JMAN GROUP

Skill Matrix System (SMS)

Saket Singh

Final Project

Contents

Table of Contents

1 Project Overview

1.1.1 Brief introduction of the project.

The "Skill Matrix System" is a full-stack application tailored for corporate environments, streamlining the management of employee skills and experiences. Administrators can create user profiles, and users can update their skills, certificates, and project experiences. The system employs an approval workflow where users can review and approve each other's entries, ensuring data accuracy and reliability. Powered by a MongoDB Atlas database backend, it offers a robust platform for skill tracking and management.

1.1.2 Purpose of the project.

The purpose of this project is to empower organisation and enhance their experience by providing them with a comprehensive employee technical expertise tracking solution.

The key goals of the project include:

- The project aims to provide a centralized platform for corporate entities to streamline the management of employee skills, certificates, and project experiences, facilitating better talent utilization and resource allocation..
- By incorporating an approval workflow, the system fosters collaboration among employees by allowing them to review and validate each other's skill entries. This promotes accountability and ensures the accuracy and credibility of the information within the system.

1.1.3 Goals of the project:

- Design of the UI/UX components for Skill Matrix System supporting both user and admin functionality.
- The exact scope of work for the Development Phase will be determined by the Requirements Analysis Phase. By the end of development phase, a Minimum Viable Product (MVP) – as scoped in the Requirements phase.

2 Project Scope

The project scope encompasses the development of a comprehensive "Skill Matrix System" that includes features such as user management, skill tracking, certification management, project experience recording, an approval workflow, and secure authentication. It involves the creation of both frontend and backend components, integration with a MongoDB Atlas database, and implementation of user-friendly interfaces for administrators and regular users.

2.1.1 Key features and functionalities of the mobile app.

- **User Authentication:**User-friendly authentication process for User to their accounts.Secure authentication mechanisms, including email/password
- **Personalized User Dashboard**:Customized dashboard for each user, displaying relevant information such as the skills ,certificates , projects experiences and the approves status. Also an approver will see task to approve.
- **Responsive Design:**User-friendly design to support various devices, including Laptops, Smartphone.Responsive user interface for optimal viewing and usability across different screen sizes
- **Admin Functionalities:**Admin dashboard for keeping track of all the user.Functionalities for user creation, making and removing approvers and also being able to see all the users created with their skill , certificates and projects.

3 Architecture and Technology Stack

3.1.1 Overall architecture of the Skill Matrix System.

1. Presentation Layer:

Description: The presentation layer is the topmost layer of the architecture, responsible for handling user interactions and displaying information to users.

Components:

- User Interface (UI): Provides interactive dashboards and interfaces for users, administrators, and approvers.
- Client-Side Logic: Manages client-side interactions and communicates with the backend server.

Technologies:

- React.js for frontend development..
- Tailwind CSS for styling.

2. Application Layer:

Description: The application layer contains the business logic and functionality of the Skill Matrix System.

Components:

- Controllers: Handle incoming requests from the presentation layer and orchestrate business operations.
- Services: Implement core business logic, including user management, skill tracking, certification management, and project submissions.

Technologies:

- Node.js with Express.js for building RESTful APIs.
- JavaScript for server-side logic.

3. Domain Layer:

Description: The domain layer encapsulates the core domain concepts and entities of the Skill Matrix System.

Components:

- User: Represents user accounts within the system, including roles, skills, certifications, and project contributions.
- Skill: Represents individual skills and proficiency levels tracked within the system.
- Certification: Represents certifications and qualifications uploaded by users.
- Project: Represents project contributions documented by users.

4. Data Access Layer:

Description: The data access layer manages interactions with the underlying data storage systems.

Components:

- Data Access Objects (DAOs): Encapsulate database interactions and provide an abstraction layer for querying and manipulating data.

- Database Connection Pool: Manages connections to the database to ensure efficient data access.

Technologies:

- MongoDB for storing user data, skill information, certifications, and project details.

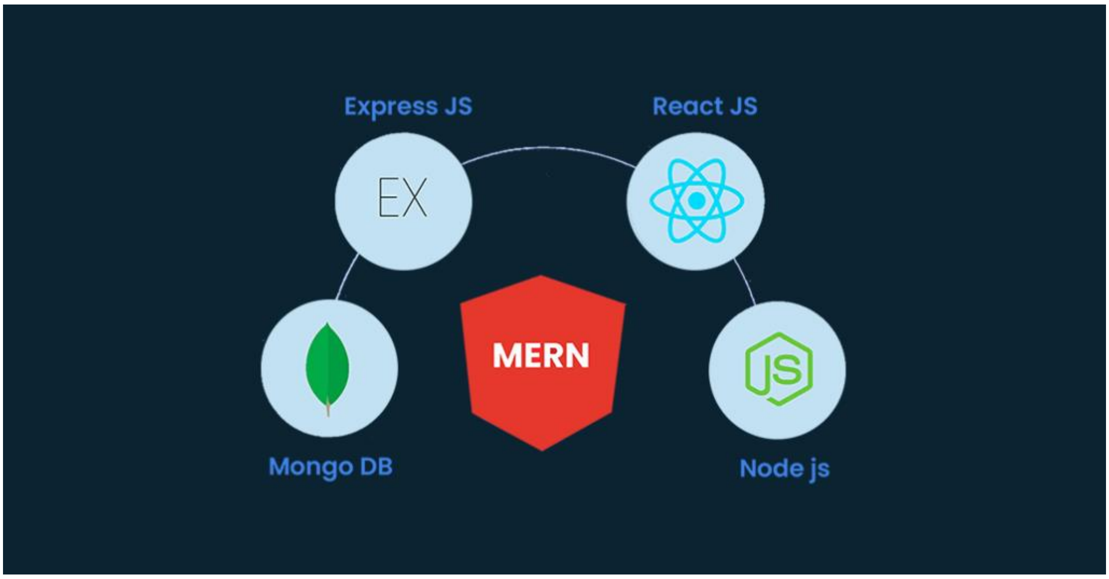
3.1.2 Technology stack (e.g., programming languages, frameworks, libraries).

Frontend:

- Framework: React.js
- Styling: Tailwind CSS
- Using axios (for making HTTP requests to the backend)

Backend:

- Framework: Node.js with Express.js
- Database: MongoDB
- Authentication: JSON Web Tokens (JWT) for securing API endpoints



Data Engineering:

- Python: Used for data injection to Snowflake, data preprocessing, and any custom data manipulation tasks.
- Snowflake: Cloud data platform used for storing and managing structured and semi-structured data.
- Dbt (Data Build Tool): Used for staging and transforming data in snowflake as per the KPIs.

Data Analytics:

- Power BI: Business intelligence tool used for creating reports and dashboards.

Data Science:

- Programming Language: Python (for its extensive libraries and frameworks for data science)

- Used Gradient Boosting Classifier

4 Project Stages:

4.1.1 Front-End (React.js):

Description:

- Develops the user interface components of the Skill Matrix System using React.js, a popular JavaScript library for building UIs.

Activities:

- Designing and implementing UI components such as forms, tables, and dashboards.
- Managing state and user interactions using React.js features like state management and hooks.
- Integrating with backend APIs to fetch and update data asynchronously.
- Ensuring responsiveness and cross-browser compatibility.

Tools:

- React.js, React Router (for navigation), React Hooks (for state management), Axios (for HTTP requests).

4.1.2 Database (MongoDB):

Description:

- Manages data storage and retrieval for the Skill Matrix System using MongoDB, a flexible and scalable NoSQL database solution known for its JSON-like document storage and high- performance capabilities.

Activities:

- Designing JSON-like document structures to represent entities such as admins, users, skills, certifications, and projects.
- Creating collections, indexes, and validations to organize and enhance data storage efficiency.
- Implementing data integrity and relationships between collections using MongoDB's flexible schema design.
- Performing operations to interact with the database.

Tools:

- MongoDB Atlas, mongoose (for query and data manipulation).

4.1.3 Back-End (Node.js):

Description:

- Implements the server-side logic and APIs of the Skill Matrix System using Node.js, a JavaScript runtime environment.

Activities:

- Developing RESTful API endpoints to handle requests from the frontend and interact with the database.
- Implementing authentication and authorization mechanisms to secure access to resources.
- Writing middleware functions for request validation, error handling, and logging.
- Integrating with external services and APIs as needed.

Tools:

- Node.js, Express.js (as the web application framework), JWT (for authentication), mongoose (for communication to the database).

4.1.4 Data Engineering:

1. **Python Code for Pipeline:**

   - Develop Python code using libraries like Pandas, numpy , faker to first create fake data for ingestion in MongoDB and then to create a pipeline for transferring data from MongoDB to Snowflake Warehouse.

   - Use Snowflake's Python connector to establish a connection and load data into Snowflake tables.

2. **Snowflake Data Warehouse:**

   - Create a Snowflake database named JMAN_FINAL_PROJECT.

   - Designate three schema within the database:

     - Public : Raw data layer for storing data directly ingested from sources.

     - Public_Staging: Intermediate layer for performing data transformation and processing.

     - Public_Mart: Final layer for storing processed and cleaned data ready for analysis.

3. **DBT (Data Build Tool):**

   - Utilize DBT to perform data modeling, transformation, and management tasks.

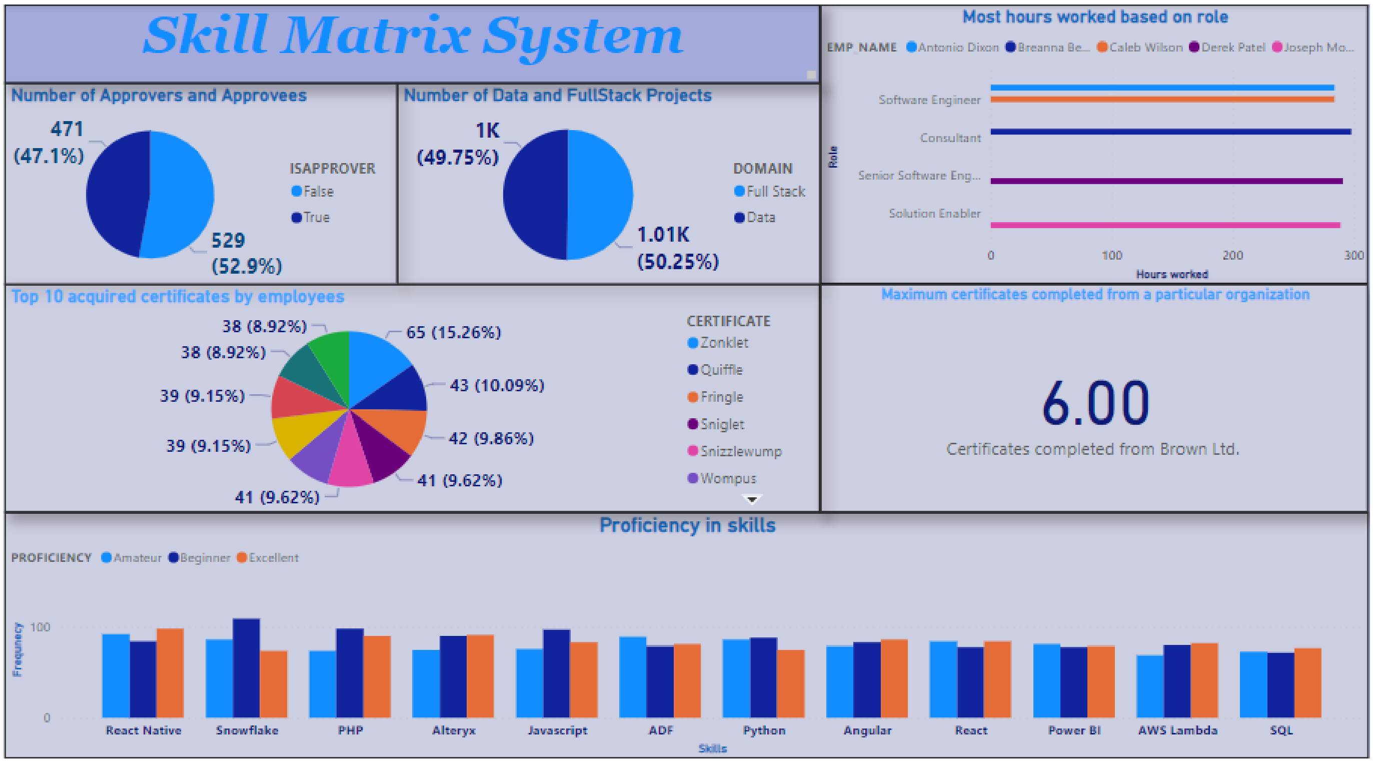
   - Write DBT models first for staging and then mart to make the data ready for analysis.

   - Execute DBT commands to run models, generate SQL scripts, and deploy changes to Snowflake.

4.1.5 Data Analytics:

1. **Power BI Dashboard:**

- Connect Power BI to the Snowflake warehouse's Public Mart layer to access cleaned and processed data.
- Design interactive dashboards using Power BI's drag-and-drop interface to visualize insights derived from the data.
- Incorporate various visualization types like charts, graphs, tables, and maps to represent different aspects of the data.



4.1.6 Data Science:

Python Code for Data Science:

- Conducted thorough data cleaning, handling missing values, and performing exploratory data analysis using Python libraries such as Pandas, NumPy, and Matplotlib..
- Utilized advanced feature engineering techniques to enhance the predictive power of the model. This included creating new features and extra columns based on domain knowledge and data analysis, as well as augmenting the target variable to improve model performance and interpretability.
- Imported machine learning models from the Scikit-learn library and trained them on the preprocessed data. This involved selecting appropriate algorithms based on the problem domain, tuning hyperparameters, and evaluating model performance using techniques such as cross-validation and performance metrics like accuracy, precision, recall, and F1-score..

5 APP Guide

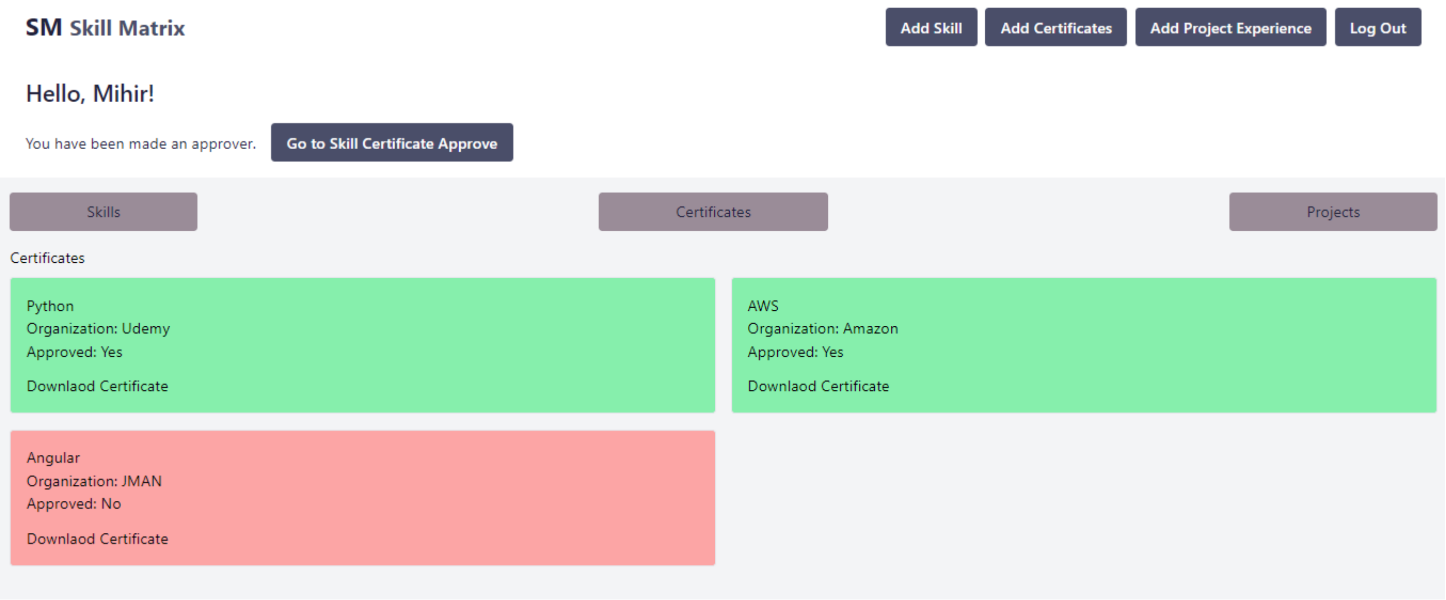
5.1.1 User Dashboard:

Overview: The User Dashboard is tailored for individual users within the organization to manage their skills, certifications, and project contributions.

Features:

- **Profile**: The user can view his personal info and his skills , certifictaes nd projects
- **Addition of Skills:** The user can add new skills and get the proficiency by a test
- **Addition of Certificates**: The user can add new certificates with the certificate pdf.
- **Addition of Projects**: User can add the projects he has worked on.
- **Approve Skills :** If made an approver , the user can approve certificates and projects of other employees.

User dashboard:



Hello, Mihir!

You have been made an approver.

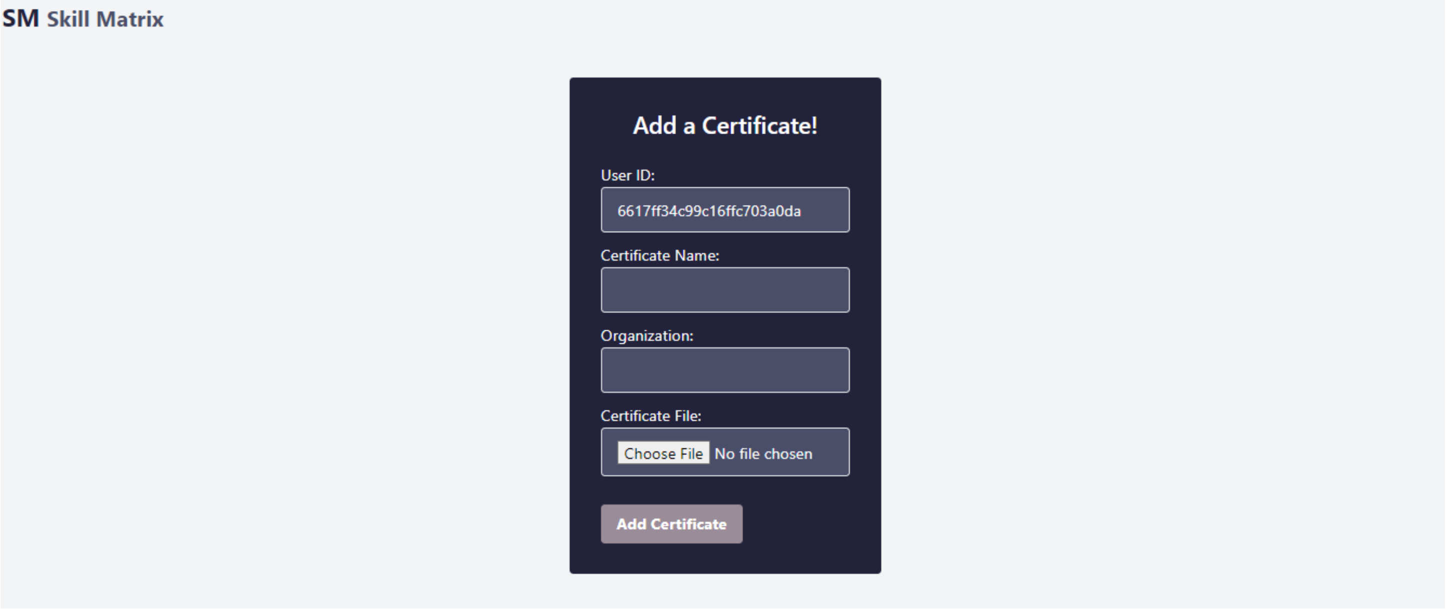
Certificates

Python
Organization: Udemy
Approved: Yes

AWS
Organization: Amazon
Approved: Yes

Angular
Organization: JMAN
Approved: No

Add Certificate:



Add a Certificate!

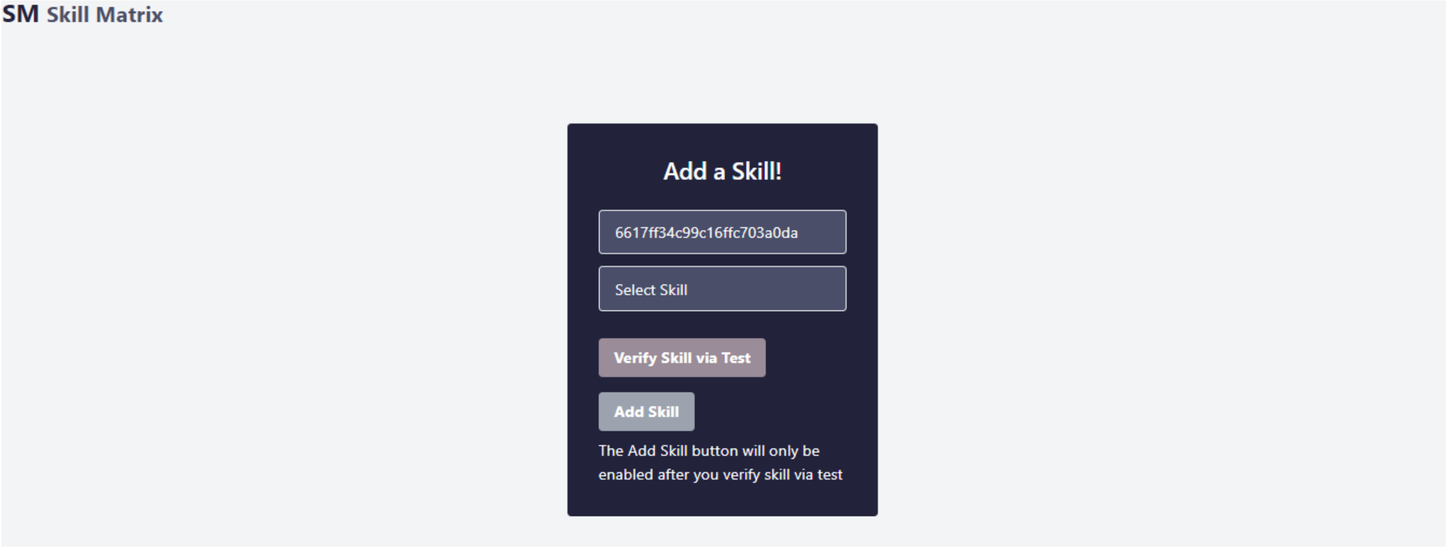
User ID: 6617ff34c99c16ffc703a0da

Certificate Name:

Organization:

Certificate File: No file chosen

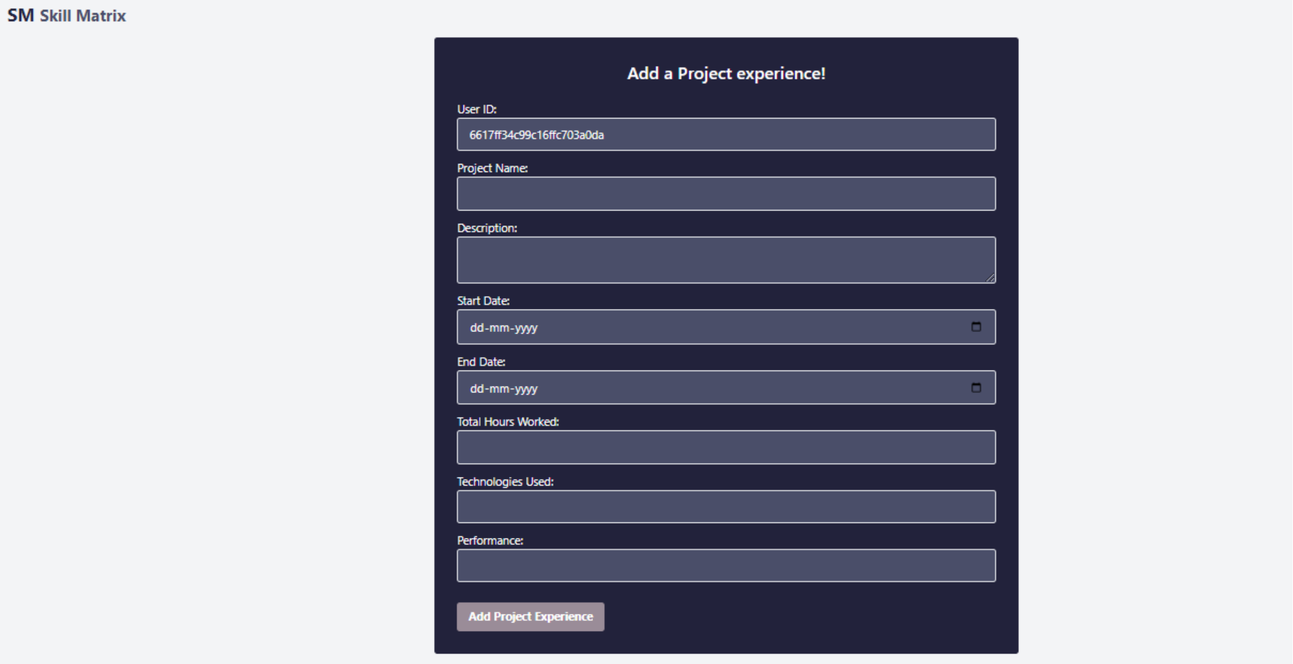
Add Skill:



6617ff34c99c16ffc703a0da

The Add Skill button will only be enabled after you verify skill via test

Add Project:



User ID: 6617ff34c99c16ffc703a0da

Project Name:

Description:

Start Date: dd-mm-yyyy

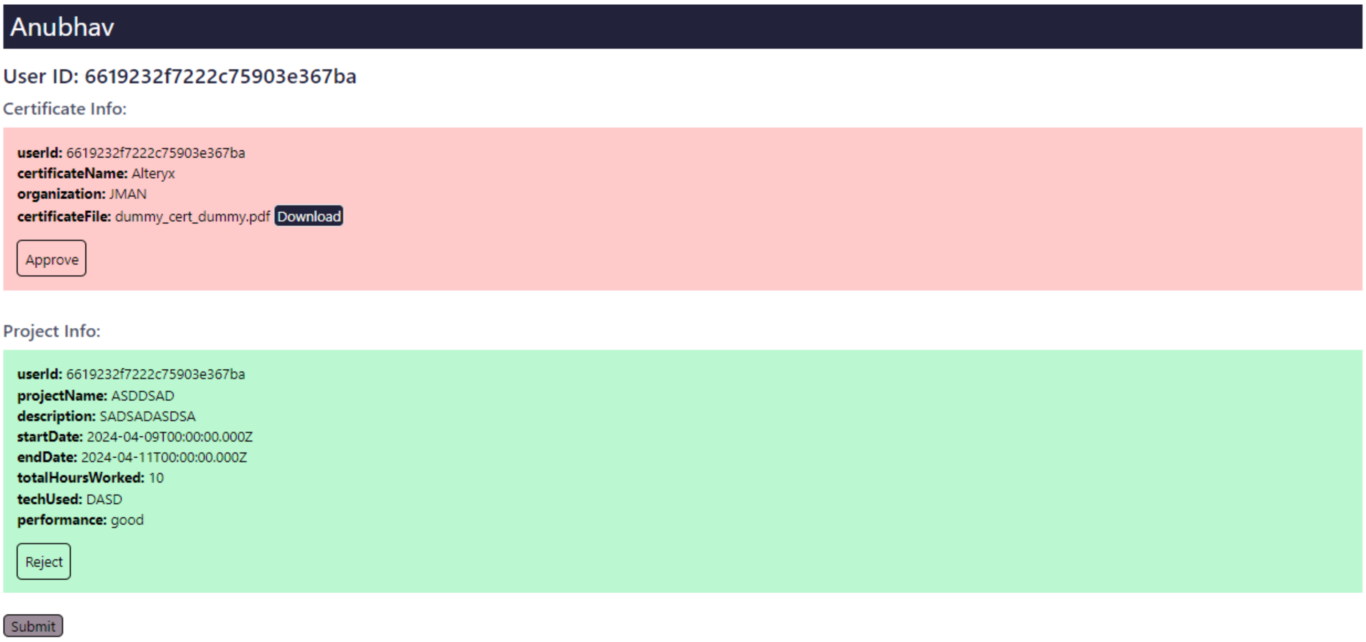
End Date: dd-mm-yyyy

Total Hours Worked:

Technologies Used:

Performance:

Approve Page:



Anubhav

User ID: 6619232f7222c75903e367ba

Certificate Info:

userId: 6619232f7222c75903e367ba
certificateName: Alteryx
organization: JMAN
certificateFile: dummy_cert_dummy.pdf

Project Info:

userId: 6619232f7222c75903e367ba
projectName: ASDDSAD
description: SADSADASDSA
startDate: 2024-04-09T00:00:00.000Z
endDate: 2024-04-11T00:00:00.000Z
totalHoursWorked: 10
techUsed: DASD
performance: good

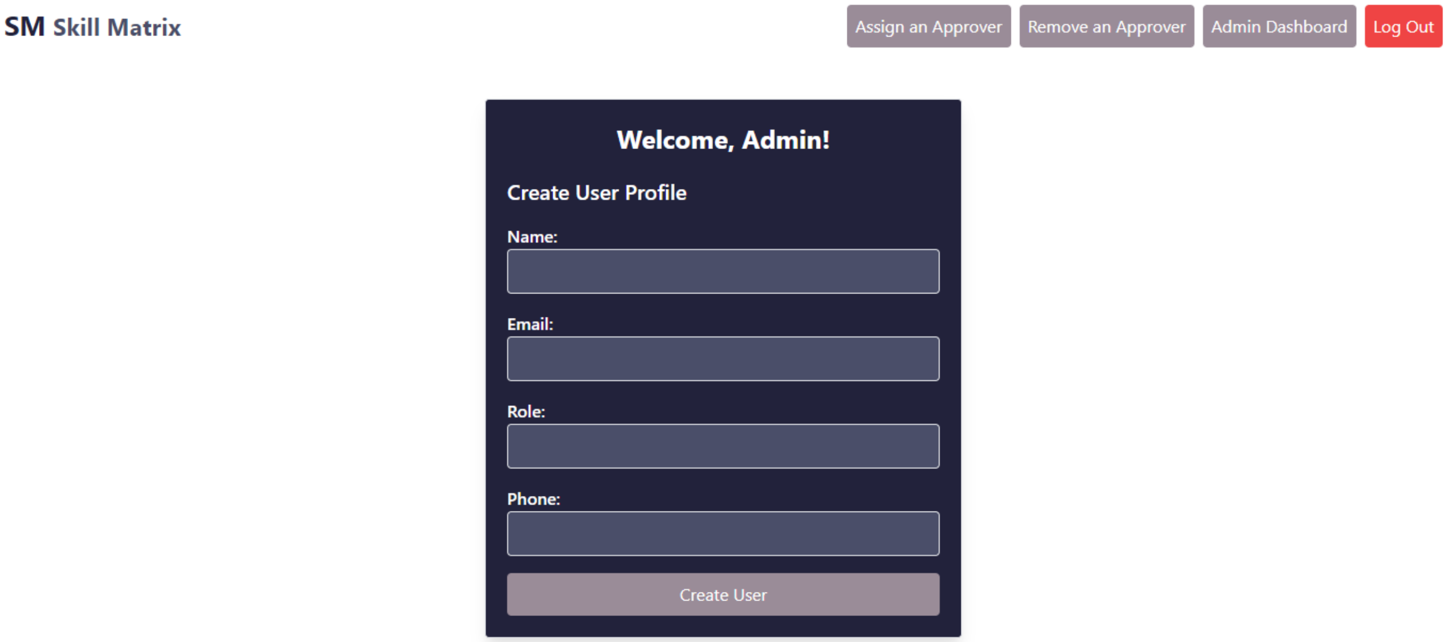
5.1.2 Admin Dashboard:

Overview: The Admin Dashboard provides administrators with comprehensive control and management capabilities over the Skill Matrix System.

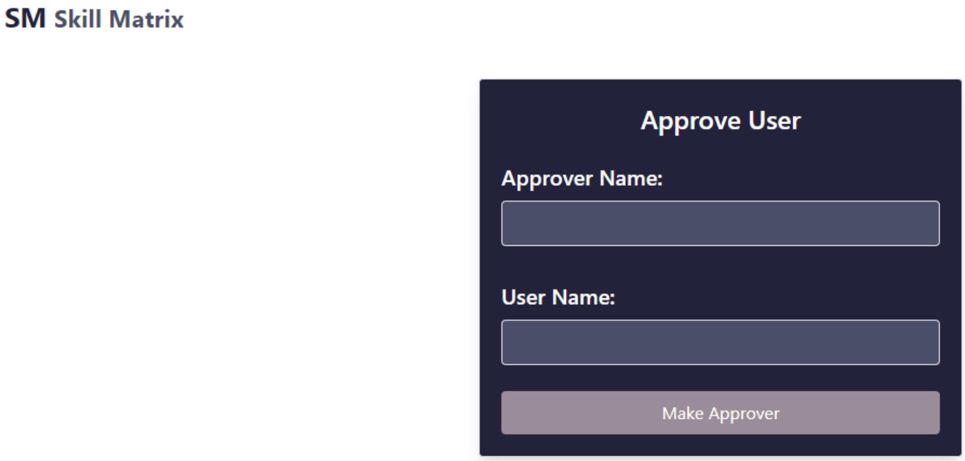
Features:

- **User Creation:** Admin cab create users for the system who can then login.
- **Assigning Approver:** Admin can assign approvers to users .
- **Removing Approvers:** Admin can also remove the approver status of the approver.
- **All User Dashboard:** Admin can see the mini dashboard of all users which will help him to assign and remove approvers accordingly.

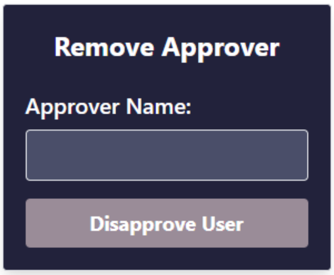
User Creation:



Assigning an Approver:



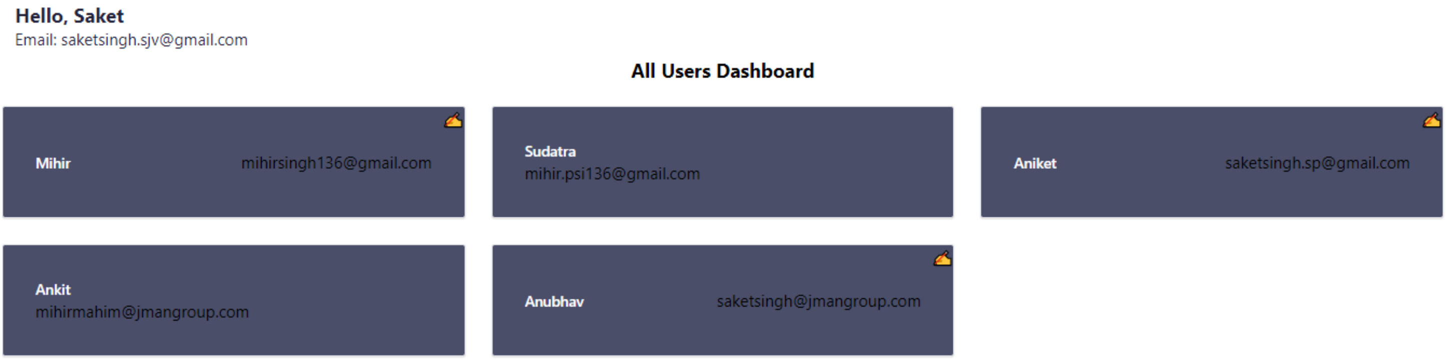
.**Un-assign an Approver**:



Remove Approver

Approver Name:

All User Dashboard:

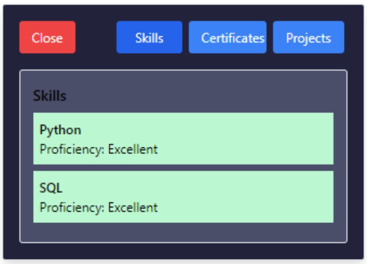


Hello, Saket

Email: saketsingh.sjv@gmail.com

All Users Dashboard

Mihir	mihirsingh136@gmail.com
Sudatra	mihir.psi136@gmail.com
Aniket	saketsingh.sp@gmail.com
Ankit	mihirmahim@jmangroup.com
Anubhav	saketsingh@jmangroup.com



Skills

Python
Proficiency: Excellent

SQL
Proficiency: Excellent

.

6 Workflow:

For Admin:

- Login: Admin logs in with the credentials.
- User Creation: Admin can create a user.
- Assign and Un-assign Approvers: Admin can assign approvers to approve certificates and project experiences.
- All User Dashboard: Admin can also see a mini dashboard for all users.

For User:

- Login : User logs in with credentials.
- Add Skill: User can choose a skill from the dropdown and then a give a short test for that skill to get the proficiency for that skill.

- Add Certificate: User can also add a certificate with a pdf of that certificate.
- Add Project Experience: User can also add project experiences with all the relevant information.
- Approve: If the user has been made an approver by the admin then he can also approve certificates and projects.

7 Data Model

7.1 Data Model Description:

Admin Schema:

- **Fields:** `name`, `password`, `email`.
- Represents the data structure for admin users in the system.
- Stores information about administrators who have access to the system.

User Schema:

- **Fields:** `name`, `password`, `email`, `role`, `phone`, `isPasswordSet`, `isApprover`, `madeApproverFor`, `resetPasswordToken`, `resetPasswordExpires`.
- Represents the data structure for regular users in the system.
- Stores information about users, including their credentials, contact details, roles, and password-related data.

Skill Schema:

- **Fields:** `userId`, `skillName`, `proficiency`, `isVerified`, `marksScored`.
- Represents the skills possessed by users.
- Links to the user who owns the skill via `userId`.
- Contains information about the skill name, proficiency level, verification status, and any associated marks scored.

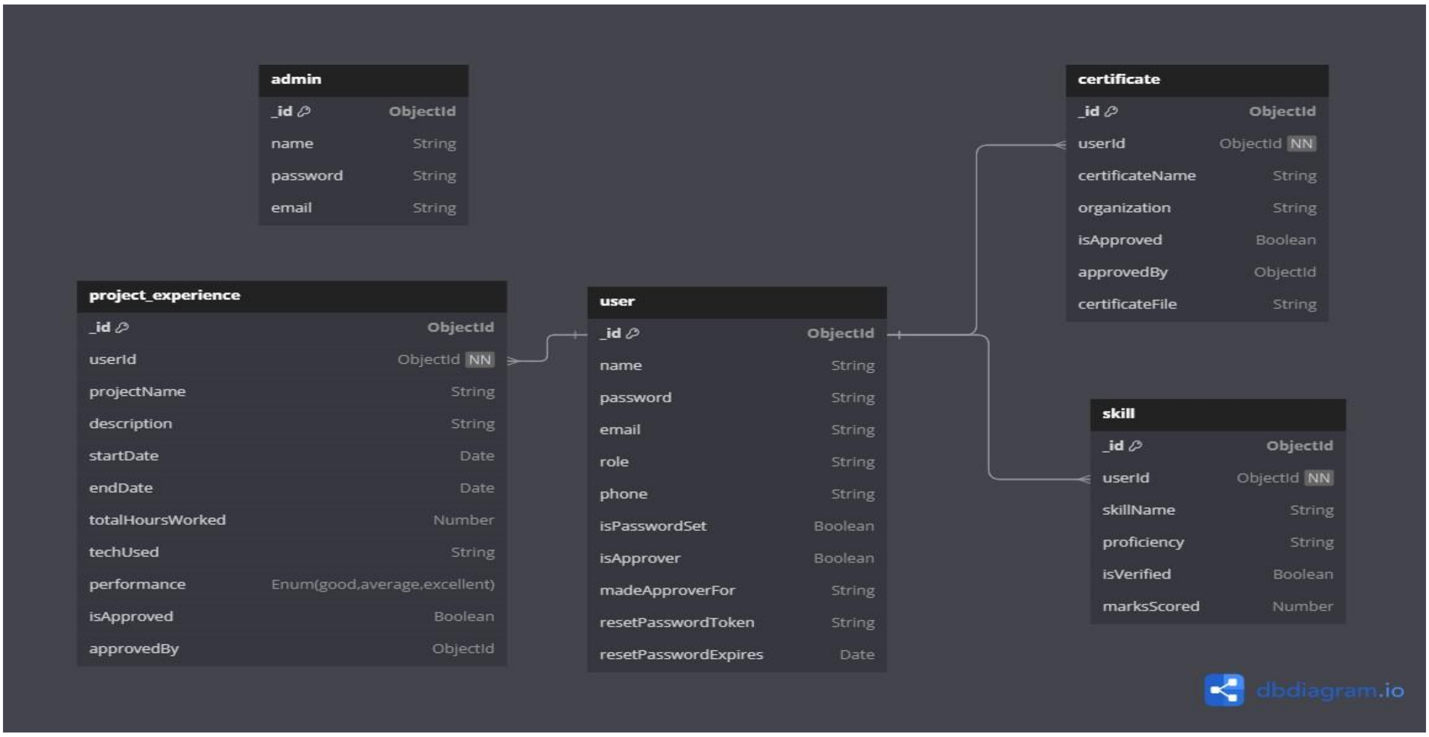
Certificate Schema:

- **Fields:** `userId`, `certificateName`, `organization`, `isApproved`, `approvedBy`, `certificateFile`.
- Represents certificates earned by users.
- Links to the user who earned the certificate via `userId`.
- Contains info about the certificate name and the organization that issued that certificate.

Project Experience Schema:

- **Fields:** `userId`, `projectName`, `description`, `startDate`, `endDate`, `totalHoursWorked`, `techUsed`, `performance`, `isApproved`, `approvedBy`.
- Represents the project experiences of users.
- Links to the user who was involved in the project via `userId`.
- Contains details about the project name, description, duration, technologies used, performance assessment, approval status, and approving authority.

7.2 Entity-Relationship Diagram (ERD):



admin	
_id	ObjectId
name	String
password	String
email	String

certificate	
_id	ObjectId
userId	ObjectId NN
certificateName	String
organization	String
isApproved	Boolean
approvedBy	ObjectId
certificateFile	String

project_experience	
_id	ObjectId
userId	ObjectId NN
projectName	String
description	String
startDate	Date
endDate	Date
totalHoursWorked	Number
techUsed	String
performance	Enum(good,average,excellent)
isApproved	Boolean
approvedBy	ObjectId

user	
_id	ObjectId
name	String
password	String
email	String
role	String
phone	String
isPasswordSet	Boolean
isApprover	Boolean
madeApproverFor	String
resetPasswordToken	String
resetPasswordExpires	Date

skill	
_id	ObjectId
userId	ObjectId NN
skillName	String
proficiency	String
isVerified	Boolean
marksScored	Number

8 Conclusion:

In conclusion, the "Skill Matrix System" represents a pivotal advancement in workforce management within corporate environments. By streamlining the process of skill and experience tracking, this system offers a centralized platform for administrators to efficiently create and manage user profiles while empowering employees to showcase their skills, certificates, and project experiences. Through meticulous data preprocessing and exploratory data analysis, the system ensures the accuracy and reliability of the information stored within its database. This enables organizations to make informed decisions regarding talent deployment and resource allocation, ultimately enhancing productivity and competitiveness in today's dynamic business landscape.

Moreover, the incorporation of feature engineering techniques adds depth and nuance to the data, allowing for the creation of additional variables that enrich the predictive capabilities of the system. By augmenting the target variable and fine-tuning machine learning models imported from libraries like Scikit-learn, the system can provide valuable insights into talent development and succession planning. This not only facilitates better decision-making at the organizational level but also fosters a culture of continuous learning and professional development among employees.

Furthermore, the "Skill Matrix System" prioritizes security and user experience, offering robust authentication mechanisms to safeguard sensitive corporate data and ensure compliance with privacy regulations. Through an intuitive interface and seamless integration with MongoDB Atlas, the system enhances collaboration and accountability among team members, driving innovation and fostering a sense of ownership over individual skill development. In essence, the "Skill Matrix System" stands as a testament to the transformative power of technology in optimizing workforce management and unlocking the full potential of human capital in today's digital age.