

<b>Name</b>	DHARANI M
<b>Roll no</b>	7376222AD124
<b>Seat no</b>	4
<b>Project ID</b>	4
<b>Problem Statement</b>	BITHACK'S 24 -Hackathon Events Forum

## 1. Introduction

### 1.1. Purpose:

The purpose of this platform is to provide a centralized solution for organizing and managing internal college events. It aims to facilitate team formation among students, fostering collaboration and team-based project development while encouraging the exchange of knowledge, insights, and best practices among event participants. Administrators can easily add new event domains or project categories, review and approve project abstracts submitted by student teams, and enhance the overall event experience for both organizers and participants.

### 1.2. Scope of Project:

- User management for students and administrators
- Event registration and domain selection
- Team formation and project abstract submission
- Administrative features for managing events and approving projects
- Analytical dashboard for monitoring event activities and performance

## **2. Users:**

### **2.1 Students:**

Register, log in, participate in events, form teams, submit project abstracts, and track application status.

### **2.2 Admins:**

1. Administrators (college staff) can add new domains or project categories to the platform.
2. They can review the abstracts submitted by the student teams and approve or reject them.

## **3. Features:**

### **1. Event Categories and Domains:**

The platform clearly categorizes events, such as software or hardware. Students can choose from various domains within each category, such as AR/VR, Full-stack, etc.

### **2. Team Formation:**

Students can join an existing team with fewer members or create a new one. This feature facilitates collaboration and team-based project development.

### **3. Abstract Submission:**

After forming a team, students can submit an abstract of their project. The abstract should include details about the project, its objectives, and any relevant information.

### **4. Admin Review and Approval:**

Administrators can review the submitted abstracts and either approve or reject them. This process ensures the quality and relevance of the projects being undertaken.

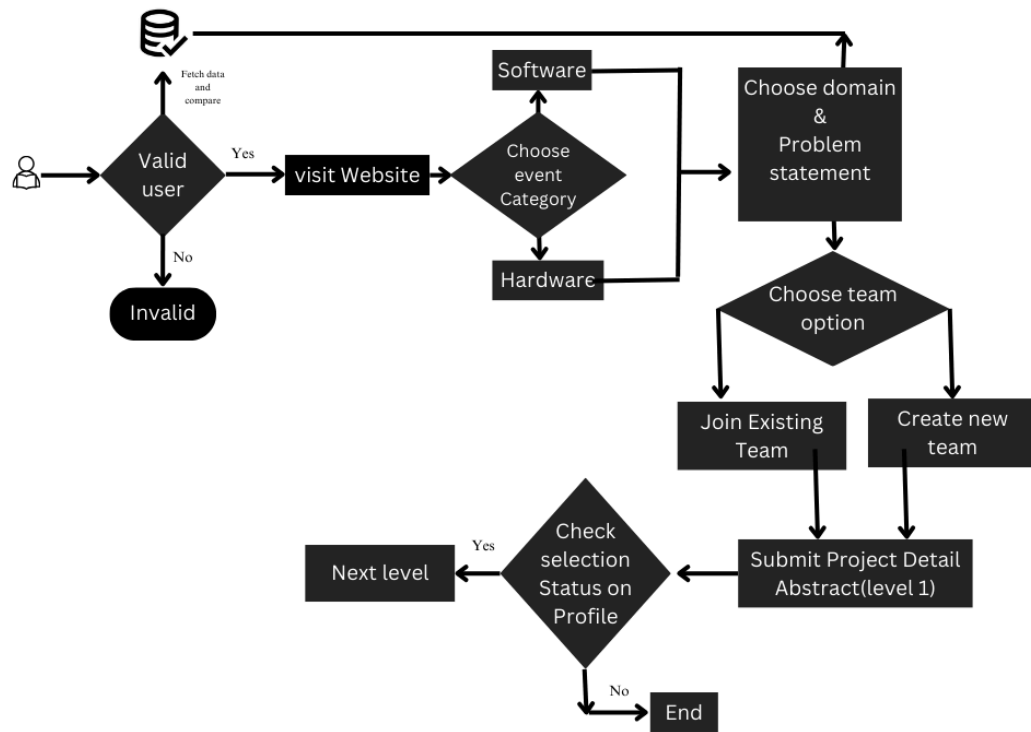
### **5. Knowledge Sharing:**

The platform can provide a space for students to share their knowledge, insights, and best practices related to the event domains. This can be facilitated through forums, discussion boards, or other collaborative features.

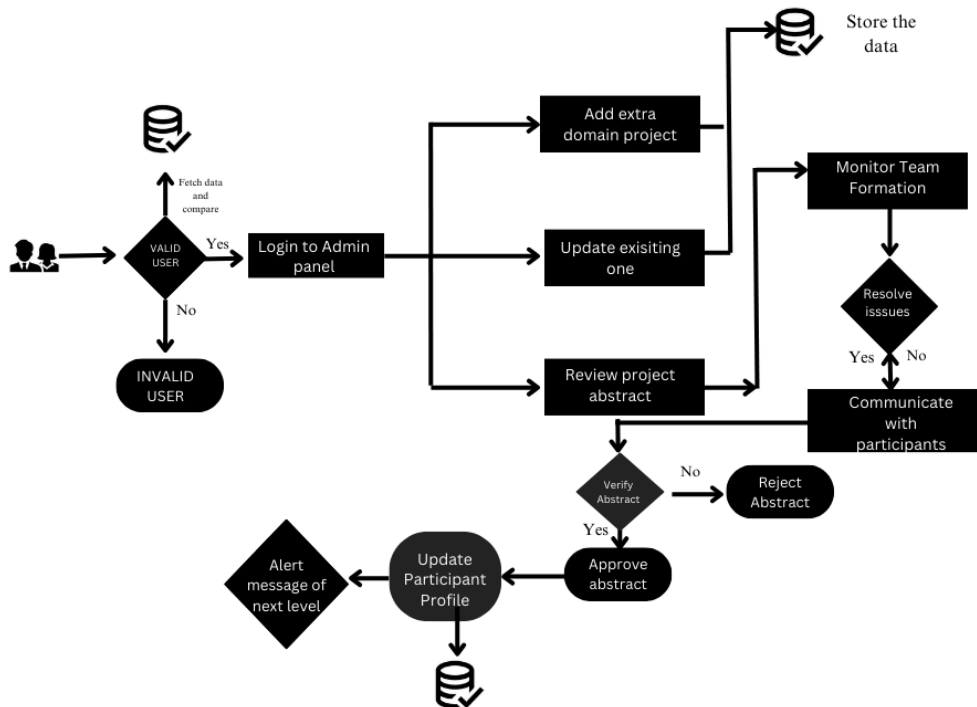
### **6. Analytical Dashboard:**

Administrators can access an analytical dashboard to monitor the event progress, track team participation, and gather insights for future improvements.

## USER INTERFACE



## ADMIN INTERFACE



### 3. System Requirements Specification:

#### 3.1 Functional Requirements:

1. User Management: Registration, login, and access control.
2. Event Registration: Category selection, domain choices, team formation.
3. Team Formation and Abstract Submission: Project abstract submission with details.
4. Admin Dashboard: Event domain management, project abstract review, approval/rejection.
5. Analytics Dashboard: Monitoring applications by category, tracking application requests and approvals.

#### 3.2. Non-Functional Requirements:

1. Usability: User-friendly GUI, clear instructions.
2. Performance: Handling concurrent user sessions, and fast data retrieval.
3. Security: Robust authentication, and compliance with security protocols.
4. Scalability: Accommodating user growth, scaling based on requirements.
5. Maintainability: Modular architecture, comprehensive documentation.
6. Reliability: High uptime, robust error handling.

### 4. System Architecture

The software will follow a client-server architecture, with the client-side (web application) communicating with the server-side (backend) through secure APIs. A relational database management system will be used for data storage and retrieval.

#### Stack:

Front End	React.js
Backend	Node.js with Express.js
Data Base	MongoDB

## Backend:

### 1. Student entity

name	String
email	String
password	Hash code
Roll no	String

### 2. Admin entity

Admin_id	String
name	String
email	String

### 3. Project entity

project_id	Integer
project_title	String
Abstract	Hash code
Team-id	Integer
category	string
Domain	string