**Employee Engagement and Learning Behaviour**

**Name: RITISH**

**ID: JMD344**

**Table of contents**

1. **Problem statement**
2. **Overview**
3. **Solutions**
   1. **Full Stack**
   2. **Data Engineering**
   3. **Data Science**
4. **Conclusion**

# Problem statement

In today's fast-paced work environment, organizations strive to enhance employee engagement and learning outcomes. However, many existing learning management systems fail to provide comprehensive insights into how employees interact with learning materials. This gap results in challenges such as:

* **Low Engagement:** Employees may not actively participate in courses or discussions, leading to underutilization of available learning resources.
* **Lack of Tracking:** Organizations struggle to monitor employee progress, time spent on courses, and participation rates, hindering their ability to tailor learning experiences.
* **Feedback Mechanisms:** There is often no structured way for employees to provide feedback on courses, which limits opportunities for course improvement and adaptation to employee needs.

To address these challenges, there is a need for a system that tracks employee engagement with learning materials, enables active participation, and provides valuable insights for both employees and administrators.

# Overview

The **Employee Engagement and Learning Behavior** project aims to create a comprehensive system that monitors and enhances employee engagement with learning materials. The system provides a user-friendly platform for employees to enroll in courses, track their progress, and engage in discussions. It also offers visual analytics for employees to see their participation rates, time spent on learning materials, and quiz scores.

Key features of the system include:

* **Engagement Dashboard:** Employees can view their enrolled courses, participation metrics, and time spent on each course. A progress bar visually represents their course completion status.
* **Discussion Platform:** Employees can interact with peers by asking questions and providing answers within the course context, fostering a collaborative learning environment.
* **Feedback System:** Employees can submit feedback on courses, helping administrators identify areas for improvement and adapt learning materials to better meet employee needs.
* **Admin Interface:** Administrators can analyze engagement levels across teams, track time spent on courses, and review employee feedback to improve overall learning outcomes.

By implementing this system, organizations can promote a culture of continuous learning and development, ultimately leading to higher employee satisfaction, retention, and productivity.

# Solutions

# (Full Stack)

**Overview**

This project aims to create a system that tracks employee engagement with learning materials, focusing on metrics like time spent on the platform, participation in discussions, and quiz scores.

**1. Features**

**Employee Features**

* **Course Listing:** Employees can view all available courses.
* **Enrollment:** Employees can enroll in courses of interest.
* **Course Progress Tracking:**
  + A progress bar displays the completion percentage of each course.
  + Time spent on each chapter is recorded.
* **Discussion Section:**
  + Employees can raise questions and answer others' questions related to the course.
* **Engagement Dashboard:**
  + Visual representation of:
    - Participation rate in every course.
    - Time spent on courses.
    - Scores in quizzes displayed in a bar chart.
* **Feedback Submission:**
  + Employees can provide feedback on the courses they enrolled in.

**Admin Features**

* **Dashboard Overview:**
  + Admins can view metrics on total courses and total employees.
* **Engagement Analysis:**
  + Admins can see time spent and participation rates for every team.
  + Individual employee metrics are available for detailed analysis.
* **Feedback Review:**
  + Admins can view feedback from employees on various courses.
* **Course Management:**
  + Ability to create, update, and delete courses as needed.

**2.Database Design**

**Schema Overview**

The database is designed using Prisma and PostgreSQL, with the following models:

* **Employee:** Stores employee information and relationships with enrollments, engagements, discussions, and feedback.
* **Course:** Contains course details and relations to modules, enrollments, discussions, and feedback.
* **Module:** Represents chapters within each course.
* **Enrollment:** Records employee enrollments in courses and tracks quiz scores.
* **Engagement:** Logs time spent by employees on each module.
* **Discussion:** Captures employee discussions related to courses.
* **Feedback:** Stores ratings and comments provided by employees on courses.

**Entity Relationships**

* **Employee ↔ Enrollment:** One-to-many relationship where one employee can have multiple enrollments.
* **Course ↔ Module:** One-to-many relationship where one course can have multiple modules.
* **Course ↔ Feedback:** One-to-many relationship where one course can receive multiple feedback entries.
* **Module ↔ Engagement:** One-to-many relationship where one module can have multiple engagements logged.

**3.Technology Stack**

* **Frontend:** [React] for building the user interface.
* **Backend:** [Node.js] for server-side logic and API development.
* **Database:** [PostgreSQL] for storing data.
* **ORM:** [Prisma] for database interactions.

**4. User Authentication**

* **Login Types:**
  + Admin and Employee login systems to ensure appropriate access to features and data.

**5. Visualizations**

* **Employee Dashboard:** Displays charts for:
  + Overall participation rates.
  + Time spent on courses.
  + Scores from quizzes.
* **Admin Dashboard:** Visualizes team-level engagement metrics.

# (Data Engineering)

## Overview

This project aims to implement a data engineering pipeline using a four-layer architecture to manage and analyze employee engagement data from various courses. The pipeline consists of the Raw Layer, Clean Layer, Load Layer, and Reporting Layer.

**1. Raw Layer (Data Sources)**

* **Database Connection:**
  + Data is extracted from a relational database containing employee and course information.
* **Generated Data:**
  + Additional data is generated using Python scripts to enhance the dataset and provide more comprehensive analysis.

**2. Clean Layer (Data Cleaning Operations)**

* **Null Value Removal:**
  + Identify and remove any rows with null values to ensure data integrity.
* **Duplicate Removal:**
  + Eliminate duplicate records to maintain unique entries within the dataset.

**3. Load Layer (Data Modeling)**

* **Fact Tables:**
  + Create fact tables to capture quantitative data related to employee engagement, such as course completion rates and quiz scores.
* **Dimension Tables:**
  + Develop dimension tables to store descriptive attributes, including employee details and course metadata. This structure allows for efficient querying and reporting.

**4. Reporting Layer (Reports Generated)**

* **Top 3 Employees in Every Course:**
  + Generate a report that lists the top 3 employees based on performance metrics for each course.
* **Top 10 Courses:**
  + Create a report showcasing the top 10 courses with the highest engagement levels or ratings.
* **Enrollment Statistics:**
  + Analyze and report the number of employees enrolled in each course, providing insights into course popularity and participation.

# (Data Science)

## 1. Exploratory Data Analysis (EDA)

EDA is the initial step to understand the data structure, spot anomalies, and derive insights. Below is a summary of the tasks performed during EDA:

**1.1 Data Types and Missing Values**

* **Goal**: Check data types and identify missing values.
* **Action**:
  + Printed the data types of each column to ensure proper data types (e.g., categorical or numerical).
  + Checked for missing values in the dataset to assess the need for imputation or handling.
* **Result**: No missing values were found, and data types were as expected for analysis.

**1.2 Descriptive Statistics**

* **Goal**: Summarize the central tendency, dispersion, and shape of the dataset’s distribution.
* **Action**:
  + Generated descriptive statistics for all numeric columns.
  + Used .describe() to capture summary statistics for each feature (mean, std, min, max, etc.).
* **Result**: Provided a broad understanding of the distribution of numeric features.

**1.3 Visualizations**

* **Goal**: Visualize data distributions and relationships between features.
* **Action**:
  + **Count Plot**: Visualized the distribution of the target variable (Engagement Level).
  + **Histograms**: Visualized the distribution of numeric features.
  + **Boxplots**: Visualized the distribution of numeric features by engagement level.
  + **Correlation Heatmap**: Visualized correlations between features to check for multicollinearity.
  + **Pair Plot**: Visualized relationships between pairs of features with the target variable.
  + **Violin Plots**: Showed the distribution of numeric features across different engagement levels.
* **Result**: Gained insights into feature distributions and relationships, identifying possible feature interactions for further analysis.

## 2. Feature Engineering

Feature engineering involves transforming raw data into features that better represent the underlying structure of the problem to improve model performance.

**2.1 Handling Categorical Data**

* **Goal**: Convert categorical variables (e.g., Department) into a format suitable for machine learning models.
* **Action**:
  + Used One-Hot Encoding to transform categorical features (Department) into numerical format.
* **Result**: The categorical Department feature was successfully encoded.

**2.2 Scaling Numerical Features**

* **Goal**: Normalize or standardize numeric features to ensure they are on a comparable scale.
* **Action**:
  + Applied StandardScaler to normalize all numerical features such as DiscussionCount, AverageRating, etc.
* **Result**: All numeric features were successfully scaled.

**2.3 Feature Interaction**

* **Goal**: Consider interactions between features to capture complex relationships.
* **Action**:
  + Evaluated interactions through visual inspection and plotted potential interactions between important features (e.g., DiscussionCount vs. EngagementLevelScore).
* **Result**: Identified some useful interactions that could potentially improve model predictions.

**3. Model Training**

This section involves the process of model selection, training, evaluation, and interpretation.

* 1. **Model Selection**
* **Goal**: Choose a suitable model for predicting engagement levels.
* **Action**:
  + Used a **Random Forest Classifier** due to its ability to handle both categorical and numerical data and provide feature importance.
* **Result**: The Random Forest model was selected and trained successfully.

**3.2 Handling Class Imbalance**

* **Goal**: Address class imbalance in the target variable (EngagementLevel).
* **Action**:
  + Applied **SMOTE** (Synthetic Minority Over-sampling Technique) to balance the dataset by generating synthetic examples for the minority class.
* **Result**: Balanced the dataset to prevent bias in model predictions.

**3.3 Model Evaluation**

* **Goal**: Evaluate the performance of the trained model.
* **Action**:
  + Assessed the model using metrics such as accuracy, classification report, and confusion matrix.
* **Result**: The model performed well with a reasonable balance between precision, recall, and F1-score.

**3.4 Model Interpretation**

* **Goal**: Understand the influence of features on model predictions.
* **Action**:
  + Used **LIME** (Local Interpretable Model-agnostic Explanations) to interpret model predictions and provide insights into feature contributions.
* **Result**: LIME provided detailed insights into how features impacted the model’s predictions for specific instances.

**Appendix Title**

Document Title