**Employee Learning Platform**

**Major Project**

Contents

[1 Project Overview 2](#_Toc163829857)

[1.1 Brief introduction of the project. 2](#_Toc163829858)

[1.2 Purpose of the project. 2](#_Toc163829860)

[1.3 Goals of the project: 2](#_Toc163829861)

[1.4 Key features and functionalities of the web app. 3](#_Toc163829866)

[2 Architecture and Technology Stack 3](#_Toc163829874)

[2.1 Overall architecture of the application. 3](#_Toc163829875)

[2.2 Technology stack (e.g., programming languages, frameworks, libraries) 4](#_Toc163829894)

[3 Authentication and Authorization 5](#_Toc163829895)

[4 Data Model 8](#_Toc163829896)

[4.1.1 Entity Relationships: 8](#_Toc163829897)

**5**  **Recommendation Model:……………………………………………………………………………………...10**

# Project Overview

## Brief introduction of the project.

## The Employee Learning Platform project seeks to revolutionize the learning and development landscape within our organization by providing a comprehensive and user-centric platform. It serves as a centralized hub for employees to access training resources, register for events, and engage in continuous learning. Through a secure login page, users can confidently explore the platform's offerings, knowing their data is protected. The platform's Event Creation feature empowers administrators to orchestrate impactful training sessions, while User Registration capability enables employees to curate their learning path based on their interests and career goals. Automated Email Notifications ensure timely updates on event registrations, fostering active participation and engagement. Additionally, Capacity Management functionality enhances efficiency by monitoring event capacity and notifying interested employees when events reach full capacity. Altogether, the Employee Learning Platform drives a culture of learning excellence within our organization, empowering employees to take ownership of their growth and supporting administrators in achieving organizational learning goals.

## Purpose of the project.

The purpose of the Employee Learning Platform project is to revolutionize the learning and development landscape within our organization. Through the creation of a centralized hub, it aims to provide employees with easy access to training resources and opportunities for professional growth. By facilitating seamless event registration and personalized learning paths, the platform seeks to empower employees to take ownership of their development journey. Additionally, the project aims to enhance communication and engagement through automated email notifications and proactive capacity management. Ultimately, the Employee Learning Platform endeavors to foster a culture of learning excellence, where employees are equipped with the knowledge and skills needed to drive organizational success.

## 1.3 Goals of the project:

# Enhance transparency: Establish a clear overview of employees' skillsets, certifications, and project experiences to facilitate informed decision-making.

# Improve resource allocation: Enable efficient matching of employee skills and interests with available training opportunities and project needs, optimizing team composition and enhancing project outcomes through strategic deployment of talent.

# Strengthen compliance: Ensure adherence to organizational learning standards and regulatory requirements by providing visibility into employees' certification statuses and ensuring that training activities align with compliance mandates, thereby mitigating risks and upholding quality assurance standards..

# Foster professional growth: Empower employees to proactively manage their development by offering personalized learning pathways, opportunities to identify skill gaps, and access to targeted training resources, fostering a culture of continuous learning and supporting career advancement within the organization.

## Key features and functionalities of the web app.

# Secure User Authentication Implement robust login functionality to ensure that only authorized personnel can access the platform securely, safeguarding sensitive data and maintaining confidentiality.

# User Profile Management: Allow administrators to efficiently create, update, and manage user profiles within the platform, encompassing essential information such as basic details, roles, and permissions.

# Skill Tracking: Empower users to input and regularly update their skills, certifications, and project experience directly into the platform, with options to specify proficiency levels and relevant details to accurately reflect their expertise.

# Approval Workflow: Establish an efficient approval process where designated approvers can review and approve users' submissions regarding certifications and project experience, ensuring accuracy and reliability of information..

# Email Notifications: Configure automated email notifications to promptly inform users and admins about pending approvals, updates, and other relevant actions within the platform, promoting timely responses and smooth workflow.

# Dashboard View: Provide intuitive dashboards tailored for both users and administrators, offering comprehensive visualizations of skill profiles, certification statuses, pending approvals, and summaries of project experience, events enhancing accessibility and facilitating informed decision-making..

# Reporting and Analytics: Enable the generation of insightful reports and analytics within the platform to track skill trends, monitor certification compliance, analyze project allocations, and other key metrics, empowering stakeholders with valuable insights for strategic planning and optimization..

# Architecture and Technology Stack

## Overall architecture of the application.

## Frontend (React, Material-UI, Bootstrap):

## React forms the foundational frontend library for constructing user interface components, fostering dynamic and interactive experiences.

## Leveraging Material-UI and Bootstrap enhances the aesthetics and responsiveness of the UI components, elevating the overall visual appeal and user experience.

## The frontend seamlessly communicates with the backend server via RESTful APIs, facilitating efficient data exchange and updates.

## Node.js furnishes the server-side runtime environment, empowering streamlined execution of asynchronous tasks and bolstering performance.

## Express.js, a versatile Node.js web application framework, empowers the creation of resilient APIs adept at managing HTTP requests and delivering timely responses.

## Integrating with MongoDB database and Snowflake data warehouse, the backend orchestrates seamless data storage, retrieval, and processing operations, ensuring robust functionality and scalability.

## Data Storage and Processing (MongoDB, Snowflake):

## MongoDB acts as the cornerstone NoSQL database, housing user profiles, skill sets, project specifics, and various other vital application data.

## The Snowflake data warehouse stands as a robust solution for handling vast amounts of both structured and semi-structured data, offering scalability and optimal performance crucial for analytics and reporting needs.

## Bridging MongoDB with Snowflake is facilitated through tailored connectors or by implementing efficient ETL (Extract, Transform, Load) procedures.

## Analytics and Reporting (Power BI):

## Power BI is seamlessly woven into the system, offering sophisticated data visualization, analytics, and reporting functionalities.

## Information sourced from Snowflake and MongoDB undergoes extraction, transformation, and loading processes into Power BI, empowering the creation of dynamic dashboards, reports, and visual representations.

## Through Power BI dashboards, comprehensive insights into employee skills, certification statuses, project involvements, and other pertinent metrics are readily available, facilitating informed decision-making processes.

## ETL Processes (dbt - Data Build Tool):

## dbt serves as the data modeling and transformation layer, providing a SQL-based workflow for building data pipelines and transformations.

## dbt can be used to perform transformations on raw data from MongoDB and Snowflake, creating structured datasets optimized for reporting and analytics in Power BI.

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

* + **React**: Utilized for building dynamic and responsive user interfaces, React provides a component-based architecture and virtual DOM for efficient rendering.
  + **MongoDB**: Employed as the database solution for its flexibility with unstructured data, MongoDB stores user profiles, skill information, and project details in JSON-like documents.
  + **Node.js**: Used as the backend runtime environment, Node.js enables server-side logic execution with its non-blocking I/O model, integrating seamlessly with JavaScript on both frontend and backend.
  + **Express**: Chosen as the web application framework for Node.js, Express simplifies the process of building robust APIs with its minimalist design and middleware support.
  + **Bootstrap**: Employed for frontend development to ensure rapid prototyping and consistent UI design across different devices, Bootstrap offers a plethora of pre-designed components and responsive layouts.
  + **Power BI**: Integrated for data visualization and analytics, Power BI empowers users to create interactive reports and dashboards, extracting insights from the Skill Matrix System's data.
  + **Material-UI**: Utilized for UI component libraries, Material-UI provides pre-built React components following Google's Material Design guidelines, facilitating the creation of visually appealing interfaces.
  + **DBT (Data Build Tool):** Incorporate DBT into your technology stack for modeling, transforming, and testing your data in Snowflake. DBT's SQL-based approach enables you to define data transformations, manage dependencies, write tests, and generate documentation within your data warehouse environment.
  + **Python**: To generate fake data with Python, utilize libraries like Faker or Random for user profiles, skills, certifications, project experiences and events . Define functions to generate data based on predefined templates. Then, establish a connection to MongoDB using the pymongo library with appropriate parameters like host, port, database name, and credentials if needed. Finally, insert the generated data into MongoDB collections using pymongo functions tailored to each collection.

# Authentication and Authorization

**1. User Roles:**

1. **Admin:**
   * **Responsibilities:**
     + Create new user accounts.
     + Approve or reject certificates and project experiences.
     + Create , Delete and Modify Events.
     + Manage system settings and configurations.
   * **Permissions:**
     + Full access to all administrative functionalities.
     + Can access user's profile data.
2. **User:**
   * **Responsibilities:**
     + Maintain their own profile, including skills and project experiences.
     + Register in events.
     + Submit certificates and project experiences for approval.
   * **Permissions:**
     + View their own profile data.
     + Submit certificates and project experiences for approval.
     + View approved certificates and project experiences.

**2. Authentication:**

• Describe the authentication process used to verify user identities, such as:

• JWT (JSON Web Tokens): Explain how users obtain JWT tokens upon successful login.

• Authentication API: Detail the endpoints and methods used for user authentication.

**3. Authorization:**

• Specify the authorization rules for different endpoints and actions, including:

• Only admins can access endpoints for creating new users, approving certificates, project experiences and creating Events.

• Users can only access and manage their own profile data.

A screen shot of a sign in

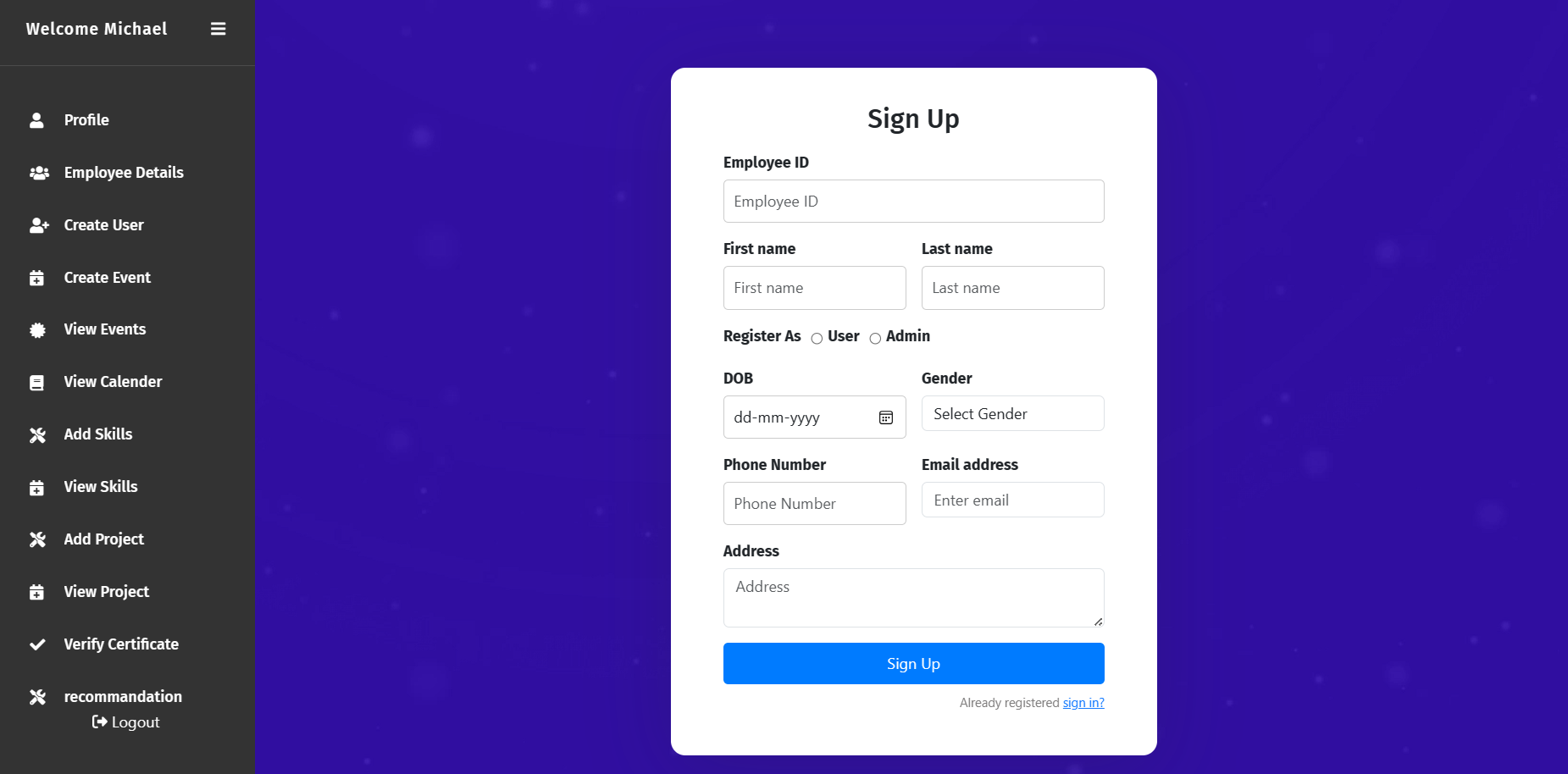
Description automatically generated

**4. New User Creation:**

• Outline the process for creating a new user account:

• Only admins can create new user accounts through a dedicated admin interface or API endpoint.

• Upon creation, a temporary password is generated and emailed to the new user.



A blue and black text

Description automatically generated

**5. Password Reset:**

• Describe the process for resetting forgotten passwords:

• Users can request a password reset by providing their email address.

• An email containing a password reset link is sent to the user's email address.

• The link expires after a set period or upon successful password reset.

**6. Admin Workflow:**

• Explain the workflow for approving certificates and project experiences:

• Certificates and project experiences submitted by users are marked as pending approval.

• Admins review and approve/reject pending submissions.

• Admins can create events , which users can register.

**7. Email Notifications:**

• Detail the email notifications sent to users and admins for important events, including:

• **New user creation**: Notify users of their account creation and provide instructions to change their temporary password.

• **Password reset**: Notify users of their password reset request and provide a link to reset their password.

• **Certificate approval**: Notify Admins to check for Users Certificate.

• **Event registration** : Notify users that they have registered for the event

A white background with a black and white text

Description automatically generated with medium confidence

# Data Model

### Entity Relationships:

A screenshot of a computer

Description automatically generated

* **TABLE EVENTS:** 
  + NAME, TOTAL\_CAPACITY, START\_DATE, END\_DATE, TRAINEE\_NAME, MODE, MEETING\_LINK, VENUE
* **TABLE PROJECTDATAILS:** 
  + EMPLOYEE\_ID, PROJECT\_NAME, TECHSTACK\_USED, STATUS
* **TABLE CERTIFICATEDETAILS:**
  + EMPLOYEE\_ID, CERTIFICATE\_NAME, ISSUING\_ORGANIZATION, ISSUE\_DATE, EXPIRE\_DATE: CREDENTIAL\_ID(Primary Key) , CERTIFICATE\_STATUS
* **TABLE SKILLDETAILS:** 
  + EMPLOYEE\_ID, SKILLS, RATING
* **TABLE USEREVENT:** 
  + EVENT\_NAME, USER\_EMAIL
* **TABLE USERINFOS:** 
  + EMPLOYEE\_ID (primary key), FIRST\_NAME, LAST\_NAME, EMAIL, USERTYPE, DATE\_OF\_BIRTH, GENDER, PHONENO, ADDRESS

1. **Recommendation Model:**

**1 Overview:**

The recommendation model is designed to suggest suitable Learning Events based on their skills, certifications, and proficiency levels. Leveraging text preprocessing techniques and machine learning algorithms, the model identifies employees whose profiles align closely with the user's specified requirements.

**2 Data Preparation:**

The model assumes the existence of a dataset, merged\_data, containing comprehensive information about employees, including their certificates, skills, and proficiency levels. This dataset is transformed into a pandas DataFrame (df) to facilitate further processing and analysis.

**3 Defining Required Tech Stack and Proficiency:**

The recommendation model incorporates user-defined requirements regarding the desired tech stack and proficiency levels. These requirements are converted to lowercase and concatenated into a single string, representing the user's criteria for candidate selection.

**4 Feature Extraction with TF-IDF:**

Text data is converted into numerical vectors using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer from the scikit-learn library. The fit\_transform method is applied to the stemmed text data (df['stemmed\_text']) to generate feature vectors (X) representing each employee profile. TF-IDF assigns weights to terms based on their frequency in a document and inverse frequency across documents, capturing their importance in distinguishing profiles.

**5 Calculating Similarity:**

Cosine similarity is computed between the feature vectors of employee profiles and the feature vector representing the user's specified requirements. This metric quantifies the similarity between two vectors by measuring the cosine of the angle between them. Higher cosine similarity scores indicate greater alignment between employee profiles and user preferences.

1. **Training Recommendation:**

For each Employee , model gives the best 5 learning Events that they should participate in order to boost their skills

**Appendix Title**

Document Title