Training Management System (TMS) - Microservices Architecture

1. Introduction

A well-structured **Training Management System (TMS)** is essential to streamline training allocation, trainer availability, and vendor partnerships. The system will provide insights into available trainers, ongoing trainings, and vendor collaborations while leveraging a microservices architecture to ensure scalability and modularity.

---

2. System Overview

The **Training Management System (TMS)** consists of three independent microservices:

1. **Trainer Service** - Handles trainer details, skill sets, and availability.
2. **Training Service** - Manages training sessions, including required skills, duration, and participant information.
3. **Vendor Service** - Oversees vendor information and associated training programs.

Each microservice operates with its own database and communicates with other services via REST APIs.

---

3. Entity and Database Design

3.1 Trainer Entity

- *Trainer ID (Primary Key)*
- *Trainer Name*
- *Skills (List of Strings)*
- *Budget*
- *Available From (Date)*
- *Available To (Date)*
- *Phone Number*
- *Email (Unique, Validated)*

3.2 Training Entity

- *Training ID (Primary Key)*
- *Training Name*

- *Skills Required (List of Strings)*
- *Duration*
- *Start Date*
- *End Date*
- *Student Type*
- *Number of Batches*
- *Number of Students per Batch*
- *Budget*
- *Point of Contact (POC)*
- *Organization Name*
- *Table of Contents*
- *Status*
- *Vendor id*

3.3 Vendor Entity

- *Vendor ID (Primary Key)*
- *Vendor Code*
- *Organization Name*
- *Country*
- *Headquarters Location*
- *Founded Year*

---

4. Microservices Architecture

4.1 Trainer Service

- Manages trainer records and their skill sets.
- Tracks trainer availability.
- Provides trainer information to Training and Vendor Services.

4.2 Training Service

- Manages training program details.
- Maintains skill requirements, duration, and budget allocations.
- Links training programs with assigned trainers and vendors.
- Provides training details, including associated trainer information.

4.3 Vendor Service

- Manages vendor profiles and organizational details.
- Maintains a list of training programs conducted by the vendor.
- Provides vendor details, along with associated training and trainer information.

---

5. Services and API Endpoints

5.1 Get Training Details by Training ID

Endpoint: `GET /training/{id}`

- **Description:** Retrieves training details along with associated trainer information.
- **Conditions:** Ensures that only available trainers are considered.

5.2 Get Vendor Details by Vendor ID

Endpoint: `GET /vendor/{id}`

- **Description:** Fetches vendor details, including associated trainings and trainer information.
- **Conditions:** Filters results based on:
  - Vendor budget constraints.
  - Trainer salary requirements.
  - Trainer availability (excluding those assigned to ongoing trainings).

5.3 Filter Trainings by Training Name

Endpoint: `GET /training/filter?name={trainingName}`

- **Description:** Retrieves training details filtered by training name.

5.4 Filter Trainings by Date

Endpoint: `GET /training/filter?startDate={startDate}&endDate={endDate}`

- **Description:** Retrieves training sessions within a specified date range.

5.5 Assign a Trainer to a Training

Endpoint: `POST /training/{trainingId}/assign-trainer/{trainerId}`

- **Description:** Assigns an available trainer to a training session.
- **Conditions:** Ensures that the trainer is available and meets the budget and skill requirements.
- **Effect:** Marks the trainer as unavailable for overlapping training sessions.

5.6 Get Available Trainers for a Training

Endpoint: `GET /trainer/available?trainingId={trainingId}`

- **Description:** Retrieves a list of trainers available for a given training session.
- **Conditions:** Filters based on skills, budget, and availability.

5.7 Create a New Vendor

Endpoint: `POST /vendor`

- **Description:** Adds a new vendor to the system.
- **Request Body:** Vendor details including name, code, country, and headquarters.

5.8 Create a New Training

Endpoint: `POST /training`

- **Description:** Adds a new training session to the system.
- **Request Body:** Training details such as name, skills required, duration, start date, end date, budget, and associated vendor.

5.9 Create a New Trainer

Endpoint: `POST /trainer`

- **Description:** Adds a new trainer to the system.
- **Request Body:** Trainer details including name, skills, budget, availability, and contact details.