**SKILL MATRIX**

Final Project – Ashwati Karunanidhi

Contents

[1 Project Overview 2](#_Toc137748514)

[1.1.1 Brief introduction of the project. 2](#_Toc137748515)

[1.1.2 Purpose of the project. 2](#_Toc137748516)

[1.1.3 Goals of the project: 2](#_Toc137748517)

[2 Project Scope 3](#_Toc137748518)

[2.1.1 Key features and functionalities of the mobile app. 3](#_Toc137748520)

[3 Architecture and Technology Stack 3](#_Toc137748521)

[3.1.1 Overall architecture of the website. 3](#_Toc137748522)

[3.1.2 Architecture 4](#_Toc137748523)

[3.1.3 Technology stack (e.g., programming languages, frameworks, libraries). 4](#_Toc137748524)

[3.1.4 Rationale behind the chosen technology stack. 5](#_Toc137748525)

[4 Mobile App Components 6](#_Toc137748526)

[4.1.1 Main components of the mobile app. 6](#_Toc137748527)

[4.1.2 Purpose of each component. 6](#_Toc137748528)

[5 User Interface Design 8](#_Toc137748529)

[5.1.1 User interface (UI) design approach. 8](#_Toc137748530)

[6 Testing and Quality Assurance 12](#_Toc137748531)

[6.1.1 Testing approach for the mobile app. 12](#_Toc137748532)

[6.1.2 Types of testing to be performed. 13](#_Toc137748533)

[6.1.3 Quality assurance processes and tools to ensure app reliability. 14](#_Toc137748534)

[7 Project Timeline and Resources 15](#_Toc137748535)

[7.1.1 Estimated project timeline, including major milestones. 15](#_Toc137748536)

# Project Overview

### Brief introduction of the project.

Skill Matrix is a MERN stack-based platform facilitating efficient management of user skills and projects. Admins can add users, who then independently input their skills and projects. Admin approval transitions entries from pending to approved status. Data engineering involves migrating MongoDB data to Snowflake via DBT for enhanced data processing. Subsequently, data science techniques such as EDA, feature engineering, model development, and explainability are applied for insights generation.

### Purpose of the project.

The purpose of Skill Matrix is to streamline talent management processes within organizations, enabling efficient tracking and utilization of employee skills and project contributions. The key goals of the project include:

* Improving Talent Management: By providing a centralized platform for users to input their skills and projects, Skill Matrix aims to enhance talent management processes within organizations. Admins can efficiently oversee and approve user submissions, ensuring accurate and up-to-date skill profiles for each employee.
* Facilitating Skill Visibility: Skill Matrix empowers employees to showcase their skills and project contributions, fostering a culture of transparency and collaboration within the organization. Users can easily browse and search for colleagues with specific skills, facilitating skill-sharing and knowledge transfer.
* Enhancing Decision-Making: With access to comprehensive skill and project data, organizations can make more informed decisions regarding resource allocation, project staffing, and skills development initiatives. Skill Matrix provides valuable insights into the skillsets available within the organization, enabling strategic workforce planning and optimization.

### Goals of the project:

* UI/UX Design: Craft intuitive UI/UX components for Skill Matrix, in line with project specifications.
* MVP Development: Create a Minimum Viable Product (MVP) based on scoped features from requirements analysis.
* Core Application Design: Develop front-end and integrate provided back-end API for seamless user experience.
* Deployment to Data Platforms: Implement Skill Matrix on designated data engineering and data science platforms for internal use.
* QA and UAT: Conduct rigorous QA testing and user acceptance testing (UAT) to ensure functionality and address any issues promptly.

# Project Scope

### 2.1 Key features and functionalities of the mobile app.

**User Management:**

* Admin interface for adding new users and managing user roles and permissions.
* User registration process with authentication mechanisms such as username/password and possibly other security features like multi-factor authentication.

**Skill and Project Submission:**

* User interface for users to input their skills and project details separately.
* Submission forms with validation to ensure accurate data entry.

**Approval Workflow:**

* Admin approval process for submitted skills and projects, transitioning entries from pending to approved status.
* Notifications to users regarding approval status updates.

**Data Integration:**

* Integration with external APIs or databases for importing or exporting data related to skills and projects.
* Seamless communication between the Skill Matrix platform and external systems for data synchronization and workflow automation.

**Responsive Design:**

* Mobile-friendly design ensuring compatibility across various devices and screen sizes.
* Responsive user interface for optimal viewing and usability on desktops, laptops, and tablets.

# Architecture and Technology Stack

### Overall architecture of the website.

**Client-Side Interface:**

* The client-side interface serves as the user-facing component of the web application. It provides an intuitive and responsive user interface, allowing users to interact with the platform. The interface supports various devices and screen sizes, ensuring a consistent experience across different platforms.

**Application Logic Layer:**

* The application logic layer forms the core of the Skill Matrix application, encompassing the business logic and functionality. It processes user requests, manages data retrieval and storage, and performs necessary calculations and validations. This layer ensures the seamless execution of operations within the application.

**Database Management:**

* The database management component handles the storage and retrieval of data within the Skill Matrix application. It includes databases for storing user information, skills, projects, and approval statuses. The database management system ensures data integrity, security, and efficient access to information.

**Server-Side Processing:**

* The server-side processing component manages the communication between the client-side interface and the application logic layer. It handles incoming requests from users, processes them through the application logic, and generates responses to be sent back to the client. This component ensures the efficient flow of data and operations within the application.

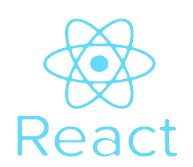
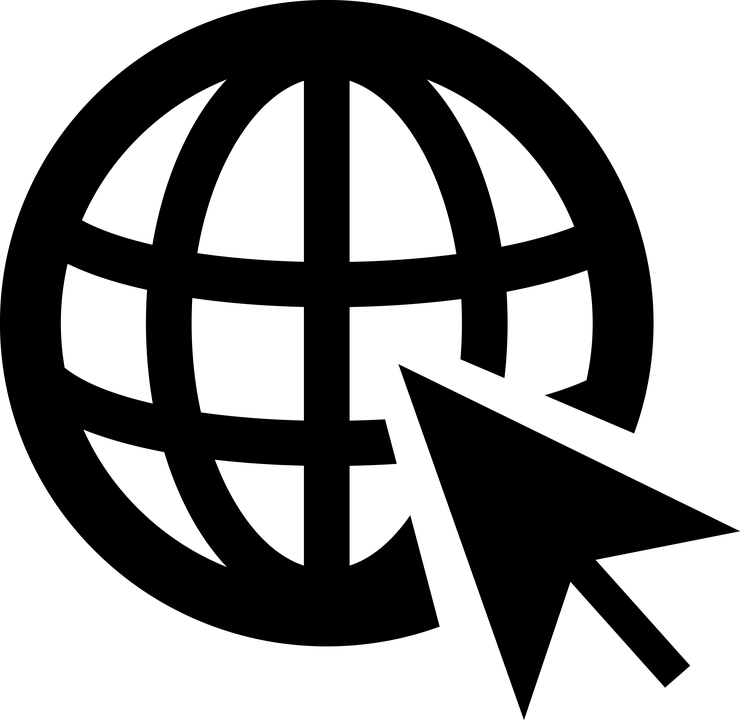
**Security and Authentication:**

* Security and authentication mechanisms are integrated throughout the Skill Matrix application to ensure the protection of user data and secure access to the platform. This includes features such as user authentication, encryption of sensitive information, and role-based access control to safeguard the application against unauthorized access and data breaches.

### Architecture

**BACKEND and DATABASE**

**FRONTEND**



**API**



API interceptor – Axios

Nodemailer – Mail management

### Technology stack (e.g., programming languages, frameworks, libraries).

React:

* React is a JavaScript library for building user interfaces, providing a component-based approach for creating interactive web applications.

Node.js:

* Node.js is a JavaScript runtime environment that allows for server-side development, providing an event-driven architecture for building scalable and efficient web servers.

Express.js:

* Express.js is a web application framework for Node.js, providing a minimalist and flexible approach to building web applications and APIs.

MongoDB:

* MongoDB is a NoSQL database system that stores data in flexible, JSON-like documents, making it suitable for handling unstructured or semi-structured data.

Mongoose:

* Mongoose is an object modelling tool for MongoDB and Node.js, providing a schema-based solution for modelling application data and simplifying interactions with the MongoDB database.

React Router:

* React Router is a routing library for React applications, enabling navigation between different components and managing application URLs.

Bootstrap or Material-UI:

* Bootstrap or Material-UI can be used as UI libraries to provide pre-designed components and styles for creating responsive and visually appealing user interfaces.

Nodemailer:

A module for Node.js applications that enables email sending functionality, used for features like password reset in the Skill Matrix application.

### Rationale behind the chosen technology stack.

React:

* React was selected as the frontend technology due to its component-based architecture, facilitating code reusability, efficient UI management, and easy maintenance. Its virtual DOM implementation ensures fast rendering and updates, enhancing the application's performance.

Node.js and Express.js:

* Node.js and Express.js were chosen for the backend to leverage JavaScript across the full stack, enabling seamless communication between the frontend and backend components. Node.js offers non-blocking I/O operations, enhancing scalability and performance, while Express.js provides a minimalist framework for building robust APIs.

MongoDB and Mongoose:

* MongoDB, a NoSQL database, was chosen for its flexibility, scalability, and compatibility with JavaScript-based development. Mongoose serves as an object modelling tool, simplifying interactions with MongoDB and ensuring data integrity through schema-based validation.

Nodemailer:

* Nodemailer was integrated into the stack to facilitate email sending functionality, such as password reset and account verification. Its ease of use, robustness, and support for various email providers make it a suitable choice for handling email communications within the application.

This technology stack offers a cohesive and scalable solution for building the Skill Matrix application, leveraging the strengths of each technology to ensure optimal performance, security, and maintainability.

# Web app Components

### Main components of the app.

Main components of the Skill Matrix application:

* React.js
* Node.js
* Express.js
* MongoDB
* Mongoose

Additional Libraries and Dependencies:

* Axios
* React Router
* DBT
* React Modal
* React Dropdown
* Nodemailer
* Snowflake connector

### Purpose of each component.

Main components of the Skill Matrix application:

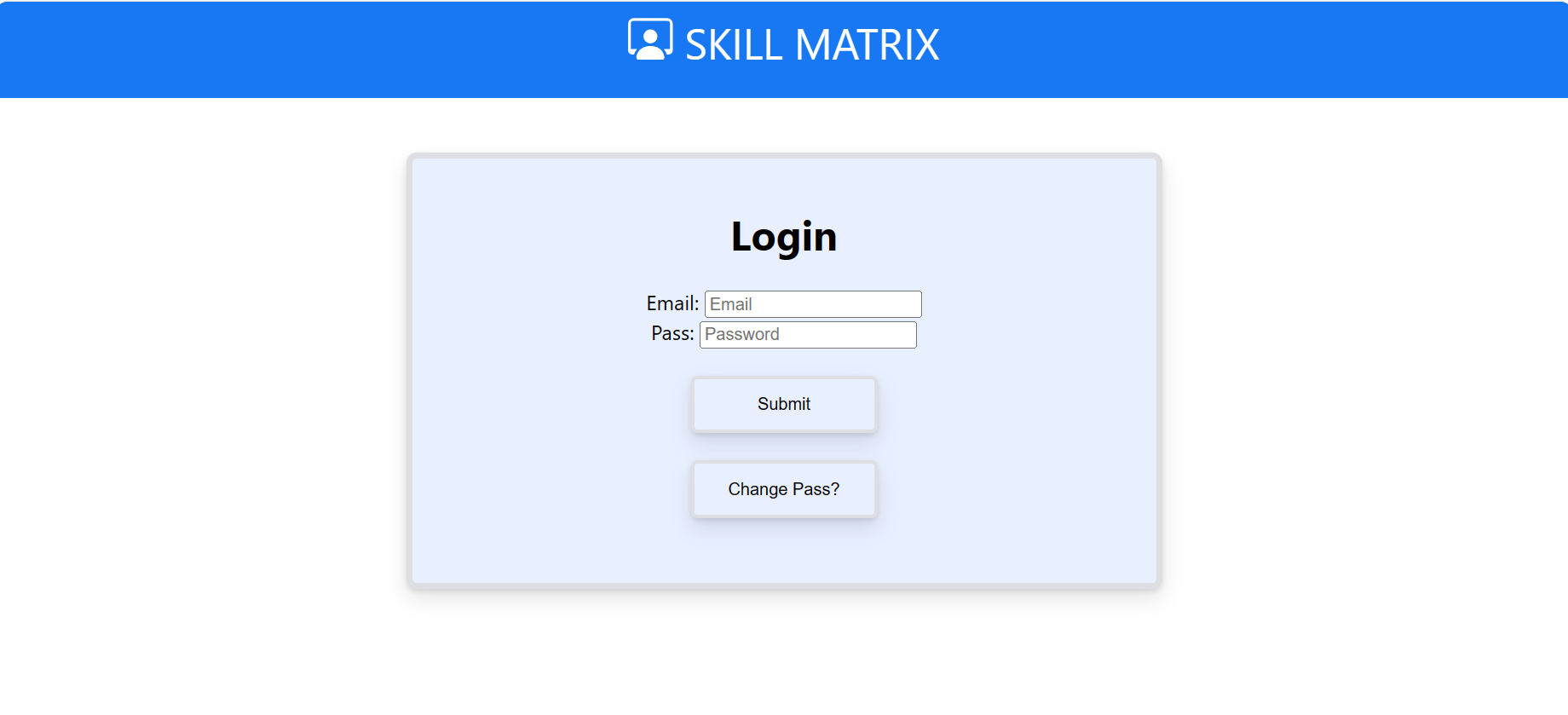
* React.js: JavaScript library for building user interfaces.
* Node.js: JavaScript runtime for server-side development.
* Express.js: Web application framework for Node.js.
* MongoDB: NoSQL database for data storage.
* Mongoose: Object modelling tool for MongoDB and Node.js.

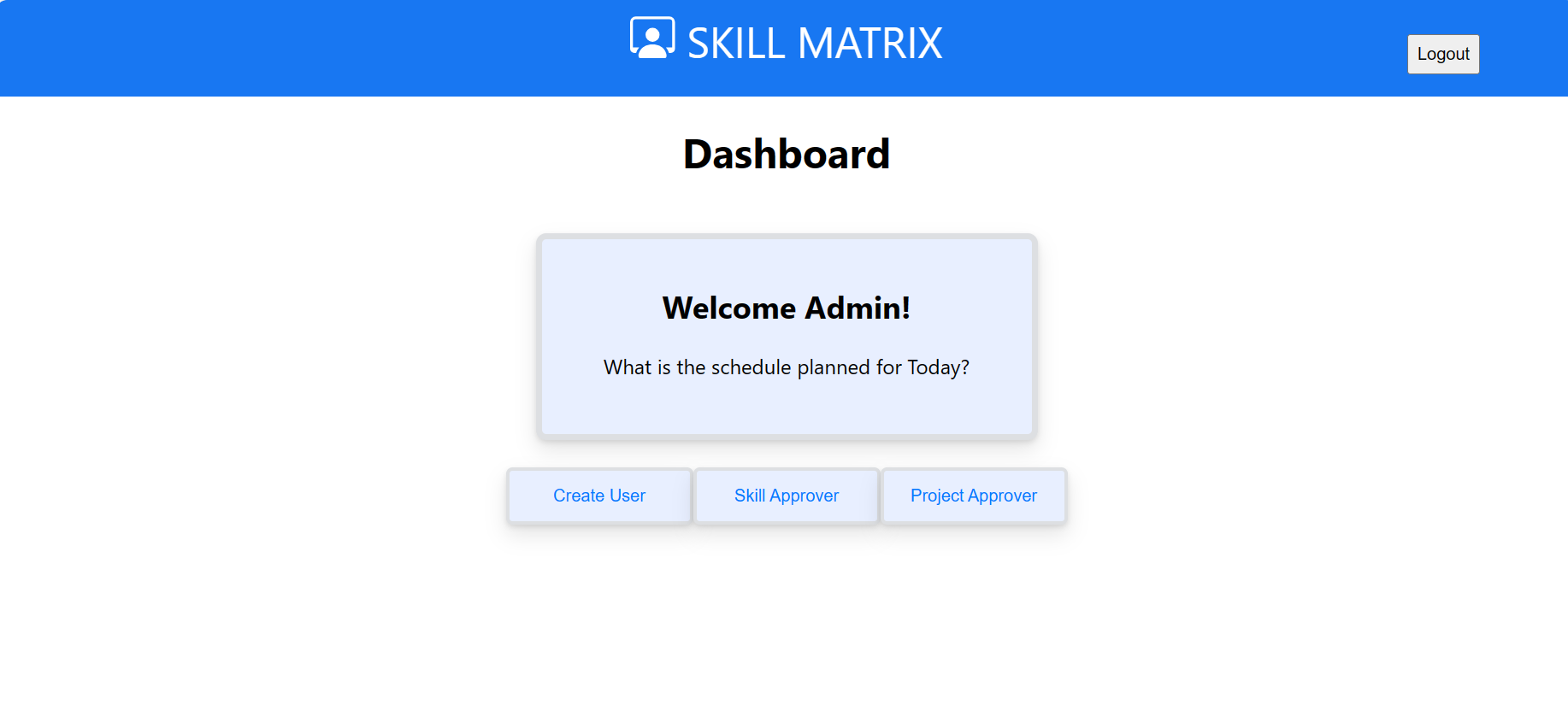
Additional Libraries and Dependencies:

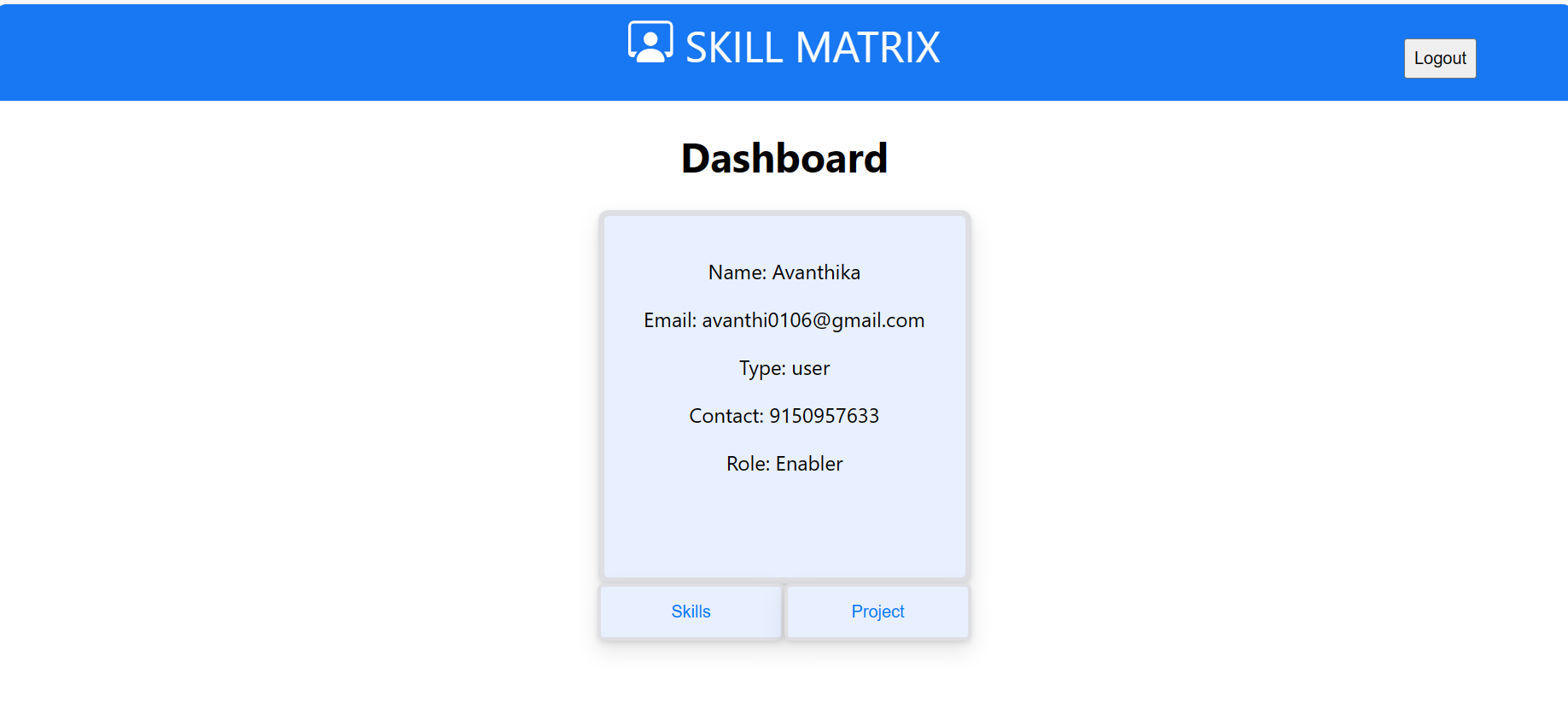
* Axios: JavaScript library for making HTTP requests.
* React Router: Routing library for React applications.
* DBT: Preprocessing the staging and mart layers of data.
* React Modal: Library for creating modal dialogs in React applications.
* React Dropdown: Component for creating dropdowns in React applications.
* Nodemailer: Triggering emails in Node applications.
* Snowflake connector: Establishing the connection between DBT and Snowflake.

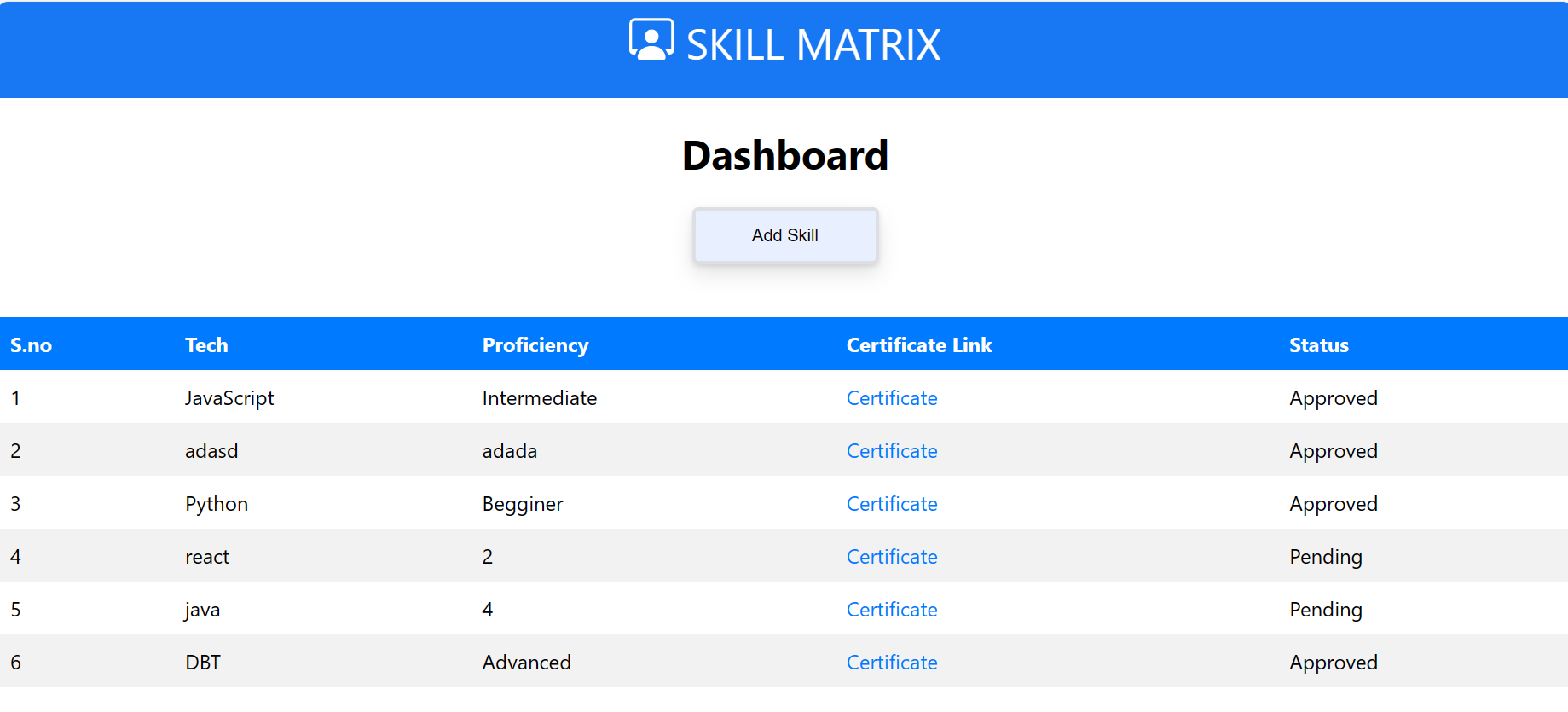
# User Interface Design

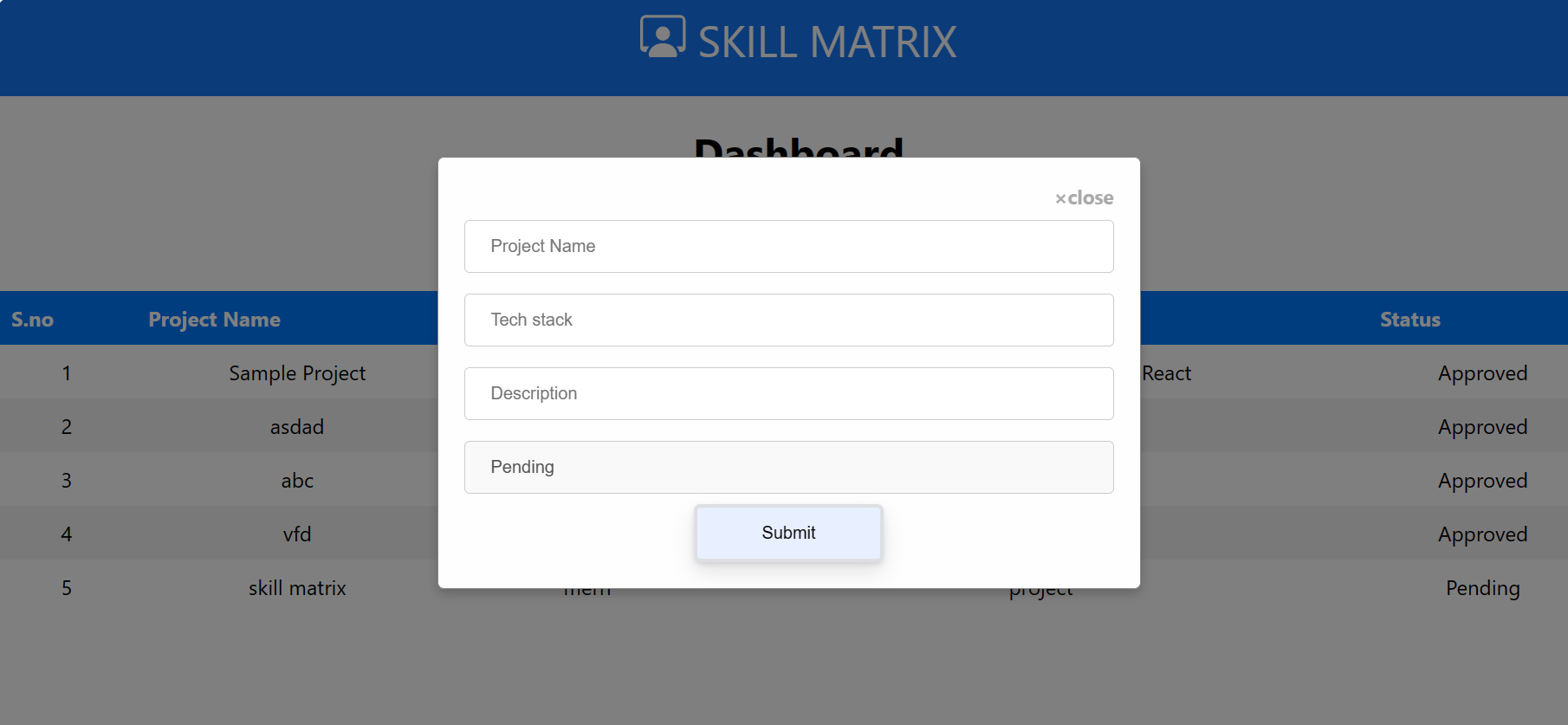
### User interface (UI) design approach.











# Testing and Quality Assurance

Testing and quality assurance are essential processes in software development that aim to ensure the reliability, functionality, and overall quality of a software product. These processes involve systematically examining and validating the software at various stages to identify defects, bugs, and any deviations from the expected behaviour. By conducting thorough testing and quality assurance, developers can uncover and rectify issues early in the development cycle, enhancing user satisfaction, minimizing risks, and ultimately delivering a stable and dependable software product to the end users.

### Testing approach for the app.

Requirement Analysis:

* Thoroughly understand the requirements of the Skill Matrix application, including its functionality, user roles, and data management needs.

Test Planning:

* Develop a comprehensive test plan outlining the testing objectives, scope, environments, test cases, and techniques. Determine the types of testing to be performed, such as functional testing, usability testing, and performance testing.

Test Environment Setup:

* Set up the necessary test environments, including development, staging, and production environments. Ensure that the test environment closely mimics the production environment to accurately simulate real-world usage scenarios.

Functional Testing:

* Verify that the Skill Matrix application meets its functional requirements. Test various features and functionalities, including user authentication, user management, skill and project submission, approval workflow, and data integration.

User Interface Testing:

* Test the user interface (UI) of the Skill Matrix application to ensure consistency, responsiveness, and usability across different devices and screen sizes. Check for proper alignment of elements, accurate rendering of fonts and images, and intuitive navigation.

Integration Testing:

* Validate the integration between different components of the Skill Matrix application, including frontend and backend systems, databases, and external APIs. Ensure seamless communication and data exchange between these components.

Performance Testing:

* Assess the performance of the Skill Matrix application under various load conditions to identify any bottlenecks or performance issues. Measure response times, throughput, and resource utilization to ensure optimal performance and scalability.

User Acceptance Testing (UAT):

* Involve end-users in the testing process to validate the functionality, usability, and overall satisfaction with the Skill Matrix application. Gather feedback and address any issues identified during UAT to ensure a successful deployment.

By following this testing approach, the Skill Matrix application can be thoroughly evaluated for quality, functionality, performance, and security, ensuring a robust and reliable solution for users.

### Types of testing to be performed.

Functional Testing:

* Verify that each function in the application behaves as expected.
* Ensure that all requirements are met and that all functionalities work properly.

Usability Testing:

* Assess user satisfaction, navigation, intuitiveness, and learnability of the application.
* Identify areas where users may face difficulties or frustrations.
* Evaluate error frequency and severity to enhance user experience.

System Testing:

* Evaluate the entire system's compliance against specified requirements.
* Perform end-to-end testing to ensure proper functionality across the application.

Retesting:

* Re-execute previously failed tests against new software to verify if issues are resolved.
* Verify that problems identified in earlier testing phases have been adequately addressed.

Regression Testing:

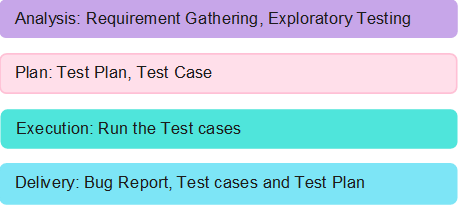
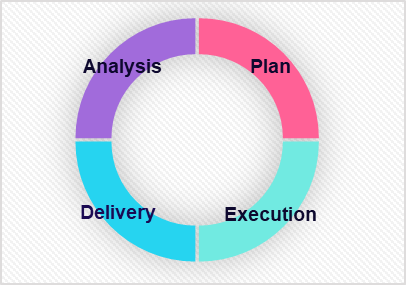
* Verify that existing features continue to function correctly after changes or additions to the application.
* Ensure that fixes do not adversely impact the existing functionality.

Compatibility Testing:

* Check if the application functions correctly on different devices, operating systems, and screen sizes.
* Identify and resolve compatibility issues early to minimize user complaints and negative reviews.
* Ensure a seamless user experience across various platforms to enhance customer satisfaction and market acceptance.

These types of testing ensure that the Skill Matrix application meets quality standards, functions properly across various scenarios, and provides an optimal user experience for its intended audience.

### Quality assurance processes and tools to ensure app reliability.

The objective of the test is to define the goals and purpose of the testing effort. It aims to provide a comprehensive and focused statement of what is to be accomplished through testing. The test objectives serve as a guiding principle for the testing activities and ensure that they are aligned with the overall objectives of the project  
  


# Timeline and Resources

### Estimated project timeline, including major milestones.

|  |  |  |  |
| --- | --- | --- | --- |
| TASKS | W1 | W2 | W3 |
| Wireframe and requirement gathering |  |  |  |
| Login and Signup page creation |  |  |  |
| User and Admin Dashboard creation |  |  |  |
| Add Skills and Project Feature |  |  |  |
| Connect to Database and Warehouse |  |  |  |
| Data Engineering for the project |  |  |  |
| Model development for the problem |  |  |  |
| UI/UX design of the website |  |  |  |

**Appendix Title**

Document Title