EMPACT

Team Members: Muhammed Basith VP, Prashanth Rapolu

Problem Statement:

In many organizations, the Learning & Development (L&D) department faces the challenge of managing and tracking employee training programs, certifications, and development progress. With multiple ongoing programs, it becomes difficult to measure the effectiveness of training, track employee engagement, or align learning initiatives with the company's strategic goals. A comprehensive system is needed to manage training schedules, track progress, automate certification processes, and provide insightful reports on employee development and skill enhancement.

1.Understanding of the Problem statement

a. **Explanation of the Problem Context:**

The Learning & Development (L&D) department in many organizations manages diverse and complex training requirements, including tracking employee training programs, and development progress. The dynamic nature of training initiatives makes it difficult to measure effectiveness, track engagement, and align learning goals with organizational objectives.

Managers, employees, and training administrators face unique challenges:

- **Managers** need a streamlined, user-friendly way to submit training requests, specify training types, duration, and target audiences, and monitor employee progress.
- **Employees** require tailored training paths that align with their skill levels and ensure meaningful progress.
- **Training Administrators** need real-time insights into training batch performance and employee progress to adjust programs as needed.

A comprehensive solution would address these challenges by managing training schedules, tracking employee development, automating certification processes, and delivering detailed reports on skill enhancement, ultimately aligning learning initiatives with strategic goals.

b. **Key Requirements Identified**

Key requirements to address the challenges include:

1. **Streamlining Training Requests**: Provide a user-friendly interface for managers to submit detailed training requests.
2. **Automating Baseline Assessments**: Use generative AI to create baseline assessments, allowing employees to showcase their skill levels.
3. **Dynamic Batching**: Assign employees to training batches based on performance scores, customizing training duration and content accordingly.

4. **Cost Optimization**: Minimize training costs by avoiding redundant training sessions for advanced employees.
5. **Real-Time Tracking and Evaluation**: Develop an intuitive dashboard for real-time monitoring of batch progress and assessment results.
6. **Robust Backend Services**: Employ microservices architecture to handle specific functionalities like requirement management, assessment generation, evaluation, notification, and batching.

2.Solution Overview

a. Solution Summary

The proposed solution is an integrated Learning Management System (LMS) designed to streamline and automate employee training management. It consists of a React and TypeScript-based frontend for user interactions, and a backend powered by Node.js microservices, each responsible for a specific functionality. The system leverages generative AI (OpenAI) for baseline assessments, supports real-time progress tracking, and features dynamic batching to tailor training experiences based on employee skill levels.

b. Objectives

The core objectives of the LMS solution are:

1. **Streamline Training Requirements**: Simplify the process for managers to submit training requests through an intuitive UI.
2. **Automate Baseline Assessments**: Implement generative AI to automatically create skill-based assessments.
3. **Dynamic Batching for Personalized Training**: Tailor training duration and content by categorizing employees based on skill levels.
4. **Cost Optimization**: Reduce training costs by curating personalized, skill-level-appropriate training paths.
5. **Real-Time Monitoring and Insightful Reporting**: Provide administrators with a dashboard to track batch formation, employee progress, and training effectiveness.
6. **Robust, Scalable, and Secure Platform**: Ensure high performance, scalability, and secure access to safeguard user data and maintain operational reliability.

3.Features and Functionalities:

a. Core Features:

1. Training Requirement Submission

- **Training Requirement Form**: A user-friendly form where managers can submit training needs.
  - **Fields**: Includes mandatory fields such as training type, duration, audience, and intended outcomes.
  - **Form Validation**: Ensures all required fields are filled correctly, enhancing data quality.
- **Real-Time Dashboard**: Displays an overview of all training requests, showing statuses, deadlines, and batch assignments.

2. Automated Baseline Assessment Generation

- **Generative AI for Question Creation**: Utilizes Gemini API to automatically generate baseline assessment questions, tailored to the training requirements.
- **Question Types**: Supports multiple-choice, coding tasks, and open-ended questions.
- **Customizable Difficulty Levels**: Adjusts question complexity based on the training goals and employee skill levels.

3. Dynamic Batching of Employees

- **Performance-Based Batch Assignment**: Automatically assigns employees to different batches based on baseline assessment scores, creating customized training groups.
- **Optimized Training Duration**: Determines optimal training duration for each batch, allowing advanced employees to progress faster.
- **Personalized Learning Paths**: Adjusts training content and pace to better match the skill levels of each batch.

4. Progress Tracking and Reporting

- **Real-Time Progress Dashboard**: Displays ongoing batch formation, employee progress, and assessment scores for managers and administrators.
- **Skill Gap Analysis**: Provides insights into individual and batch performance, helping L&D teams identify training needs and effectiveness.
- **Completion and Engagement Metrics**: Shows employee engagement levels, completion rates, and time spent on training, assisting with engagement and program optimization.

b. User Flows

1. Training Request Submission (Manager)

- **Description**: Allows a manager to submit training requirements for their team members.
- **Steps**:

  1. **Login**: Manager logs in through the **Authentication Microservice**.
  2. **Access Training Form**: Manager navigates to the **Training Requirement Management** section.
  3. **Submit Details**: Fills out the training request form, including training type, duration, audience, and desired outcomes.
  4. **Submit Form**: Validates and submits the form, triggering a notification to the L&D team.
  5. **Confirmation**: Receives a confirmation email and dashboard update on the request status.

2. Batch Assignment and Training Initiation (Employee)

- **Description**: Based on the assessment score, the employee is assigned to a batch and begins training.

- **Steps**:

    1. **Batch Notification**: Employee is notified of their batch assignment via the **Notification Microservice**.
    2. **Access Training Materials**: Navigates to the **Dashboard** to view assigned batch and training schedule.
    3. **Start Training**: Begins training sessions assigned to their batch, personalized based on skill level.
    4. **Progress Tracking**: Tracks progress through the **Training Progress Tracking** module, where completion metrics and engagement are logged.

3. Training Progress Monitoring (Training Administrator)

- **Description**: Training administrators monitor the progress and engagement levels of batches.
- **Steps**:

    1. **Login**: Training Administrator logs in via the **Authentication Microservice**.
    2. **Access Dashboard**: Opens the **Dashboard** to view training requests, batch formations, and real-time employee progress.
    3. **View Batch Performance**: Checks individual and batch metrics such as completion rates, scores, and feedback.
    4. **Generate Reports**: Exports reports on employee progress and training effectiveness for review and analysis.

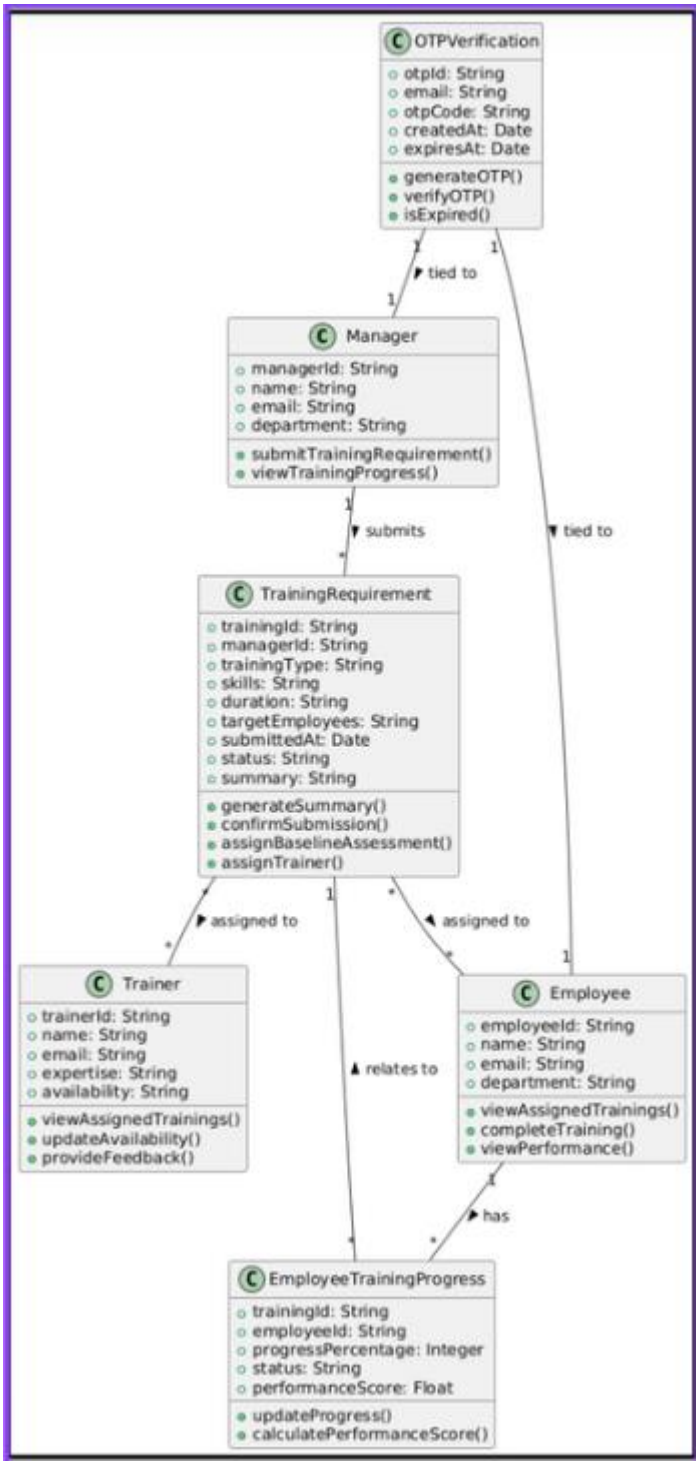
4. Trainer Assignment (Training Administrator)

- **Description**: Allows training administrators to assign trainers to specific batches.
- **Steps**:

    1. **Login**: Training Administrator logs in via the **Authentication Microservice**.
    2. **Access Trainer Management**: Opens the **Trainer Management** section on the dashboard.
    3. **Assign Trainer**: Selects trainers based on availability, expertise, and batch requirements.
    4. **Confirmation**: Trainer assignment is finalized.

4. Architecture Diagram

a. System Architecture:



b. Key Components and Their Roles

1. **Frontend Interface (React & TypeScript)**:
    - **Role**: The user-facing layer, displaying forms, dashboards, and progress tracking. It also handles input validation and sends requests to the backend via the API Gateway.
2. **Authentication Microservice**:
    - **Role**: Authenticates users and enforces role-based access control (RBAC) to ensure only authorized users can access specific parts of the system.
3. **Training Requirement Management Microservice**:
    - **Role**: Manages training requests submitted by managers, validates input data, and stores requests in the database.
4. **Baseline Assessment Generation Microservice**:
    - **Role**: Generates baseline assessments using the Gemini API for new training programs and customizes assessments based on training type and employee skills.
5. **Batch Management Microservice**:
    - **Role**: Dynamically groups employees into training batches based on their baseline scores, ensuring that training is tailored to skill levels.
6. **Notification Microservice**:
    - **Role**: Sends automated email and in-app notifications to employees and managers for training updates, batch assignments, and certification completions.
7. **Training Progress Tracking Microservice**:
    - **Role**: Monitors training progress for employees, tracking completion rates, engagement, and other key metrics. Information is relayed to the dashboard for administrators.
8. **Databases (NoSQL)**:
    - **Role**: Stores all system data, including user information, training requests, assessments, batch assignments, notifications, and logs.
9. **Gemini API**:
    - **Role**: Provides automated assessment generation capabilities by generating questions and open-ended tasks tailored to employee skills and training requirements.
10. **Admin Dashboard**:
    - **Role**: Provides an overview of training progress, assessments, batches, and notifications. Allows training administrators to monitor and manage ongoing training programs.

5. Technical Stack

a. Frontend

- **Technologies and Frameworks**:
    - **React**: A powerful JavaScript library for building user interfaces. React is chosen for its component-based structure, which makes the development of modular and reusable UI components efficient.
    - **TypeScript**: Adds type safety to JavaScript, helping catch errors at compile time and making the codebase more maintainable, readable, and scalable.

- **CSS/Tailwind**: Used for styling the interface. Tailwind enables nesting and modular styles, which are essential for large applications to manage design consistency and customizations.
- **Axios**: A promise-based HTTP client used to handle API requests from the frontend to the backend through the API Gateway, managing requests and handling errors efficiently.

b. Backend

- **Technologies and Frameworks**:
  - **Node.js**: Provides an efficient, non-blocking, and scalable server-side environment, ideal for handling numerous simultaneous requests.
  - **Express.js**: A minimal and flexible Node.js web application framework that provides robust routing and middleware options, used to create the core microservices structure.
  - **Microservices Architecture**: Separates backend functionality into independent services (e.g., Training Requirement, Baseline Assessment, Batch Management) that communicate through APIs. This approach allows for scalable, isolated deployments and updates.
  - **Gemini Api:** It uses a generative AI model (Google's Gemini) to automatically generate summaries based on provided training requirement data.
  - **Cognito Authentication:** Using Cognito for user management and JWT for secure communication. It uses MongoDB to store user data and supports roles likes manager, employee, trainer & admin.

c. **Databases**

NoSQL Database:

- **MongoDB**: Used for storing unstructured data such as assessment responses, logs, and potentially large datasets from batch formations and training records. MongoDB's flexible schema design and scalability make it suitable for handling high-volume write operations.

6 Prerequisites and Requirements:

a. **Technical Requirements:**

1. Hardware Requirements:

- **OS**: Windows/Linux/MacOS, with necessary tools and IDEs installed (e.g., Visual Studio Code, IntelliJ IDEA for backend).
- **CPU**: At least 4 cores (Intel i5 or equivalent for development and testing).
- **RAM**: Minimum 8GB of RAM (16GB recommended for optimal performance).
- **Storage**: SSD with at least 50GB free space for managing source code, databases, and development tools.

2. **Software Requirements:**

Frontend **Development**:

- **React.js**: For building interactive UIs.
- **TypeScript**: For static typing and improved development experience.
- **CSS/Tailwind**: For styling the frontend.
- **Axios**: For making API requests from the frontend to the backend.

Backend **Development**:

- **Node.js**: The runtime environment for backend development.
- **Express.js**: A web framework for building the REST API and microservices.
- **Gemini API**: Integration for generating baseline assessment questions (requires API keys and proper access).
- **JWT (JSON Web Tokens)**: For user authentication and authorization.
- **MongoDB**: NoSQL database for managing user data, training information, and assessments.
- **AWS Cognito**: For managing user authentication and roles (e.g., manager, employee, admin).

b. Data Requirements

i. **Data Required**:
   a. **Training Requirement Data**: Data gathered from managers via the training request form (e.g., training type, duration, target audience, outcomes) will be used by the generative AI model to produce assessment questions.
   b. **Assessments**: Data on the structure of the baseline assessments will be required for AI to understand how to generate effective questions that assess the employees' existing skills. This could include predefined question templates and skills required for each training.

ii. **AWS Cognito**:
   a. **User Profiles**: Each user (manager, employee, trainer, admin) will need to be stored in AWS Cognito for identity management.
   b. **Roles and Permissions**: Defining roles (manager, employee, trainer, admin) with associated access rights to various parts of the platform.

iii. **JWT (JSON Web Tokens)**:
   a. **Access Tokens**: After users authenticate, JWT tokens will be issued, carrying user data and roles, to ensure secure access to the APIs.

c. **Sample Data for Development and Testing:**

- **Training Requests**: Sample data representing various types of training, expected outcomes, and audience categories.
- **Employee Data**: Sample profiles for employees, including their roles, training needs, and performance data for testing batching functionality.
- **Assessment Questions**: Predefined sets of baseline assessment questions for various training programs.
- **Batch Data**: Sample data showing how employees are grouped into different batches based on their skills and assessment scores.

d. Access Permissions and Requirements

For the effective development, testing, and deployment of the Learning Management System, the following access permissions and tools/platforms are necessary for the involved team members (developers, testers, and administrators).

1. **Version Control (Git Repositories):**

- **GitHub//GitLab Access**: Developers and team members involved in the project will need appropriate access to the Git repositories hosting the project's source code. This includes:
    - **Repository Read/Write Access**: Developers will require the ability to clone, pull, commit, and push changes to the repository.
    - **Branch Permissions**:
        - **Main/Master Branch**: Only authorized personnel (e.g., senior developers or team leads) should be allowed to merge code into the main branch after proper review (pull requests).
        - **Feature Branches**: Developers should be able to create and push feature branches for implementing individual functionalities.
- **Access Levels**:
    - **Admin**: For project managers or team leads to manage repository settings, manage access to team members, and merge pull requests.
    - **Collaborator**: For developers to push code changes and collaborate on feature development.
    - **Read-Only**: For stakeholders who need to review the project without making changes.

2. **Cloud Platform Access.**

AWS Cognito:

- **User Access Management**: Permissions to configure, modify, and manage user authentication, roles, and access permissions.
- **IAM for Cognito Access**: Developers will need to configure Cognito for user authentication, including user pools and identity pools.

3. Testing Tools and Platform.

- **Postman**: Developers will need access to Postman collections for testing APIs and debugging issues during development.
    - **API Testing Permissions**: Developers and QA testers will need permissions to access the testing suite, update test cases, and track results.

4. Other Platforms:

- **Gemini API (for Baseline Assessment Generation & AI Summary Generation)**:
    - **API Key Management**: Developers will need access to Gemini API credentials for assessment generation. Proper authentication (API keys) must be managed securely.

7.Future Improvements:

- Personalized User Experience with Generative AI
- Automated Insights and Reports Using GenAI
- Enhanced Manager Verification Using Multi-Factor and Behavioral Authentication

8.Conclusion

The comprehensive Learning and Development (L&D) platform addresses the core challenges faced by organizations in managing employee training programs by providing a streamlined, automated, and data-driven solution. Here's how the solution provides value to users and effectively tackles the core problems:

1. **Streaming Training Requirements**

- The platform offers an intuitive interface for managers to submit and specify training requirements, including type, duration, audience, and expected outcomes. This simplifies the request process and ensures alignment with the organization's strategic goals. By automating this process, the solution eliminates manual paperwork and reduces the risk of errors, enhancing the overall efficiency of the training request process.

2. Dynamic Employee Batching

- The dynamic batching system divides employees into training batches based on their performance in baseline assessments. This enables the training to be tailored to the specific needs of different groups. For instance, advanced employees will be placed in shorter, more advanced training programs, while those with more gaps in knowledge will be assigned more foundational training. This ensures that each employee receives the appropriate level of training, optimizing both time and resources.

3. Cost Optimization

- By optimizing training durations based on employee performance, the platform helps minimize unnecessary training costs. Employees who are more advanced in certain skills can skip foundational courses, which helps reduce training time and cost. This not only saves money for the organization but also prevents employees from becoming disengaged by irrelevant content, resulting in higher overall engagement and productivity.

In Summary

This Empact platform provides tremendous value by streamlining and automating key training processes, ensuring efficient resource allocation, enhancing training effectiveness, and improving employee engagement. It eliminates manual tasks, reduces administrative overhead, optimizes training costs, and provides data-driven insights to monitor progress. By offering personalized and scalable training experiences, the solution addresses the fundamental challenges of training management while aligning with organizational goals for employee development and performance improvement.

FIGMA URL:

https://www.figma.com/design/DOGrOCyKTYxWa5IzgmBxHZ/Untitled?m=auto&t=6ToH7WT20cMkh2lV-6