

The following are the Test Cases we have considered:

FEATURE 1: <u>User Login and profiles</u>

1. **UT:**
    - Verify that users can successfully login with valid credentials.
    - Invalid username
    - Verify unsuccessful login with invalid username
    - Invalid password
2. **IT:**
    - Valid login and authorization.
    - Invalid login and authorization. Verify unsuccessful login and authorization for non-admin users.

FEATURE 2: <u>Event Creation</u>

1. **UT:**
    - Verify that club head can create an event
    - Verify if the mandatory fields are filled which are required for successful event creation
    - Verify if when the mandatory fields are unfilled there is an error with event creation
    - No two events can be conducted on the same day and same venue
2. **IT:**
    - Confirm that the event creation module integrates seamlessly with the database to store event details.

FEATURE 3: **Approval from faculty mentor of club**

1. **UT:**
   - Confirm that the faculty approval status is initially set to pending.
   - Validate that the approval status is updated correctly when the faculty mentor approves or rejects the event.
2. **IT:**
   - Ensure that the faculty approval process integrates with the event creation module.
3. **ST:**
   - Verify that only approved events are displayed in the public event listings.
   - Check that pending events are appropriately communicated to the club heads.

FEATURE 4: Event Discover

1. **UT:**
   - Ensure that the event discovery module integrates well with the event creation and approval features and displays for participants
2. **ST:**
   - Event is created and sent for dean's approval

FEATURE 5: Event Registration and Ticketing

1. **UT:**
   - Verify that users can register for an event successfully.
2. **IT:**
   - Confirm that the registration module interacts seamlessly with the user authentication system.

7. Test Cases and Test results Matrix- including screenshots of Inputs and resulting outputs of execution of test cases:

FEATURE 1: User Login and profiles

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT-01	User (participant, club head , dean) authentication module	Verify that users can successfully login with valid credentials.	User (participant, club head , dean) account exists with valid username and password	1: Navigate to login page 2: Enter valid Username and Password 3: Click the login button	Valid username and password.	Login successful with "welcome message" displayed	Login should be successful with "welcome Participant/ Club Head/ Dean/ Club Mentor"	Pass
UT-02	User authentication module	Invalid password (Club head/ Dean/ Participant)	Verify unsuccessful login with invalid password. User account with a valid username exists but with the wrong password .	1. Navigate to the login page. 2. Enter valid username and invalid password. 3. Click the login button.	Valid username and invalid password	Error message indicating invalid password	Error message indicating invalid password displayed	pass
IT-01	User (Club head/ Dean/ Participant) authentication module and authorization module	Valid login and authorization Verify successful login and authorization for user	User (participant, club head , dean) account exists with valid username and password	1. Navigate to the login page. 2. Enter valid user username and password. 3. Click the login button.	Valid user username and password	Successful login and redirect to user (Club head/ Dean/ Participant) dashboard	Successful login and redirect to user (Club head/ Dean/ Participant) dashboard	Pass

FEATURE 2: Event Creation

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT-01	Creation of an event	Verify that club head can create an event	Club Head is logged and enters valid event details to create an event	1.Navigate to the drop down 2.Select event creation 3.Enter event details.	Valid event details	Event successfully created.	Event successfully created.	pass
UT-02	Validate that mandatory fields	Verify if the mandatory fields are filled which are required for successful event creation	Club Head is logged in and enters valid event details in all the mandatory fields.	1.Navigate to the drop down 2.Select event creation 3.Enter event details in the mandatory fields.	Valid event details.	Event successfully created.	Event successfully created.	pass
UT-03	Validate that mandatory fields	Verify if when the mandatory fields are unfilled there is an error with event creation	Club Head is logged in and enters invalid event details in the mandatory fields.	1.Navigate to the drop down 2.Select event creation 3.Enter event details in few of the mandatory fields and leave a few empty	Leave a few fields blank	Error- enter all the details	Error- enter all the details	pass
IT-01	Integration of event creation with database.	Confirm that the event creation module integrates seamlessly with the database to store event details.	Club Head is logged in and the server is connected to the database.	1. Open event creation form 2. Enter event details and submit. 3. Verify database entry. 4. Compare the details.	Valid event details	Event creation module should seamlessly integrate with the database and the entered details should be reflected in the database.	Event details are reflected int the backend database.	pass
UT-04	Collision of events	No two events can be conducted on the same day and same venue	Club Head tries to create an event on the same day as another event in the same venue	1. Open event creation form. 2. Enter same day and venue as another event	Duplicate day and venue	This Slot is occupied by another event	This Slot is occupied by another event	pass

FEATURE 3: **Approval from faculty mentor of club**

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT-01	Status of approval	Confirm that the faculty approval status is initially set to pending.	While event creation the event status is pending by default	1.Create an event. 2. Check the status of the event	Enter details and create event	Pending under status of event approval column	Pending under status of event approval column	pass
UT-02	Status of approval	Validate that the approval status is updated correctly when the faculty mentor approves or rejects the event.	Event is created and the faculty gets the proposal for approval	1. Create an event 2. Faculty updates the status by checking the event proposal.	Faculty approves the event and changes the status	Approved under status of event approval	Approved under status of event approval	pass
IT-01	Status of approval	Ensure that the faculty approval process integrates with the event creation module.	Event is created and the faculty approves the event.	1. Create an event 2. Faculty approves the event.	Faculty approves the event and changes reflect on the backend and frontend	Approved status updated on front end and backend	Approved status updated on front end and backend	pass
ST-01	Event list on participants dashboard	Verify that only approved events are displayed in the public event listings.	Event is created and approved by faculty	1. Login as a participant. 2. Navigate to dashboard 3. View the events list.	Login as participants and view events	Only Approved events are displayed on the dashboard	Only Approved events are displayed on the dashboard	pass
ST-02	Unapproved events to be communicated to the club heads.	Check that pending events are appropriately communicated to the club heads.	Event is created and dean send with updates	1. Login as club head 2. Check status of approval and 3. View message with changes to be made sent by dean	Dean sends updates for a particular event	Approval pending with message displayed on the event status columns	Approval pending with message displayed on the event status columns	pass.

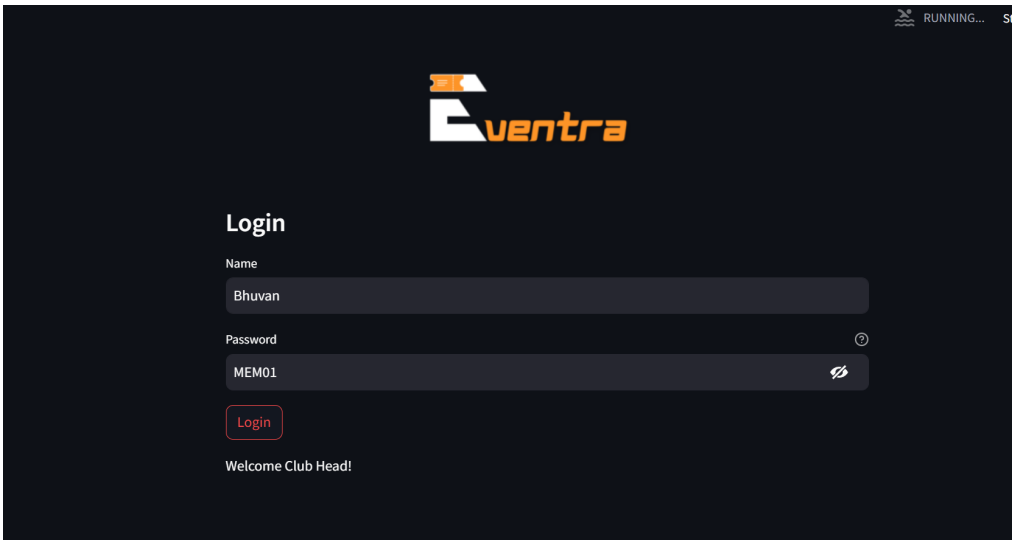
Feature 4: Event Discover

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT-01	Event view.	Ensure that the event discovery module integrates well with the event creation and approval features and displays for participants	Event is created and approved	1. Login as a participant. 2. Navigate to dashboard 3. View the events list.	View events on dashboard	Display the approved events	Displays the approved events	Pass
ST-01	Event view and event discovery filter for dean	Confirm that events are displayed accurately on the platform for the dean to approve.	Event is created and sent for dean's approval	1. Login as Dean 2. Navigate to pending events approval	View events on deans dashboard	Display the events using the filter(for clubs or days or venues)	Displays the events based on the filters	pass

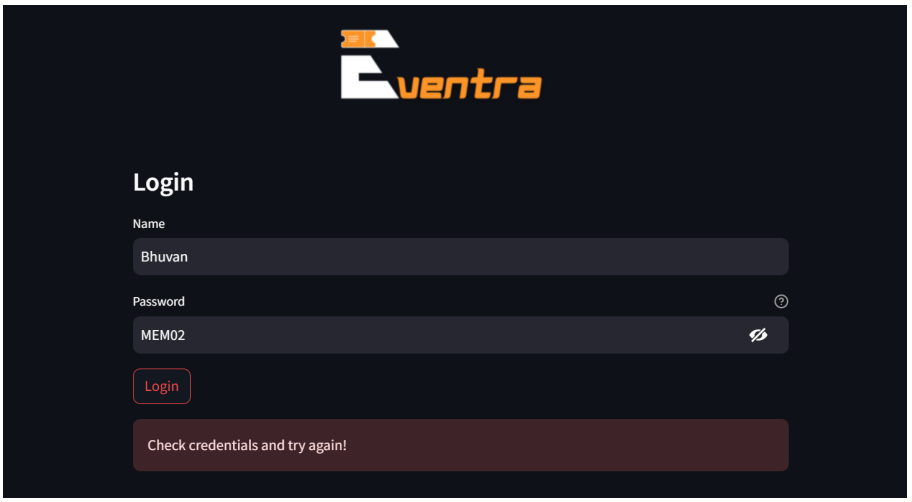
Feature 5: Event Registration and Ticketing

Test Case ID	Name of Module	Test Case Description	Pre-conditions	Test Steps	Test Data	Expected Results	Actual Results	Test Result
UT-01	Event registration	Verify that users can register for an event successfully.	Event is displayed on participant dashboard to register	1. Login as a participant. 2. View events 3. Select the event to register and fill the details.	Login and enter the details to register for event.	Successfully registered for event	Successfully registered for event	pass
IT-01	Participant authentication and database check	Confirm that the registration module interacts seamlessly with the user authentication system.	Events are created and participants register for them.	1. Login as club head 2. Navigate to registered details for events. 3. Verify with the backend database.	Participant registers for event and club heads checks the database and his dashboard	View participant details on club head dashboard	Views participants details in dashboard and backend	pass

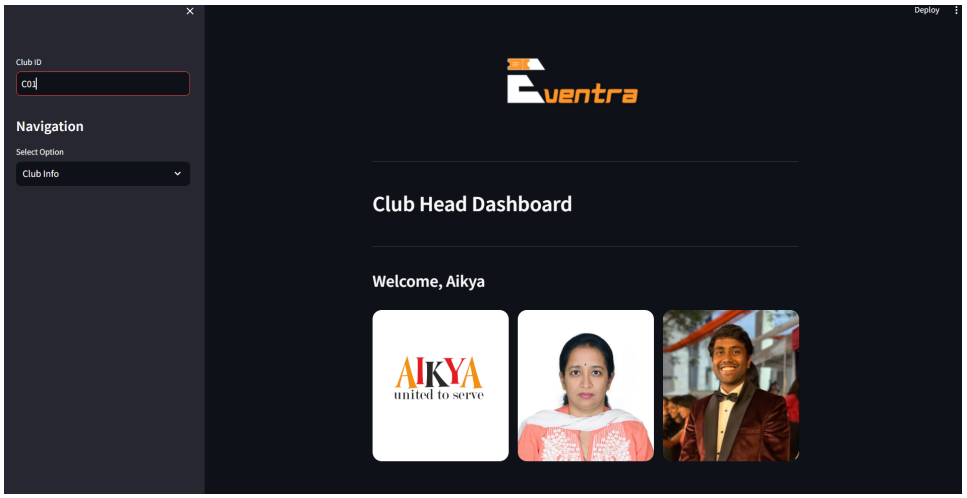
UT-01: Verify that users **(participant, club head , dean)** can successfully login with valid credentials.



UT-02: Verify unsuccessful login with invalid username.(Invalid password)

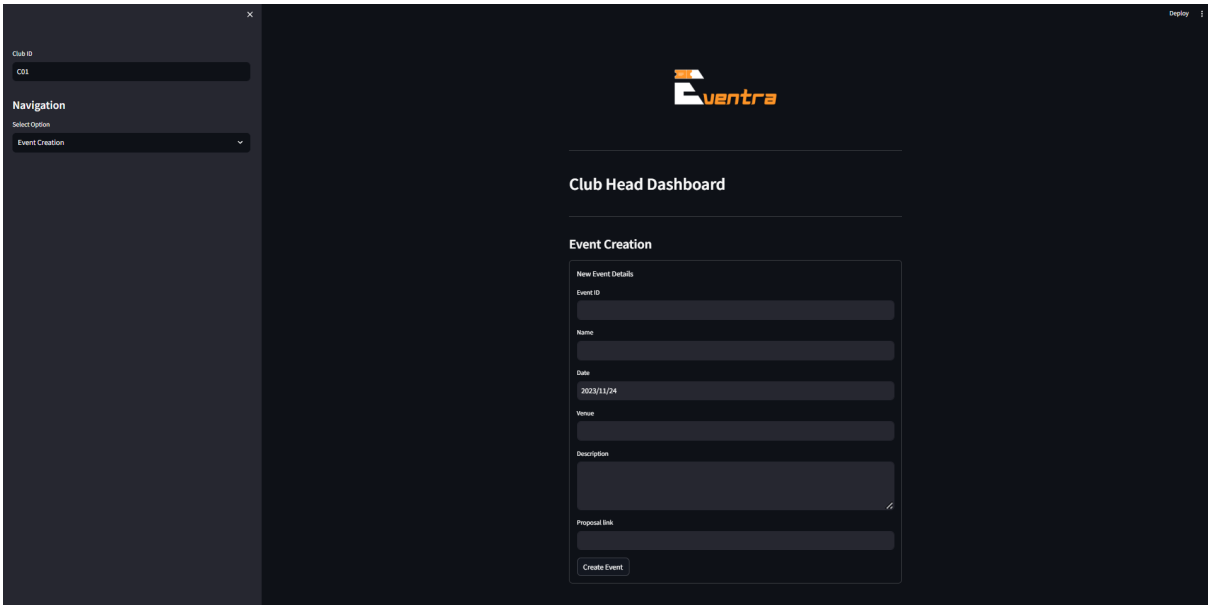


IT-01: Valid login and authorization.

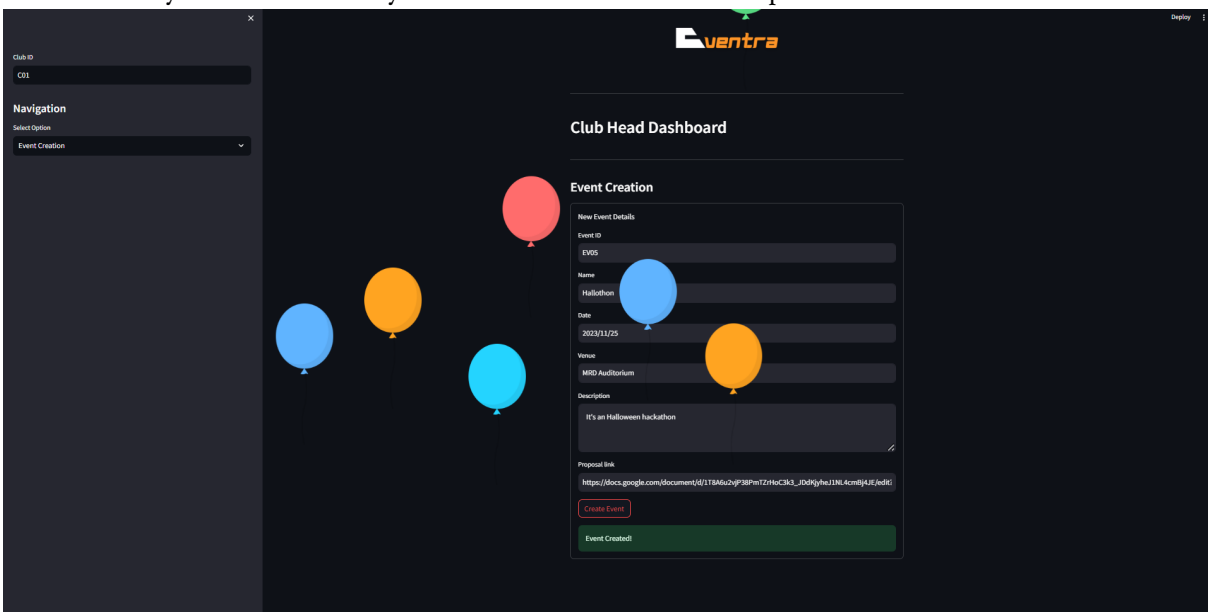


FEATURE 2: Event Creation

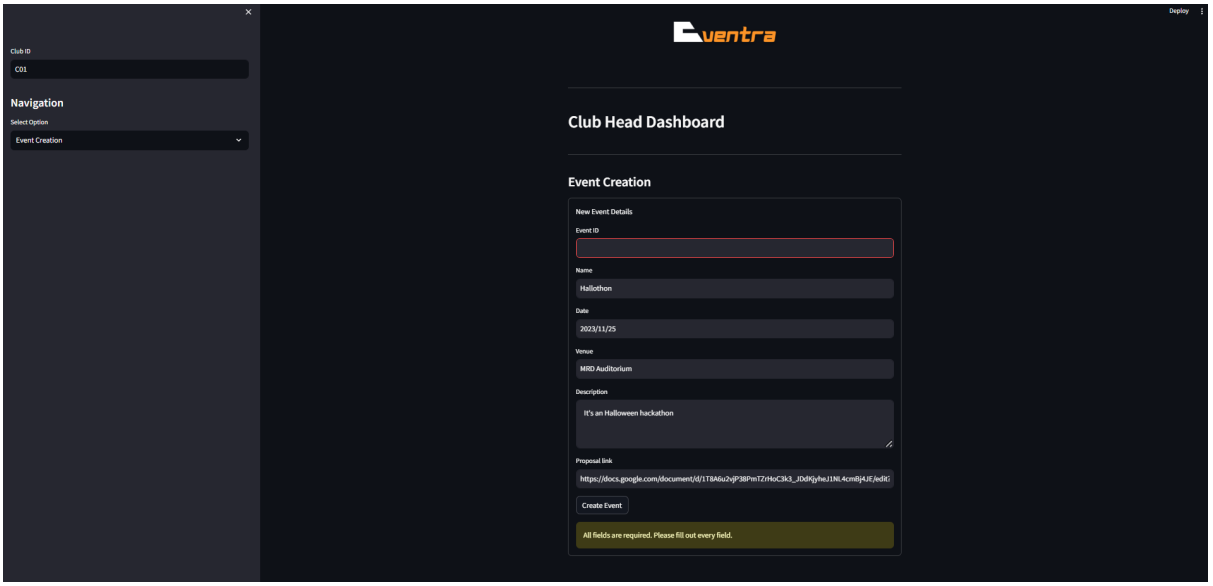
UT-01: Verify that club head can create an event



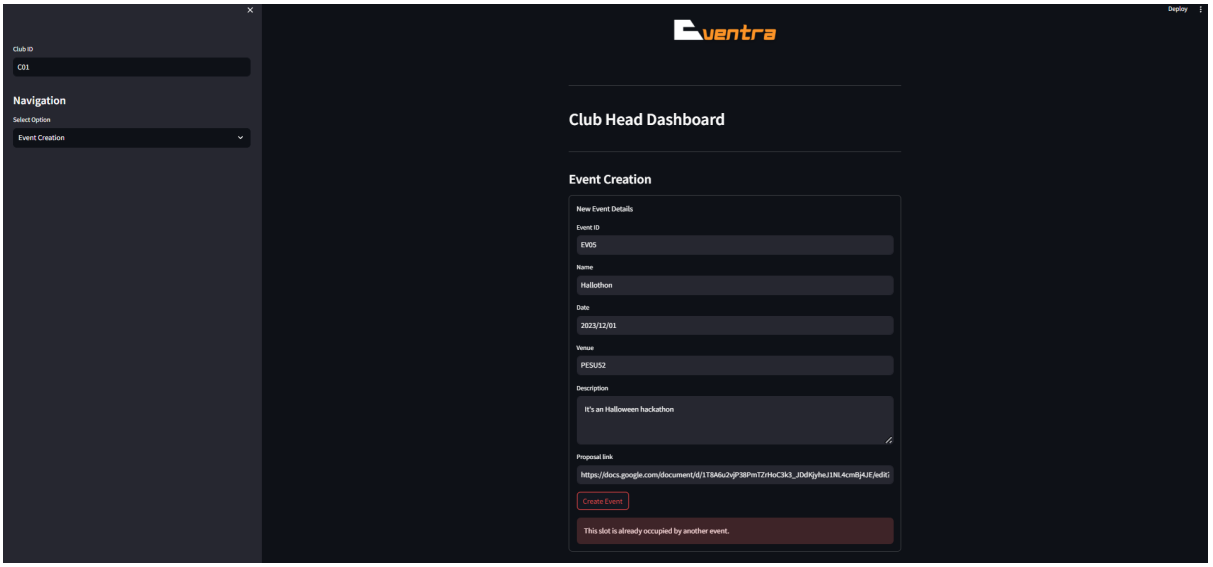
UT-02: Verify if the mandatory fields are filled which are required for successful event creation



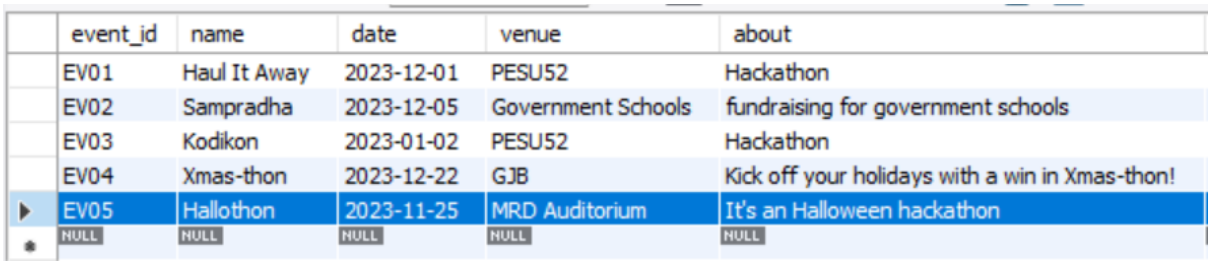
UT-03: Verify if when the mandatory fields are unfilled there is an error with event creation



UT-04: No two events can be conducted on the same day and same venue



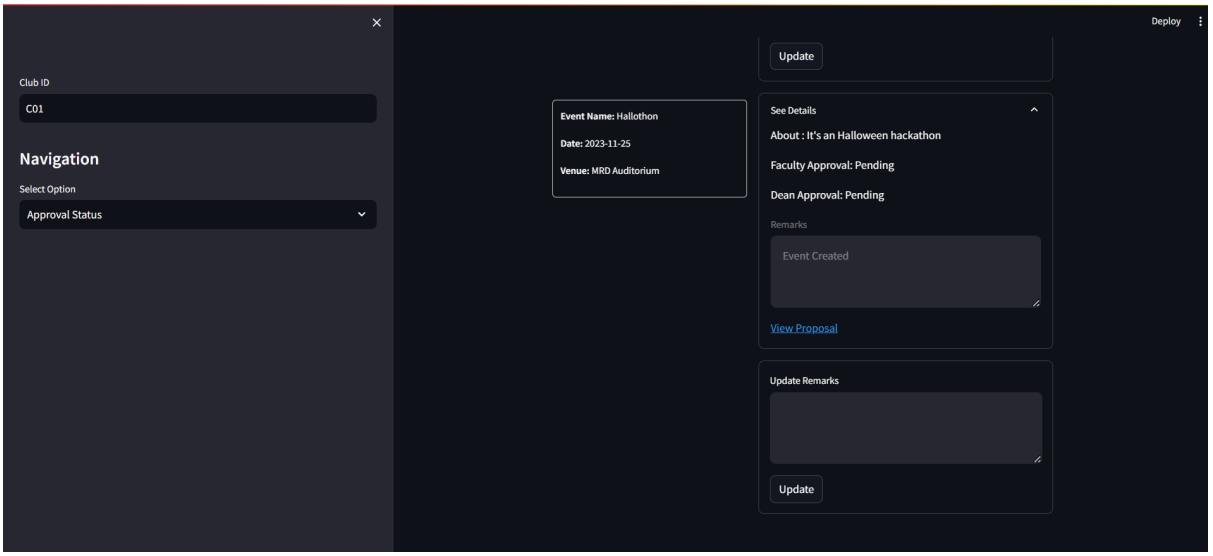
IT-01: Confirm that the event creation module integrates seamlessly with the database to store event details.



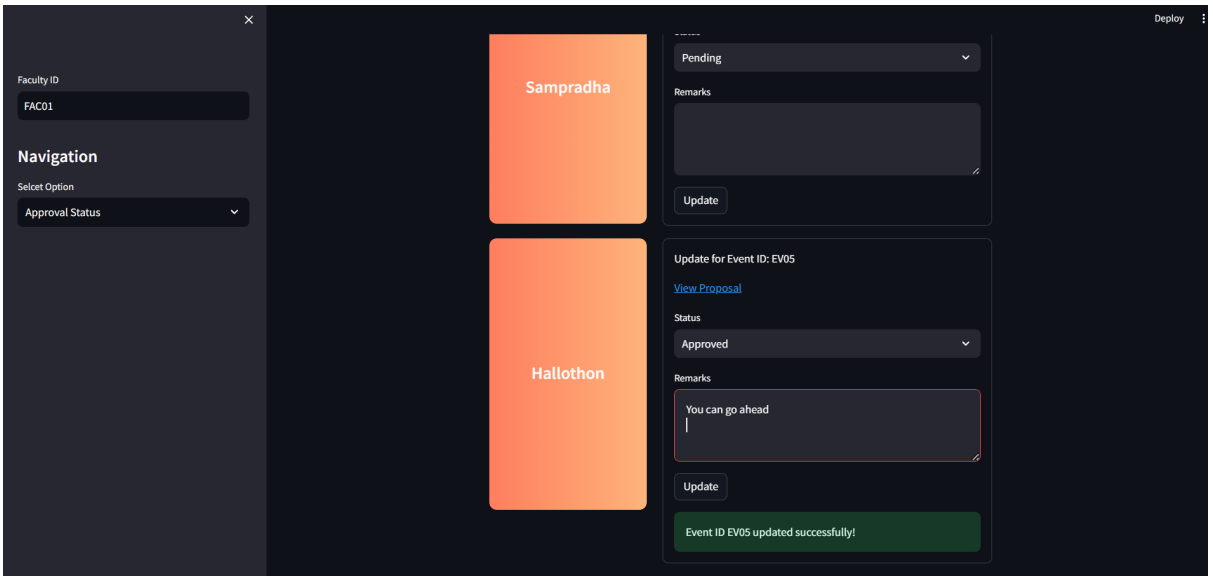
event_id	name	date	venue	about
EV01	Haul It Away	2023-12-01	PESU52	Hackathon
EV02	Sampradha	2023-12-05	Government Schools	fundraising for government schools
EV03	Kodikon	2023-01-02	PESU52	Hackathon
EV04	Xmas-thon	2023-12-22	GJB	Kick off your holidays with a win in Xmas-thon!
EV05	Hallothon	2023-11-25	MRD Auditorium	It's an Halloween hackathon
NULL	NULL	NULL	NULL	NULL

FEATURE 3: **<u>Approval from faculty mentor of club</u>**

UT-01: Confirm that the faculty approval status is initially set to pending.

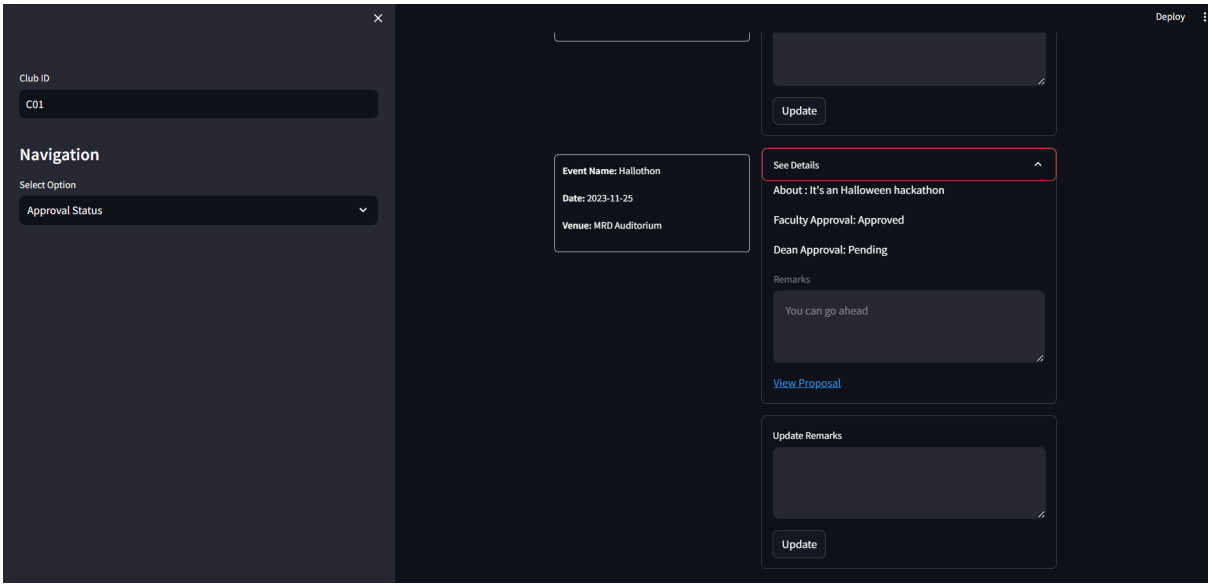


UT-02: Validate that the approval status is updated correctly when the faculty mentor approves or rejects the event.

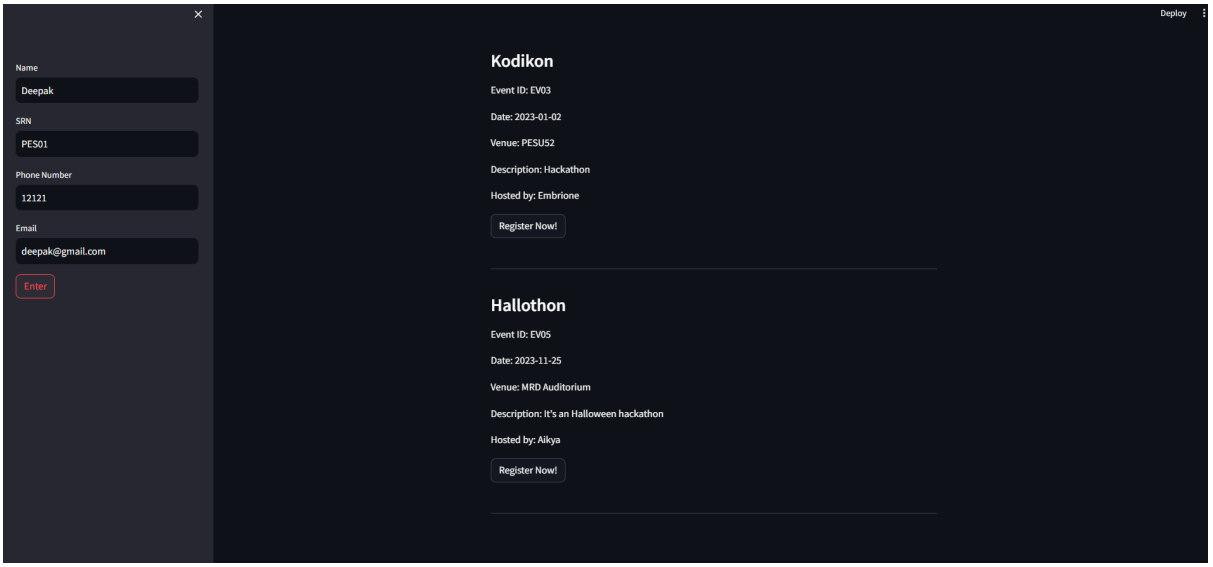


IT-01: Ensure that the faculty approval process integrates with the event creation module.

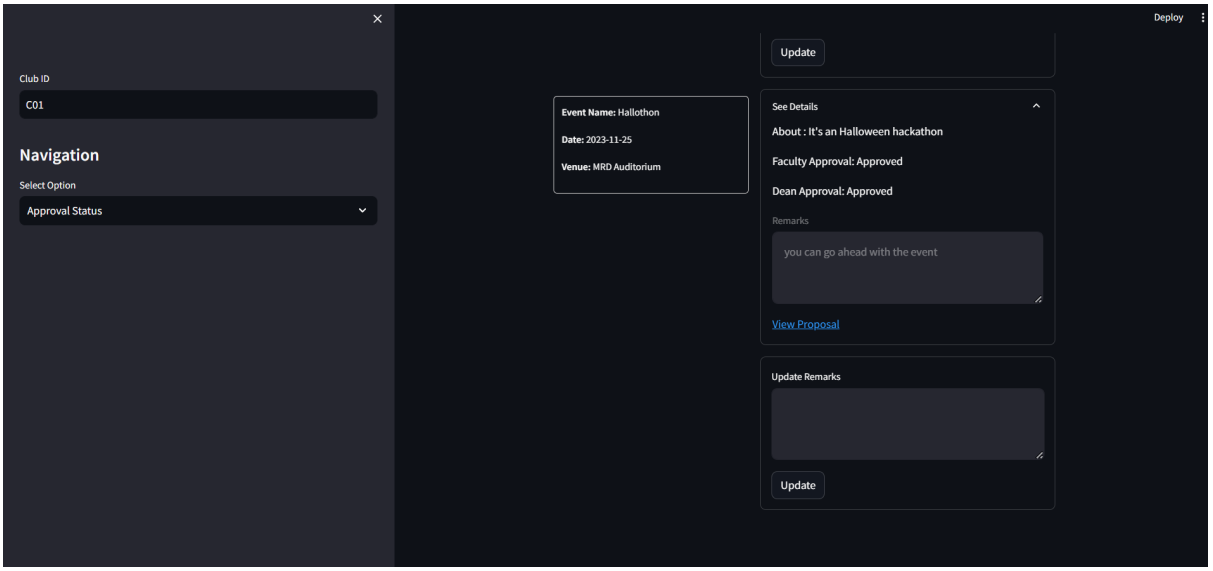
	proposal	fac_approval	dean_approval	remarks
	https://docs.google.com/document/d/1T...	Approved	Approved	Event preparations commer
	https://docs.google.com/document/d/1T8A6u2...	Pending	Pending	Make them
	https://docs.google.com/document/d/1T20EA7...	Approved	Approved	NULL
n!	https://docs.google.com/document/d/1cwovA7...	Pending	Pending	Event Created
▶	https://docs.google.com/document/d/1T8A6u2...	Approved	Pending	You can go ahead



ST-01: Verify that only approved events are displayed in the participant event listings.



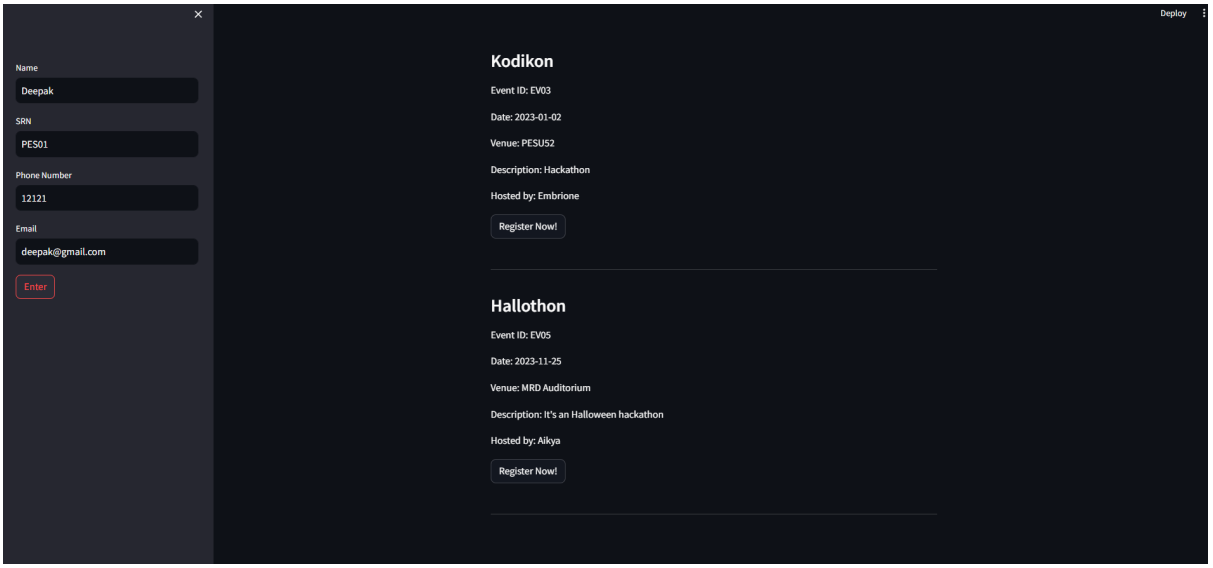
ST-02: Check that pending events are appropriately communicated to the club heads.



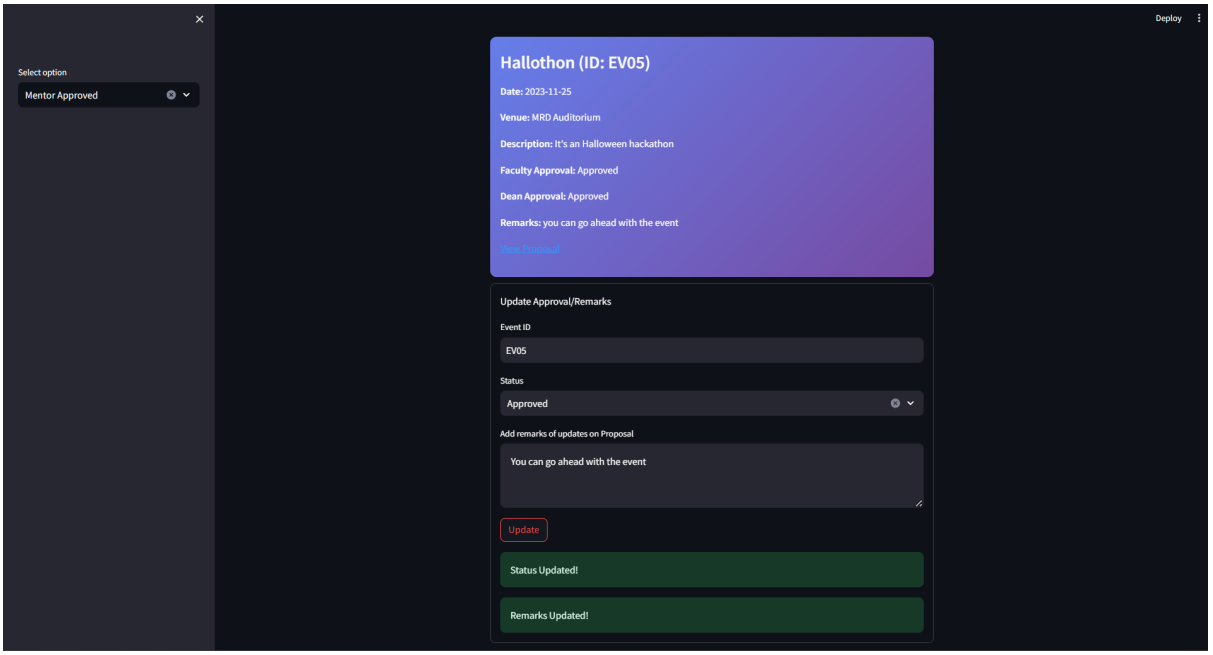
	proposal	fac_approval	dean_approval	remarks
▶	https://docs.google.com/document/d/1T...	Approved	Approved	Event preparations commenc
	https://docs.google.com/document/d/1T8A6u2...	Pending	Pending	Make them
	https://docs.google.com/document/d/1T20EA7...	Approved	Approved	NULL
	https://docs.google.com/document/d/1cwovA7...	Pending	Pending	Event Created
	https://docs.google.com/document/d/1T8A6u2...	Approved	Approved	you can go ahead with the e

FEATURE 4: <u>Event Discover</u>

UT-01: Ensure that the event discovery module integrates well with the event creation and approval features and displays for participants

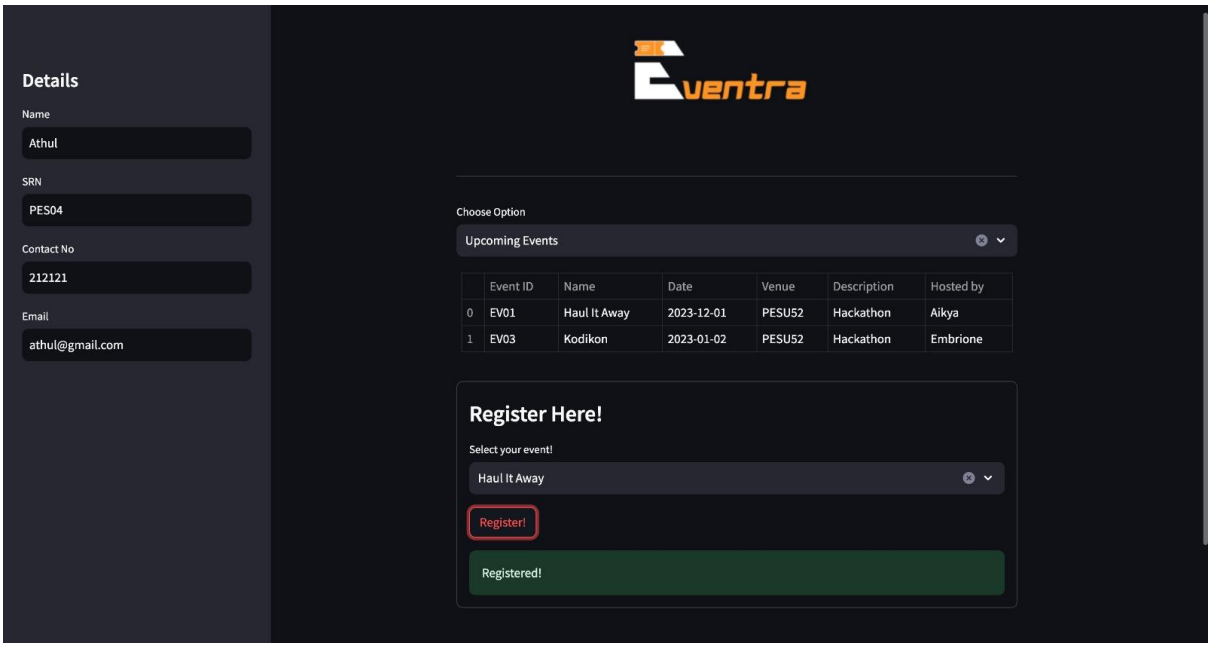


ST-01: Confirm that events are displayed accurately on the platform for the dean to approve.

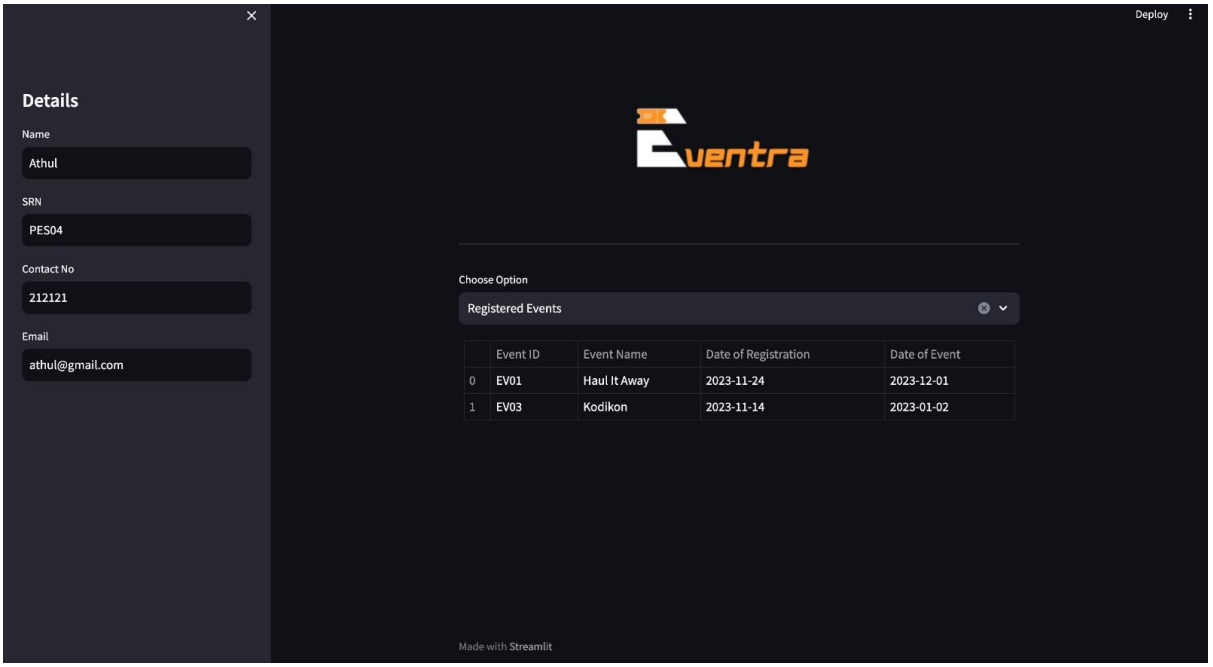


FEATURE 5: Event Registration and Ticketing

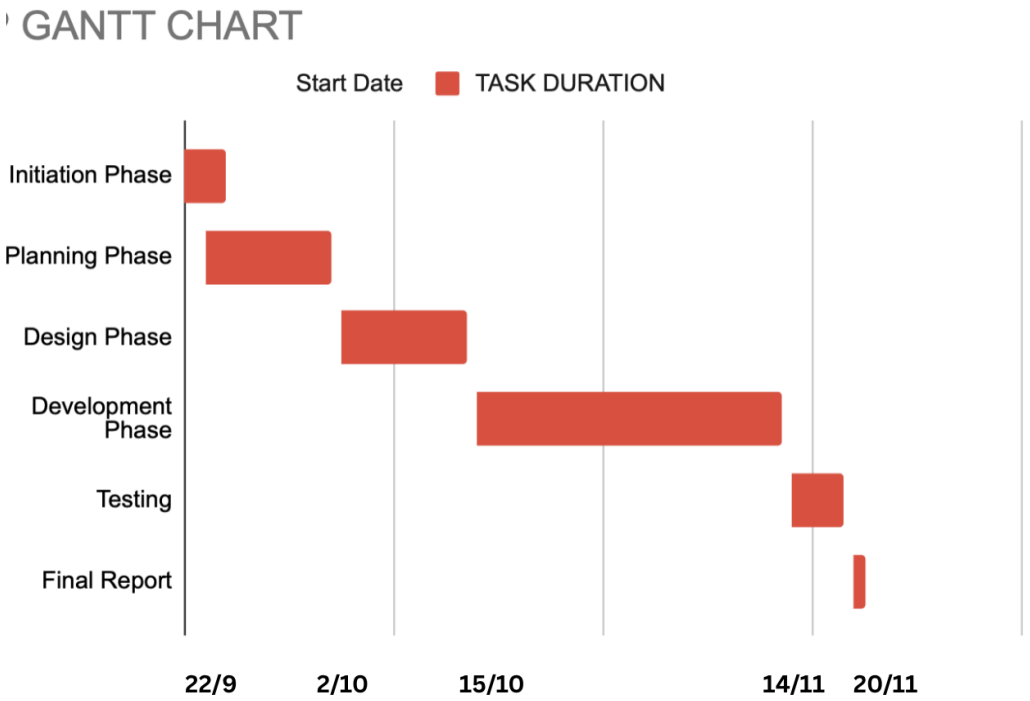
UT-01: Verify that users can register for an event successfully.



IT-01: Confirm that the registration module interacts seamlessly with the user authentication system.



8. Final Gantt Chart:



9. Conclusion:

The successful implementation of the college club event management system, EVENTRA, marks a significant achievement in streamlining event organisation within the college community. While the system has been implemented successfully, it is acknowledged that full deployment has not been realised entirely due to time constraints. Key features, including user login, event creation, faculty mentor approval, event discovery, and event registration, have been designed and implemented to meet specific functional requirements. The utilisation of Streamlit for the user interface and MySQL for database management has contributed to the development of a robust and user-friendly system.

The testing phase, encompassing User Testing (UT), Integration Testing (IT), and System Testing (ST), has validated the functionality of individual components and their seamless interaction. Test cases have been designed to cover various scenarios, ensuring the system's reliability and identifying areas for further refinement.
The project team recognises the importance of ongoing efforts to address any remaining issues, gather user feedback, and perform a thorough deployment when time permits. While the current state represents a successful implementation, the full deployment will require additional considerations, such as user training, system optimization, and potential bug fixes.

In conclusion, the successful implementation of EVENTRA is a testament to the dedication and effort put forth by the project team. The system's functionalities are poised to make a positive impact on event management within the college community, and future deployment efforts will enhance its accessibility and usability for all stakeholders.

Appendix A(a): Glossary

SRS: Software Requirements Specification- A document that defines the functional and non-functional requirements of a software system.

API: Application Programming Interface-A set of rules and protocols that allow different software applications to communicate with each other.

DBMS: Database Management System- Software that manages storage retrieval, and organisation of data in a database.

HTTP: HyperText Transfer Protocol-A protocol used for transferring data over the internet, commonly used for web browsing.

HTTPS: HyperText Transfer Protocol Secure- A secure version of HTTP that encrypts data transferred between a user's browser and a web server.

Bug: A defect or error in the software that causes it to behave unexpectedly or incorrectly.

Encryption: The process of converting data into a code to protect it from unauthorised access.

Backups: copies of data that are stored separately to prevent data loss in case of system failures or
disasters.

User Manual: Documentation that provides instructions and guidance on how to use a software
Application.

Appendix A(b): Field Layouts

Field	Length (bytes)	Data type	Description	Is Mandatory
Event ID	10	Numeric	Unique identifier for the event	Y
Event Name	60	String	Name of the event	Y
Event Date	8	Date	Date of the event	Y
Event Time	8	Time	Time of the event	Y
Event Location	60	String	Location of the event	Y
Event Organiser	60	String	Name of the event organiser	Y
Event Description	255	String	Description of the event	Y
Event Capacity	10	Numeric	Maximum number of attendees for the event	N
Event Registration Fee	10	Numeric	Fee for registering for the event	N
Registration Deadline	8	Date	Deadline for registering for the event	Y

Registration Report
Event Name
Event Date
Event Time
Event Location
Event Organiser
Registrant Name
Registrant Email Address
Registrant Phone Number
Registration Date
Payment Status

Transaction Report
Event Name
Registrant Name
Registration Fee
Payment Method
Payment Date
Payment Status

Additional Report Requirements
Attendance Report
Payment Summary Report
Event Feedback Report

Appendix B: Requirement Traceability Matrix(RTM)

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	REQ-01	The system shall allow users to create, edit, and manage events.	event_management.md	event_management_design.pdf	EventManager.py	TC-01	TC-02
2	REQ-02	The system shall allow users to register for events.	registration.md	registration_design.pdf	RegistrationManager.py	TC-03	TC-04
3	REQ-03	The system shall allow users to pay for event registration fees online.	payment_processing.md	payment_processing_design.pdf	PaymentProcessor.py	TC-05	TC-06
4	REQ-04	The system shall generate reports on attendance, registration, and payments.	reporting.md	reporting_design.pdf	ReportGenerator.py	TC-07	TC-08
5	REQ-05	The system shall provide status of event approvals.	approval.md	approval_design.pdf	ApprovalStatus.py	TC-09	TC-10

Appendix C: Technology stack and References

- **Streamlit**
- **MySQL**
- **HTML**
- **CSS**