SOFTWARE REQUIREMENT SPECIFICATION

NAME	ANTONY VIJAY S
ROLL NO	7376221CD103
SEAT NO	202
PROJECT ID	2
PROBLEM STATEMENT	STUDENT PROGRESS TRACKER

INTRODUCTION :

This project implements the idea of tracking the students right from the first year of their academics/skills till their final year, so tracking student progress effectively is crucial. Students submit proof of acquired skills and self-reporting performance. Faculty verify those submitted marks and adjust marks as needed. For administrators, a centralized dashboard offers a complete view of student progress. Filtering options and reporting tools make them to identify top performers and make decisions for sorting in company placement requirements.

SCOPE OF THE PROJECT :

Admins will have access to a centralized dashboard displaying all student details, including submitted work, instructor-verified marks, and self-reported performance. Further, the system will allow admins to check the performance of the students. This allows admins to easily identify top-ranked students based on marks. Rank students based on performance, enabling admins to quickly pinpoint top performers within the college for placement activities.

SYSTEM OVERVIEW:

USERS :

1. STUDENT:

- Students must log in using their college email IDs exclusively, ensuring database verification.
- This process guarantees that only students with valid college email addresses can access the portal, maintaining data integrity and security.
- By verifying the database, authenticate the identity of users and prevent unauthorized access.

2. FACULTY

- Faculty members must log in using their college email IDs to access the portal and verify students' marks.
- This ensures authentication and data security, as only faculty members with valid college email addresses can enter the platform.
- This verification process involves cross-checking the submitted marks with the concerned final assessments given by skill team.
- Faculty members may also have the option to adjust marks if necessary, ensuring accuracy and fairness in the evaluation process.

3. ADMIN

- Admins must log in using their college email IDs to gain access to the system, ensuring authentication and data security.
- Once logged in, admins have complete access to all the students and faculty details stored in the system.
- This includes information such as student profiles, skill records, and faculty credentials.
- Additionally, admins can filter students based on their performance metrics, such as grades or skill assessments.

FEATURES OF THE PROJECT:

1. STUDENT SUBMISSION:

- Students can upload proof of their acquired skills.
- They can prove their known skills using online courses they have completed in certified exam portals like NPTEL.
- Students can submit various proofs like,
  1. Technologies known example (blockchain, cybersecurity, etc.)
  2. Projects done
  3. Number of internships attended
  4. Ranking in coding platforms
  5. Numbers of levels in PS cleared.
- They can also self-report their performance on assessments or projects.

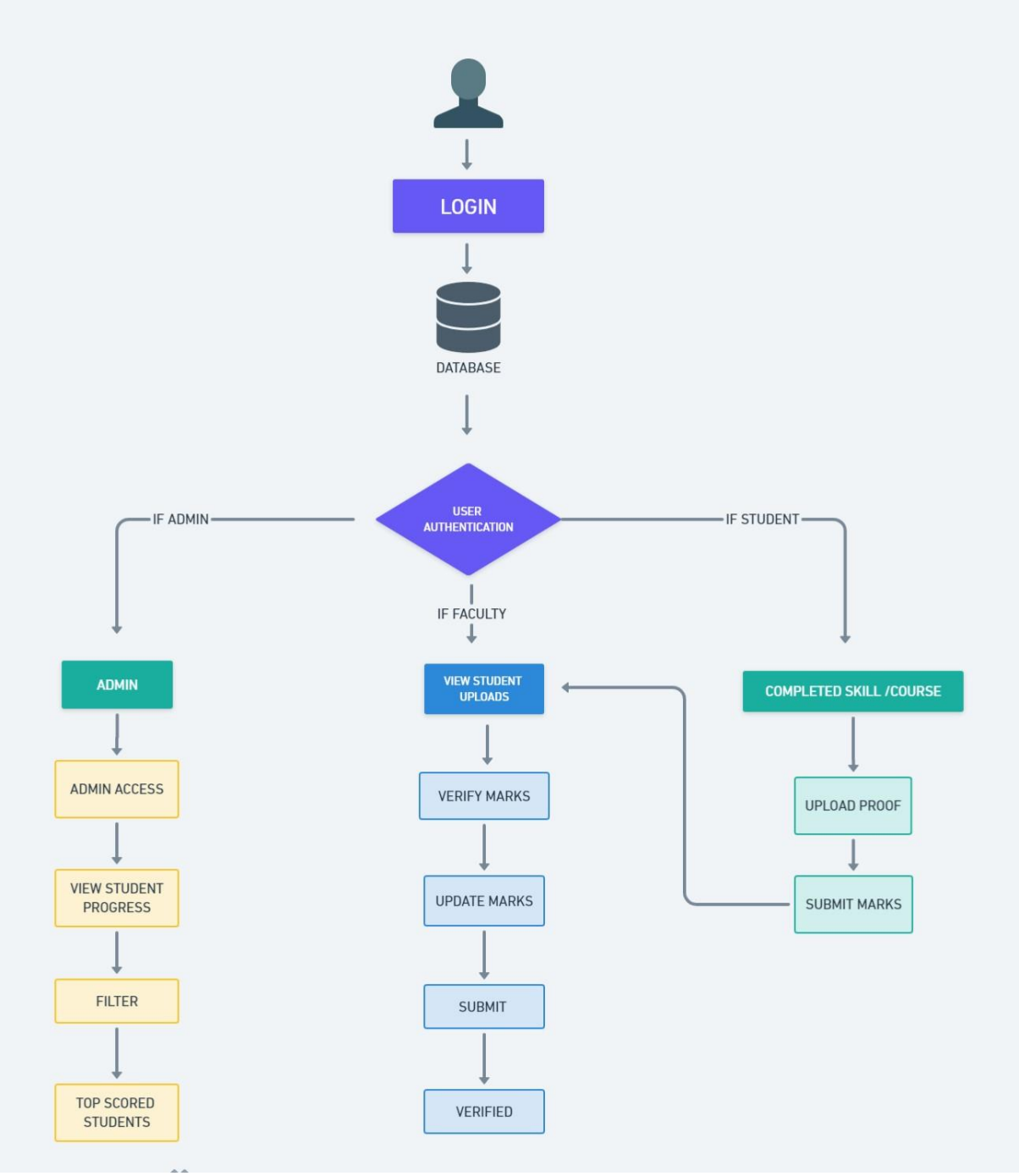
2. STAFF VERIFICATION:

Faculty members play a crucial role by reviewing the submitted proof. They can verify the authenticity of the materials and assess the student's understanding of the acquired skills. Based on their evaluation, Faculty members may adjust the self-reported marks to ensure a fair and accurate assessment.

3. ADMIN OVERSIGHT:

Administrators have access to a centralized dashboard displaying all student details. This includes submitted proof, instructor-verified marks, and self-reported performance. They can leverage filtering options to identify specific student groups or utilize reports to pinpoint top performers.

PROJECT FLOW :



FUNCTIONAL REQUIREMENTS

1. **STUDENT:**

- **Skill Proof Submission:**

  Students can upload various file formats (documents) to demonstrate the validity of learned skills. This flexibility offers them different learning factors and project types.

- **Self-Reported Performance:**

  Students can submit their performance assessment on completed tasks, and projects. This allows for initial evaluation and reflection on their Ranking progress.

2. STAFF:

- **Verification and Assessment:**

  Faculty members can review uploaded proofs to ensure authenticity and assess the student's performance of the demonstrated skills.

- **Mark Adjustment:**

  Based on their review, instructors can adjust the self-reported marks to ensure a fair and accurate assessment.

3. ADMIN:

- **Centralized Dashboard:**

  Administrators have access to the complete dashboard that displays all student details. This includes information like uploaded proof, instructor-verified marks, and self-reported performance, offering a complete view of student progress.

- **Filtering System:**

  Filtering system allows admins to identify specific student groups based on various criteria (course, skill area, performance range). This enables targeted analysis and support for different student populations.

- **Reporting Functionality:**

  The platform provides the ability to generate reports highlighting top performers across the system.

NON FUNCTIONAL REQUIREMENTS

1. SCALABILITY

- The Student Progress Tracker website is built to grow with user needs. As more students join, the website remains efficient, handling increased traffic smoothly.

- This scalability ensures the website stays reliable and responsive, no matter how many users or data it handles.

2. SECURITY

- To keep data safe, we use MongoDB to store it securely. We also use a hash algorithm to protect passwords.
- This means that passwords are converted into a special code that's hard to decode.
- By doing this, even if someone gets access to the database, they can't easily see the passwords.
- This helps to keep user accounts and their information safe from unauthorized access.

3. PERFORMANCE

- It takes a few seconds for students to be able to log in.
- To provide seamless platform access, the system handles their login request during this time.
- To execute authentication and security checks and make sure that only authorized users may access the platform, a waiting period is required.
- Students can log in and use the platform's features and services after the waiting period has ended.

STACK :

Frontend	Vue js & Quasar framework
Backend	Node js & Express js
Database	Mongo DB
Restful API	AXIOS