

ST. ANN'S COLLEGE OF ENGINEERING & TECHNOLOGY



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

→ **TITLE :** College Hackathon And Judging Portal



Batch NO:A17



Team Members :



A.Harika (22F01A0504)
K.Daniel (22F01A0566)
B.John (22F01A0514)

Under The Guidance Of
Dr.P.V.S.SARMA

COLLEGE HACKATHON AND JUDGING PORTAL

Presented by: Trinity



OVERVIEW

01

Abstract

02

Introduction

03

Goals & Scope

04

System & Issues

05

Our Solution

06

Research

07

Modules

08

UML Diagrams

09

Requirements

10

Algorithms

11

Testing & Results

12

Conclusion



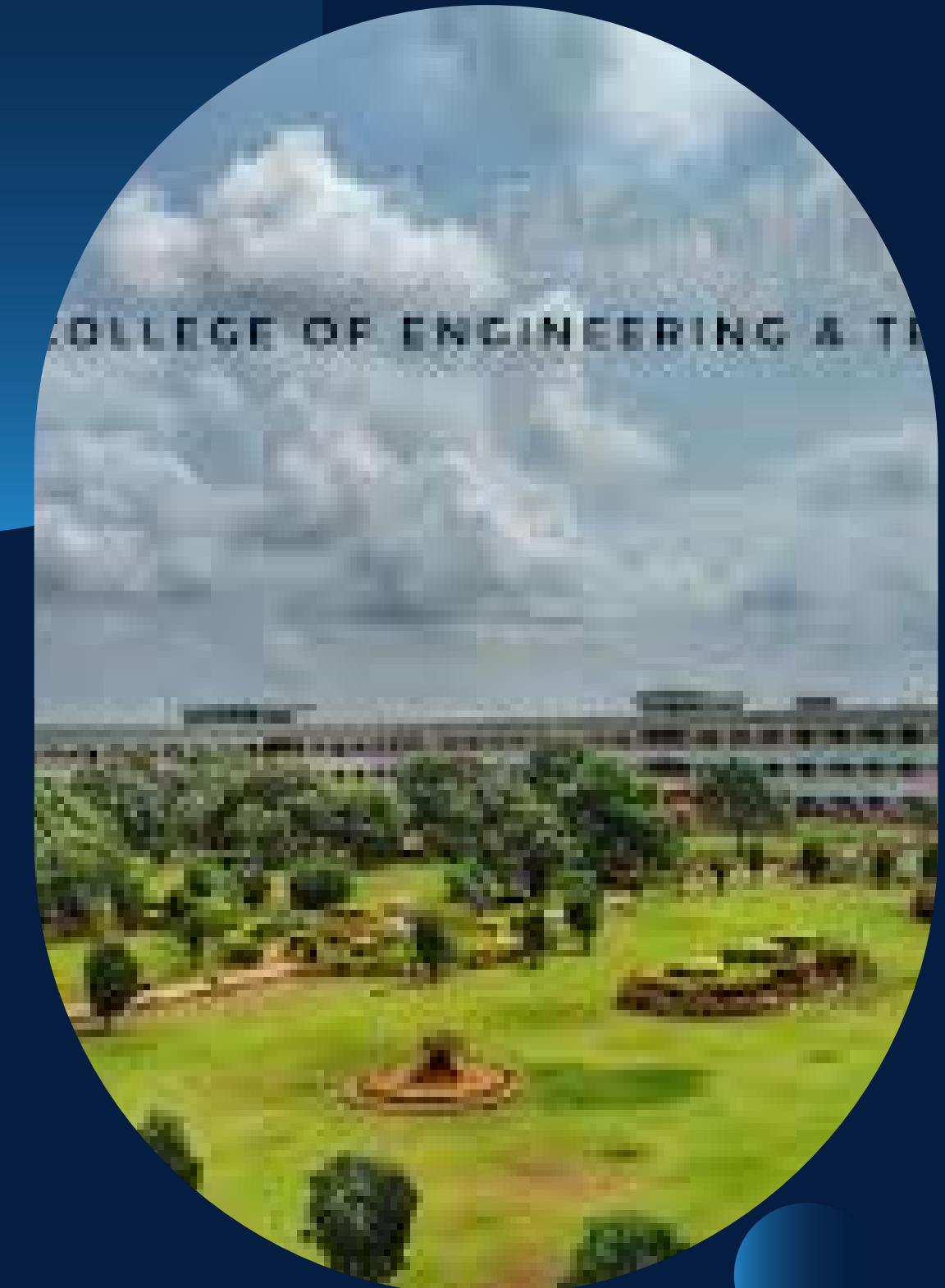
01 ABSTRACT



The College Hackathon and Judging Portal is a web-based platform designed to manage hackathons efficiently. It allows students to register, form teams, submit ideas, and upload project prototypes online. Judges can evaluate entries using a rubric-based scoring system, ensuring fair and transparent assessment. The portal automates score aggregation, ranking, and result declaration, reducing manual effort. By digitalizing the process, it promotes efficiency, transparency, and collaboration, fostering a culture of innovation in academic institutions.

02 INTRODUCTION

- 01 Overview** Platform to manage college hackathons from registration to judging.
- 02 Purpose** Simplify event organization and boost collaboration
- 03 Features** Real-time submissions, auto-scoring, leaderboards.
- 04 Benefits** Fair judging, less manual work, more engagement.



03 SCOPE & OBJECTIVES

SCOPE

- Online platform for registration, submissions, and judging.
- Real-time leaderboard for transparency.
- Communication channel for participants, mentors, and judges.

OBJECTIVES

- Simplify hackathon management.
- Ensure fair and unbiased scoring.
- Encourage teamwork and creativity.



04

CURRENT SYSTEM & ISSUES



CONCEPT 1 – MANUAL MANAGEMENT

Hackathons are mostly organized using manual processes or simple tools like Google Forms and spreadsheets, making coordination harder.

CONCEPT 2 – SUBMISSION & JUDGING PROBLEMS

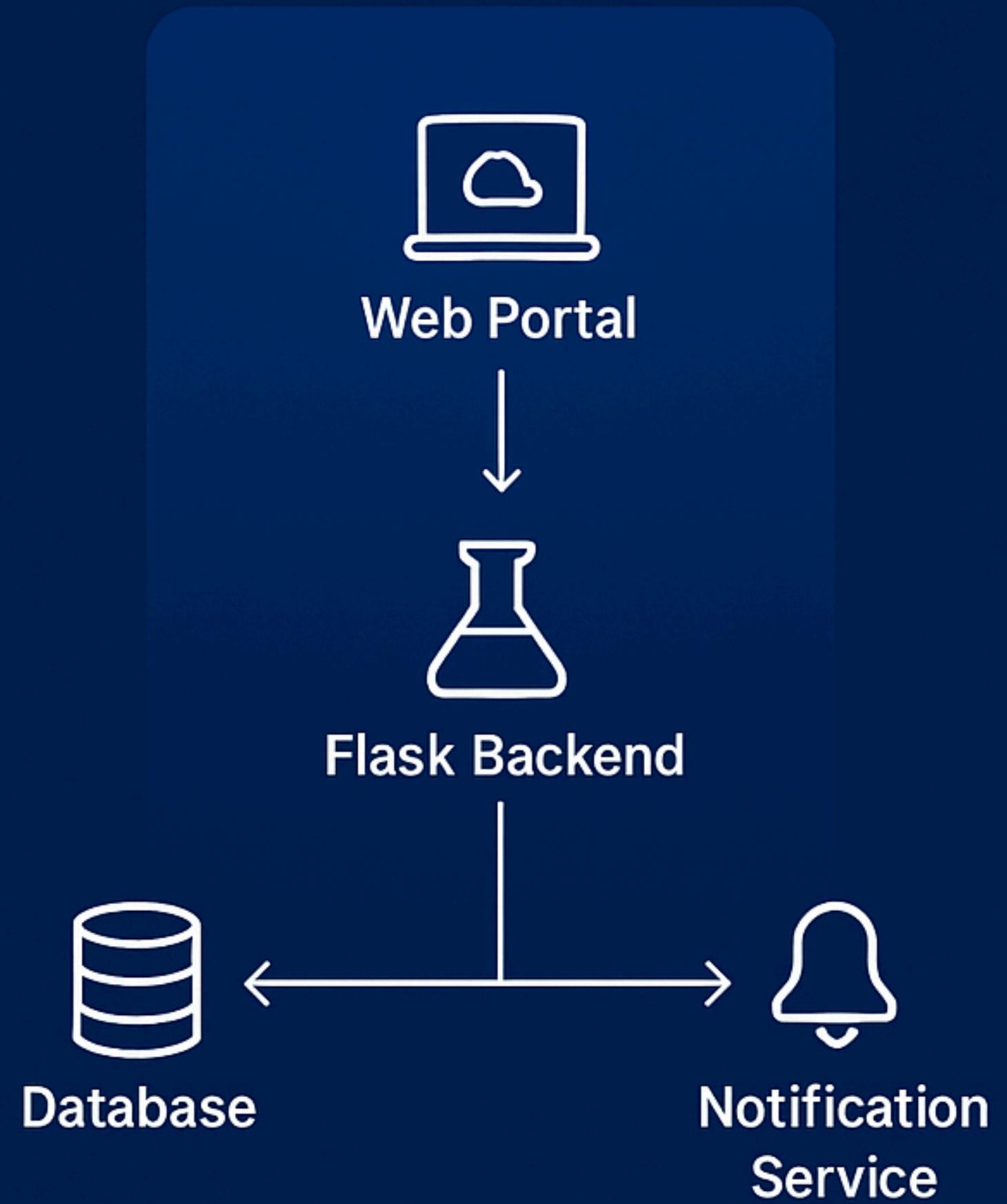
Project submissions happen via email or file links, and judging is manual, which can cause delays and human errors.

CONCEPT 3 - COMMUNICATION GAPS

No single platform for registration, submissions, judging, and communication, leading to confusion and inefficiency.

05 ARCHITECTURE

- 01 Streamlined submissions for easy project uploads
- 02 Efficient judging with real-time scoring
- 03 Transparent results with automated scoring



06 LITERATURE REVIEW

Web-Based College Event Registration System – Sharma et al. (2019) In order to manage large-scale technical and cultural events in universities, Sharma et al. (2019) developed a two-phase registration portal with pre-verification of participants followed by database storage in a cloud environment. The system used secure authentication, email confirmation, and automated ID generation. The study showed that the use of an online registration system reduced administrative workload by 65% compared to traditional manual processes.



Data-Driven Participant Grouping for Hackathons – Mehta and Raghavan (2020) Mehta and Raghavan (2020) proposed a role-based participant classification method for hackathons, dividing participants into categories such as individuals, teams, and mentors based on skill set and event role. The study adapted clustering-style grouping techniques from data mining to assign participants to suitable event tracks, thereby improving judge allocation and time management during the event.

Automated Judge Allocation Model – Khan et al. (2021) In multi-track hackathons, selecting the right judge for the right category is crucial. Khan et al. (2021) developed an automated allocation system that uses expertise keywords and project category tags to assign judges. The research compared static

6.1 LITERATURE REVIEW

- ◆ Web-Based College Event Registration System – Sharma et al. (2019) In order to manage large-scale technical and cultural events in universities, Sharma et al. (2019) developed a two-phase registration portal with pre-verification of participants followed by database storage in a cloud environment. The system used secure authentication, email confirmation, and automated ID generation. The study showed that the use of an online registration system reduced administrative workload by 65% compared to traditional manual processes.
- ◆ Real-Time Scoring and Leaderboard System – Li and Banerjee (2020) Li and Banerjee (2020) designed a real-time scoring API for coding competitions where judges' scores were instantly pushed to a central leaderboard using WebSocket technology. The study found that this live feedback loop increased audience engagement and reduced the time between judging and result announcement by 75%.
- ◆ Retention and Engagement Strategies in Hackathon Platforms – Gupta et al. (2018) Gupta et al. (2018) studied long-term engagement in recurring hackathon events and proposed gamified dashboards, achievement badges, and personalized reminders. Using historical participation data, the system identified and targeted likely returnees, resulting in a 20% increase in repeat participants.
- ◆ Geographical and Institutional Participant Analysis – Das and Rao (2019) Das and Rao (2019) implemented a database filtering model to categorize participants by institution, city, and state for logistics planning and diversity analysis. This geographical segmentation helped organizers identify underrepresented regions and direct targeted invitations to improve participation equity

07

MODULES OF OUR PROJECT

01

User Authentication & Registration

- Allows participants, judges, and admins to securely log in and register for the platform.

02

Hackathon Management

- Admins can create, schedule, and manage hackathon events, Including registration deadlines, themes, and rules setup.

03

Project Submission

- Participants can upload project pct details, source code, and documentation.

04

Judging & Evaluation

- Judges can view assigned p̄i projects and score them based on criteria.

05

Leaderboard & Results

- Displays participant rankings dynamically
- Publishes final results after evaluation is complete

UML DIAGRAMS

08



- ▶ USECASE DIAGRAM
- ▶ CLASS DIAGRAM
- ▶ ACTIVITY DIAGRAM
- ▶ SEQUENCE DIAGRAM

8.1 USECASE DIAGRAM

→ student

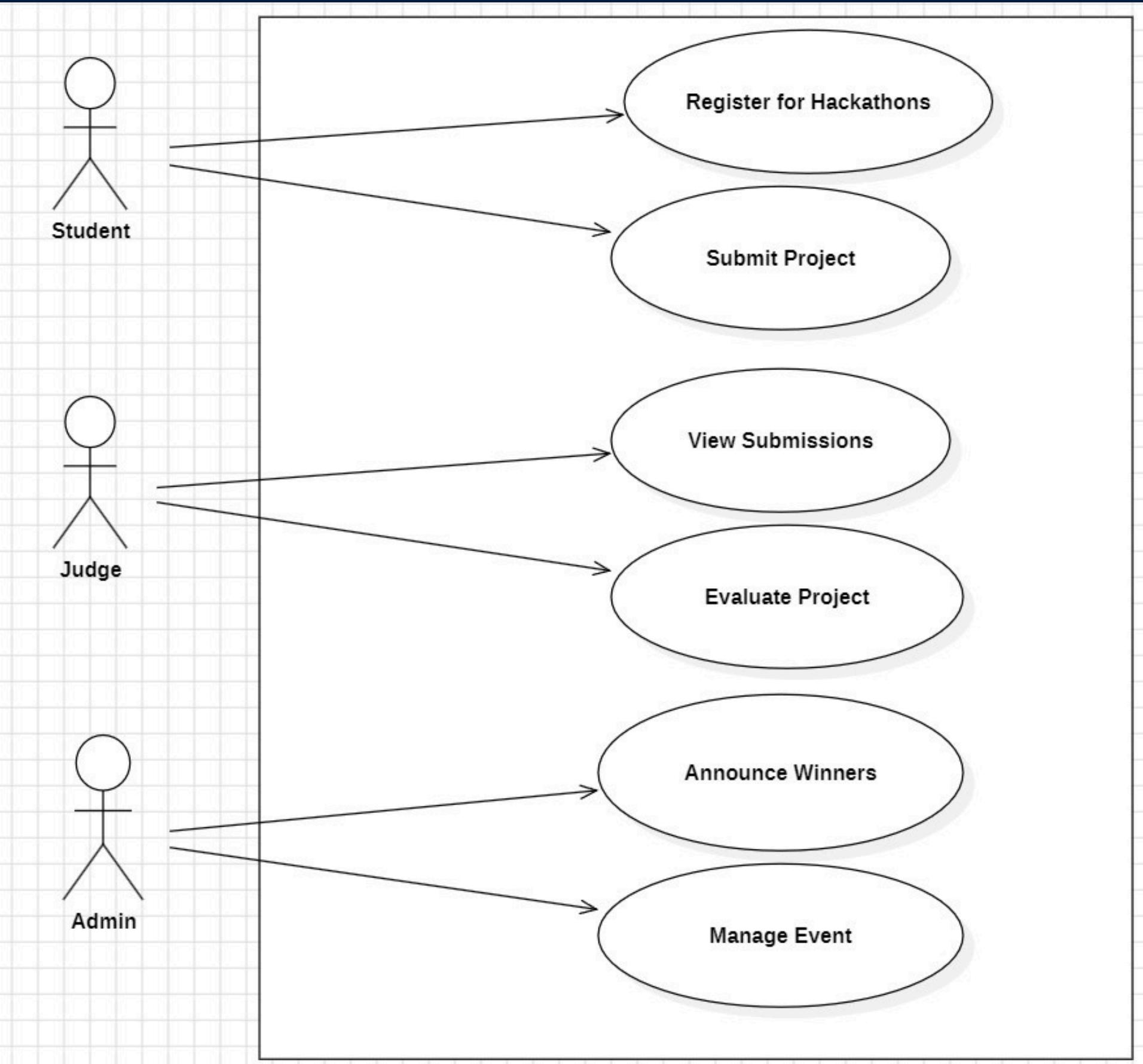
- Register for Hackathons – Students can sign up for available hackathon events.
- Submit Project – Participants can upload their project submissions for evaluation.

→ Judge

- View Submissions – Judges can access and review all submitted projects.
- Evaluate Project – Judges assess each project based on predefined criteria.

→ Admin

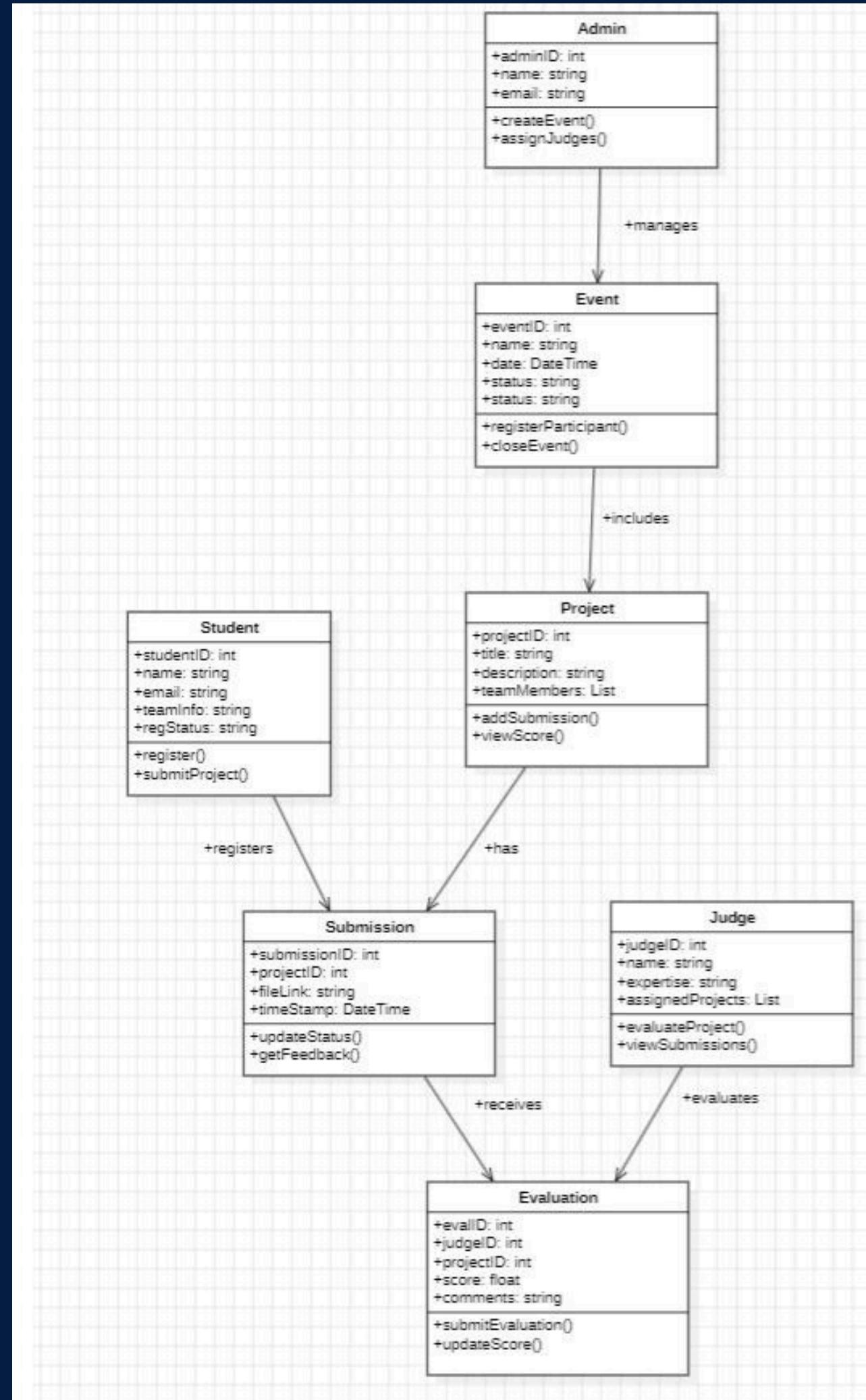
- Announce Winners – Admins publish the final results after evaluation.
- Manage Event – Admins handle event scheduling, participant management, and overall coordination.



8.2

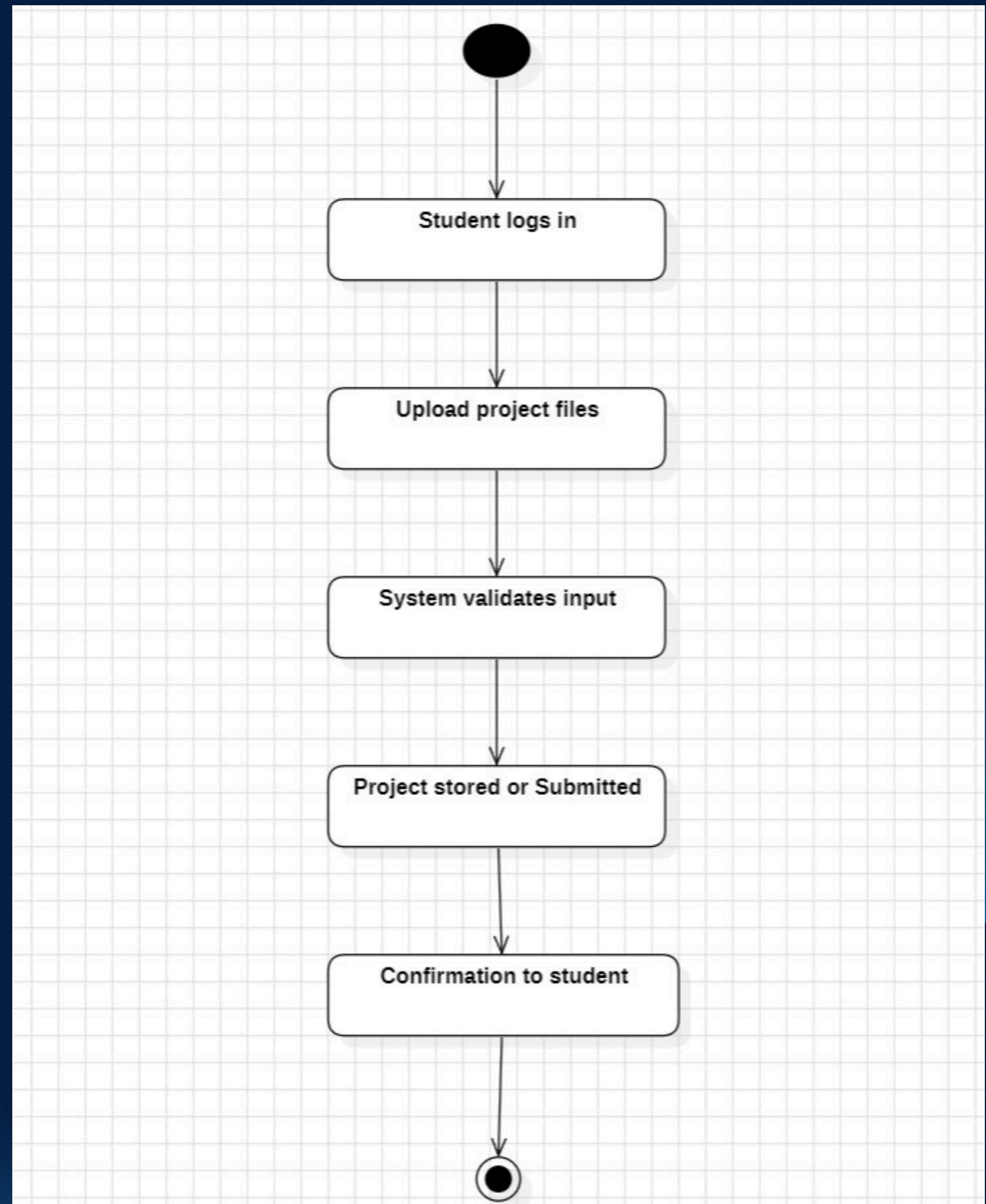
CLASS DIAGRAM

- **Admin** – Creates hackathon events and assigns judges.
- **Event** – Stores event details such as date, status, and participants.
- **Student** – Registers for events and submits projects.
- **Project** – Contains project information and links to submissions.
- **Submission** – Holds uploaded project files, status, and timestamps.
- **Judge** – Evaluates projects and provides scores.
- **Evaluation** – Stores scores and feedback given by judges.



8.3 ACTIVITY DIAGRAM

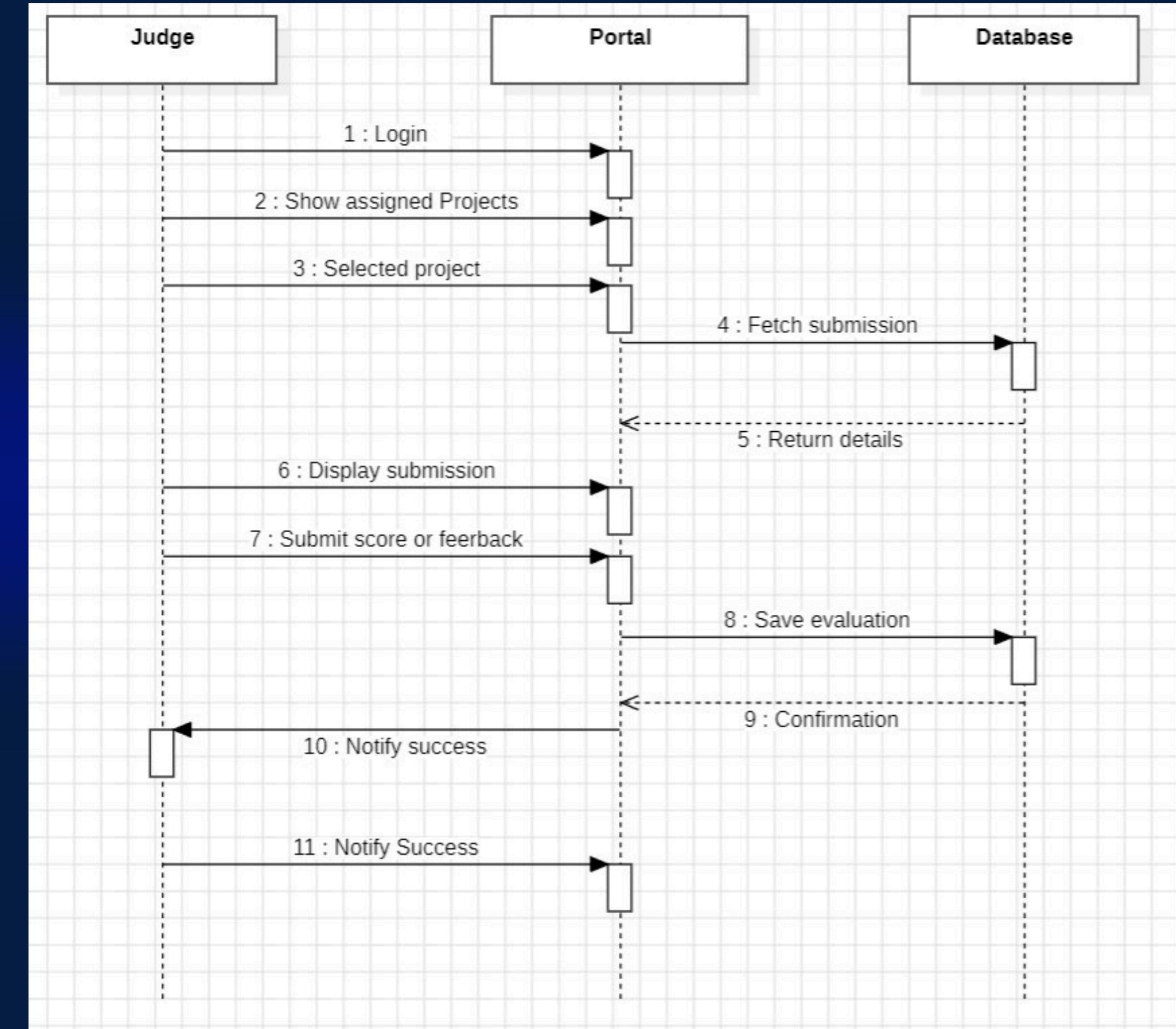
This activity diagram shows the project submission process in the hackathon portal. It starts with the student logging in, uploading project files, and the system validating the input. If valid, the project is stored or submitted, followed by a confirmation message to the student, ending the process.



8.4

SEQUENCEDIAGRAM

→ The sequence diagram illustrates the Judge logging in, selecting a project, retrieving details from the database, submitting scores, and receiving confirmation.



09 SYSTEM REQUIREMENTS



SOFTWARE REQUIREMENTS

- Operating System: Windows 10 / Linux / macOS
- Backend: Python with Flask Framework
- Frontend: HTML, CSS, JavaScript
- Database: Mongodb
- Tools & Libraries: Bootstrap, jQuery, Flask extensions
- Browser: Chrome, Firefox, or Edge (latest version)



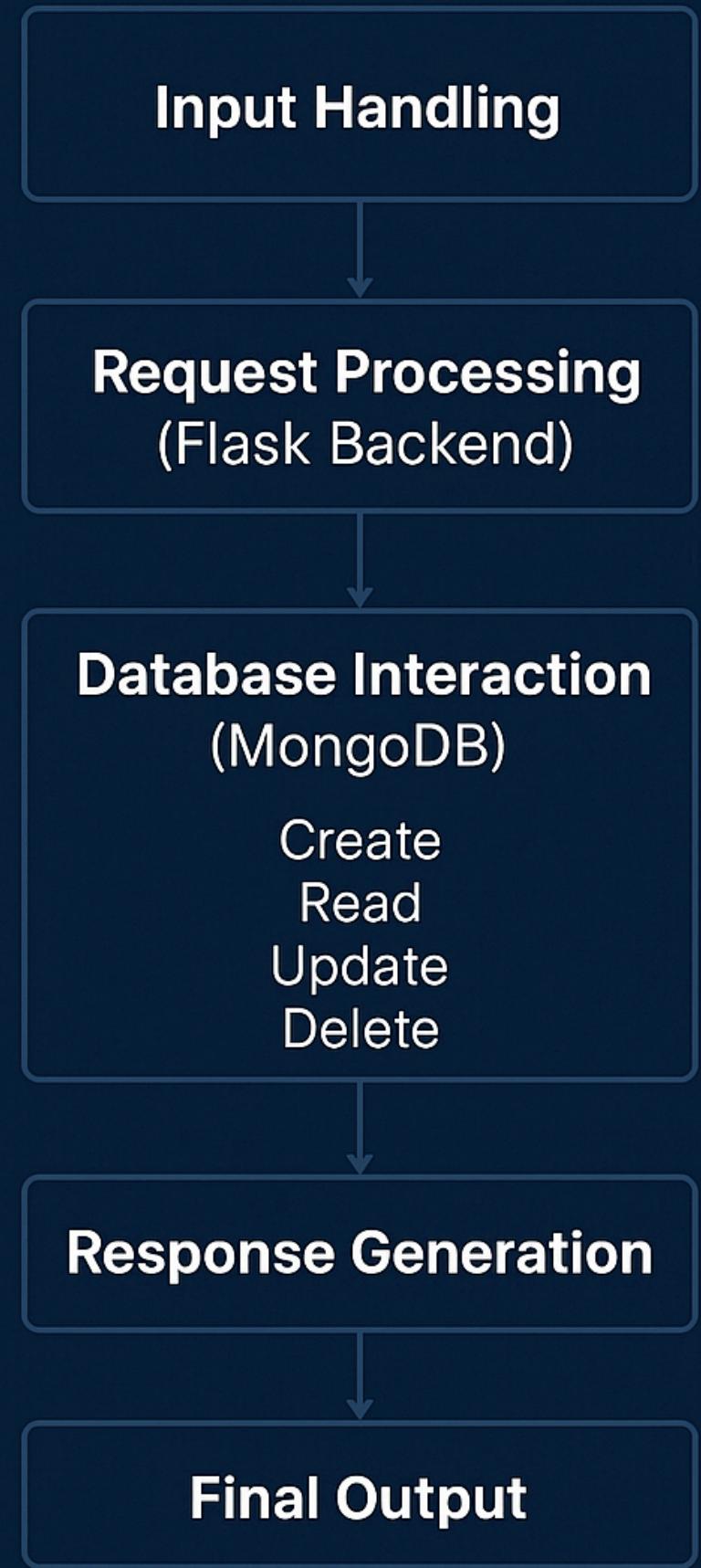
HARDWARE REQUIREMENTS

- Processor: Intel i5 or higher
- RAM: Minimum 8 GB
- Storage: Minimum 256 GB SSD
- Display: 1366×768 resolution or higher
- Peripherals: Keyboard, Mouse, and Internet Connectivity

ALGORITHMS

- The algorithm follows a CRUD (Create, Read, Update, Delete) operation model for handling application data efficiently.
- - **Flask (Backend)**
 - **MongoDB (Database)**
 - **Tailwind CSS (Frontend)**

ALGORITHM USED



ALGORITHM STEPS & ADVANTAGES

10.1



ALGORITHM STEPS

- 1 **User Login**
Judge logs in using valid credentials.
- 2 **Fetch Assigned Projects**
System retrieves a list of projects assigned to the judge from the database.
- 3 **Select Project**
Judge chooses a specific project to evaluate.
- 4 **Retrieve Submission**
Portal fetches project details and submission files from the database.
- 5 **Display Submission**
Submission is shown on the judge's interface for review.
- 6 **Submit Evaluation**
Judge provides scores or feedback.
- 7 **Store Evaluation**
The system saves the evaluation results into a database.
- 8 **Centralized Management**
All project data and evaluations stored in one place.

ADVANTAGES

- **Efficient Workflow** – Streamlined process for judges to evaluate projects.
- **Secure Data Handling** – Uses secure login and database operations.
- **Real-time Updates** – Judges get updated project details instantly.

IMPLEMENTATION

- 1 Project setup
- 2 Backend development
- 3 Frontend development
- 4 Database integration
- 5 Deployment

11.1 Implementation

Project Setup

- Installed required tools: Python, Flask, MongoDB, Tailwind CSS.
- Created virtual environment and installed dependencies (Flask, pymongo, etc.).

Backend Development

- Designed REST API endpoints for user login, project submission, and judging.
- Integrated MongoDB for storing project details, scores, and feedback.
- Implemented authentication and session management for judges.

Frontend Development

- Built responsive UI for login, project listing, submission view, and evaluation forms.
- Styled components using Tailwind classes for quick design customization.
- Used JavaScript to fetch data dynamically from the Flask backend.

Database Integration

- Created collections for Users, Projects, and Scores.
- Defined schema structure for storing submissions and evaluation results.

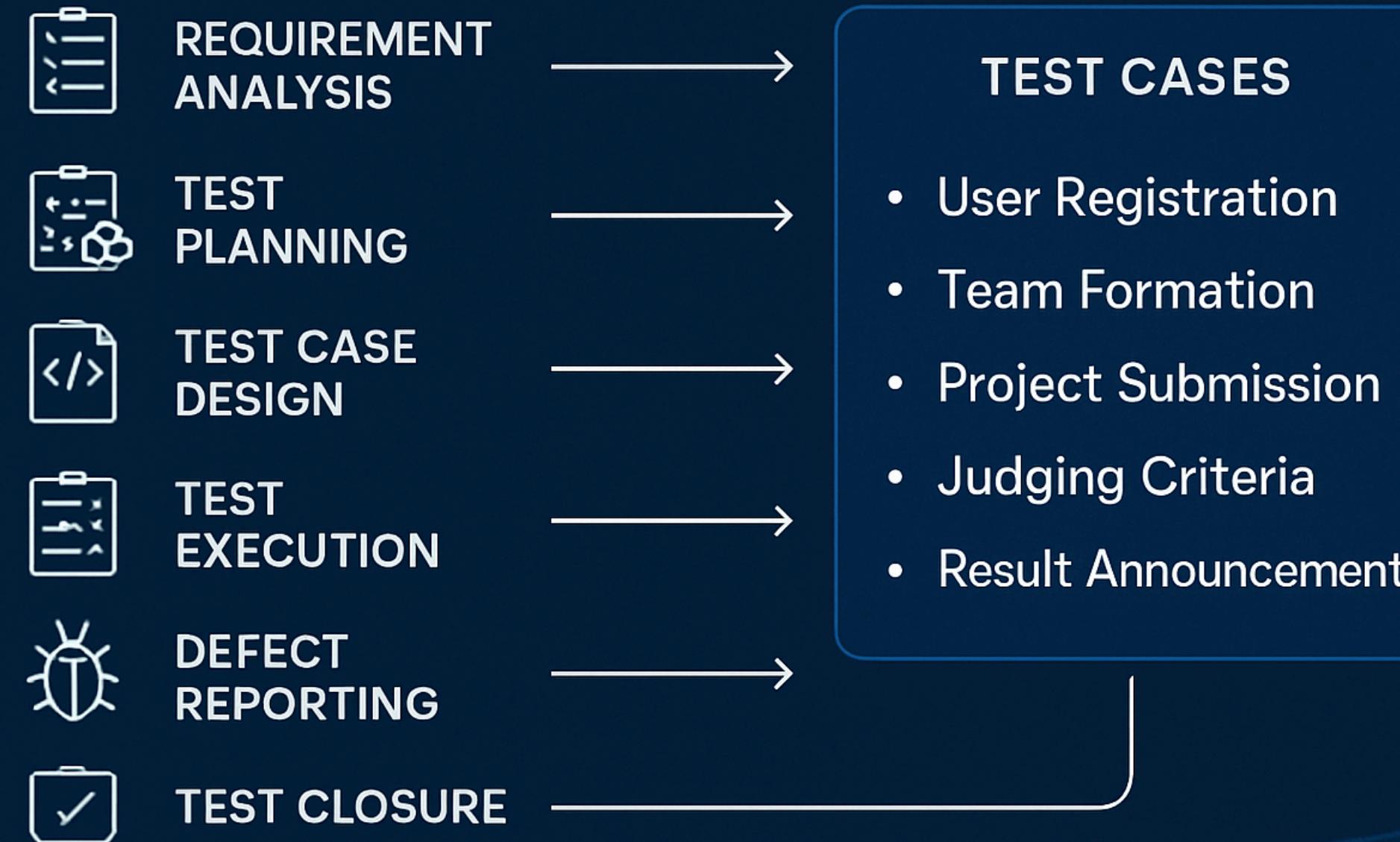
Deployment

- Deployed backend on a hosting service (e.g., Render/Heroku).
- Hosted frontend and connected to the live backend API.
- Configured MongoDB Atlas for cloud database storage.





COLLEGE HACKATHON PORTAL TESTING PROCESS



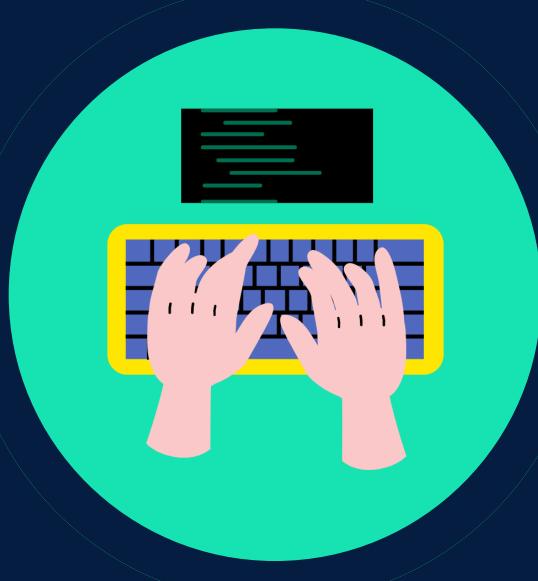
12.1

TEST CASES



ID	Description	Input	Expected Output	Status
TC01	User Registration	Valid details	Registration success	Pass
TC02	Registration Missing Fields	Incomplete details	Error message	Pass
TC03	Project Submission	Valid project info	Submission success	Pass
TC06	User Registration with invalid email	Wrong email format	Show error message	Fail
TC09	Project Submission with wrong format	Invalid file type (.exe)	Display "Invalid file type"	Fail

13 SAMPLE CODE



```
import os
from flask import Flask, render_template, request, redirect, url_for, session, flash, send_file
from werkzeug.utils import secure_filename
from datetime import datetime
from bson import ObjectId
from pymongo import MongoClient
import gridfs

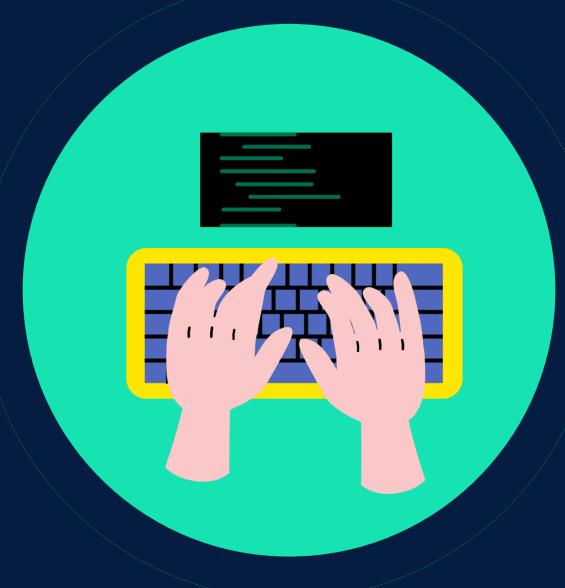
# MongoDB Atlas connection
MONGO_URI = "mongodb+srv://john:<db_password>@hackathon.qp5qlat.mongodb.net?retryWrites=true&w=majority&appName=Hackathonn"
client = MongoClient(MONGO_URI)
db = client["hackathon_portal"]
fs = gridfs.GridFS(db)
app = Flask(__name__)
app.secret_key = os.urandom(24)
app.config['MAX_CONTENT_LENGTH'] = 5 * 1024 * 1024 # 5MB max upload

# Allowed poster extensions
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg', 'gif'}

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

# Home Page
@app.route('/')
def index():
    return render_template('index.html')
```

13.1 SAMPLE CODE



```
@app.route('/judge/logout')
def judge_logout():
    session.clear()
    return redirect(url_for('index'))
```

```
# Endpoint to serve images from GridFS
@app.route('/image/<file_id>')
def serve_image(file_id):
    try:
        file = fs.get(ObjectId(file_id))
        return send_file(file, mimetype=file.content_type)
    except Exception as e:
        flash('Image not found.', 'danger')
        return "", 404

if __name__ == '__main__':
    app.run(debug=True)
```

▶ HOMEPAGE – COLLEGE HACKATHON PORTAL

- The homepage provides role-based access for College, Student, and Judge.
- Each user can Login / Sign Up securely based on their role.
- The portal allows colleges to create hackathons, students to participate and submit projects, and judges to evaluate entries.
- Clean and user-friendly interface with clear navigation.

Welcome to the College Hackathon Portal

Empowering innovation, collaboration, and creativity across campuses. Organize, participate, and judge hackathons with ease!

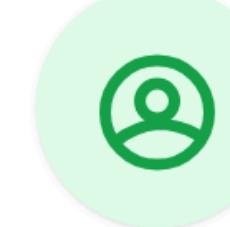
Colleges can launch events, students can showcase their brightest ideas, and judges can reward the best minds—all in one seamless, secure, and beautiful platform.



College

Create and manage hackathons, set up judges, and review student innovations. Be the hub of campus creativity!

[Login](#) [Sign Up](#)



Student

View and join your college hackathons, submit your ideas, and track your scores. Let your talent shine!

[Login](#) [Sign Up](#)

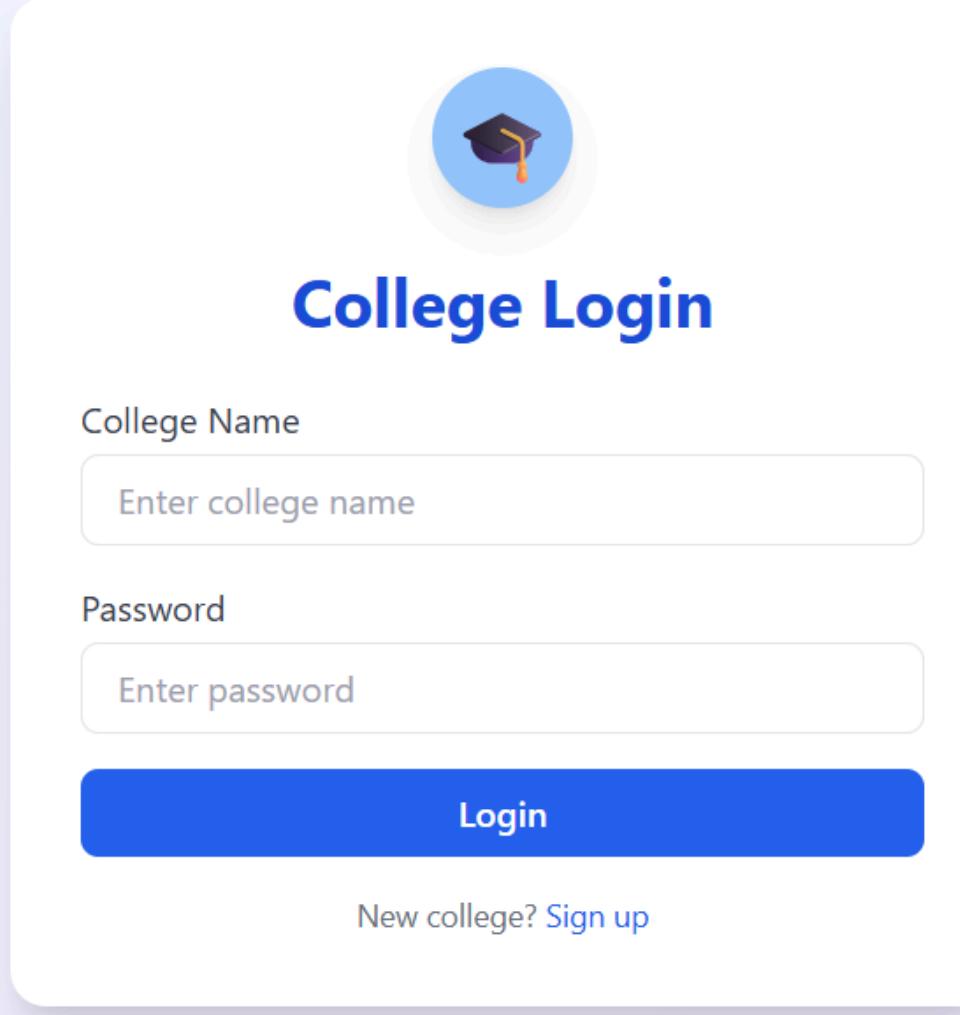


Judge

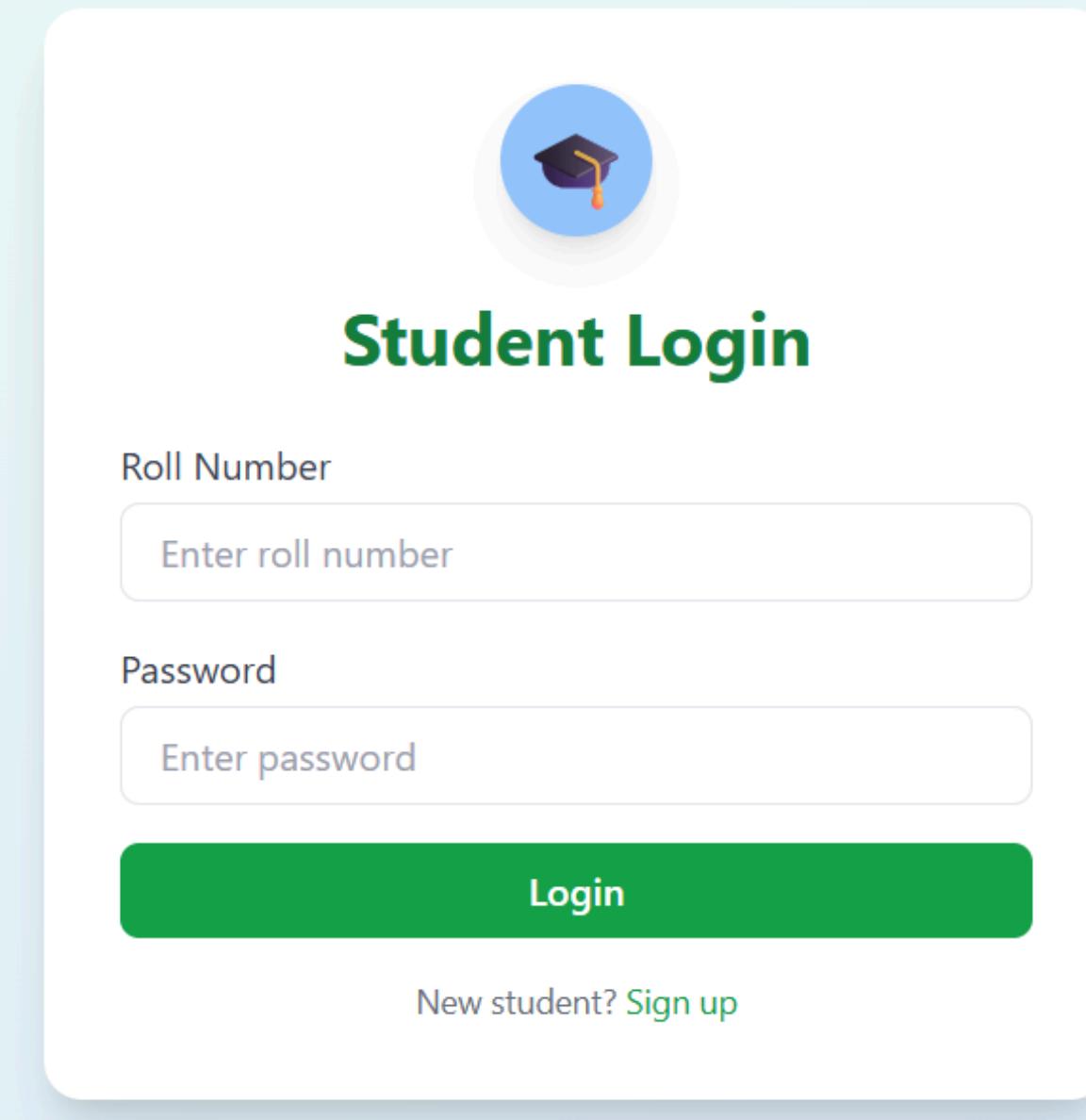
Score student ideas for your assigned college hackathons. Help reward the best and brightest!

[Login](#)

14.1 LOGIN MODULES OUTPUT



The College Login interface features a blue circular icon with a graduation cap inside. Below it is the text "College Login" in blue. The form includes fields for "College Name" (with placeholder "Enter college name") and "Password" (with placeholder "Enter password"). A large blue "Login" button is at the bottom, and a link "New college? Sign up" is at the very bottom.



The Student Login interface features a blue circular icon with a graduation cap inside. Below it is the text "Student Login" in green. The form includes fields for "Roll Number" (with placeholder "Enter roll number") and "Password" (with placeholder "Enter password"). A large green "Login" button is at the bottom, and a link "New student? Sign up" is at the very bottom.

Login Modules of College And Student

14.2 COLLEGE DASHBOARD – HACKATHON MANAGEMENT



Welcome, SACET
Manage your hackathons and review student ideas!

Post a New Hackathon

Title

Description

Deadline

Prizes (optional)

Banner / Poster Image
 No file chosen

Add Judge

Judge ID (unique)

Name

Password

Add Judge

Current Judges
Harini (1234)

Post Hackathon

Your Hackathons

full stack Deadline: 2025-07-29

AI using python full stack



Prizes: 10000000

Student Ideas

Student	Hackathon	Title	Description
John	full stack	NEW IDEA	XYZ

- Create and manage hackathons (title, description, deadline, prizes).
Upload event banner/poster.
- Add judges with unique IDs and credentials.
View current judges list.
- Centralized interface for organizing events.

14.3 Student Hackathon Portal

- Explore upcoming hackathons, check prizes and deadlines, and submit your ideas easily.
- Manage your profile and participate in challenges—all in one place.
- Collaborate, innovate, and track your progress with a simple, user-friendly portal.

The screenshot shows the homepage of the Student Hackathon Portal. At the top, there is a welcome message "Welcome, John" with a graduation cap icon, and links for "Profile" and "Logout". Below this, the heading "Your College Hackathons" is displayed. A card for the "full stack" hackathon is shown, featuring a thumbnail of a mountain range, the title "full stack", the subtitle "AI using python full stack", the deadline "Deadline: 2025-07-29", and the prize amount "Prizes: 10000000". Below this section is a form titled "Submit Your Idea" with fields for "Select Hackathon" (a dropdown menu currently set to "Choose Hackathon"), "Idea Title", "Idea Description", and "Prototype (optional link/text)". A large green "Submit Idea" button is at the bottom of the form.

14.4 JUDGES PORTAL

Judge Login

Judge ID
Enter Judge ID

Password
Enter password

Login

Welcome, Judge Harini
Score student ideas and help reward innovation!

Student Ideas to Score

Student	Hackathon	Title	Description	Prototype	Score	Action
John	full stack	NEW IDEA	XYZ	1	Score	Assign

Logout

- Secure judge login
- View assigned hackathons
- Evaluate & score projects
- Transparent scoring system
- Real-time leaderboard updates



- A complete Hackathon Management Portal has been successfully implemented.
- Provides end-to-end functionality including event setup, registration, project submission, and judging.
- Ensures seamless interaction between organizers, participants, and judges.
- The judging portal enables unbiased evaluation and transparent scoring.
- Integration with Flask (backend), MongoDB (database), and Tailwind (frontend) ensures scalability and responsiveness.
- Testing with realistic test cases confirmed the system's reliability and efficiency.
- Enhances the overall hackathon process by reducing manual effort and increasing accuracy in results.
- Future enhancements can include AI-driven project analysis, automated feedback, leaderboards, and email notifications.
- The portal is a step towards digitizing hackathon management for a smoother experience.



- ◆ AI-Powered Evaluation – Use AI/ML to auto-analyze project submissions and provide instant feedback.
- ◆ Advanced Analytics – Detailed insights on participation, judging patterns, and performance metrics.
- ◆ Gamification – Add badges, points, and rewards to encourage active participation.
- ◆ Mobile Application – Extend platform accessibility through Android/iOS apps.
- ◆ Cloud Deployment – Ensure scalability for large-scale hackathons with global participation.
- ◆ Enhanced Security – Stronger authentication and secure data handling for participants' projects.
- ◆ Automated Notifications – Email/SMS alerts for deadlines, results, and updates.
- ◆ Integration with Tools – Connect with GitHub, Figma, or other collaboration platforms.



THANK YOU



PRESENTED BY: TRINITY

→ { 22F01A0504
22F01A0566
22F01A0514

