Explaining About POM Dependencies

## Spring-boot-starter-JPA

JPA is nothing but “Java persistent API” it is used to perform CRUD operations to entities using a repository interface by implementing JPARepository<Class Name, Datatype>.

## Spring-boot-starter-validation

It is used for building web applications using Spring Mobile. It is **used for Java Bean Validation with Hibernate Validator**. It is used to build a hypermedia-based RESTful web application with Spring MVC and Spring HATEOAS.

## Spring-boot-starter-web

Spring Boot Starter Web is used for **building RESTful applications using Spring MVC**. Spring Boot Starter Tomcat is the default embedded container for Spring Boot Starter Web. We cannot exclude it while using web services. We can exclude it when we want to use another embedded container

## Spring-boot-starter-security

Spring Boot provides a spring-boot-starter-security starter which **aggregates Spring Security related dependencies together**.

## Spring-boot-starter-test

The spring-boot-starter-test is the **primary starter dependency for testing our spring boot application**. It contains the majority of libraries that are required for tests including JUnit Jupiter, Hamcrest, and Mockito.

## Spring-boot-devtools

The spring-boot-devtools module includes an embedded Live Reload server that can be used **to trigger a browser refresh when a resource is changed**. Live Reload browser extensions are freely available for Chrome, Firefox and Safari from livereload.com.

## MySQL-connector-j

This is used to get a required jar files and drivers to connect with MySQL

## Lombok

It is used to autogenerate setters, getters and constructors for entities to avoid boilerplate of code if we generate it manually.

## Model Mapper (version: 3.1.0)

To avoid having to write cumbersome/boilerplate code to map DTOs into entities and vice-versa, we are going to use a library called ModelMapper. The goal of ModelMapper is **to make object mapping easy by automatically determining how one object model maps to another**.

## JJWT (version: 0.9.1)

The use of JWT token **for authorization** is the most common of its applications. The token is usually generated in the server and sent to the client where it is stored in the session storage or local storage. To access a protected resource the client would send the JWT in the header as given above.