PLACEMENT PORTAL

Project directory : E:\ITVedant\SpringProject\Placement\_Portal

Step 1

https://spring.io/tools

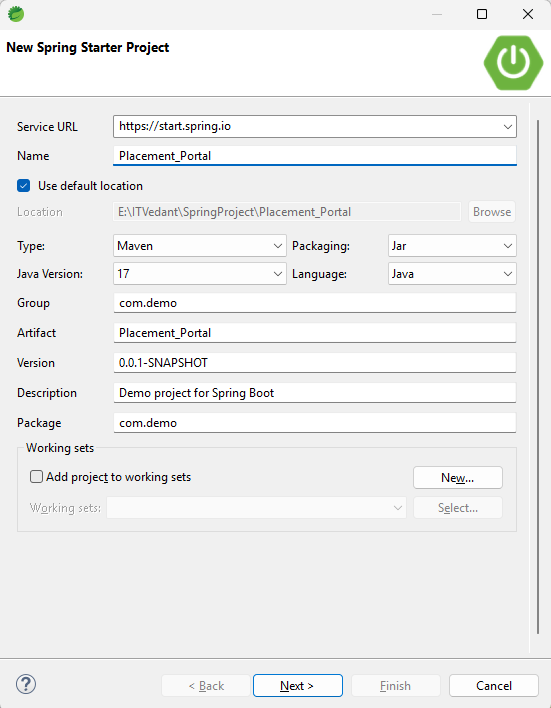
download - 4.25.0 windows x86\_64

extract file inside E drive ITVedant folder - open stringToolSuite4

pre request – we need

STS, MySQL workbench & Lombok

Step 2 : spring start project

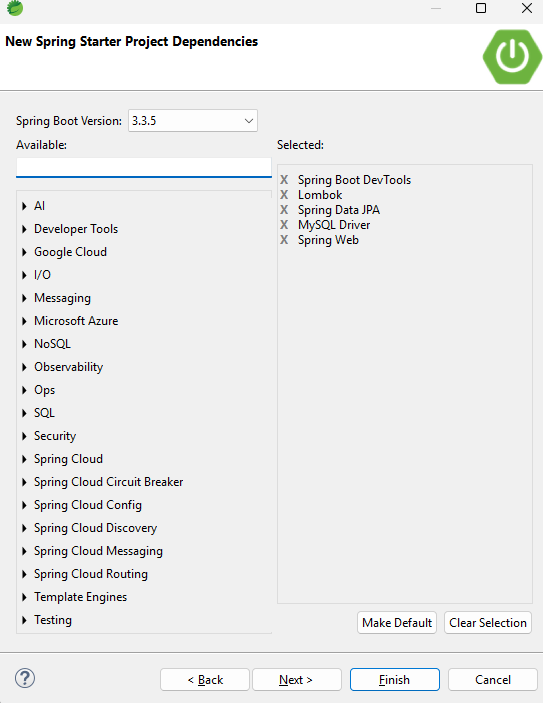


Step3: add dependency web, mysql(for db), jpa(for generating table),

Lombok: we use Lombok - to reduce code - **loosely couple** – do with getter setter or contructors - to skip the multiple getter setter

devtools : act as live server(to run project automatic)

visible inside Pom.xml



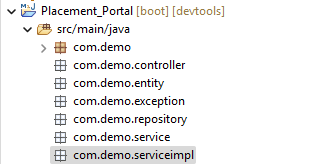
Step4 : create DB(database)

Workbench

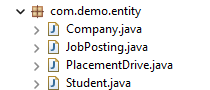
CREATE DATABASE Anitaplacement;

use Anitaplacement;

step 5 basic structure of project



Step 6 make entity



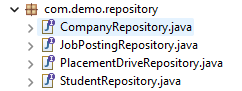
Step 7 make repository

**Repository** – help to do all action line CRUD operation – give all logic for CRUD operation. Ex – save(), findAll(), getById(), **for all this method we extends JpaRepository**

**package** com.demo.repository;

**public** **interface** CompanyRepository {

}



Step 8 database connectivity

spring.application.name=Placement\_Portal

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/Anitaplacement //database name

spring.datasource.username=root

spring.datasource.password=anita@123

spring.datasource.driver-class-name =com.mysql.cj.jdbc.Driver

server.error.include-message=always

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL8Dialect

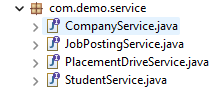
step 9

create interfaces for entities (declare CRUD() )

**package** com.demo.service;

**public** **interface** CompanyService {

}



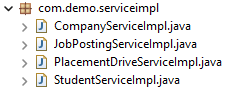
Step 10

create classes for serviceimp interfaces( logic of CRUD() )

**package** com.demo.serviceimpl;

**public** **class** CompanyServiceImpl {

}



Step 11

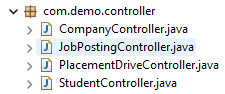
Controller : handle incoming request and share response based on request

Get : fetch data from application (status code :200)

Post – update, insert data into application (status code :201)

Delete – used to remove data from application (status code :204)

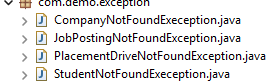
Put : used to modify existing data from application (status code :200)



Step 12 exception :

Exception : if entered data is correct , which is already entered

Validation : if user enter correct data or not



Step13 add validation

Add pom.xml dependency

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-validation</artifactId>

</dependency>