**JTalentify**

Employee Skill and Competency Tracking System

Contents

[1 Introduction 2](#_Toc163818487)

[1.1 Project Overview. 2](#_Toc163818488)

[1.2 Technolog Stack. 2](#_Toc163818489)

[1.3 Project Architecure Diagram. 3](#_Toc163818489)

[2 Full Stack Development 4](#_Toc163818491)

[2.1 Frontend Development 4](#_Toc163818492)

[2.1.1 Technologies. 4](#_Toc163818493)

[2.1.2 Key Features. 4](#_Toc163818493)

[2.1.3 Component Structure 5](#_Toc163818493)

[2.1.4 API Integration. 7](#_Toc163818493)

[2.2 Backend Development 7](#_Toc163818492)

[2.2.1 Technologies. 7](#_Toc163818493)

[2.2.2 API Endpoints. 7](#_Toc163818493)

[2.2.3 Authentication (JWT). 8](#_Toc163818493)

[2.2.4 Database Models 8](#_Toc163818493)

[3 Data Engineering 8](#_Toc163818494)

[3.1 ETL Process. 8](#_Toc163818495)

[3.1.1 Raw Layer 8](#_Toc163818496)

[3.1.2 Staging Layer 8](#_Toc163818497)

[3.1.3 Transformation Layer 8](#_Toc163818497)

[3.1.3 Reporting Layer. 9](#_Toc163818497)

[3.1.4 Python ETL Scripts. 9](#_Toc163818498)

[4 Data Science 9](#_Toc163818499)

[4.1 Problem Definition. 9](#_Toc163818500)

[4.2 Exploratory Data Analysis(EDA). 9](#_Toc163818501)

[4.3 Feature Engineering. 9](#_Toc163818493)

[4.4 Model Building. 9](#_Toc163818493)

[5 Reports and Dashboards 10](#_Toc163818502)

[5.1 Reporting Folder Structure 10](#_Toc163818503)

[5.2 Admin Dashboards. 10](#_Toc163818493)

[5.3 Visualizations. 10](#_Toc163818493)

[6 Conclusion 10](#_Toc163818504)

[7 References 10](#_Toc163818507)

# Introduction

## Project Overview

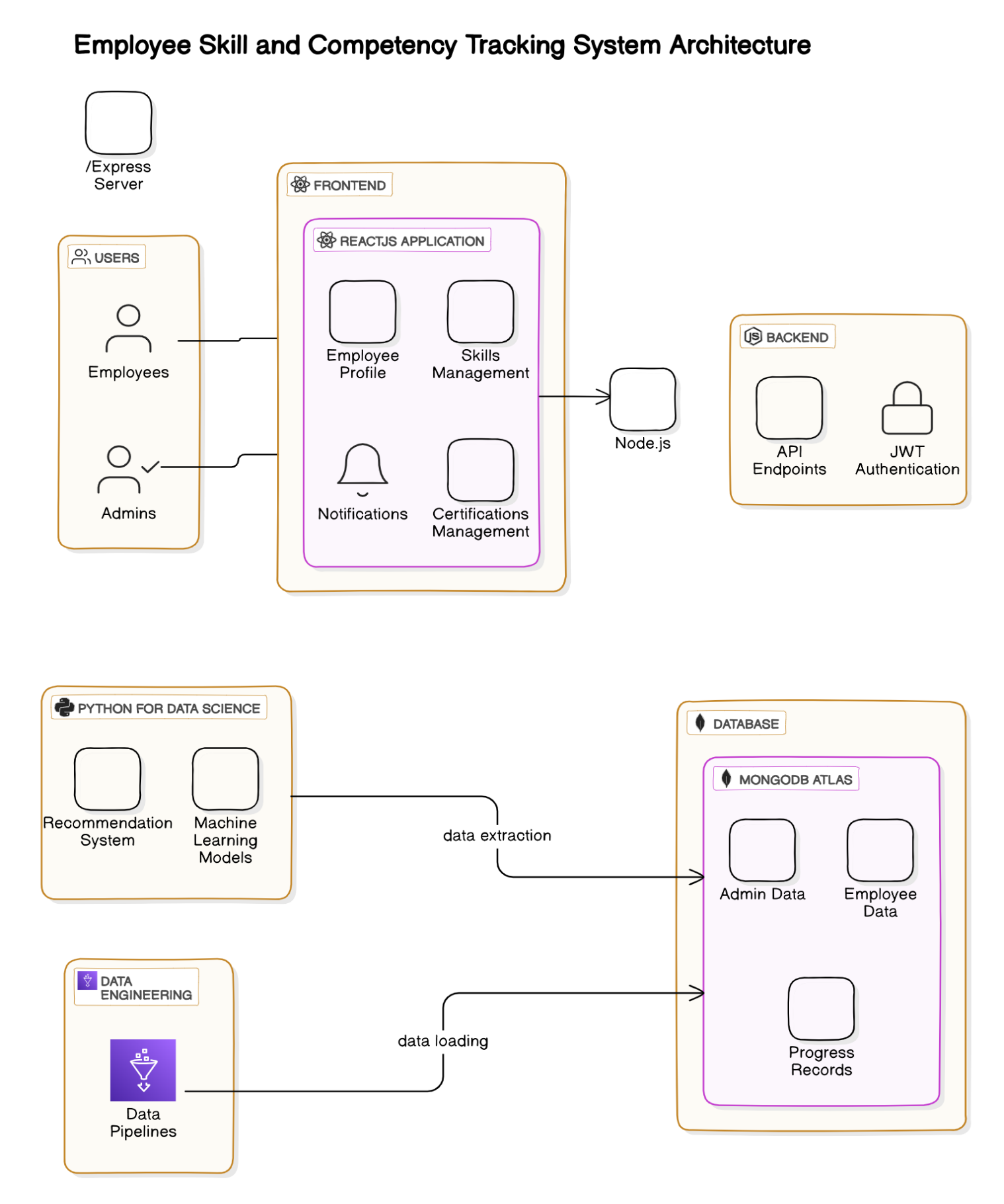
JTalentify Employee Skill and Competency Tracking System is designed to assess and manage employee skills and certifications within an organization. It offers an admin interface to track employees' skills and certification competency levels, skill progress, and skill status, while employees can view their profiles, add skills & certifications, and receive recommended courses based on their current roles and skill sets.

This system provides a complete cycle of operations, from data collection to machine learning-based skill recommendations, helping to ensure that employees remain aligned with their career development and organizational needs.

## Technology Stack

* **Frontend**: ReactJS
* **Backend**: Node.js, Express.js
* **Database**: MongoDB Atlas
* **API Communication**: Axios
* **Authentication**: JWT (JSON Web Token)
* **Data Engineering**: Python (ETL scripts, Data Transformation)
* **Data Science**: Python (pandas, Sklearn, etc.)

## Project Architecture Diagram



# Full Stack Development

## Frontend Development

### Technologies

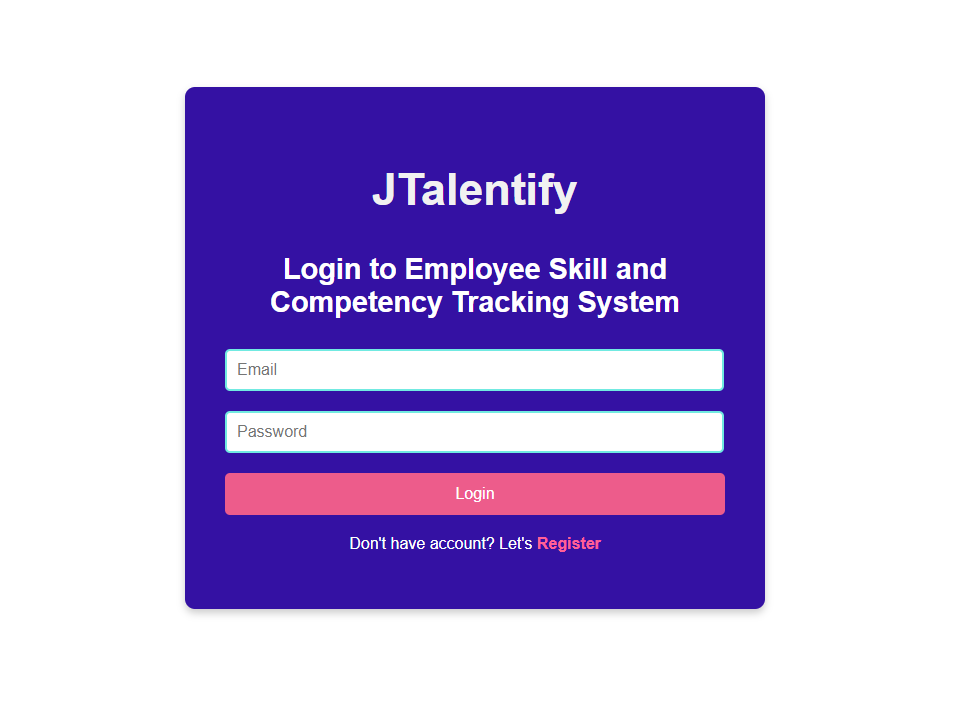
* **ReactJS:** For building the user interface.
* **React Router**: Used for routing between different pages (Login, Register, Dashboard).
* **Axios**: For handling API calls.

### Key Features

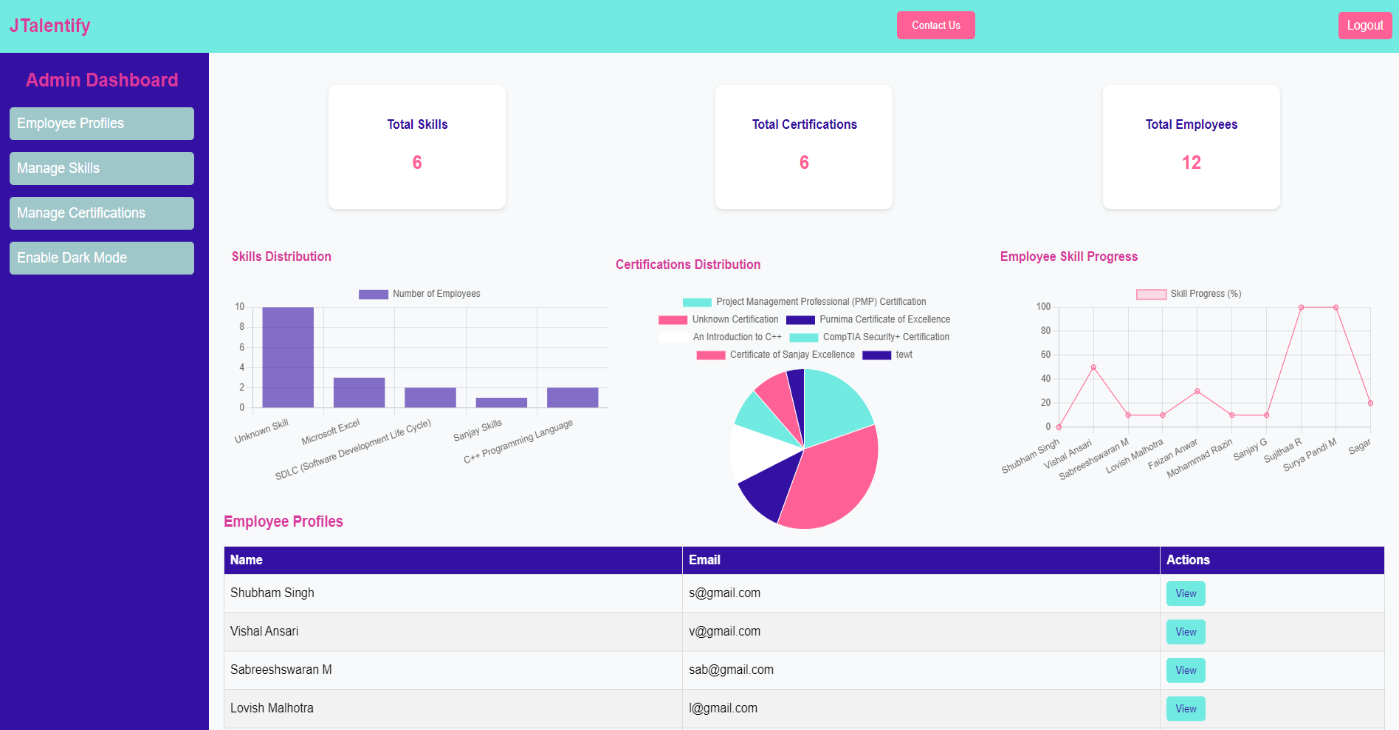
* **Login and Authentication:** Employees and admins log in via JWT. The app redirects based on the user role.
* **Admin Dashboard:** Admins can manage employees, skills, and certifications. They can also visualize employee data.
* **Employee Profile Management:** Employees can view and update their skill set, certifications, and track their progress.
* **Data Visualization:** Interactive charts and graphs representing skill distribution, certification statistics, and employee progress.
* **Role-Based Access Control:** Different access rights and functionalities are available based on user roles (admin vs. employee). Ensures data security and integrity by restricting sensitive actions to authorized users.
* **Notifications:** Real-time notifications for users when skills or certifications are added or updated, enhancing user engagement.

### Component Structure

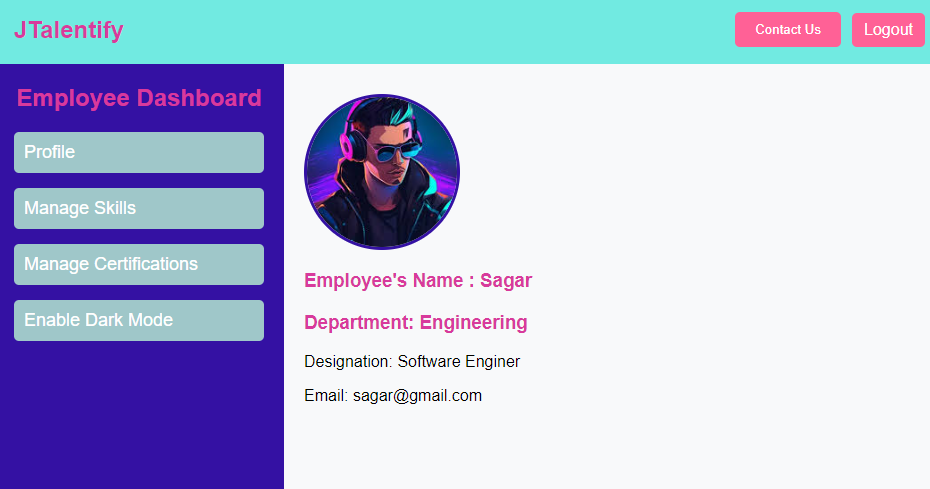
* **Login Page**: Handles user login and authentication.

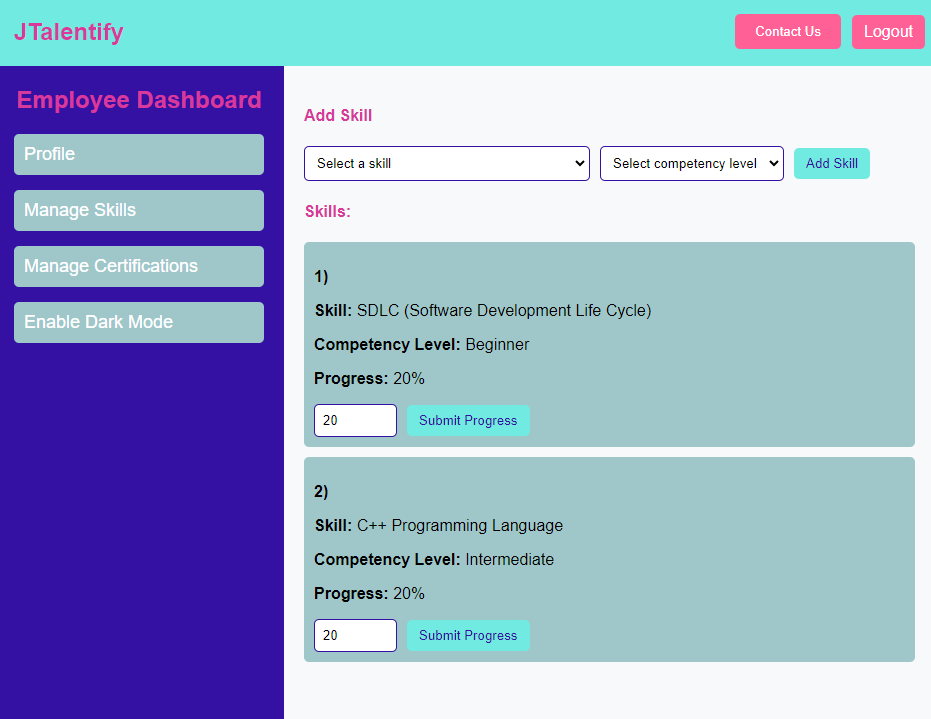


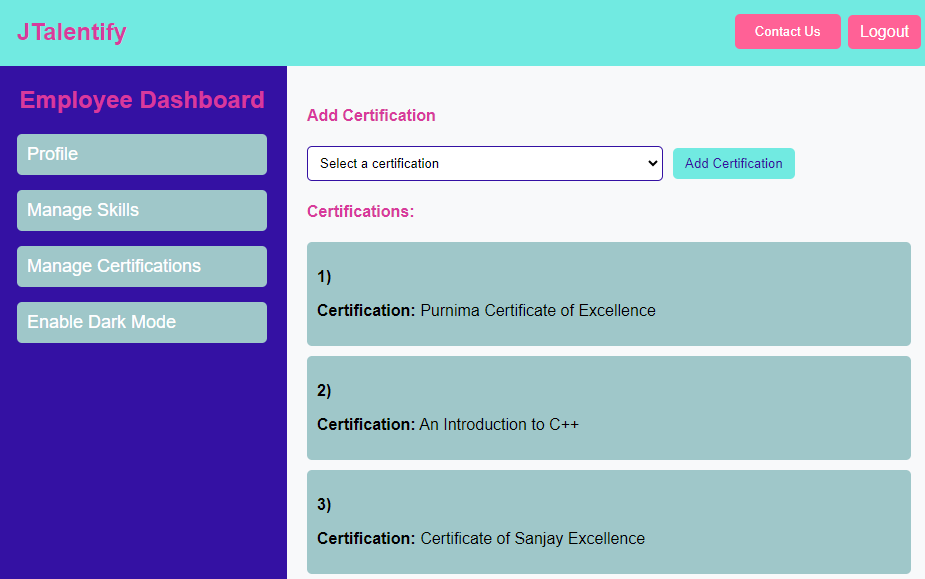
* **Admin Dashboard**: Displays employee data, allows skill/certification management.

****

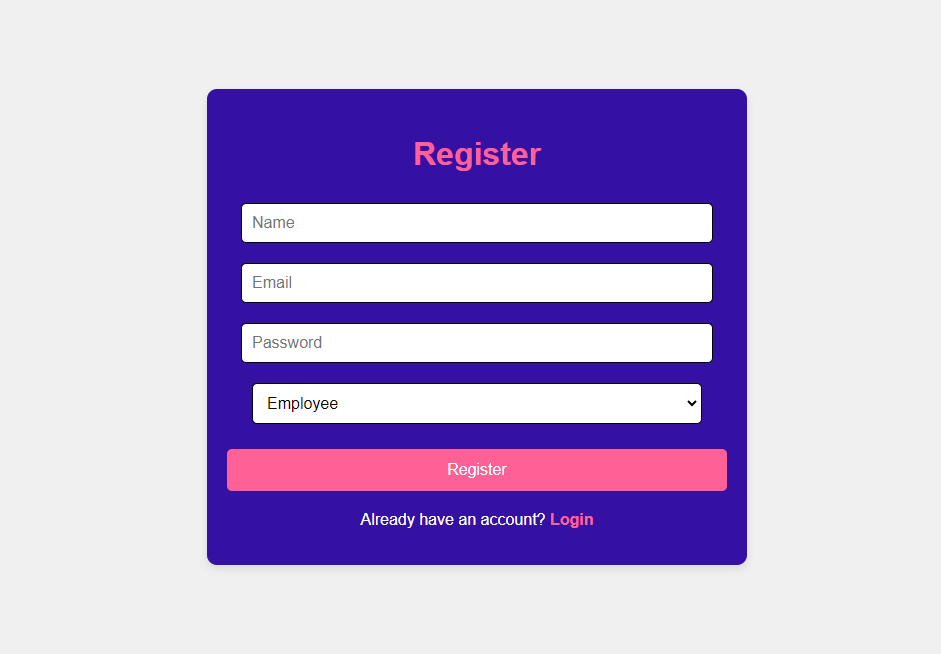
* **Employee Profile Pages**: Displays current employee data and allows employees to add/update skills & certifications.







* **Register Page**: Allows new users to create accounts with necessary details such as name, email, password, and role selection.



### API Integration

* **Login/Authentication API**: JWT-based authentication for both employees and admins.
* **Employee Management APIs**:

1. /add-skill: Adds a skill to an employee.
2. /view-certifications: Fetches available certifications.
3. /update-skill: Updates an employee's skill level.
4. /fetch-all-employees: Retrieves all employee records for admin view.

## Backend Development

### Technologies

* **Node.js**: Backend server.
* **Express.js**: API routing and middleware.
* **MongoDB Atlas**: NoSQL database for employee, skills, and certification data.

### API Endpoints

* **POST /login**: Authenticates users with JWT.
* **GET /employees**: Retrieves all employees.
* **DELETE /admin/remove-certifications**: Removes a certification.
* **DELETE /admin/remove-skills**: Removes a skill.
* **GET /skills**: Fetches a list of all available skills.
* **GET/certification**: Fetches a list of all available certifications.
* **POST /certifications/add**: Admin can add new certifications.
* **POST /skills/add**: Admin can add new skills.

### Authentication (JWT)

JWT tokens are generated upon login and stored in local storage. Each request to protected routes (employee profile, admin dashboard) is verified with the token for security.

### Database Models

* **Employee**: Stores employee information.
* **Admin**: Stores Admin information.
* **Skills**: Stores skill information.
* **Certifications**: Stores certification details.
* **EmployeeSkills**: Links employees to their skills, with information about proficiency.
* **EmployeeCertificates**: Links employees to the certifications they have completed.

# Data Engineering

## ETL Process

### Raw Layer

Data is extracted from the MongoDB database, which stores employee data (skills, certifications, etc.) in JSON format. Generated data, specific to departments and roles, is also collected.

### Staging Layer

In the staging layer, the raw data is cleaned and standardized:

* **Cleaning**: Removing duplicates (e.g., duplicate skills).
* **Standardizing**: Converting all text to lowercase, trimming whitespaces, ensuring consistency in department and role names.
* **Handling Missing Values**: Filling missing skill level or progress information with appropriate defaults.

### Transformation Layer

**Transformations**:

* **Normalization**: Normalizing skills data (e.g., splitting multi-word skill names).
* **Feature Engineering**: Creating new columns for skill progression and assigning certification completion status.
* **Aggregations**: Aggregating employee data by department to assess skill distribution.

For Reporting-ready datasets are exported. These include:

* transformed\_skills.csv
* transformed\_certifications.csv
* transformed\_courses.csv
* transformed\_employees.csv
* transformed\_employee\_certificates.csv
* transformed\_employee\_skills.csv
* transformed\_employee\_courses.csv

### Reporting Layer

The final transformed data is stored in CSV or other tabular formats.

Reporting-ready datasets are exported for data science processing. These include:

* curated\_final\_ale.csv
* competency\_summary\_report.csv
* course\_enrollment\_summary\_report.csv
* certification\_summary\_report.csv
* skill\_summary\_report.csv
* employee\_certificate\_summary\_report.csv
* employee\_course\_summary\_report.csv
* average\_course\_enrollment\_by\_department.csv
* average\_skill\_progress\_by\_department.csv

### Python ETL Scrips

* **Extract**: Scripts for reading from the MongoDB database and loading raw JSON.
* **Transform**: Data cleaning and transformation using pandas.
* **Load**: Final data exported into CSV for reporting and further analysis.

# Data Science

## Problem Definition

The goal is to recommend courses and learning paths to employees based on their current skill set, completed certifications, and job requirements. This helps employees grow professionally by aligning their skill development with the organization's needs.

## Exploratory Data Analysis (EDA)

* **Skill Distribution**: Analyzed the number of employees with each skill and certification.
* **Progress Analysis**: Assessed the average skill progression across departments.
* **Visualization**: Used matplotlib and seaborn for bar charts, heatmaps showing correlations between roles and required skills.

## Feature Engineering

* **Skills Proficiency**: Created new columns for skill proficiency (beginner, intermediate, advanced).
* **Role-Specific Skill Requirements**: Added columns showing skills relevant to specific roles.
* **Certification Trends**: Engineered features representing the trend of employees obtaining certifications over time.

## Model Building

* **Recommendation Model**: Trained a KNN model to recommend the top 3 courses to an employee to pursue next based on their role, current skill set, and certifications.
* **Input Features**:
  + Employee role
  + Department
  + Completed certifications
  + Current skill level
* **Model Evaluation**: Achieved overall average distance between test employees and their nearest neighbours 0.027.

# Reports and Dashboards

## Reporting Folder Structure

* **Raw Data**: Contains raw JSON data from the MongoDB database.
* **Staging Data**: Contains cleaned, intermediate data files used for transformations.
* **Transformed Data**: Final transformed CSVs ready for reporting and model training.
* **Reports**: Final CSV reports generated for admin viewing.

## Admin Dashboards

* **Skills Overview**: A dashboard showing skill distribution across departments.
* **Certification Trends**: A dashboard tracking certifications earned over time.
* **Employee Progress**: A dashboard highlighting employees' skills and certifications progress.

## Visualizations

* **Certification Breakdown**: A pie chart showing the percentage of employees with certifications.
* **Skills Breakdown**: A bar chart showing the percentage of employees with skills.

# Conclusion

JTalentify Employee Skill and Competency Tracking System provides a comprehensive solution to manage employee skill sets and offer personalized skill development recommendations. By combining full stack development, data engineering, and data science, the system not only helps track skills and certifications but also proactively supports employee growth aligned with organizational needs.

# References

* **ReactJS Documentation**: <https://reactjs.org/docs/getting-started.html>
* **ExpressJS Documentation**: <https://expressjs.com/>
* **Pandas Documentation**: <https://pandas.pydata.org/docs/>
* **Scikit-learn Documentation**: <https://scikit-learn.org/stable/>