

|                      |                              |
|----------------------|------------------------------|
| <b>STUDENT NAME</b>  | PRADEEP A                    |
| <b>ROLL NO</b>       | 7376221CD131                 |
| <b>SEAT NO</b>       | 209                          |
| <b>PROJECT ID</b>    | 09                           |
| <b>PROJECT TITLE</b> | REWARD POINTS PORTAL FOR TAC |

### Implementation Timeline:

| Phase   | Deadline   | Status      | Notes                              |
|---------|------------|-------------|------------------------------------|
| Stage 1 | 03.05.2024 | Completed   | Planning and Requirement gathering |
| Stage 2 |            | On going    | Design and Prototyping             |
| Stage 3 |            | Not started | DB Designing                       |
| Stage 4 |            | Not started | Backend Implementation             |
| Stage 5 |            | Not started | Testing & Implementation           |

## **PROBLEM STATEMENT:**

Build the portal to automatically compute the final reward points of the TAC projects when the reviewers submit the review marks. The RP includes

- (i) Initial Submission
- (ii) Final Report
- (iii) Plagiarism of final report
- (iv) Review Marks
- (v) Team Communication
- (vi) Work Log

## **PURPOSE:**

Develop a Reward Point Calculation Portal for TAC (Technical Approval Committee) to streamline and automate the process of computing final reward points for projects. The portal should integrate various project evaluation criteria including Initial Submission, Final Report quality, detection of plagiarism in the final report, Review Marks provided by reviewers, effectiveness of Team Communication, and adherence to Work Log.

## **STAKEHOLDERS:**

Only the Faculty from Bannari Amman Institute of Technology with their Bitsathy mail.id can access the Website. And Some of the things can be applicable only to Admin to edit the Information.

1. TAC Administrators: Admin can view overall portal management, including project submission, review coordination, and reward point calculation.
2. Reviewers: Tasked with evaluating projects, providing Review Marks, and inputting Team Communication scores.

| S.No | Marks                   | access           |
|------|-------------------------|------------------|
| 1    | Initial Mark Submission | Admin            |
| 2    | Final Report Submission | Admin            |
| 3    | Plagiarism checker      | Admin            |
| 4    | Reviewer Mark           | Admin & Reviewer |
| 5    | Team Communication Mark | Admin & Reviewer |
| 6    | Work log                | Admin            |

### Technical Component:

Full Stack-Python Stack (AI):

|           |   |
|-----------|---|
| FRONT END | <ul style="list-style-type: none"> <li>● HTML</li> <li>● CSS</li> <li>● JS</li> </ul>                   |
| BACK END  | <ul style="list-style-type: none"> <li>● Python</li> <li>● Django (Python Web)</li> </ul>               |
| DATA BASE | <ul style="list-style-type: none"> <li>● PostgreSQL</li> <li>● MYSQL</li> </ul>                         |
| API       | <ul style="list-style-type: none"> <li>● OpenAPI</li> <li>● SOAP APIs</li> <li>● RESTFUL API</li> </ul> |

## **Functional Requirements:**

### **1.User Authentication:**

- Admin Login: Granting access to all functionalities related to Initial Submission, Final Submission, and Work Log also reviewer status.
- Reviewer Login: Enabling reviewers to input review marks and assess team communication.

### **2.Reward Point Calculation:**

- Initial Submission: Automatically assign 5 points upon submission.
- Final Report Submission: Award 10 points by default.

### **3.Plagiarism Detection:**

- Less than 30% plagiarism: Deduct 5 points.
- Less than 50% plagiarism: Deduct 3 points.
- More than 50% plagiarism: No points awarded.

**4.Reviewer Marks:** Allocate 60 marks based on evaluations (scaled from 100).

**5.Team Communication:** Allocate 10 marks based on evaluations(scaled from 30)

**6.Work Log:** Assign points based on hours logged:

- 0-10 hours: 2 points.
- 10-30 hours: 4 points.
- 30-70 hours: 8 points.
- Above 70 hours: 10 points.

## **Non-Functional Requirements:**

### **1.Security:**

- Keep user data and project info safe.
- Make sure only authorized people can access certain parts.

### **2.Usability:**

- Make it easy for people to use with clear navigation.
- Work well on different devices and screen sizes.

### **3.Performance:**

- Make it work fast.
- Test it to handle lots of people using it at once.

## MARK DETAILS:

| S.No | Mark Details   | Marks    |
|------|--|----------|
| 1.   | Initial Submission   | 5 Marks  |
| 2.   | Final Submission   | 10 Marks |
| 3.   | If any plagiarism found in final submission<br>Mark will provided by following conditions, <ul style="list-style-type: none"><li>• Less than 30% - 5 Mark</li><li>• Less than 50% - 3 Mark</li><li>• Greater than 50% - 0 Mark</li></ul> | 5 Marks  |
| 4.   | Reviewer Mark<br>(Evaluated in $10 \times 10 = 100$ & convert to 60 Mark)  | 60 Marks |
| 5.   | Team Communication Mark<br>(Evaluated in $6 \times 5 = 30$ & convert to 10 Mark)   | 10 Marks |
| 6.   | Work log, <ul style="list-style-type: none"><li>• 0 to 10 hrs - 2 Marks</li><li>• 10 to 30 hrs - 4 Marks</li><li>• 30 to 70 hrs - 8 Marks</li><li>• Above 70 hrs - 10 Marks</li></ul>  | 10 Marks |

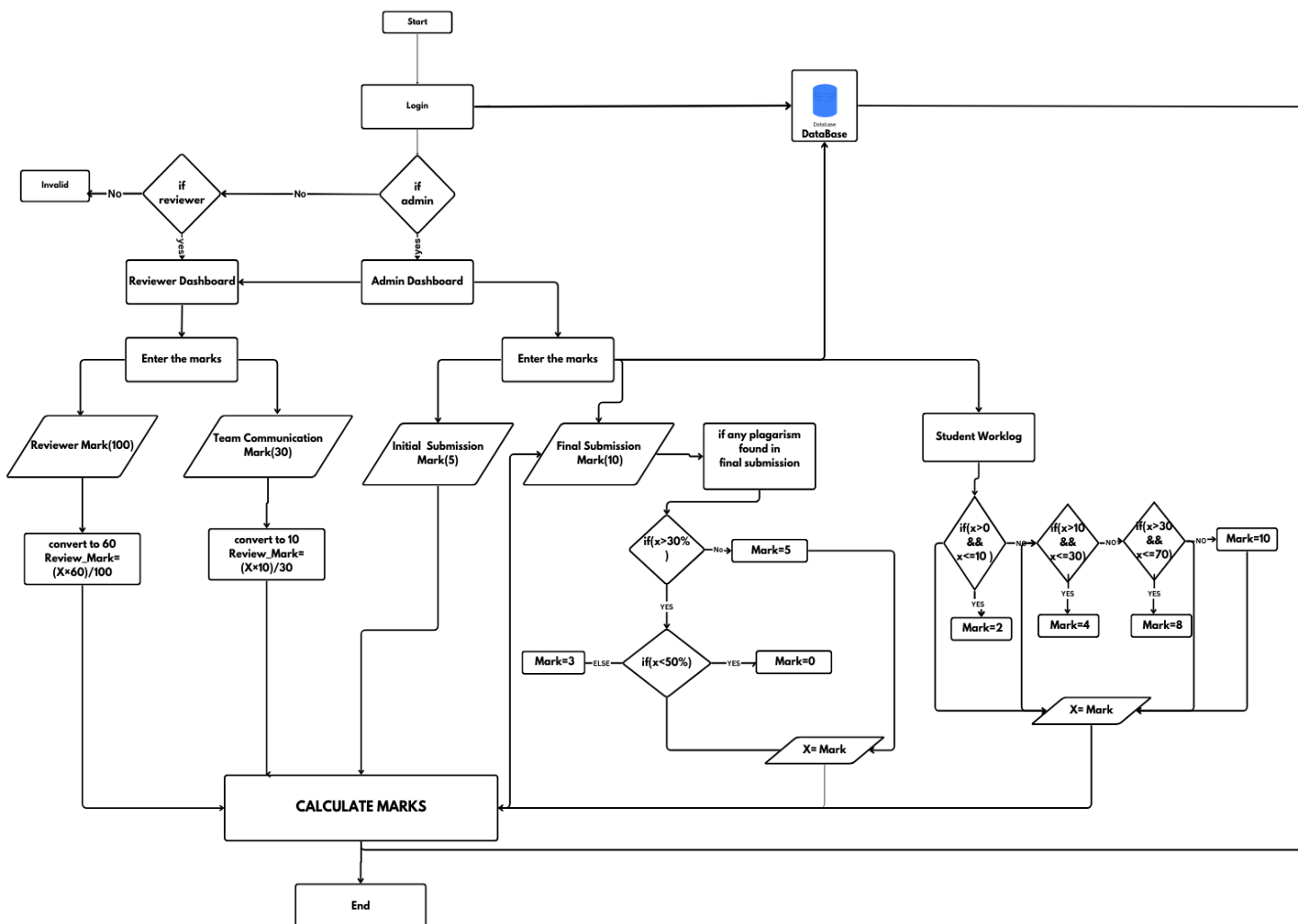
**Total -100**

## PROJECT WORKFLOW:

- **PLANNING AND REQUIREMENTS:** Understand what the client needs for the portal. Make a clear plan based on their requirements.
- **UI/UX AND PROTOTYPING:** Show the client how the portal will look and work, using tools like Figma. Get their feedback and make any needed changes.
- **FRONTEND:** Once the design is approved, start building the part of the portal that users see and interact with. Use HTML, CSS, and JavaScript to create a user-friendly experience.
- **DATABASE:** Collect and store project data and user information in a database. Choose a database system like PostgreSQL or MySQL to manage this data securely.

- **BACKEND:** Create the logic that connects the frontend with the database. Verify user credentials and manage project data. Python with Django can be used for this part.
- **API:** Set up APIs to let different parts of the portal communicate with each other. This ensures smooth interaction between features like plagiarism detection and review marks.

## FLOW CHART:



## FRONT END:

