**SKILL MATRIX SYSTEM**

Final Project

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 2](#_Toc137748518)

[2.1.1 Key features and functionalities of the Web Application. 2](#_Toc137748520)

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

[3.1.1 Overall architecture of the Web Application 3](#_Toc137748522)

[3.1.2 Architecture 4](#_Toc137748523)

[3.1.3 Technology Stack: 4](#_Toc137748517)

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

[4 User Interface Design 6](#_Toc137748529)

[4.1.1 User interface (UI) design approach. 7](#_Toc137748530)

[5 Data Engineering 9](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464944)

[5.1.1 Components 9](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464945)

[5.1.2 PostgreSQL to Snowflake Data Transfer 10](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464946)

[5.1.3 Configuring DBT for Staging and Tranformations 10](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464946)

[5.1.4 Load Transformed Data into Snowflake 10](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464946)

[6 Data Science 11](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464944)

[6.1.1 EDA (Exploratory Data Analysis) 11](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464946)

[6.1.2 Feature Engineering 11](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464946)

[6.1.3 Model Development 11](file:///C:\Users\friends%20mobiles\Downloads\Major%20project%20report%201.docx#_Toc160464946)

7 Timeline 12

# Project Overview

### Brief introduction of the project.

This project aims to create a simple Skill Matrix System to manage employee skills, certifications, and project experience. It includes features like a login page, user creation, tech stack details input, and email notifications for updates.

### Purpose of the project.

The purpose of this project is to enhance organizational efficiency and talent management and Centralize and automate the management of employee skills, certifications, and project experience

The key goals of the project include:

* Efficient Skill Management: The project aims to streamline the management of employee skills, enabling better resource allocation, project planning, and talent development initiatives across the organization..
* Certification Tracking: By providing a centralized system for tracking and managing employee certifications, the project ensures that employees maintain relevant qualifications and compliance with industry standards or regulatory requirements.
* Project Experience Management : The system facilitates the tracking of employees' project experience, allowing for better utilization of expertise and more informed project assignments. It also aids in identifying skill gaps and areas for professional development.

### Goals of the project:

* The overarching goal of the Skill Matrix System project is to modernize and optimize talent management practices within the organization by implementing a comprehensive platform for managing employee skills, certifications, and project experience.
* Enable managers to identify skill gaps and opportunities for professional development among employees, fostering a culture of continuous learning and growth.
* Utilize skill data to effectively assign employees to projects based on their expertise and experience, maximizing project success rates and resource utilization.
* Empower employees to take ownership of their skill development by providing visibility into their skill profiles and opportunities for advancement within the organization.

# Project Scope

### Key features and functionalities of the Web Application.

**User Authentication:**

Provide a secure login page for administrators to access the system.

Authentication mechanism to verify admin credentials before granting access.

**User Management**:

Admin-only access to create, view, and manage user profiles.

View user details, including basic information like name, role, and contact details.

**User Skills:**

User interface for users to upload and manage their skills, certifications, and projects.

Users can upload details such as skill names, proficiency levels, certification names, expiration dates, project titles, roles, and durations.

**Approver Role:**

Designate certain users as approvers who can review and approve/reject the skills, certifications, and projects uploaded by other users

Approval workflow to ensure that uploaded skills, certifications, and projects undergo evaluation before being accepted into the system.

**Admin View:**

Admin dashboard displaying a list of users and their uploaded skills, certifications, and projects.   
Detailed view for admins to review individual user profiles and associated details.

**Notifications:**

Automated email notifications to inform users when their skills, certifications, or projects have been approved or rejected by the approver.

# Architecture and Technology Stack

### Overall architecture of the Web Application.

The Web Application used to manages the abstraction of user profiles, skill sets, certifications, and project experience data. This layer ensures that data is represented in a uniform and structured manner across the application.

**Client-Side Interface:**

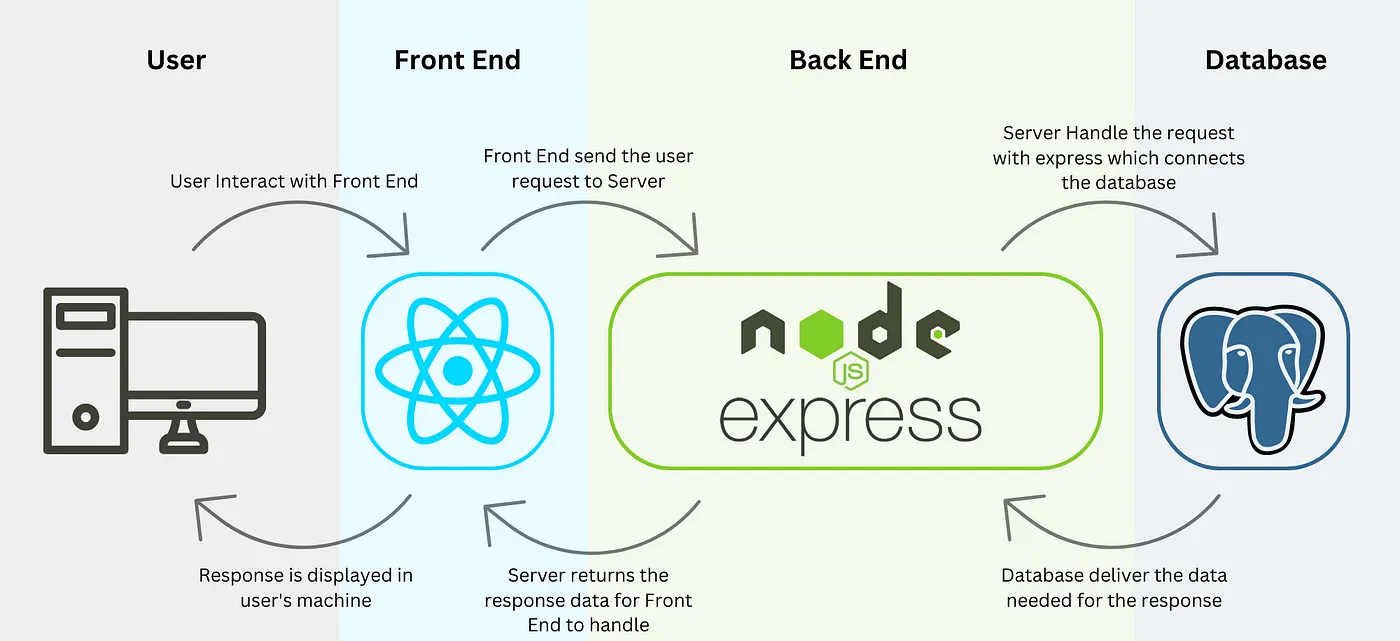
Develops an intuitive and user-friendly interface for users to interact with the Skill Matrix System. This includes dashboards, forms, and visualizations for managing skills, certifications, and projects

Ensures compatibility with various client devices and platforms, such as web browsers on desktops, laptops, and mobile devices

**Application Logic Layer:**

The Application Logic Layer forms the core functionality of the Skill Matrix System, orchestrating user interactions, data processing, and system workflows to ensure efficient management of employee skills, certifications, and project experience.

**3.1.2 Architecture**

**FULL STACK ARCHITECTURE** 

**DATA ENGINEERING ARCHITECTURE**

A diagram of a data processing process

Description automatically generated

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

**Technology Stack for the Web Application:**

The technology stack for the Web Application for Skill Tracking in Skill Matrix System will include the following components:

**PostgresSql**:

PostgreSQL is a powerful, open-source relational database management system (RDBMS) known for its reliability, robust feature set, and extensibility.It offers support for complex queries, transactions, data integrity constraints, and advanced data types like JSONB for storing semi-structured data.PostgreSQL is widely used in production environments for various applications ranging from small-scale projects to large-scale enterprise systems.

**Express:**

Express.js is a minimalist web application framework for Node.js, designed for building web applications and APIs.It provides a robust set of features for handling HTTP requests, routing, middleware, and template engines.Express simplifies the process of building server-side logic and RESTful APIs, allowing developers to focus on business logic rather than low-level details.

**React :**React is a popular JavaScript library for building user interfaces, developed by Facebook.It allows developers to create reusable UI components and build complex UIs efficiently through its component-based architecture.React's virtual DOM (Document Object Model) and reconciliation algorithm optimize performance by minimizing DOM updates.It supports a declarative programming style, making it easier to understand and maintain code.

**Axios:**

Axios is a widely used JavaScript library for making HTTP requests from the application to the server. It simplifies the process of sending and receiving data, handling API calls, and managing network requests in the mobile app.

**Nodemailer:**

Nodemailer is a module for Node.js applications that allows sending emails.It supports various transport methods such as SMTP, Sendmail, and Amazon SES, making it flexible for different email sending needs. Nodemailer simplifies the process of sending transactional emails, notifications, and other types of emails from your Node.js application.

**JWTToken:**

JWT (JSON Web Token) is an open standard for securely transmitting information between parties as a JSON object.It consists of three parts: a header, a payload, and a signature, which are encoded and can be digitally signed to verify the integrity of the token.JWT tokens are commonly used for authentication and authorization in web applications, allowing users to securely access protected resources by exchanging tokens.

### Rationale behind the chosen technology stack.

In developing the Mobile Application Platform for Instrument Tracking in Healthcare, we have chosen React Native as the primary technology stack. The rationale behind this selection is based on the following considerations:

**Cross-Platform Compatibility:**

Choose technologies that allow for cross-platform development to ensure the Skill Matrix System can be accessed from various devices and operating systems.Consider frameworks like React Native for mobile applications or web technologies like React.js for browser-based access

**Time and Cost Efficiency:**

Opt for technologies that streamline development efforts and minimize duplication of work.Use frameworks or libraries that enable code reusability across different platforms or components within the system.

**Native-Like Performance:**

Select technologies that provide smooth performance and responsiveness, ensuring a seamless user experience.Look for frameworks or tools that leverage native components or optimize performance for the target platforms.

**Large Developer Community and Ecosystem:**

Prioritize technologies with active communities and extensive support resources.

Choose frameworks or platforms with a wide range of libraries and components that facilitate development and problem-solving.

**Code Reusability:**

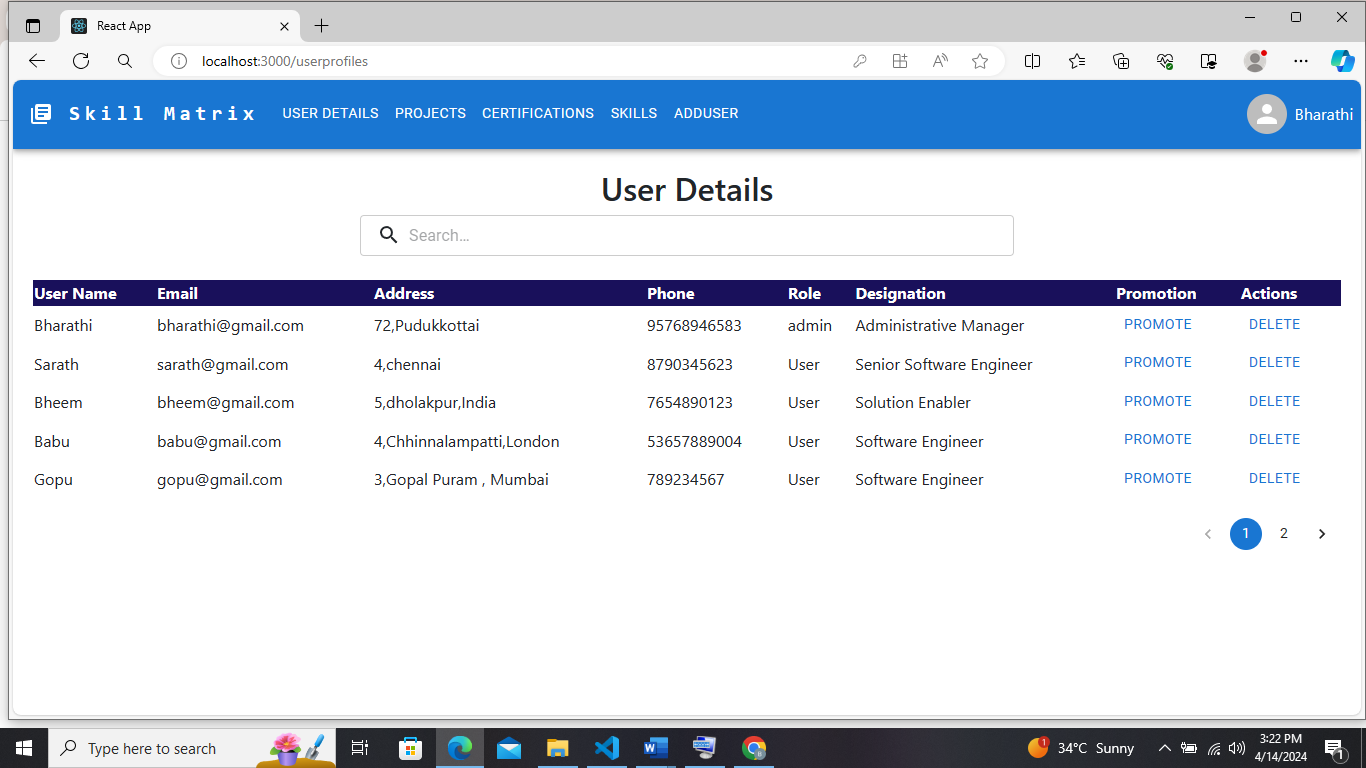
Emphasize technologies with component-based architectures that promote code reusability.Select frameworks or libraries that allow for the efficient management and reuse of UI elements and logic across different parts of the system.

# User Interface Design

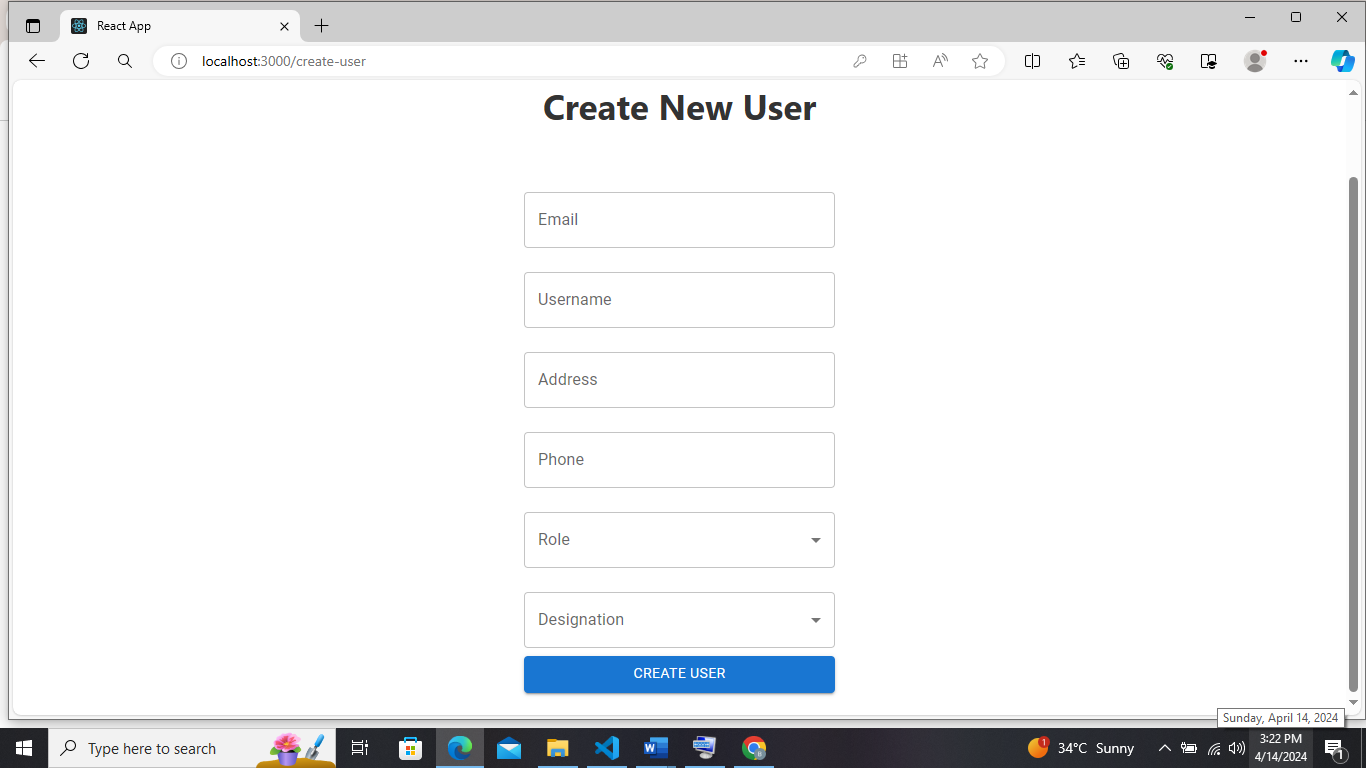
### User interface (UI) design approach.

**LOGIN PAGE**

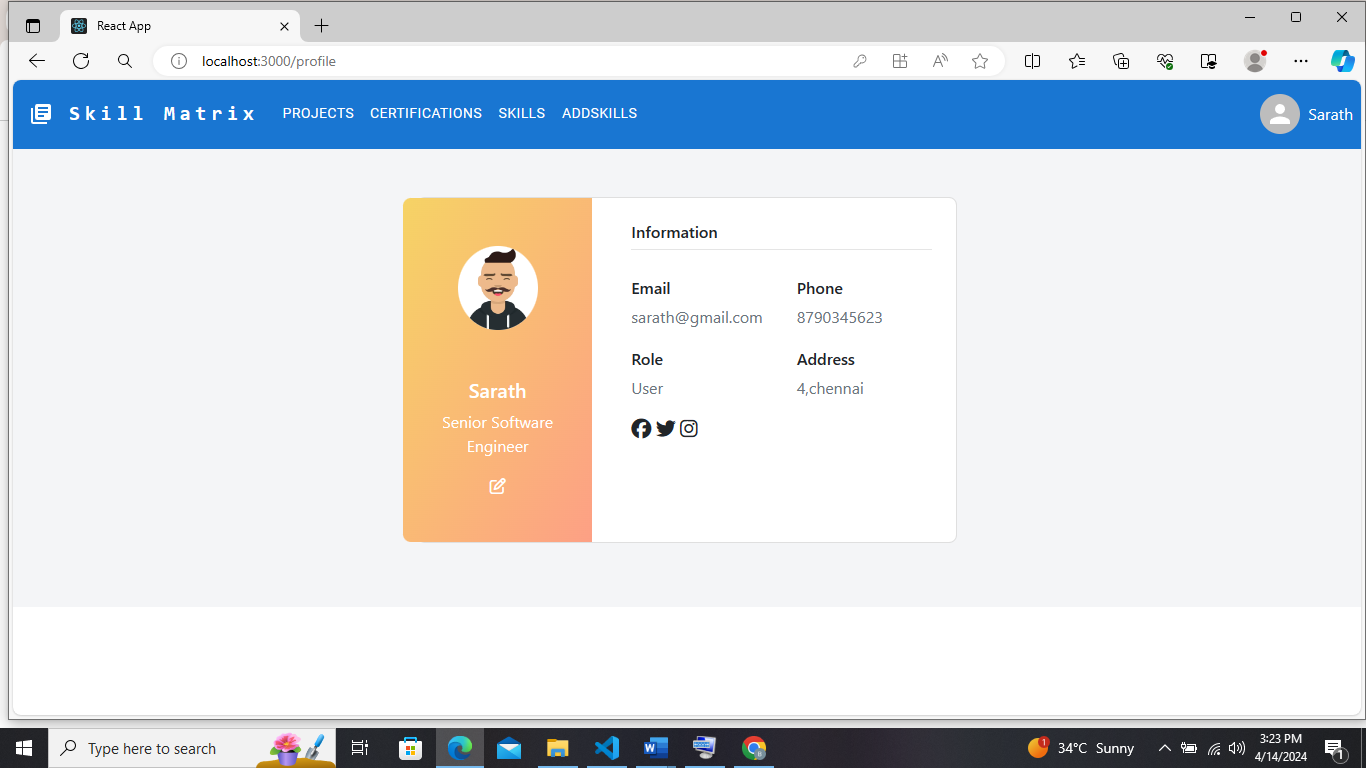
|  |  |
| --- | --- |
|  |  |
| **ADMIN PAGE** |  |
|  |  |



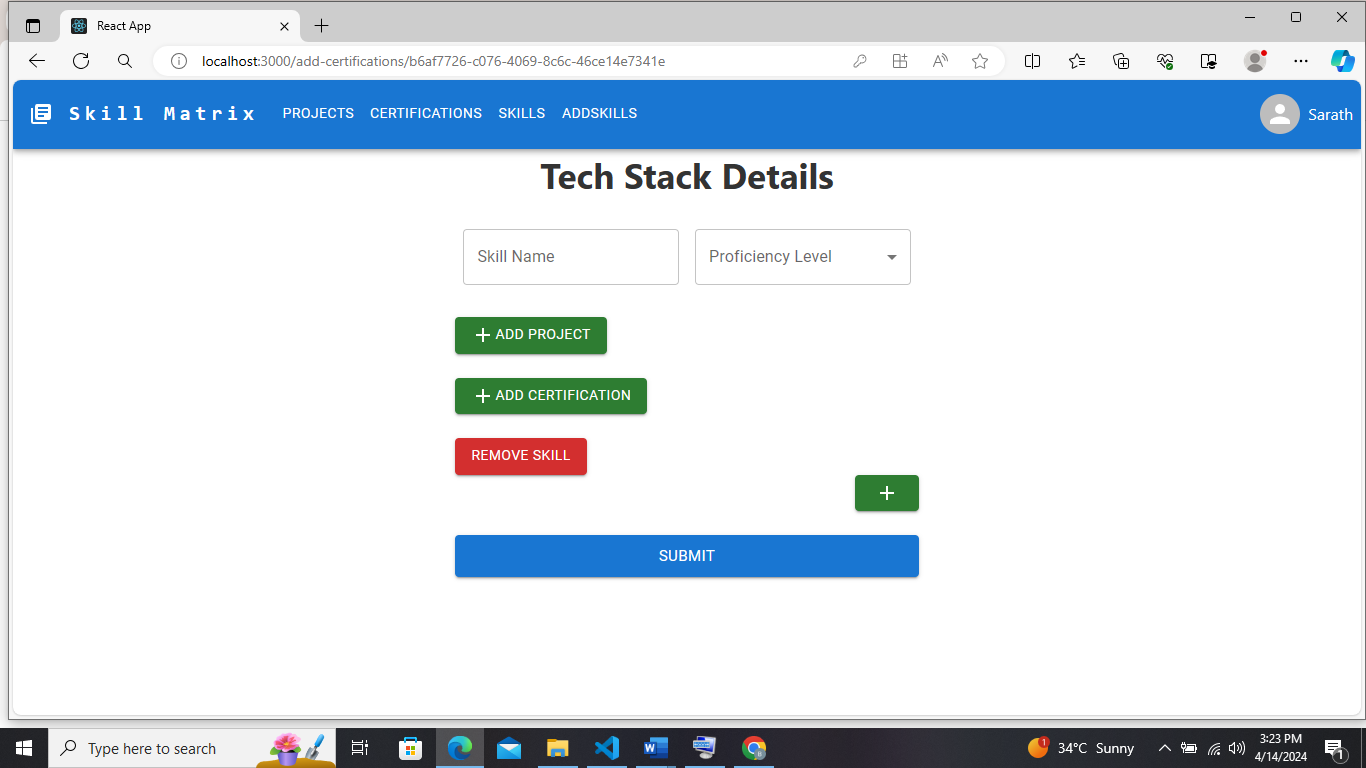
**USER CREATION PAGE**



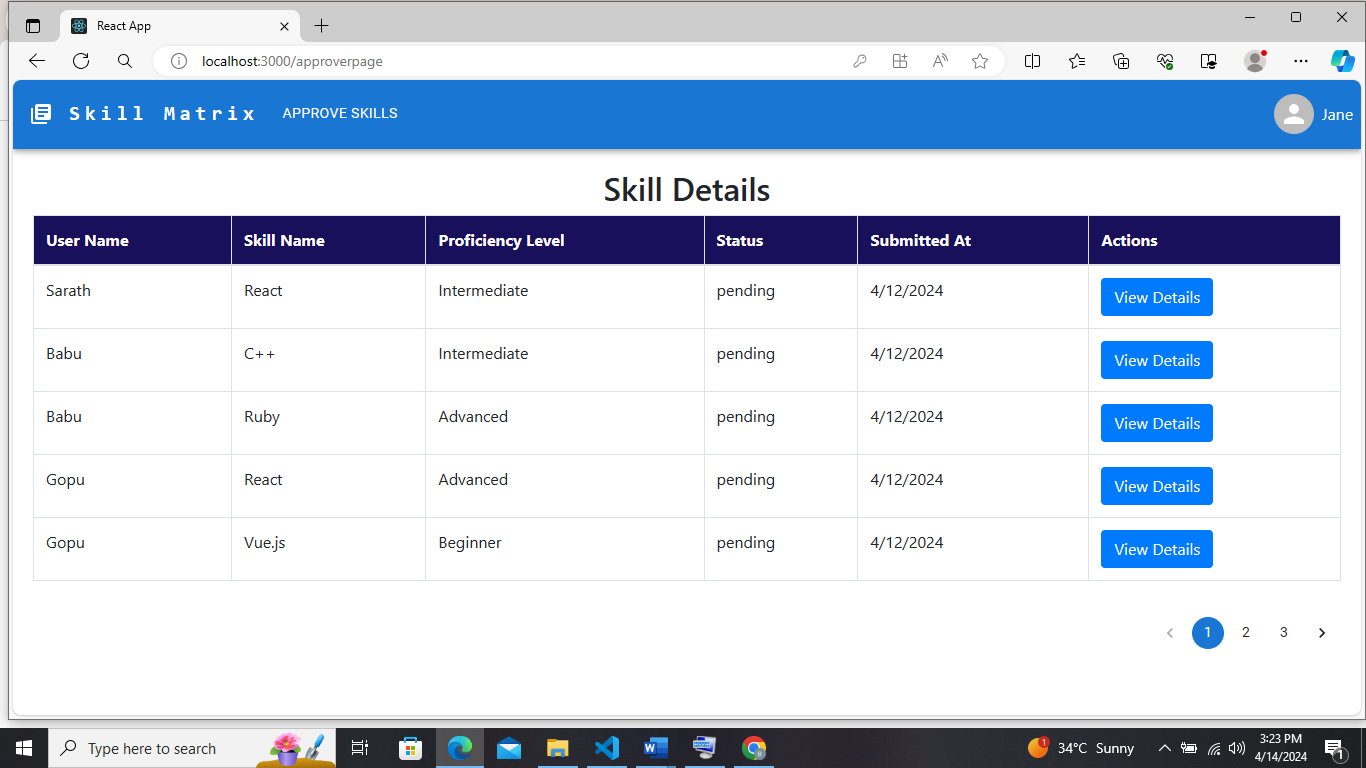
**USER PROFILE PAGE**



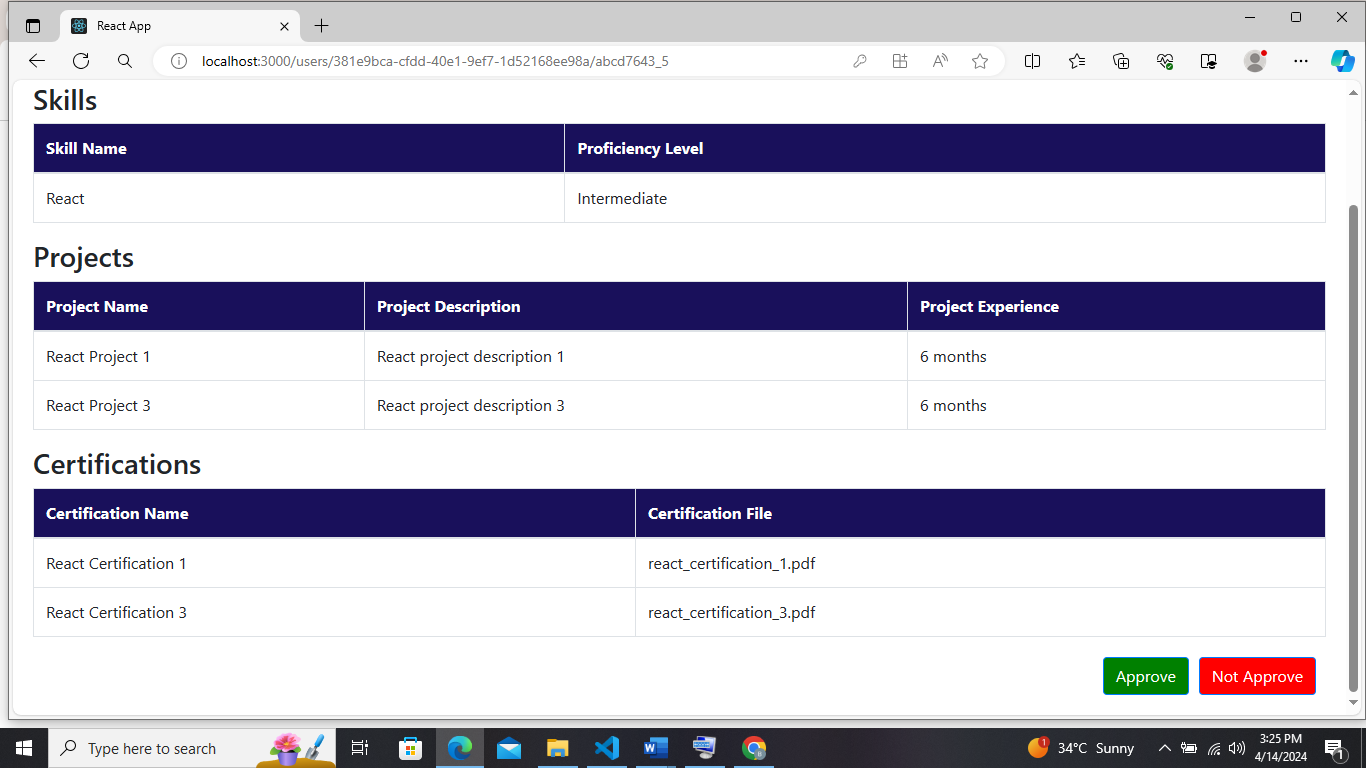
**ADD SKILLS PAGE**



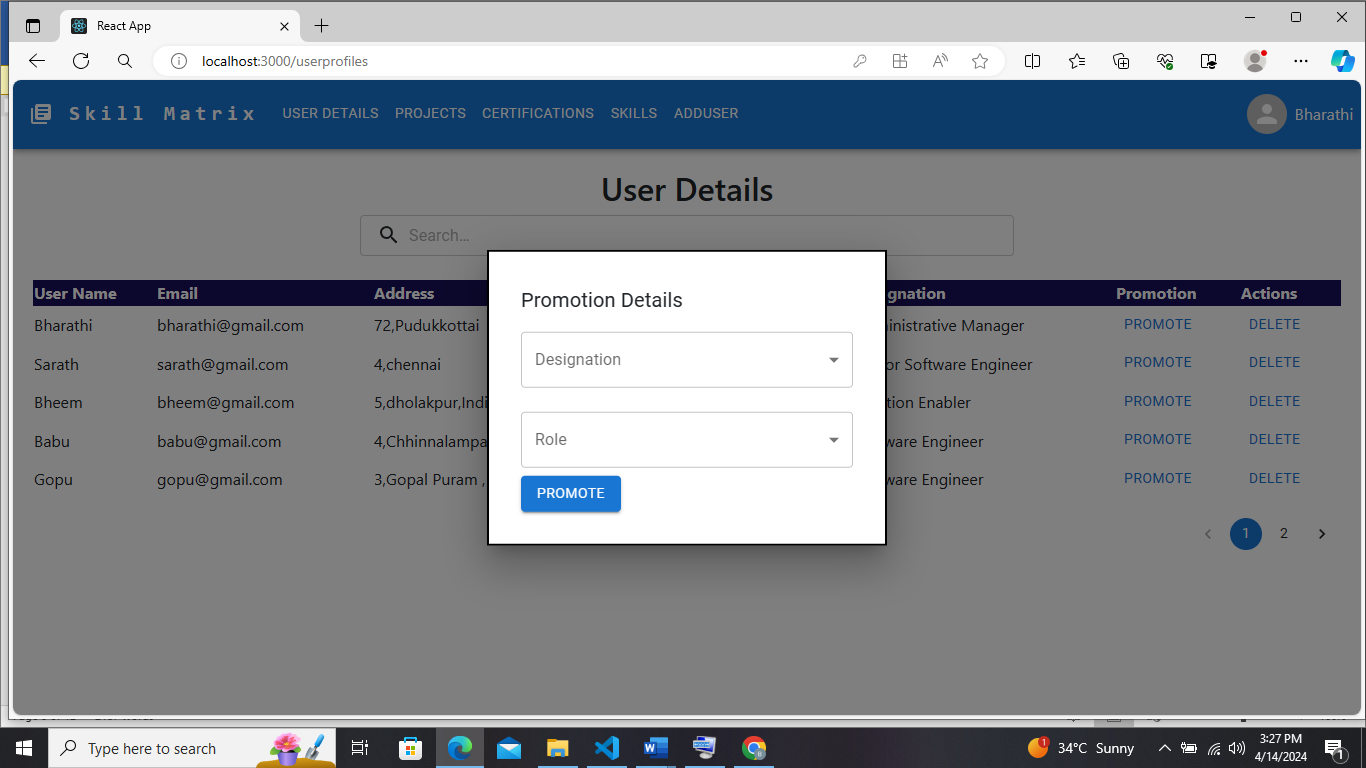
**APPROVER PAGE**



**HANDLE APPROVE PAGE**



**PROMOTE PAGE**



# Data Engineering

### Components

**Source Database:** **PostgreSQL**

The PostgreSQL database serves as the source of raw data.

It contains tables with transactional data that need to be processed for analytics.

**ETL Pipeline: DBT (Data Build Tool)**

DBT is used for Extract, Transform, and Load (ETL) operations.

It connects to the PostgreSQL database, extracts data, performs transformations using SQL queries and Jinja templating, and stages the transformed data.

DBT provides a modular approach to building and managing data transformation pipelines.

**Target Data Warehouse: Snowflake**

Snowflake acts as the target data warehouse for storing transformed data.

It provides scalable and flexible storage for analytics and reporting purposes.

The transformed data from DBT is loaded into Snowflake tables for further analysis.

### PostgreSQL to Snowflake Data Transfer:

**Establish Connection:** Set up a secure connection between PostgreSQL and Snowflake, ensuring proper authentication and authorization mechanisms.

**Data Transfer**: Utilize tools like AWS Data Pipeline, Apache NiFi, or custom scripts to transfer data from PostgreSQL to Snowflake.

**Data Validation:** Validate the integrity and completeness of transferred data to ensure accurate analysis.

### Configuring DBT for Staging and Transformations:

**Installation:** Install DBTon your local machine or server by following the installation instructions provided in the **DBT** documentation.

**Connection Profiles:** Configure DBTprofiles (profiles.yml) to connect to both PostgreSQL and Snowflake databases securely.

**Staging Models:** Write DBTmodels to stage raw data from PostgreSQL tables into Snowflake staging tables, preserving the source data structure.

**Transformation Models**: Define DBTtransformations using SQL and Jinja templating to process staged data and create analytical datasets tailored to your business requirements.

The metrics that have been included for transformation are:

1. User Engagement score based on user’s previous projects , skills and certifications
2. Raking based on user’s skills proficiency levels with appropriate skills
3. User details with appropriate project details , skills details and certification details
4. Total count of users in a particular designation

### Loading Transformed Data into Snowflake:

**Loading Strategy:** Use DBT’sbuilt-in functionality to load transformed data from DBTmodels into Snowflake tables efficiently.

**Snowflake Configuration:** Configure Snowflake warehouses, databases, and schemas to optimize performance and resource utilization.

**Automation:** Schedule DBTruns using cron jobs, Airflow, or similar tools to automate data transformation and loading processes, ensuring timely updates.

**Monitoring:** Monitor DBTruns and Snowflake usage to track pipeline performance, identify bottlenecks, and optimize resource allocation for better efficiency.

# Data Science

### EDA (Exploratory Data Analysis):

Exploratory Data Analysis (EDA) is an essential step in the data analysis process, where we aim to understand the data characteristics, patterns, and relationships within the dataset. Through EDA, we can uncover insights, identify anomalies, and prepare the data for further analysis and modelling. The analyses that have been made are:

* Count of number of users in a particular designation
* Count of number of users based on their roles such as Admin , User and Approvers
* Checking the null values in the dataset
* Remove the null values in the features
* Top users who have completed more projects , skills and certifications

### 6.1.2 Feature Engineering :

Feature engineering is the process of transforming raw data into informative features that can be used by machine learning algorithms to make accurate predictions. It involves selecting, extracting, and transforming features to improve the performance of a model.

The new feature that has been considered is to check whether the user is suitable for the project or not based on user engagement score and skills proficiency level. The feature name is set as “suitable\_for\_project”.

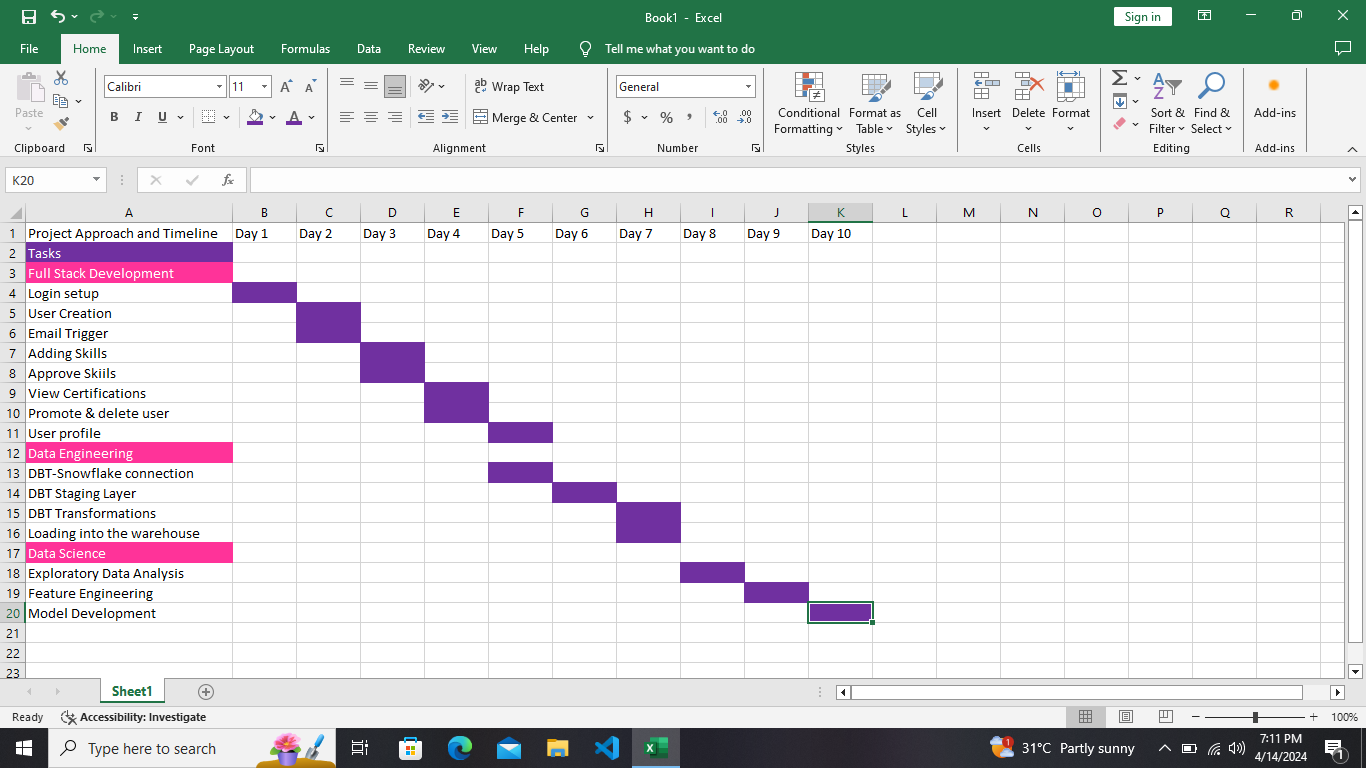
### Model Development:

Model development in machine learning involves the process of creating, training, evaluating, and refining machine learning models to perform specific tasks or solve particular problems.

Here the model predicts whether the user is suitable for the upcoming projects or not, the model takes the inputs of skills, certifications and previous projects and skills proficiency.

Here I am using Logistic Regression with accuracy of 98% to predict the user is suitable for upcoming projects.

# Project Timeline



**Appendix Title**

Document Title