



# Institute of Computer Engineering Technology

## Industry Training Selection Test

Registration Number :

## Employee Directory Project

### Project Overview

#### Description:

A web application designed for a corporate company to manage employee records, enabling HR personnel or administrators to manage employee details. The application needs to be developed using **Spring Boot** (backend), **Angular or React** (frontend), and **MySQL** (database), and is scoped for completion within a single day (8 am-6 pm hours). The project aligns with the operational model of **Fortium Partners**, a leading provider of interim and fractional technology leadership services, headquartered in Texas, USA, founded in 2003. Fortium Partners specializes in supplying mid-sized and large companies with experienced CIOs, CTOs, CISOs, and tech executives on a contract basis for special projects, transformations, or leadership transitions.

**Objective:** Deliver a functional CRUD (Create, Read, Update, Delete) application tailored to corporate HR needs, incorporating business logic to ensure data accuracy and compliance with organizational requirements. The application supports Fortium Partners' hands-on, executive-led approach by providing a practical tool that a fractional CIO or CTO could deploy to streamline HR processes.

**Corporate Context:** The application is intended for a mid-sized corporate company (100-500 employees) with a structured HR department, similar to **Fortium Partners'** typical clients. It supports HR processes such as onboarding, employee data management, and record updates, ensuring data integrity and usability for non-technical HR staff. The project demonstrates how Fortium's interim technology leadership can deliver rapid, high-impact solutions for corporate clients undergoing transformations or leadership transitions.

# Business Requirements

The Employee Directory addresses the following corporate business needs, reflecting the real-world requirements of a Fortium Partners client:

## 1. Centralized Employee Data Management

- Maintain a reliable repository for employee records to streamline HR operations, which is critical for mid-sized and large companies managing onboarding, offboarding, and audits.
- Enable the HR department to access employee details quickly, supporting Fortium's focus on delivering immediate, actionable solutions for clients.

## 2. Data Accuracy and Validation

- Ensure employee data (name, email, department) is valid and consistent to prevent errors in HR processes, aligning with Fortium's emphasis on operational excellence.
- Enforce unique email addresses to avoid duplicate records, a common issue in large organizations transforming.

## 3. Auditability for Compliance

- Track creation and update timestamps for employee records to support corporate governance and audit requirements.
- Provide transparency in data changes to meet compliance needs during leadership transitions or special projects.

## 4. User-Friendly Interface for HR Department

- Deliver an intuitive interface for HR staff with minimal technical expertise, reflecting Fortium's hands-on approach to embedding executive-level solutions that non-technical users can easily adopt.
- Provide clear feedback (e.g., success/error messages) to reduce user errors and ensure operational efficiency.

## 5. Scalability and Modularity

- Design a modular system that can accommodate future enhancements (e.g., authentication, integration with HR systems), mirroring Fortium’s strategic approach to building flexible solutions for mid-sized and large companies.

---

## Features

- **Create Employee:** Add a new employee with name, email, and department.
- **View Employees:** Display a list of all employees with details (ID, Name, Email, Department, Created At, Updated At).
- **Update Employee:** Edit an existing employee’s details while maintaining data integrity.
- **Delete Employee:** Remove an employee, ensuring only authorized deletions.
- **Business Logic:**
  - **Validation:**
    - Names must be non-empty, contain only alphabetic characters and spaces, and be limited to 100 characters.
    - Emails must follow a valid format (e.g., employee@company.com) and be unique across all employees.
    - Departments must be selected from a predefined list (e.g., HR, IT, Finance, Operations).
  - **Uniqueness:** Prevent duplicate email addresses to maintain data integrity.
  - **Timestamps:** Automatically record creation and update timestamps for audit purposes.
  - **Error Handling:** Provide clear error messages for invalid inputs or failed operations (e.g., “Email already exists”).

## Technology Stack

- **Backend:** Spring Boot
  - Enables rapid development of RESTful APIs with robust validation and error handling, ideal for Fortium's focus on delivering strategic technology solutions.
  - Uses Spring Data JPA for efficient database interactions.
- **Frontend:** Angular or React
  - Provides a dynamic, single-page application (SPA) for a responsive user experience, aligning with Fortium's emphasis on modern, user-friendly solutions.
  - Supports modular components for maintainability.
- **Database:** MySQL
  - Reliable relational database for structured employee data, suitable for mid-sized and large companies.
  - Supports auditability requirements with timestamp fields.

---

## Functional Requirements

### 1. Employee Data Model:

- Fields:
  - ID (auto-generated, unique identifier).
  - Name (string, non-empty, alphabetic characters and spaces, max 100 characters).
  - Email (string, valid format, unique).
  - Department (string, restricted to HR, IT, Finance, Operations).
  - Created At (datetime, auto-set on creation).
  - Updated At (datetime, auto-updated on modification).
- Validation is enforced at the backend to ensure data quality.

## 2. User Interface:

- **Employee List Page:**

- Table with columns: ID, Name, Email, Department, Created At, Updated At, Actions (Edit, Delete).
- Optional sorting by name or department, if time permits.

- **Add/Edit Employee Form:**

- Fields: Name (text), Email (text), Department (dropdown).
- Buttons: Submit (save), Cancel (return to list).
- Client-side validation for immediate feedback.

- Responsive design for desktop and tablet usability.

## 3. Business Logic:

- **Input Validation:**

- Name: Non-empty, alphabetic with spaces, max 100 characters.
- Email: Matches regex (e.g., `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$`).
- Department: Restricted to HR, IT, Finance, Operations.

- **Uniqueness Check:** Verify email uniqueness before creating or updating.

- **Timestamps:**

- Set Created At on creation.
- Update: Updated at modification.

- **Error Handling:**

- Return user-friendly error messages (e.g., “Invalid email format”).
- Log errors for debugging (optional, time-permitting).

## 4. API Endpoints:

- GET /api/employees: Retrieve all employees.
- POST /api/employees: Create a new employee.
- PUT /api/employees/{id}: Update an employee by ID.
- DELETE /api/employees/{id}: Delete an employee by ID.

## Note

- Developer has working knowledge of Spring Boot, Angular, and MySQL.
- Local development environment (Java, Node.js, MySQL) is pre-configured.
- No authentication required for the 1-day prototype, aligning with Fortium's focus on rapid delivery.
- The corporate company provides a department list (HR, IT, Finance, Operations).
- Tested with a small dataset (10-20 employees).
- Add corporate authentication (e.g., OAuth2) for HR staff, aligning with Fortium's security expertise.
- Implement search and filter options for employee data.
- Add export functionality (e.g., CSV) for HR reporting.

## Deliverables

- Functional Employee Directory application:
  - Spring Boot backend with RESTful APIs and business logic.
  - Angular frontend with employee list and form components.
  - MySQL database with audit timestamps.
- Documentation (this document).
- Tested CRUD operations with validated business logic.

## Risks and Mitigation

- **Risk:** Setup delays.
  - **Mitigation:** Use Spring Initializer, Angular CLI, and pre-configured MySQL.
- **Risk:** MySQL connectivity issues.
  - **Mitigation:** Test connection early using local MySQL or Docker.
- **Risk:** Business logic errors.

- **Mitigation:** Implement validation at the backend and the frontend, as well as test edge cases.
- **Risk:** Time overruns in the frontend.
  - **Mitigation:** Use a CSS framework, focus on core functionality.
- **Risk:** Misalignment with corporate needs.
  - **Mitigation:** Align requirements with Fortium's client model (mid-sized companies, HR focus).

