**Final Project Documentation**

**Harshavardhan Asadi**

Course Completion and Learning Path Optimization

1. **Project Overview**

**1.1. Brief introduction of the project.**

The **“Course Completion and Learning Path Optimization”** project addresses the essential need for structured employee training and development in today’s rapidly evolving job market. This web-based application is designed to monitor and analyze employee progress through various learning paths and course completions, providing valuable insights into their learning habits and achievements. With a user-friendly dashboard, employees can easily track their current enrollments, completion rates, and certifications, fostering accountability and motivation. Meanwhile, administrators benefit from an intuitive interface for managing learning paths and assigning relevant courses tailored to individual employee needs.

A key feature of this project is the integration of machine learning algorithms to optimize learning paths. By analyzing performance metrics and completion rates, the system can recommend the most effective courses for each employee, ensuring that training aligns with their personal development goals.

**1.2. Purpose of the project.**

The primary purpose of this project is to develop a comprehensive web-based application that monitors employee progress through learning paths and course completions. By providing detailed insights into employee engagement with training materials, the application aims to empower managers to identify areas of strength and opportunities for additional support. This project seeks to provide:

* A centralized platform for tracking employee learning history and course completion rates.
* Visualizations that clarify how training programs contribute to employee development and skill enhancement.
* Tools for assessing individual employee progress, offering tailored recommendations for improvement.
* Data-driven insights to optimize learning paths, ensuring that training is aligned with employee needs and organizational goals, ultimately enhancing performance and productivity.

**1.3. Goals of the project:**

The goals of this project are designed to enhance employee learning experiences and optimize training outcomes. They include:

* **Develop an Intuitive Platform:** Create a user-friendly web-based application that enables employees to track their progress through learning paths and course completions easily.
* **Implement Progress Tracking:** Establish comprehensive tracking mechanisms for employee engagement, including completion rates, certificates earned, and performance metrics.
* **Facilitate Data-Driven Insights:** Analyze training data to identify trends and areas for improvement, providing managers with actionable insights to support employee development.
* **Optimize Learning Paths:** Use machine learning algorithms to recommend personalized learning paths that align with individual employee goals and enhance overall performance.
* **Enhance Managerial Oversight:** Provide managers with tools and dashboards to monitor employee progress, helping them to allocate resources effectively and identify areas needing additional support.
* **Foster Continuous Improvement:** Create feedback loops that enable ongoing evaluation and enhancement of training programs based on employee performance and learning outcomes.
* **Support Skill Development:** Ensure that training initiatives contribute to the continuous professional development of employees, ultimately benefiting organizational productivity and growth.

# 2. Architecture

A diagram of a computer program

Description automatically generated

# 3. Tech Stacks Used

**Frontend:**

* **React JS:** Utilized for building a dynamic user interface, ensuring a smooth and responsive experience for users.
* **Tailwind CSS:** Employed for styling, enabling a modern and consistent design across the application.

**Backend:**

* **Node.js:** Used for server-side logic, ensuring scalability and efficient handling of requests**.**
* **Express.js:** Implemented for creating the RESTful API, facilitating data exchange between the client and server.

**Database:**

* **PostgreSQL:** A relational database used to store employee details, course metadata, enrollment records, and performance metrics, ensuring data integrity and complex query capabilities.

**Additional Tools:**

* **Prisma:** Utilized as an ORM (Object-Relational Mapping) tool for seamless database interactions and migrations.
* **Axios:** Employed for making API requests from the frontend to the backend.
* **Chart.js:** Used for rendering interactive charts to visualize training performance and completion data.

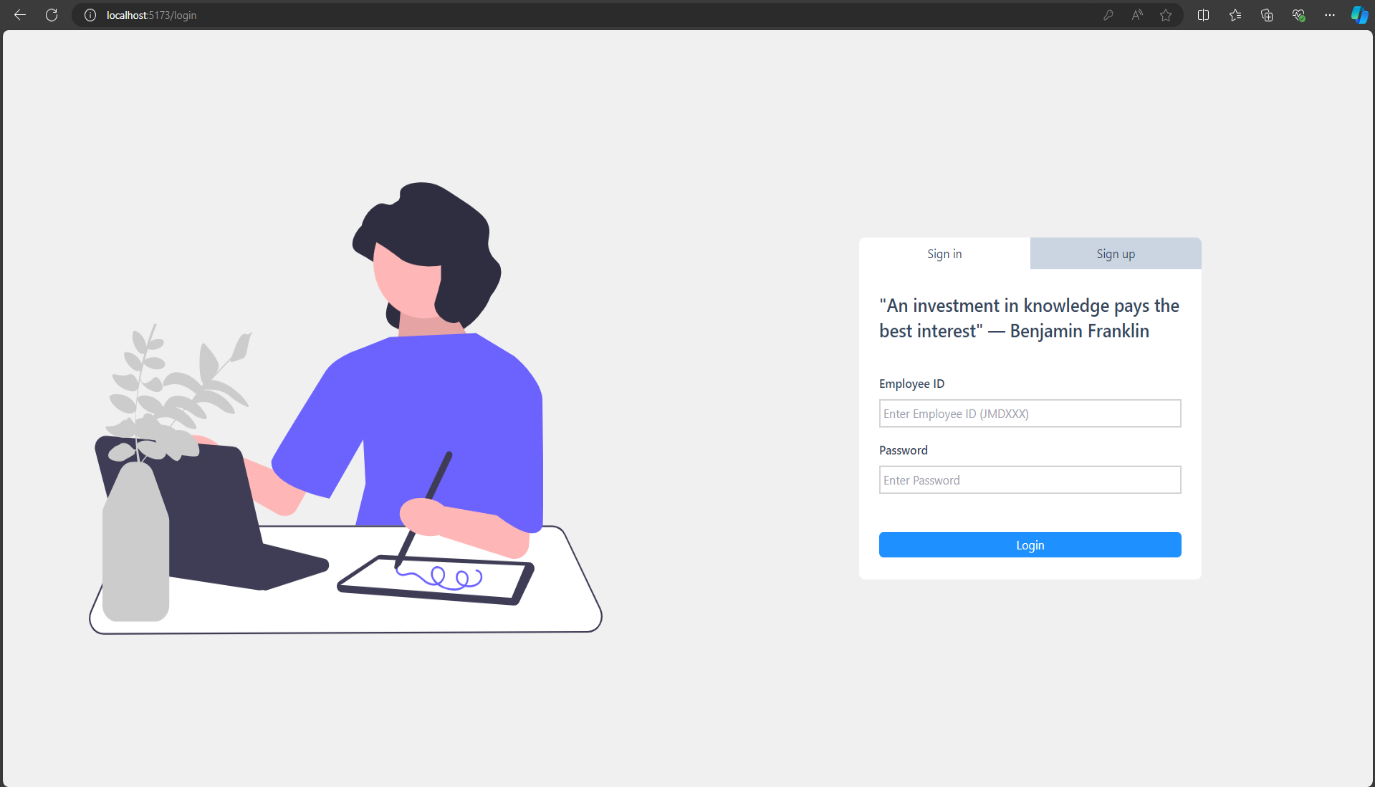
4. Key Features and Functionalities

* **User Progress Dashboards:**
* Interactive dashboards that display real-time data on employee progress through learning paths, including course completions and certificates earned.
* **Course Management:**
* Admin interfaces that allow for the creation, modification, and deletion of courses and learning paths.
* The ability to assign courses to employees based on their roles and development needs.
* **Performance Tracking:**
* Monitoring tools to assess completion rates, performance metrics, and engagement levels, helping to identify high-performing employees and those needing additional support.
* **Personalized Learning Path Suggestions:**
* Machine learning algorithms that analyze employee performance to recommend tailored learning paths, optimizing training based on individual needs and goals.
* **Certificate Generation:**
* Automatic generation of course completion certificates, which can be downloaded and shared by employees upon finishing a course.
* **Reporting and Analytics:**
* Comprehensive reporting features that provide insights into training effectiveness, employee engagement, and skill development, enabling data-driven decision-making.
* **Notifications and Reminders:**
* Automated notifications to remind employees of upcoming courses, deadlines, and any updates related to their learning paths.
* **Feedback and Evaluation Tools:**
* Mechanisms for collecting feedback from employees on courses and learning materials, aiding in continuous improvement of training programs.
* **Responsive Design:**
* A mobile-friendly interface that ensures accessibility and usability across various devices, enhancing user experience.
* **Integration Capabilities:**
* API integrations with other HR and learning management systems, facilitating seamless data exchange and enhancing overall functionality.

5. Modules Built

### 5.1 Authentication Module

* **JWT Sign-In & Sign-Up:**
* Secure user authentication using JSON Web Tokens (JWT) for sign-in and sign-up processes.
* User roles (Admin and Employee) managed through authentication tokens to control access to various features.



### Admin Module

* **Dashboard:**
  + Overview of key metrics such as top trending courses, time spent on courses, total course count, and enrollment count.

A screenshot of a graph

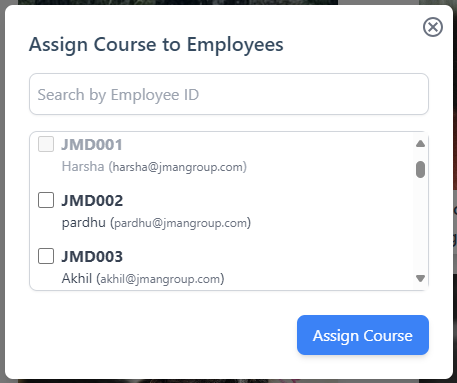
Description automatically generated

* **Course Management**:
* **All Courses**:
  + View and manage all available courses, including PDF views of course content.
  + Access detailed course information, including descriptions, prerequisites, and difficulty levels.

A screenshot of a cell phone

Description automatically generated

* **Enroll Employees & View Course deatils**:
  + Enroll employees in specific courses and view the course details.

A screen shot of a cell phone

Description automatically generated

A screen shot of a computer

Description automatically generated

* **Add Course**:
  + Create and add new courses to the system, specifying details like course name, duration, and associated content.

A screenshot of a computer

Description automatically generated

* **Add Questions to a Course**:
  + Manage the question bank by adding questions related to specific courses for assessments.

A screenshot of a computer

Description automatically generated

* **Employee Reports**:
  + Generate and view reports on employee progress, including completion rates and performance metrics.

A screenshot of a computer

Description automatically generated

* **Notifications**:
  + Send notifications to employees regarding the acceptance or rejection of tests taken, as well as course updates.

A screenshot of a computer

Description automatically generated

### Employee Module

* **Dashboard**:
  + Display metrics such as average time spent on courses each day, total certificates earned, average time spent on course completion, and average test scores.

A screenshot of a computer

Description automatically generated

* **Assigned Courses**:
  + View details of assigned courses, including the ability to complete courses and access generated certificates upon completion.

A screenshot of a computer

Description automatically generated

* + Certificate is generated if admin accepts the exam

A certificate of completion

Description automatically generated

* **Notifications**:
  + Receive notifications sent by the admin regarding the acceptance or rejection of tests taken.

A screenshot of a computer

Description automatically generated

* **Recommended Courses**:
  + Suggest courses and learning paths based on employee performance, utilizing machine learning algorithms to tailor recommendations to individual needs and goals.

A screenshot of a computer

Description automatically generated

Data engineering

# 1. Data Warehouse Architecture

The data warehouse is designed with a multi-layer architecture consisting of three key layers: Staging, Prep, and Reporting. This architecture facilitates efficient data extraction, transformation, and analysis.

A screenshot of a computer

Description automatically generated

### 1.1 Staging Layer

**Purpose**: This layer serves as the initial repository for raw data extracted from the PostgresSql database.

**Data Ingestion**: The source data tables are inserted into the staging layer.

**Data Source**:

* LearningPathMap
* LearningPath
* CourseEnrollment
* User
* Questions
* QuestionBank
* CourseEngageLogs
* Employee

### 1.2 Prep Layer

**Purpose**: This layer is responsible for data cleaning and transformation.

**Data Cleaning Process**:

**Removing Null Values**: Any rows with missing values are identified and handled according to predefined business rules.

**Removing Duplicates**: Duplicate records are identified and eliminated to ensure data integrity.

**Data Type Adjustment**: Columns are converted to appropriate data types to facilitate efficient querying and analysis.

**Data Movement**: Once the data is cleaned, it is moved from the staging layer to the prep layer.

### 1.3 Reporting Layer

**Purpose**: The reporting layer is where the final data is organized for analysis and reporting.

**Data Structure**: The reporting layer of the application consolidates data from multiple sources to provide meaningful insights and visualizations. It helps in tracking employee progress, course completion rates, and performance metrics, allowing managers to make data-driven decisions.

* **merger.csv**: Contains merged data of employee details, course enrollments, performance scores, and completion status, providing a comprehensive view of training history.
* **employee\_performance.csv**: Tracks individual employee performance metrics such as test scores, completion rates, and time spent on courses, highlighting strengths and areas needing improvement.
* **best\_learning\_path.csv**: Analyzes course engagement and outcomes to determine the most suitable learning path for employees, identifying which paths yield the highest performance improvements.

### 1.4 Schema

**Schema Design**: The data warehouse employs a fact constellation schema design, where multiple fact tables are connected to shared dimension tables. This design supports more complex queries and enhances flexibility, allowing different business processes to be analyzed within the same schema.

**Insights Generation**: By joining the fact and dimension tables, the organization can generate meaningful insights regarding employee performance, learning path optimization and course recommendation.

A screenshot of a computer

Description automatically generated

# 2. Data Flow Overview

The data flow within the data warehouse can be summarized as follows:

**Extraction**: Data is extracted from the PostgresSql database and loaded into the staging layer.

**Transformation**: The data is cleaned and transformed in the prep layer.

**Loading**: The cleaned data is then structured into fact and dimension tables in the reporting layer, ready for analysis.

Data Science

# 1. Exploratory Data Analysis (EDA)

### 1.1 Data Overview and Basic Statistics

**Data Structure**: The shape of the dataset was examined to determine the number of rows and columns, which helped in understanding the scope and depth of the data.

**Data Types (dtypes)**: Each column's data type was checked to confirm that numerical fields were properly encoded as integers or floats, and categorical fields were correctly identified as objects (strings).

**Min/Max Values**: The minimum, maximum, and range of values for each numeric column were calculated to understand the spread and variability of data.

**Descriptive Statistics**: Using describe() in Pandas, summary statistics such as mean, median, standard deviation, and quartiles for numeric columns were generated. This step helped in identifying potential outliers or skewness in the data.

* **Mean**: Average performance scores across all employees.
* **Standard Deviation**: Variability in manager ratings across the dataset.

### 1.2 Handling Missing Values and Null Data

**Null Values**: Missing or incomplete data were identified across columns. Null values were handled by either imputing values or removing rows with missing data, depending on the extent of the missingness.

**Outlier Detection**: Outliers were detected in certain columns, such as unusually high or low training\_score. These outliers were addressed through either data transformation or removal, depending on their impact on the overall analysis.

# 2. Hypothesis Framing and Visualization

In this project, we hypothesize that an employee’s performance metrics, such as time spent on courses, learning rate, and average test scores, can be strong indicators for predicting the most suitable learning path for the individual. By analyzing these performance factors, we can recommend learning paths that are most aligned with the employee’s capabilities, learning pace, and improvement trajectory.

To validate this hypothesis, the following key features of employee performance have been used as predictors:

* **Time Spent on Courses**: The amount of time an employee dedicates to completing courses is a critical factor in assessing engagement and learning efficiency.
* **Learning Rate**: This reflects how quickly an employee progresses through their assigned courses, giving insight into how fast they grasp the training material.
* **Average Test Scores**: The normalized test scores serve as a direct measure of an employee’s comprehension and retention of the material covered.

By feeding these features into a machine learning model (such as the Random Forest classifier), we aim to predict the **best-fit learning path** for each employee. This allows us to tailor recommendations based on their learning behavior and performance trends, leading to more personalized training experiences and better overall skill development within the organization.

# 3. Statistical Tests

To validate the relationships between different variables, the following statistical tests were conducted:

**Chi-Square Test**: Used to test the association between categorical variables.

**ANOVA Test**: Applied to determine if there are any statistically significant differences between the means of multiple groups (e.g., training scores and promotion status)A screenshot of a computer screen

Description automatically generated.

# 4. Model Training

Several machine learning algorithms were used to build classification models. Each of these models has its own strengths and weaknesses, and they were selected to explore how well different types of algorithms perform on the promotion prediction task.

### 4.1 Models Used

* Random Forest
* K-Nearest Neighbors

### 4.2 Model Evaluation

**4.2.1 Training and Testing Process**

**Data Splitting**: The dataset was split into training and testing sets to assess the model's performance on unseen data. 80% of the data was used for training the models, while the remaining 20% was held out for testing.

**Feature Selection**: All relevant features, such as the completion rate, test score, success rate, time spent were included in the model training.

##### **4.2.2 Metrics Used for Evaluation**

The machine learning models used to predict the best learning path for employees showed varying levels of accuracy. The **“Random Forest classifier”** achieved an accuracy of **“33%”,** while the **“K-Nearest Neighbors (KNN)”** algorithm demonstrated a better performance with an accuracy of **“47%”**. Although both models offer some predictive capabilities, the KNN model currently outperforms Random Forest in identifying the most suitable learning paths based on employee performance metrics like time spent, learning rate, and test scores. This suggests that further optimization and tuning may be required to enhance the accuracy of both models.

A blue and green rectangles

Description automatically generated

# 5. Feature Importance Visualization

After training the models, feature importance was visualized to understand the most influential factors driving employee learning path optimization.

**Appendix Title**

Document Title