**Course Completion and Learning Path Optimization**

Major Project

Contents

[1 Project Overview 2](#_Toc179812784)

[1.1 Brief introduction of the project. 2](#_Toc179812785)

[1.2 Purpose of the project. 3](#_Toc179812786)

[1.3 Goals of the project: 4](#_Toc179812787)

[1.4 Web app support 4](#_Toc179812788)

[1.5 Key features and functionalities of the web app. 4](#_Toc179812789)

[2 Architecture and Technology Stack 4](#_Toc179812790)

[2.1 Overall architecture of the web application. 4](#_Toc179812791)

[2.1.1 Presentation Layer (Front-End) 4](#_Toc179812792)

[2.1.2 Application Layer (Back-End API) 5](#_Toc179812793)

[2.1.3 Data Layer (Database and Storage) 5](#_Toc179812794)

[2.1.4 Data Engineering Layer (ETL and Reporting) 5](#_Toc179812795)

[2.1.5 Data Science Layer (Machine Learning Model) 5](#_Toc179812796)

[2.2 Technology stack (e.g., programming languages, frameworks, libraries). 5](#_Toc179812797)

[2.3 Rationale behind the chosen technology stack. 6](#_Toc179812798)

[3 Web App Components 7](#_Toc179812799)

[3.1 Main components of the web app. 7](#_Toc179812800)

[3.2 Purpose of each component. 8](#_Toc179812801)

[4 User Interface Design 9](#_Toc179812802)

[4.1 User interface (UI) 9](#_Toc179812803)

[5 Work with Data 15](#_Toc179812805)

[5.1 Data Modelling 15](#_Toc179812806)

[5.2 Approach for the Data Engineering. 15](#_Toc179812807)

[5.3 Approach for the Machine Learning. 15](#_Toc179812808)

[5.3.1 Data Exploration: 15](#_Toc179812809)

[5.3.2 Feature Engineering: 16](#_Toc179812810)

[5.3.3 Model Building: 16](#_Toc179812811)

[5.3.4 Why LSTM? 16](#_Toc179812812)

[5.3.5 Model Architecture: 16](#_Toc179812813)

# Project Overview

## Brief introduction of the project.

The Course Completion and Learning Path Optimization project focuses on creating a comprehensive system that monitors and enhances employee learning journeys. The fullstack development aspect includes building user-facing dashboards where employees can track their course enrollments, completion rates, and earned certificates. Additionally, the system features an admin interface for managing learning paths and assigning courses. To further assist employees, the project includes interactive learning path suggestions based on individual performance, helping employees choose courses that align with their improvement areas. The system stores key course data like title, duration, difficulty level, completion status, and performance metrics in a robust database architecture.

On the data engineering side, the project involves connecting the fullstack system to an existing database. Data will be cleaned, transformed, and integrated into a data model, with a reporting structure built using Python to support analytical needs. The data science component focuses on optimizing learning paths through machine learning, recommending courses that improve employee performance and speed up learning outcomes, ultimately driving personalized learning strategies for employees based on their past experiences and progress.

**Overall WorkflowA screen shot of a computer

Description automatically generated**

## Purpose of the project.

The purpose of the Course Completion and Learning Path Optimization project is to enhance employee learning and development within an organization by streamlining the monitoring of course progress and optimizing learning paths. The project has several key objectives:

**Employee Learning Progress Tracking:** Developing dashboards that allow employees to track their course enrolments, completion rates, and certificates, making it easier for them to monitor their learning journey. For administrators, the project aims to provide interfaces for managing learning paths, assigning relevant courses, and tracking overall employee learning performance.

**Learning Path Optimization:** Leveraging data science techniques to suggest personalized learning paths based on employee performance and past learning experiences. These optimized learning paths will identify courses that lead to the highest improvement and shortest completion times, helping employees focus on their growth while enabling the organization to maximize training effectiveness and employee productivity.

## Goals of the project:

The goals of the Course Completion and Learning Path Optimization project include the design and development of key components that enhance employee learning experiences.

**User Interface and Experience Design:** Developing responsive UI/UX components for employee dashboards and admin interfaces, ensuring that the platform is intuitive and easy to navigate for both employees and administrators.

**Core Application Development:** Designing the core application architecture, which consists of two main tiers: a user-friendly front-end accessible through the web for employees to track their learning progress and view recommendations, and a robust back-end API that facilitates data management for course progress, learning paths, and performance metricsProject 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 Creation and Authentication:**

Users can create their account through sign up page.

Users can enrol in courses of their choices and learn.

JWT Tokens are used for authentication.

**Personalized Admin Dashboard**:

Admin Specific dashboard for viewing reports about users and courses assigned to them.

Admin can allocate projects to the employees.

Apex Charts are used for dashboard visualization.

**Learning Path**:

User specific courses will be recommended based on user’s past courses and performance.

**Responsive Design:**

The Responsive UI is done with Next JS and styling with bootstrap v.5.

# Architecture and Technology Stack

## Overall architecture of the web application.

The architecture of the web application follows a three-tier structure with the Presentation Layer, Application Layer, and Data Layer. This design ensures modularity, scalability, and maintainability while providing a seamless user experience. Below is an overview of each layer:

### Presentation Layer (Front-End)

This layer provides the user interface for employees and administrators through a responsive web application.

**Technologies**: React.js / Next.js, HTML, CSS, bootstrap, Apexcharts.

**Features:**

**Employee Dashboard:** Displays course enrollments, progress, completion rates, and certificates.

**Admin Interface:** Manage learning paths, assign courses, and monitor employee performance.

**Interactive Learning Path Suggestions:** Provides personalized course recommendations

**Authentication:** Secure user login and role-based access control.

### Application Layer (Back-End API)

This layer serves as the business logic and communication bridge between the front-end and database, handling API calls and data processing.

**Technologies:** Node.js / Express, TypeScript

**Functions:**

**User Management:** Register, login, and manage employee profiles

**Course Management:** Create, update, and manage course data and learning paths

**Performance Tracking:** Track employee progress and generate analytics reports

**Machine Learning Integration:** Provide course recommendations based on employee performance

**Middleware:** Handle JWT-based authentication and request validations

### Data Layer (Database and Storage)

This layer handles the storage and retrieval of all course, user, and performance-related data.

**Technologies:** MS SQL, Prisma ORM

**Data Storage:**

**Course Metadata:** Stores course titles, difficulty levels, and durations

**User Progress Data:** Tracks course enrollments, completions, and certificates

**Performance Metrics:** Stores metrics for personalized learning path suggestions

### Data Engineering Layer (ETL and Reporting)

This component integrates with the back end to process data for reporting and analysis.

**Technologies:** Python, Pandas

**ETL Pipelines:** Extract, transform, and load data from internal.

**Reporting:** Generate learning analytics and performance reports stored in a structured folder system

### Data Science Layer (Machine Learning Model)

This layer focuses on Learning Path Optimization through machine learning.

**Technologies:** Python, TensorFlow / Scikit-learn, fastAPI

**Functionality:**

* Predict optimal courses based on historical data.
* Provide real-time course recommendations through the back-end API developed by python’s fastapi.

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

**Technology Stack for the Web Application Platform:**

The technology stack for the web Application Platform for employee learning platform will include the following components:

**Next.js (Frontend & Backend):**

* A React-based framework that supports server-side rendering (SSR) and static site generation (SSG) for building performant web applications.
* On the backend, it provides API routes and serverless functions, allowing seamless integration with databases and third-party services.

**ApexCharts:**

A modern JavaScript charting library for building interactive and responsive data visualizations, including bar charts, line graphs, and more.

**MS SQL:**

A relational database management system (RDBMS) by Microsoft, used to store and manage structured data with SQL querying capabilities.

**Prisma ORM:**

A type-safe Object-Relational Mapping (ORM) tool for Node.js that simplifies database interactions by generating SQL queries through JavaScript/TypeScript models.

**Python Pandas:**

A powerful data manipulation and analysis library that provides data structures like DataFrames for handling large datasets efficiently.

**FastAPI:**

A high-performance Python web framework for building APIs, known for automatic OpenAPI documentation and asynchronous support.

**TensorFlow:**

An open-source machine learning framework by Google used for building, training, and deploying deep learning models.

**Scikit-learn:**

A popular Python library for machine learning that provides tools for tasks like classification, regression, and clustering.

**JWT Authentication:**

A stateless authentication mechanism where JSON Web Tokens (JWT) are used to verify user identity, enabling secure access to APIs and resources.

## Rationale behind the chosen technology stack.

The selected technology stack is carefully chosen to meet the needs of the Course Completion and Learning Path Optimization project, ensuring scalability, performance, ease of development, and seamless integration across components.

**Next.js (Frontend & Backend):**

Next.js offers server-side rendering (SSR) and static site generation (SSG) to enhance page load speed and SEO, making it ideal for dynamic dashboards. Its ability to handle both frontend and backend logic allows easy integration of backend APIs, simplifying development with a unified framework.

**ApexCharts:**

ApexCharts provides highly customizable and interactive charts for visualizing course progress and performance metrics, delivering insights through user-friendly dashboards.

**MS SQL:**

MS SQL ensures robust storage and management of structured data like course metadata, user profiles, and completion status. It supports complex queries and high transactional throughput, essential for managing real-time course progress and analytics.

**Prisma ORM:**

Prisma simplifies interactions with the MS SQL database through type-safe queries in TypeScript, ensuring data integrity and reducing the chance of runtime errors while speeding up development with automatic schema management.

**Python Pandas:**

Pandas enables efficient data manipulation and preprocessing, making it ideal for building ETL pipelines and preparing datasets for reporting and machine learning models.

**FastAPI:**

FastAPI is chosen for its high-performance, asynchronous capabilities and automatic API documentation, which ensures fast and scalable backend services for course management and machine learning integration.

**TensorFlow:**

TensorFlow supports deep learning models for personalized learning path recommendations, enabling the project to predict optimal courses based on employee performance data.

**Scikit-learn:**

Scikit-learn is used for classical machine learning tasks like clustering and regression, which helps in deriving insights and patterns from employee learning data.

**JWT Authentication:**

JWT ensures secure, stateless authentication, enabling role-based access to different parts of the application and safeguarding user data across the platform.

This stack offers a well-rounded balance of performance, scalability, and ease of development, making it suitable for building a data-driven, interactive platform with real-time insights and machine learning capabilities.

# Web App Components

## Main components of the web app.

* api
  + auth
    - checkCookie
    - login
    - logout
    - signup
  + courses
  + metrics
  + users
* course
* dashboard
* home
* path
* tree
* user
* Reusable Components: Charts, AssignCourseForm, Button, CertificateGen. CourseCards, CourseData, CourseDataParser, CustomNode, Header, Modal, PlaceHolder, SideBar, SignInForm, Spinner, Time, UserCourseTable, UserTable.

## Purpose of each component.

* **Api:** The API layer serves as the backend of the application, exposing endpoints to manage data, authentication, and user interactions.
* **auth:** Handles user authentication and session management.
* **checkCookie:** Verifies the existence and validity of cookies for authenticated sessions.
* **login:** Authenticates users with credentials and issues a JWT token or cookie.
* **logout**: Ends user sessions by invalidating cookies or tokens.
* **signup:** Registers new users in the system and creates their profiles.
* **courses:** Manages course data such as adding, updating, and fetching available courses.
* **metrics:** Provides data on learning progress and performance metrics for dashboards and reports.
* **users:** Manages user data, including fetching employee details and roles for the platform.
* **Course:** This component manages course-related details and views, including enrollment, content display, and course status tracking for users.
* **Dashboard:** Displays an overview of learning progress, including course enrollments, completion rates, certificates earned, and personalized learning suggestions for employees.
* **Home:** Serves as the landing page of the application, offering an overview of the platform and quick links to other sections such as login, course enrollment, or dashboards.
* **Path:** Manages learning paths, showing structured sequences of courses assigned to employees. It may also offer suggestions for new paths based on performance.
* **Tree:** Visualizes course relationships and learning paths in a hierarchical format, helping employees and administrators understand course dependencies and progression.
* **User:** Handles user-related pages such as user profiles, course assignments, and progress tracking. It includes interfaces for updating user details and viewing course histories.
* **Reusable Components:** These components enhance modularity and reuse across the application, ensuring a consistent design and reducing development effort.
  + Charts: Renders data visualizations such as progress graphs and performance charts.
  + AssignCourseForm: A form for administrators to assign courses to employees.
  + Button: A reusable button component for various actions across the application.
  + CertificateGen: Generates and displays certificates for completed courses.
  + CourseCards: Displays individual course summaries with title, description, and enrollment status.
  + CourseData & CourseDataParser: Structures and parses course-related data for display and analytics.
  + CustomNode: Used in the Tree component to represent a customized node for learning paths or course relationships.
  + Header: The top navigation bar with links to key sections like dashboard, home, and user profile.
  + Modal: A popup component used for alerts, confirmations, or displaying detailed content.
  + PlaceHolder: Used to display a temporary placeholder during loading states.
  + SideBar: A navigation menu providing access to different sections such as dashboard, courses, and user settings.
  + SignInForm: A form component used for user login.
  + Spinner: A loading indicator shown while waiting for data or actions to complete.
  + Time: Handles and formats time-related data, such as tracking course completion times.
  + UserCourseTable: Displays a table of courses assigned to a user along with their progress and status.
  + UserTable: A table showing all users, with their roles and assigned courses.
  + These components form the building blocks of the application, working together to deliver a seamless user experience while ensuring efficient management of course and user data.

# User Interface Design

## User interface (UI)

Login PageA screenshot of a computer

Description automatically generated

User Home Page

A screenshot of a computer

Description automatically generated

Course Browse Page

A screenshot of a computer

Description automatically generated

Course Page

A screenshot of a computer

Description automatically generated

Course tree Page

# A computer screen shot of a computer screen Description automatically generated

Learning Path Page

A screenshot of a computer

Description automatically generated

Admin DashboardA screenshot of a computer

Description automatically generated

User Administration Page

A screenshot of a computer

Description automatically generated

User PageA screenshot of a computer

Description automatically generated

# Work with Data

Testing and quality assurance are essential processes in software development that aim to ensure the reliability, functionality and ultimately delivering a stable and dependable software product to the end users.

## Data Modelling

**A diagram of a course

Description automatically generated**

## Approach for the Data Engineering.

I have implemented a Python script to facilitate data migration from MS SQL to local directory as CSV files. Utilizing Pyodbc, we established a connection to our MS SQL database, fetched the relevant data, and performed necessary transformations to align with the medallion architecture.

Python script integration streamlines and standardizes data mart transformation, automating the process and ensuring reliable analytics.

## Approach for the Machine Learning.

### Data Exploration:

Data exploration is an essential initial phase in any data analysis or machine learning project. It encompasses examining and summarizing the primary characteristics of a dataset to comprehend its underlying structure, trends, and associations.

### Feature Engineering:

Feature engineering refers to the process of converting raw data into a set of relevant features that can serve as input for machine learning algorithms.

### Model Building:

Model building involves analysing pre-processed data from data exploration and feature engineering to predict courses for users based on their history. Using machine learning techniques, the objective is to create accurate predictive models capable of identifying courses for the future.

The project leverages Long Short-Term Memory (LSTM) networks to predict the optimal learning path for users based on their past course completions and performance metrics. Below are the key components and rationale behind using LSTM:

### Why LSTM?

LSTM is a type of recurrent neural network (RNN) that excels at processing sequential data by learning long-term dependencies.

It is well-suited for predicting sequences or time-dependent patterns, making it ideal for learning path optimization by analysing the sequence of completed courses and performance scores.

### Model Architecture:

Input Layer:

The input to the LSTM is a sequence of course IDs, represented as numerical vectors.

Each sequence reflects the order of courses completed by a user.

Embedding Layer:

Converts input course titles into dense vector representations, capturing semantic relationships between different courses.

LSTM Layer:

Processes the input sequences and learns patterns in course-taking behaviour and performance improvements over time.

Dense Layer:

Maps the output of the LSTM layer to predict the next recommended course in the sequence.

The layer outputs course titles or IDs, representing the optimal next course based on the user’s learning history.

Activation Function:

A softmax activation function is used at the output layer to return the probability distribution of the next course.

**Appendix Title**

Document Title