**Course Completion and Learning Path Optimization**

Major Project

Contents

[1 Project Overview 2](#_Toc163818487)

[1.1.1 Brief introduction of the project. 2](#_Toc163818488)

[1.1.2 Purpose of the project. 2](#_Toc163818489)

[1.1.3 Goals of the project: 2](#_Toc163818490)

[2 Project Scope 2](#_Toc163818491)

[2.1.1 Web app support 2](#_Toc163818492)

[2.1.2 Key features and functionalities of the web app. 3](#_Toc163818493)

[3 Architecture and Technology Stack 3](#_Toc163818494)

[3.1.1 Overall architecture of the mobile app. 3](#_Toc163818495)

[4 Solutions 6](#_Toc163818499)

[4.1.1 Full Stack 6](#_Toc163818500)

[4.1.2 Data Engineering 8](#_Toc163818501)

4.1.3 Data Science………………………………………………………………………………………………………..9

**5 User Interface……………………………………………………………………………………………………**10

# Project Overview

### Brief introduction of the project.

JLearn is an employee learning management system that monitors course completion and optimizes learning paths. The platform enables employees to track their learning progress while providing administrators with tools to manage learning paths and assign courses. Using machine learning, the system recommends optimal courses to improve employee performance.

### Purpose of the project.

The primary purpose of JLearn is to provide a comprehensive solution for managing and optimizing employee learning paths within an organization. With the increasing need for continuous learning and development in today’s fast-paced industries, it is crucial to ensure employees are not only completing assigned courses but are also on the most effective learning paths to improve their skills and performance. JLearn addresses this by combining full-stack development, data engineering, and machine learning technologies to:

1. **Track Employee Progress:** Monitor course completion rates, performance scores, and certification achievements in real-time, allowing employees to stay informed of their learning progress.
2. **Administer Learning Paths:** Provide administrators with tools to create, assign, and manage learning paths tailored to individual employees or groups based on their job roles, interests, and performance metrics.
3. **Enhance Employee Performance:** Leverage machine learning algorithms to recommend courses that can have the highest impact on employee performance, ensuring that each employee is on an optimized learning path that aligns with organizational goals.
4. **Data-Driven Decision-Making:** By integrating data science techniques, the platform offers insights into which courses and learning paths yield the quickest improvements or performance gains, empowering managers to make data-driven decisions on employee development.

### Goals of the project:

* Develop a learning platform that is intuitive and easy to navigate for both administrators and employees, ensuring a seamless user experience.
* Create Intuitive Learning Progress Dashboards
* Optimize Learning Paths Using Data Science
* Facilitate Performance Tracking and Certification

# Project Scope

### Web app support

|  |  |
| --- | --- |
| Platforms | Version |
| Google Chrome | 97 and above |
| Microsoft Edge | 97 and above |

### Key features and functionalities of the web app.

**User Registration and Authentication:**

User-friendly registration process for both admins and employees

Secure authentication mechanisms, including email/password.

**Personalized Admin Dashboard**:

Customized dashboard for admins to create courses, assign courses, create learning path, assign learning path to employees, track performance of employees, employee course details.

**Personalized Employee Dashboard**:

Customized dashboard for each employee, displaying relevant information such as performance stats, course details, completion percentages, detailed module view, view certificates, details profile page to view his stats.

**Responsive Design:**

Responsive user interface for optimal viewing and usability across different screen sizes

# A diagram of a software system Description automatically generated with medium confidenceArchitecture and Technology Stack

### Overall architecture of the mobile app.

JLearn is built using a modern and scalable architecture that seamlessly integrates front-end, back-end, data engineering, and machine learning components. The system is designed to be modular, allowing for easy updates, maintenance, and scalability as the user base grows. Below is a detailed breakdown of each component of the architecture:

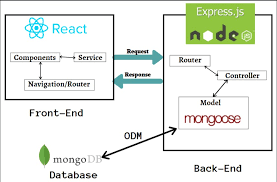
**Front end Application:**

* **Technology Stack:** React.js, Tailwind CSS, Flowbite
* **Functionality:** The front-end application of JLearn provides an intuitive and responsive user interface (UI) for both employees and administrators. It serves as the primary interaction layer between users and the system. The key front-end components include:
  + **Employee Dashboard:** Allows employees to track their course progress, view performance metrics, complete quizzes, and access certificates upon course completion.
  + **Admin Dashboard:** Enables administrators to manage courses, create learning paths, assign courses to employees, track employee performance, and receive machine learning-based recommendations for course assignments.
  + **Learning Path Visualization:** The learning paths are displayed as interactive flowcharts, allowing employees and admins to visualize courses and progress easily.

The front-end interacts with the back-end APIs to fetch employee data, course details, progress metrics, and machine learning recommendations, ensuring real-time updates and smooth user experiences.

**Back end Services:**

* **Technology Stack:** Node.js, Express.js, MongoDB, Mongoose, JWT-based authentication
* **Functionality:** The back-end of JLearn serves as the central hub for all data processing, user management, and business logic. It handles the following responsibilities:
  + **User Management:** Authentication and authorization of users (employees and admins) using JWT stored in cookies for secure access.
  + **Course and Learning Path Management:** Admins can create, update, and delete courses, as well as assign them to specific employees. The system also tracks course progress at both the course and module level, and stores metadata such as course tags, completion percentages, and performance scores.
  + **Certification Management:** The back-end handles certificate generation upon successful course completion.
  + **Learning Path Optimization:** Provides APIs to fetch machine learning-based course recommendations that optimize employee learning paths.

The back-end communicates with the front-end via RESTful APIs, ensuring smooth interactions between the two layers. MongoDB is used as the primary database, providing flexibility for storing employee and course data, including metadata on course tags, duration, and performance scores.

**Data Engineering Pipeline:**

* **Technology Stack:** Python, Pandas
* **Functionality:** The data engineering pipeline is responsible for extracting, transforming, and loading (ETL) data from the MongoDB database used by the full-stack application. It also integrates additional data sources, either generated or collected from external platforms, to enhance the machine learning models.

1. A diagram of data storage

   Description automatically generated**Data Extraction:** Data from the full-stack MongoDB database is extracted and ingested into a data warehouse for processing.
2. **Data Transformation:** The pipeline cleans and transforms the data into structured formats suitable for both reporting and machine learning.
3. **Data Modeling:** The transformed data is organized into three layers: raw, staging, and report. Data from the report layer is used to train machine learning models, providing insights into employee performance, course effectiveness, and learning path optimization.

**Machine Learning Pipeline:**

* **Technology Stack:** Python, Scikit-learn, KNN, Pandas, NumPy
* **Functionality:** The machine learning pipeline is the key to JLearn’s ability to optimize learning paths for employees. It uses historical employee data, course completion rates, and performance metrics to recommend courses that will improve employee skills and performance.
  + **Data Preprocessing:** The machine learning models preprocess employee data, filtering out noise and normalizing variables like course duration, difficulty level, and completion rates.
  + **Model Training:** Several model KNN are trained to predict which courses will lead to the highest performance improvements for employees.
  + **Learning Path Optimization:** Based on employee history, learning patterns, and course tags, the machine learning models recommend personalized learning paths for each employee. The system provides a list of courses that are most likely to improve an employee’s performance score.
  + A diagram of a neighbor

    Description automatically generated**Recommendation API:** Once trained, the machine learning models are exposed through an API, allowing the backend to fetch course recommendations when an admin views an employee's profile.

The machine learning pipeline integrates seamlessly with the data engineering pipeline, ensuring that it is constantly updated with new data to provide accurate and up-to-date recommendations.

# Solutions

### Full Stack

**Overview**

JLearn is a full-stack learning management system designed to monitor employee progress and optimize learning paths using machine learning. It features admin tools for course and learning path management, and employee dashboards for tracking course progress and performance.

**1. Features**

**Employee Features**

* **Course Listing:**
  + Employees can view all available courses, including course titles, descriptions, and relevant tags like "Web Development" and "Data Engineering.
* **Enrollment:**
  + Employees can be assigned courses by the admin, and they can view and engage with the content of these courses.
* **Course Progress Tracking:**
  + A progress bar shows the percentage completion of each course based on module completions.
  + Employees can track the completion of individual modules within courses.
* **Employee Learning Path:**
  + Displays the flowchart of learning paths assigned to employees.
  + Employees can view assigned courses, their completion percentage, and track overall learning path progress.
* **Employee Performance Dashboard:**
  + Displays a visual representation of employee performance, including course completion rates and a chart of performance scores based on various course tags.

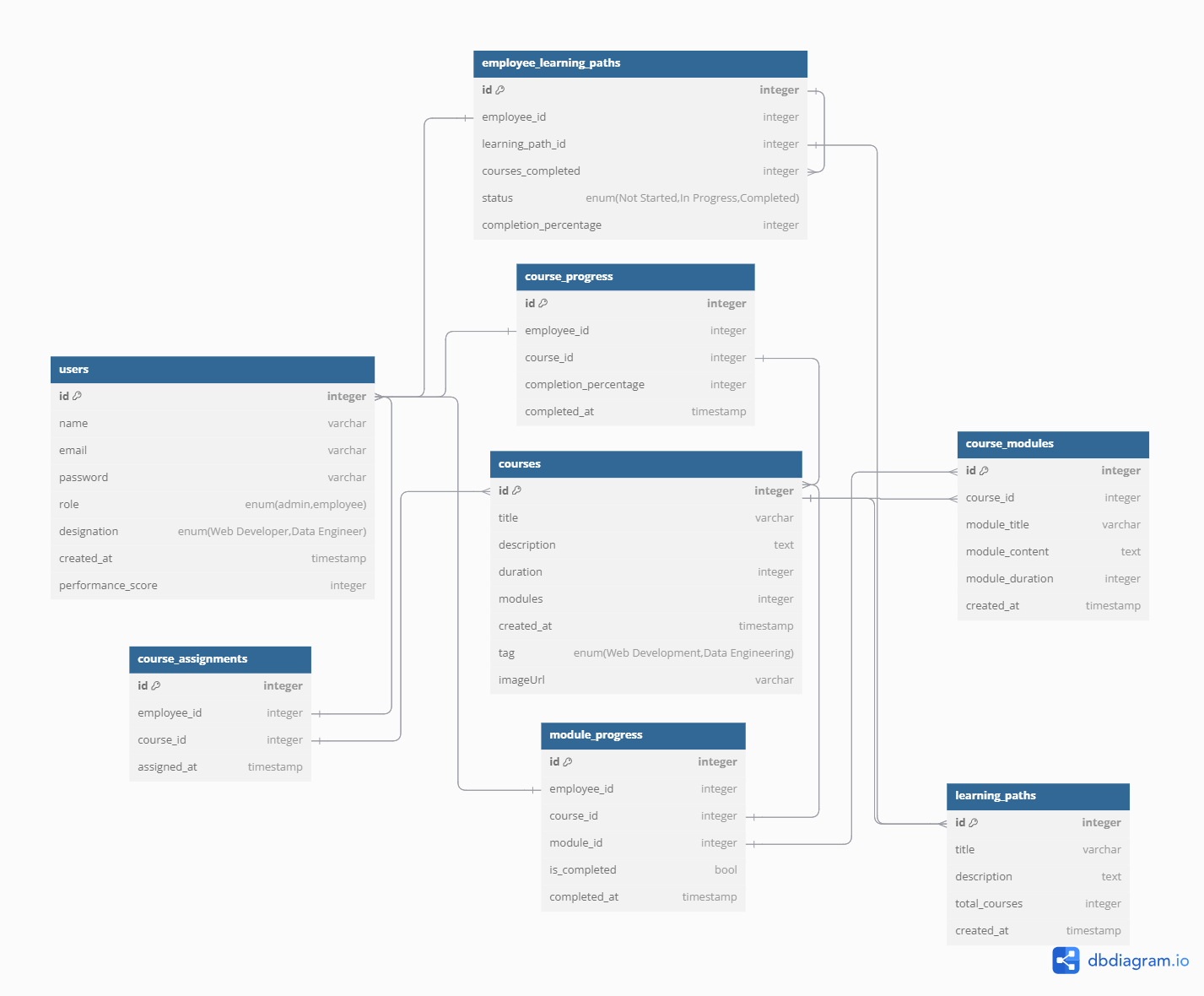
**Admin Features**

* **Dashboard Overview:**
  + Admins can view metrics such as the total number of courses, employees, and learning paths assigned, with access to detailed employee progress and performance scores..
* **Learning Path Management:**
  + Admins can create, manage, and assign learning paths to employees. The system supports tracking employee progress for each learning path, displaying course completion percentages, and marking courses as completed.
* **Course Management:**
  + Admins can create, update, and delete courses. For each course, admins can define details such as the number of modules, course duration, and relevant course tags like "Web Development" or "Data Engineering.
* **Employee Analysis:**
  + Admins can analyze employee performance by tracking time spent, module completion rates, and overall course progression. The system provides detailed insights into individual employee progress.

**2.Database Design**

**Schema Overview**

The database for **JLearn** is designed using MongoDB and Mongoose, with the following main collections and schemas:

* **Users:** Stores employee and admin information, including role (either admin or employee), designation, and overall performance score based on course completion.
* **Courses:** Stores course information such as title, description, duration, number of modules, and a tag for categorization. Course tags include options like "Web Development" and "Data Engineering."
* **Course Modules:** Contains detailed information on individual modules within each course, such as module titles, content, and duration.
* **Course Assignments:** Tracks courses assigned to employees by the admin, including timestamps of when the assignments were made.
* **Course Progress:** Logs progress for each course assigned to an employee, with fields to track the completion percentage and the timestamp of completion.
* **Module Progress:** Tracks the completion status of individual modules by employees, logging whether a module is completed and the timestamp of completion.
* **Learning Paths:** Represents learning paths that consist of multiple courses, storing details like the title, description, and the total number of courses in the path.
* **Employee Learning Paths:** Tracks learning path assignments to employees, storing the status (e.g., "In Progress," "Completed") and overall completion percentage for each learning path.

**Entity Relationships**

* **User ↔ Course Assignment:** One-to-many relationship, where each user (employee) can have multiple course assignments.
* **Course ↔ Course Modules:** One-to-many relationship, where a course can have multiple modules.
* **User ↔ Course Progress:** One-to-many relationship, where a user can have progress records for multiple assigned courses.
* **Module ↔ Module Progress:** One-to-many relationship, where a module can have multiple progress records, tracking completion by various employees.
* **Learning Path ↔ Employee Learning Path:** One-to-many relationship, where learning paths are assigned to multiple employees, tracking their progress within the path.
* **User ↔ Employee Learning Path:** One-to-many relationship, where each user (employee) can be enrolled in multiple learning paths.

### Data Engineering

**Overview**

The JLearn application incorporates a data engineering pipeline with a four-layer architecture to process, manage, and analyze employee engagement data derived from various courses. The pipeline helps to efficiently handle the data, perform cleaning operations, model it for analysis, and generate insightful reports. The layers included in the architecture are: **Raw Layer**, **Clean Layer** and **Reporting Layer**.

**1. Raw Layer**

* **Database Connection:**
  + The raw data is extracted from a relational database that houses details about employees, courses, course completions.
* **Generated Data:**
  + Python scripts are employed to create synthetic data that complements the existing dataset. This includes augmenting the dataset to enhance the scope of analysis by generating new scenarios and outcomes for testing predictive models.

**2. Staging Layer**

The Clean Layer ensures that the data is of high quality and free from inconsistencies or redundancies:

* **Null Value Removal:**
  + Rows containing null or missing values are identified and removed, ensuring that only complete and usable data proceeds to the next layers.
* **Duplicate Removal:**
  + Duplicate records are detected and eliminated to maintain unique data entries, which is essential for accurate analysis and reporting.

**3. Report Layer**

* **Fact Tables:**
  + These tables store the quantitative data that is key to understanding employee performance, such as:
    - Course completion percentages.
    - Quiz scores.
    - Time taken to complete courses.
* **Dimension Tables:**
  + Dimension tables are developed to hold descriptive attributes, enabling easier categorization and filtering of data. These tables include:
    - Employee metadata (e.g., name, designation).
    - Course metadata (e.g., course title, tags).

The Reporting Layer produces detailed reports that provide valuable insights into employee engagement and course effectiveness:

* **Top 3 Employees in Every Course:**
  + This report ranks employees based on their performance in each course. It highlights the top 3 employees by metrics such as course completion speed, quiz scores, and engagement levels.
* **Top 10 Courses:**
  + A report showcasing the 10 courses with the highest engagement. This is determined by a combination of course completion rates, quiz scores, and employee feedback ratings.
* **Enrollment Statistics:**
  + This report provides insights into the number of employees enrolled in each course, helping to identify course popularity trends and participation rates over time.

### Data Science

**1. Exploratory Data Analysis (EDA)**

EDA is the foundational step to understand the employee performance dataset, evaluate hypotheses, and visualize insights.

**1.1 Data Overview**

* **Goal:** Examine the structure and initial insights of the datasets.
* **Action:**
  + Loaded the employee\_performance\_kpi.csv and unassigned\_courses.csv datasets.
  + Reviewed the column names of both datasets.

**Result:** Acquired an understanding of the features present in the datasets.

**1.2 Hypotheses Formulation**

* **Hypothesis 1:** Employees with a higher completion percentage in courses achieve higher performance scores.
* **Hypothesis 2:** Courses with longer durations lead to greater performance score improvements.
* **Hypothesis 3:** There is a positive relationship between the number of modules completed in a course and the course score achieved by the employee.

**1.3 Performance Analysis Visualizations**

* **Goal:** Visualize relationships to validate hypotheses.
* **Action:**
  + Created scatter plots using Altair to explore:
    - The relationship between average completion percentage and performance score.
    - The effect of course duration on performance score.
    - The correlation between modules completed and course score.
* **Result:** Identified trends that support the formulated hypotheses, providing visual insights into employee performance metrics.

**2. Feature Engineering**

Feature engineering is vital for transforming raw data into meaningful features for the recommendation model.

**2.1 Data Preparation**

* **Goal:** Encode categorical features and compute normalized scores.
* **Action:**
  + Encoded the course\_tag feature using LabelEncoder.
  + Calculated a raw score as the product of completion\_percentage and course\_score.
  + Normalized the raw scores using min-max normalization.
* **Result:** Created a cleaned dataset with normalized scores ready for analysis.

**2.2 Data Aggregation**

* **Goal:** Aggregate data for recommendation.
* **Action:**
  + Created a summary of normalized scores for each user and course title.
  + Constructed a pivot table to facilitate collaborative filtering.
* **Result:** Prepared a user-item matrix suitable for building a recommendation model.

**3. Model Training and Recommendation System**

This section involves model selection, training, evaluation, and generating course recommendations.

**3.1 Model Selection**

* **Goal:** Choose a suitable model for generating course recommendations.
* **Action:**
  + Utilized the Nearest Neighbors algorithm to find similar employees based on normalized scores.
* **Result:** The KNN model was successfully implemented for recommendation purposes.

**3.2 Recommendation Generation**

* **Goal:** Generate course recommendations for a randomly selected employee.
* **Action:**
  + Randomly selected an employee and identified nearest neighbors based on the cosine distance metric.
  + Gathered courses from the nearest neighbors while ensuring recommendations were unique to the selected employee.
* **Result:** Generated tailored course recommendations for the employee, enhancing their learning path.

**3.3 Model Evaluation**

* **Goal:** Evaluate the effectiveness of the recommendations.
* **Action:**
  + Calculated the Root Mean Square Error (RMSE) to assess the predictive accuracy of the recommendations.
* **Result:** Obtained an RMSE value indicating the model's prediction performance, providing insights into the recommendation quality.

# User Interface

**Signup Page**

**A screenshot of a login form

Description automatically generated**

**Login Page**

**A login screen shot

Description automatically generated**

**A screenshot of a computer

Description automatically generatedAdmin Dashboard**

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Employee Dashboard**

A white paper with green text

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated