**Exercise 1: Employee Management System - Overview and Setup**

Business Scenario:

You are developing an employee management system that will manage employee data, departments, and their relationships.

**1. Introduction**

In this exercise, I initialized the Spring Boot Project named EmployeeManagementSystem using Spring Initializr, and added the following dependencies:

Spring Data JPA

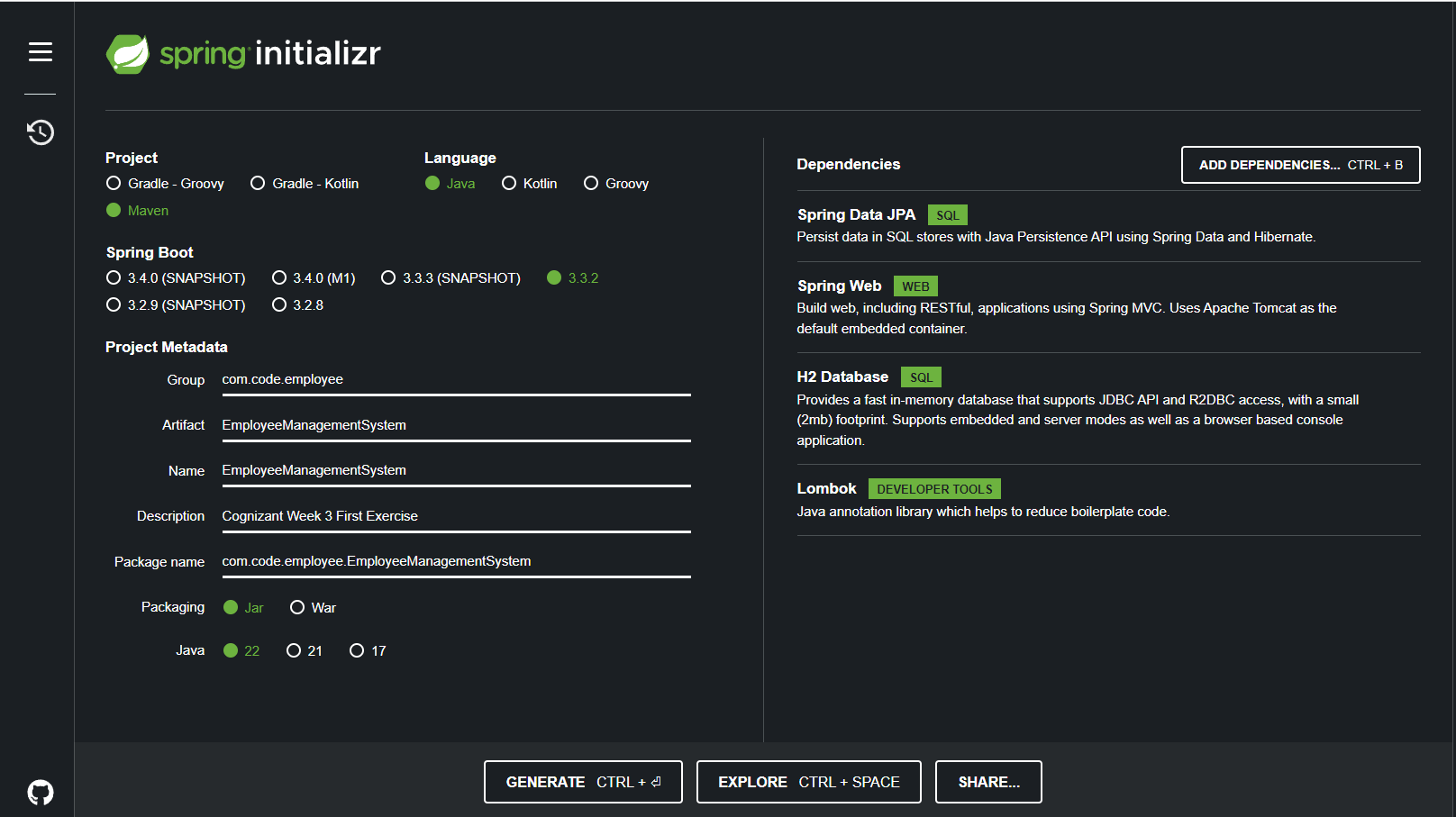
H2 Database

Spring Web

Lombok

**The necessary steps for creating the Spring Boot Project are:**

* Project: Maven Project
* Language: Java
* Spring Boot: 3.3.2
* Project Metadata:
* Group: com.code.employee
* Artifact: EmployeeManagementSystem
* Name: EmployeeManagementSystem
* Description: Cognizant Week 3 First Exercise
* Package Name: com.code.employee.EmployeeManagementSystem
* Packaging: Jar
* Java Version: 22
* Add the necessary dependencies:
* Spring Data JPA
* H2 Database
* Spring Web
* Lombok
* Click on the Generate button. This will download a .zip file.
* Extract the .zip file to Eclipse workspace.
* Open the extracted folder in Eclipse.



**2. Configuring Application Properties:**

Configured the application.Properties for H2 Database Connection:

*spring.datasource.url=jdbc:h2:mem:testdb*

*spring.datasource.driverClassName=org.h2.Driver*

*spring.datasource.username=sa*

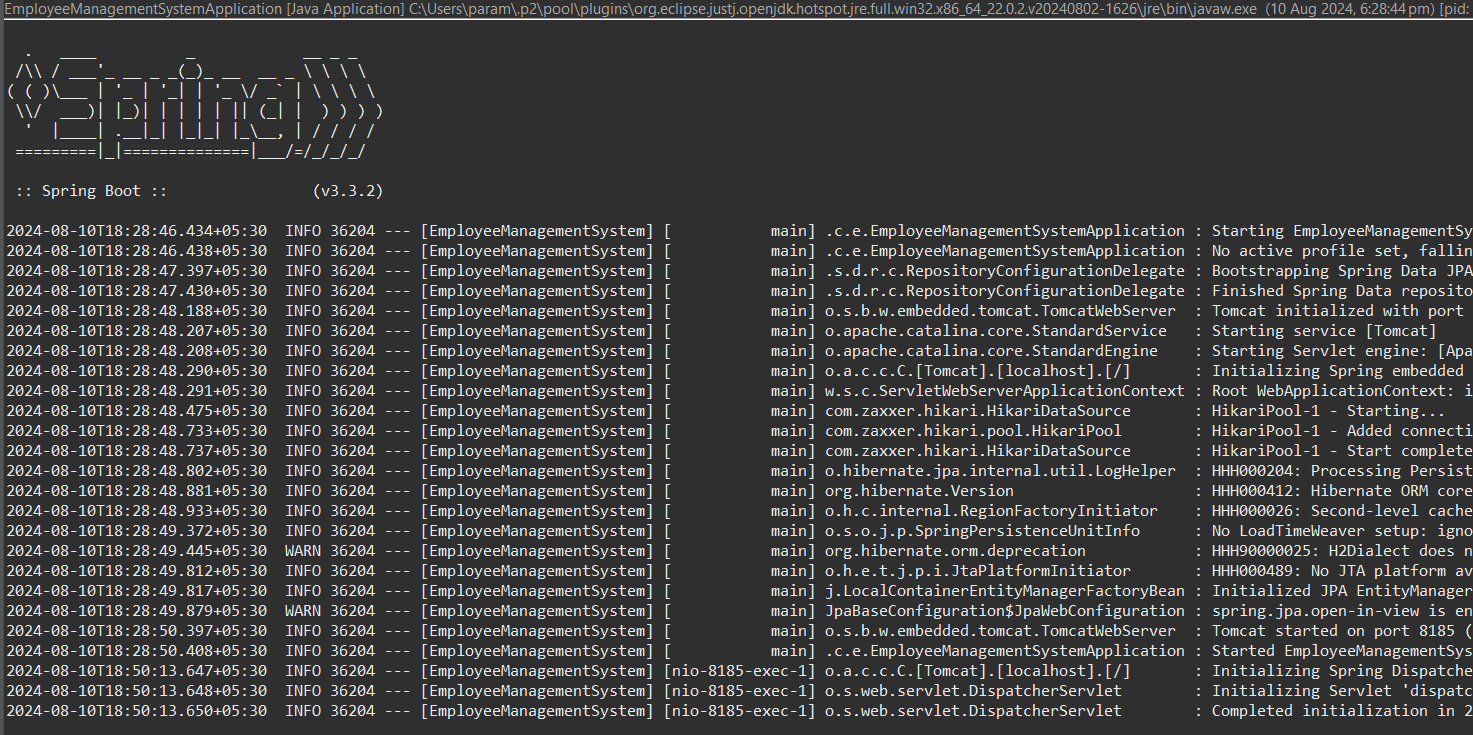
*spring.datasource.password=password*

*spring.jpa.database-platform=org.hibernate.dialect.H2Dialect*

*server.port=8185*

(server.port=8185 is set because the default server 8080 is busy)

**Output:**

****

**Analysis:**

* **Spring Data JPA**: Spring Data JPA, part of the larger Spring Data family, makes it easy to easily implement JPA-based (Java Persistence API) repositories. It makes it easier to build Spring-powered applications that use data access technologies.

Implementing a data access layer for an application can be quite cumbersome. Too much boilerplate code has to be written to execute the simplest queries. Add things like pagination, auditing, and other often-needed options, and you end up lost.

Spring Data JPA aims to significantly improve the implementation of data access layers by reducing the effort to the amount that’s actually needed. As a developer you write your repository interfaces using any number of techniques, and Spring will wire it up for you automatically. We can even use custom finders or query by example and Spring will write the query for us.

* **H2 Database:** H2 is an open-source lightweight Java database. It can operate in embedded and server modes. In-memory databases like H2 are useful for development and testing since they are fast and do not require a persistent storage layer.
* **Spring Web**: Spring Web Services (Spring-WS) is focused on creating document-driven Web services. Spring Web Services aims to facilitate contract-first SOAP service development, allowing for the creation of flexible web services using one of the many ways to manipulate XML payloads. The product is based on Spring itself, which means we can use the Spring concepts such as dependency injection as an integral part of our Web service.
* **Project Lombok:** Project Lombok is a java library that automatically plugs into our editor and build tools, spicing up java. We need not to write another getter or equals method again, with one annotation your class has a fully featured builder, Automate our logging variables, and much more.