BANNARI AMMAN INSTITUTE OF TECHNOLOGY

An Autonomous Institution Affiliated to Anna University - Chennai, Accredited by NAAC with A+ Grade

Sathyamangalam - 638401 Erode District, Tamil Nadu, India

Stay Ahead

StudentName:SRIDHARAN G

Seat No: 197

Project ID: 37

Project title: WEB APP FOR SKILL PORTAL

Technical Components

Component	Tech Stack
Backend	Node.js, Express.js
Frontend	React
Database	MongoDB
API	RESTful services

Implementation Timeline

Phase	Deadline	Status	Notes
Stage 1	04/05/2024	Under review ▾	Planning and Requirement gathering
Stage 2		In progress ▾	Design and Prototyping
Stage 3		Not started ▾	DB Designing
Stage 4		Not started ▾	Backend Implementation
Stage 5		Not started ▾	Testing & Implementation

PROBLEM STATEMENT:

The decentralized nature of building a Skill Portal Web Application for Educational Institutions

1. **Lack of Centralization:** Educational institutions lack a centralized system for students to register and verify their skill choices. This decentralization leads to inefficiencies and inconsistencies in skill management.
2. **Verification Process**: Without a dedicated portal, verifying students' skill choices against their previous semester skills becomes cumbersome and prone to errors. Manual verification processes are time-consuming and resource-intensive.
3. **Limited Access to Skill Information**: Students often face difficulty accessing comprehensive information about available skills, prerequisites, and related courses. This lack of transparency hampers informed decision-making regarding skill selection.

PROJECT-FLOW:

Purpose:

To develop a well-defined web application for registering the skill and efficiently tracking the attendance and mark updating. And other activities like visualizing the attendance percentage on the student dashboard and it ensures that the selected skill is not previously registered by the respective student. The collection of feedback from the students is also helps the team to paramount some changes in the particular system.

Scope:

The system includes the user authentication for student, staff and admin. And the instant updating of mark and attendance. This web app ensures that all the work is efficiently done in one place. The application will automate the verification process by cross-referencing students' skill choices with their previous semester skills, minimizing errors and streamlining administrative tasks. The web application will enable institutions to track students' skill progression over time, facilitating data-driven decision-making and personalized educational experiences.

Business Context:

The centralized web application for the skill portal is enhancing communication clarity and timeliness across BIT. A well-designed skill portal can encourage students to engage more deeply with their studies and skill activities. Thus, boosting organizational efficiency by minimizing scheduling conflicts. Primary stakeholders include students, faculty, administrative staff, and the IT department.

Consideration:

- All users possess active Google accounts for authentication.

- Users have regular access to internet-enabled devices.

Dependencies:

- Integration with Google OAuth for user authentication.
- Database system that aligns with your requirements in terms of scalability, performance, and data integrity in updating.

User personas:

- **Student:** Needs an up-to-date schedule to effectively plan activities and balancing skill-building with academic coursework.
- **Faculty:** Requires the ability to contribute to the skill portal by creating and sharing course materials tailored to student needs.
- **Admin Staff:** Manages system operations, resolves conflicts and assigning the course to the staffs.

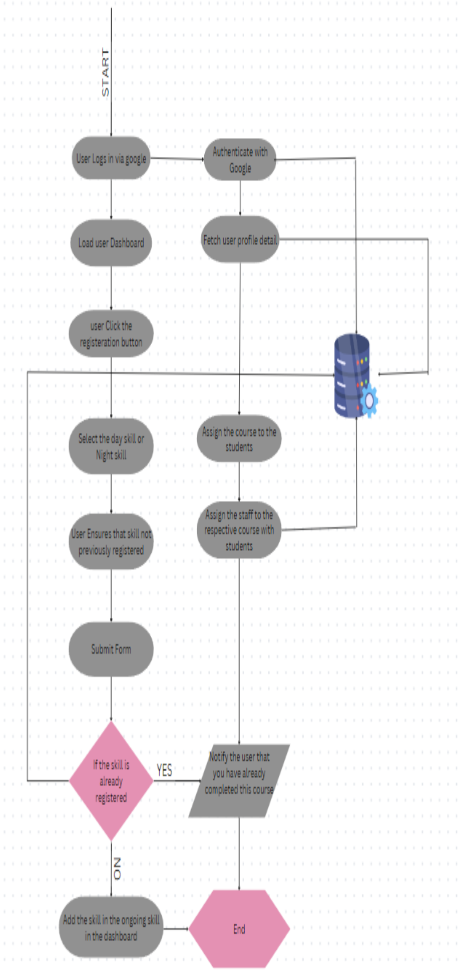
User Stories:

- As a student, I want view all the data based on my skill development and also, I need to know development in my skills so this makes me to work on my skill development effectively.
- As a faculty member, I need to validate the day assessments and updating the attendance and the marks to the student. I need to ensure that the attendance percentage is greater than 8O percentage.

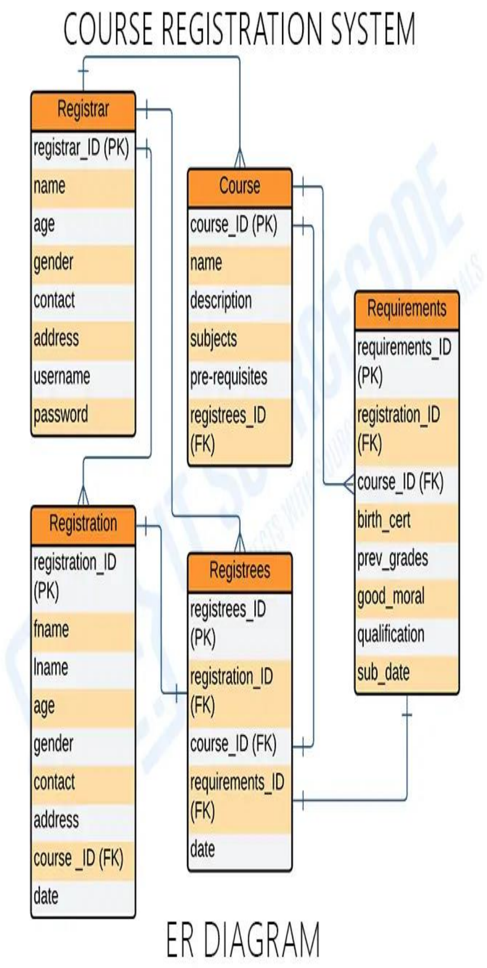
Functional Requirements:

- **User Authentication:** Secure login using Google OAuth.
- **Course Management**: Students should be able to search for and enrolled in courses, view course details, and access course materials
- **Schedule Management**: Faculty should be able to manage course schedules, update class timings, and send schedule updates and notices to students.
- **Dynamic Dashboard:** Real-time schedule viewing and interaction.

FLOW CHART:


START
User Logs in via google
Authenticate with Google
Load user Dashboard
Fetch user profile detail
user Click the registeration button
Select the day skill or Night skill
Assign the course to the students
User Ensures that skill not previously registered
Assign the staff to the respective course with students
Submit Form
If the skill is already registered
YES
Notify the user that you have already completed this course
NO
Add the skill in the ongoing skill in the dashboard
End

ER DIAGRAM (DB DESIGN):



WORK FOLW

1.User Authentication

1. User registers or logs in through the React frontend

2. Node.js backend handles authentication using Passport.js or similar library

3. User data is stored in MongoDB

2.Skill Management

1. User views available skills through the React frontend

2. Node.js backend retrieves available skills from MongoDB using Express.js API routes

3. Skills that the user has already registered for are hidden or marked as "Already Registered"

3.Skill Registration

1. User registers for a skill through the React frontend

2. Node.js backend checks if the user has already registered for the skill

   - If yes, display an error message or hide the skill

   - If no, add the skill to the user's registered skills in MongoDB

3. Update the user's registered skills in MongoDB

4.Skill Assessment

1. User takes a skill assessment through the React frontend

2. Node.js backend handles assessment logic using Express.js API routes

3. Assessment results are stored in MongoDB

5.Skill Matching

1. User's skills are matched with job requirements or learning resources

2. Node.js backend handles matching logic using Express.js API routes

3. Matching results are displayed in the React frontend

6.Hide Already Registered Skills

1. In the Node.js backend, retrieve the user's registered skills from MongoDB

2. When retrieving available skills, check if the skill is already registered by the user

3. If yes, hide the skill or mark it as "Already Registered"

7.React Frontend

1. Display available skills to the user

2. Use conditional statements to hide skills that the user has already registered for

3. Display error messages or notifications if the user tries to register for a skill they already have

7.MongoDB Schema

1. Users collection:

   - _id

   - name

   - email

   - registered Skills (array of skill IDs)

2. Skills collection:

   - _id

   - name

   - description

8.Express.js API Routes

1. GET /skills: Retrieve available skills

2. POST /skills/register: Register for a skill

3. GET /skills/assessment: Take a skill assessment

4. GET /skills/matching: Get skill matching results