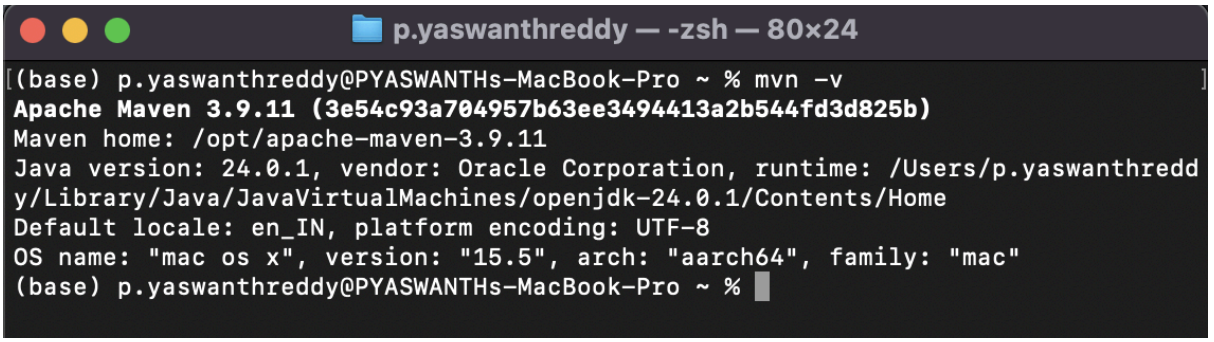
1. Maven version terminal screenshot

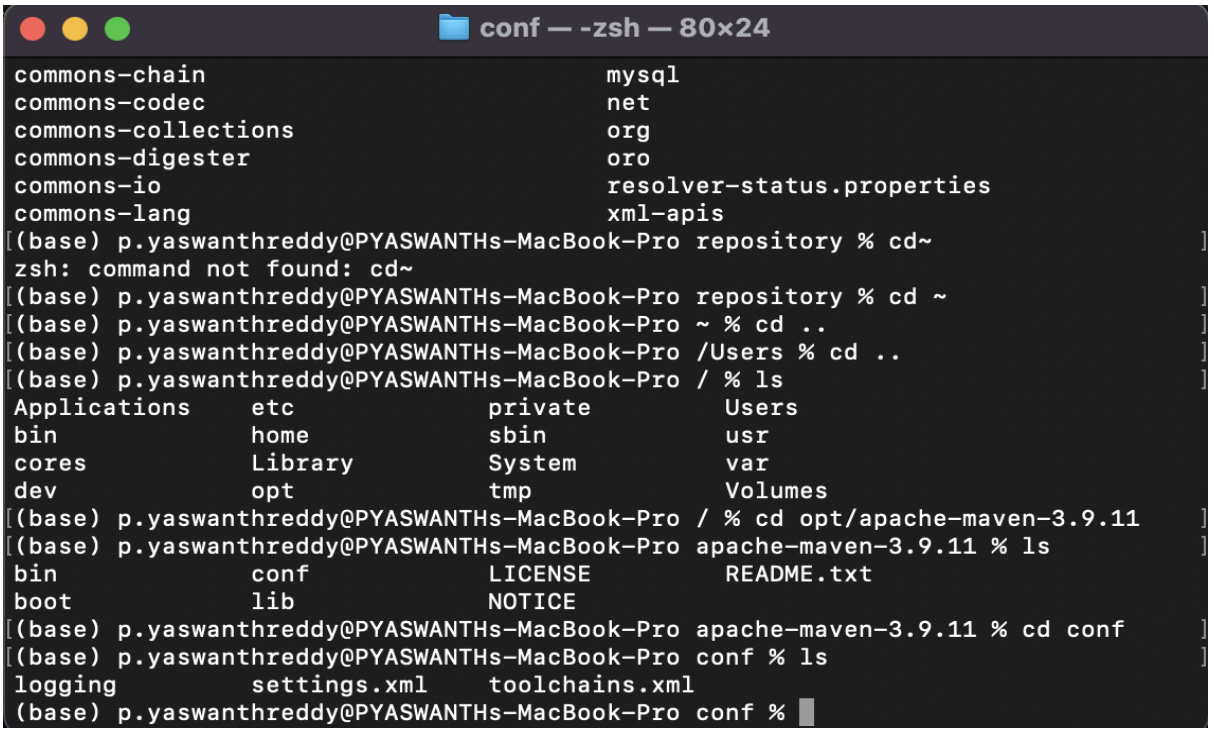


2. **Maven Central Repository**
   - A remote, online repository managed by the Maven community.
   - It Contains a vast collection of open-source libraries and dependencies.
   - Maven checks here when a dependency is not found locally.

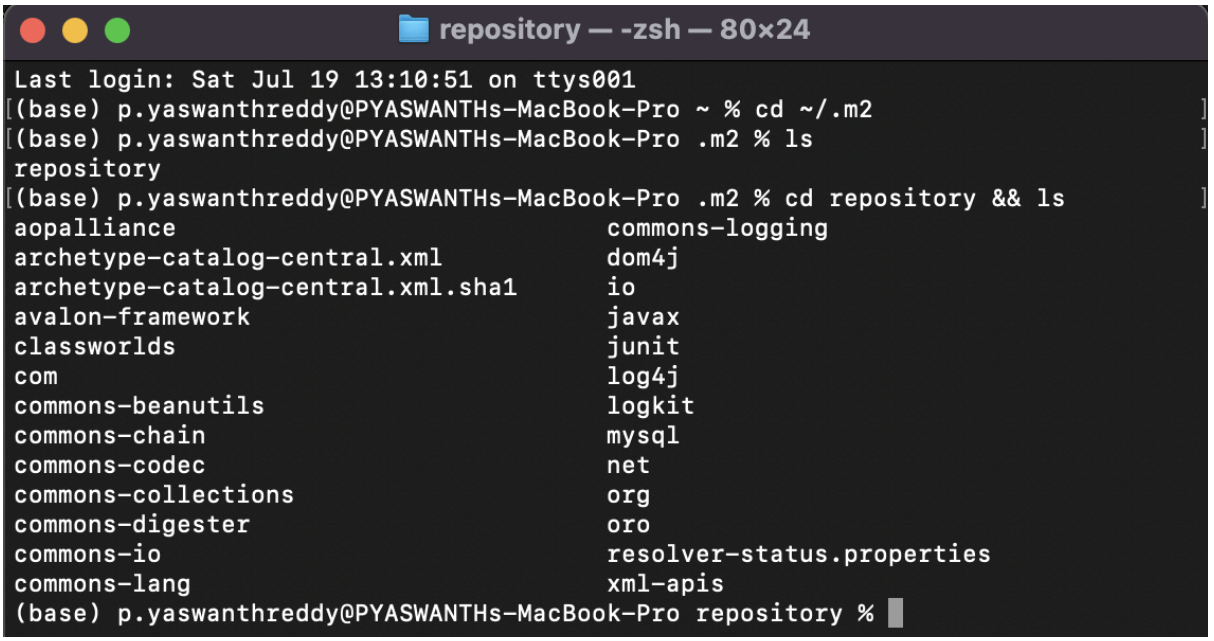
   Local Repository
   - A directory on our local machine (usually ~/.m2/repository).
   - Stores dependencies that are downloaded from central or remote repositories.
   - Maven checks the local repository first before going online.
   - Improves build speed and reduces internet dependency.

3. **Maven commands**
   a. To build the maven project - **mvn clean install**
   b. To run the maven tests - **mvn test**

4. Maven settings.xml



Maven local repository



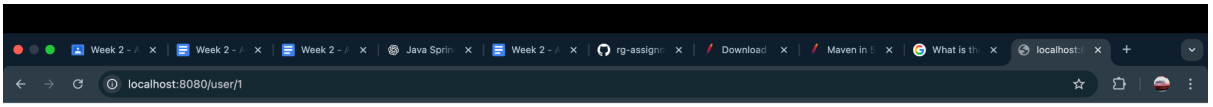
5. Different ways to Define Dependencies in DI:
    a. Setter Injection
    b. Constructor Injection
    c. Field Injection
    d. Interface Injection

6. @Autowired can be applied to fields or constructors to allow the Spring Framework to automatically inject dependencies. Similarly, @Inject serves the same purpose. The main difference is that @Inject is a standard annotation defined by JSR-330, whereas @Autowired is specific to the Spring Framework.

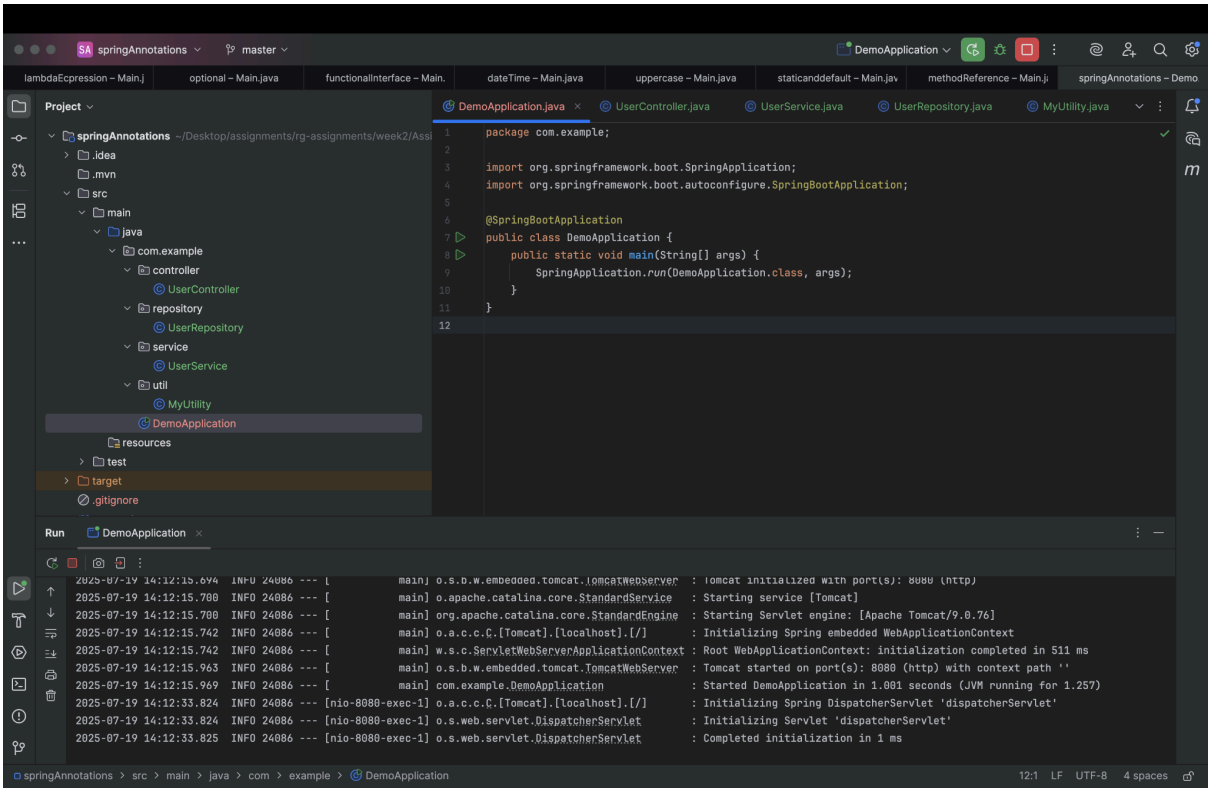
7. In Spring Framework, the annotations @Component, @Repository, @Service, and @Controller are stereotype annotations which are used to define Spring-managed beans and their specific roles in an application.

    1. @Component
        - Generic stereotype for any Spring-managed component.
        - It is the parent of all stereotype annotations.
    2. @Repository
        - Specialization of @Component.
        - Used for DAO (Data Access Object) classes.
        - Enables exception translation into Spring's DataAccessException
    3. @Service
        - Specialization of @Component.
        - Used for business logic/service layer classes.
    4. @Controller
        - Specialization of @Component.
        - Used to define a web controller in Spring MVC.

Output



USER1



Github link:
https://github.com/PathireddyYaswanthReddy/rg-assignments/blob/master/week2/Assignment%201%20Spring/springAnnotations/src/main/java/com/example/DemoApplication.java

8. The issue is with the below line
   @Value("app.name")
   The above statement is just injecting the literal string "app.name" instead of the value of the property from application.properties.

   To fix this we need to wrap the property key in ${} so Spring knows it's a property reference.

   Correct Code

```java
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component
public class AppNamePrinter {

    @Value("${app.name}") // Correct way to reference a property
    private String appName;
```

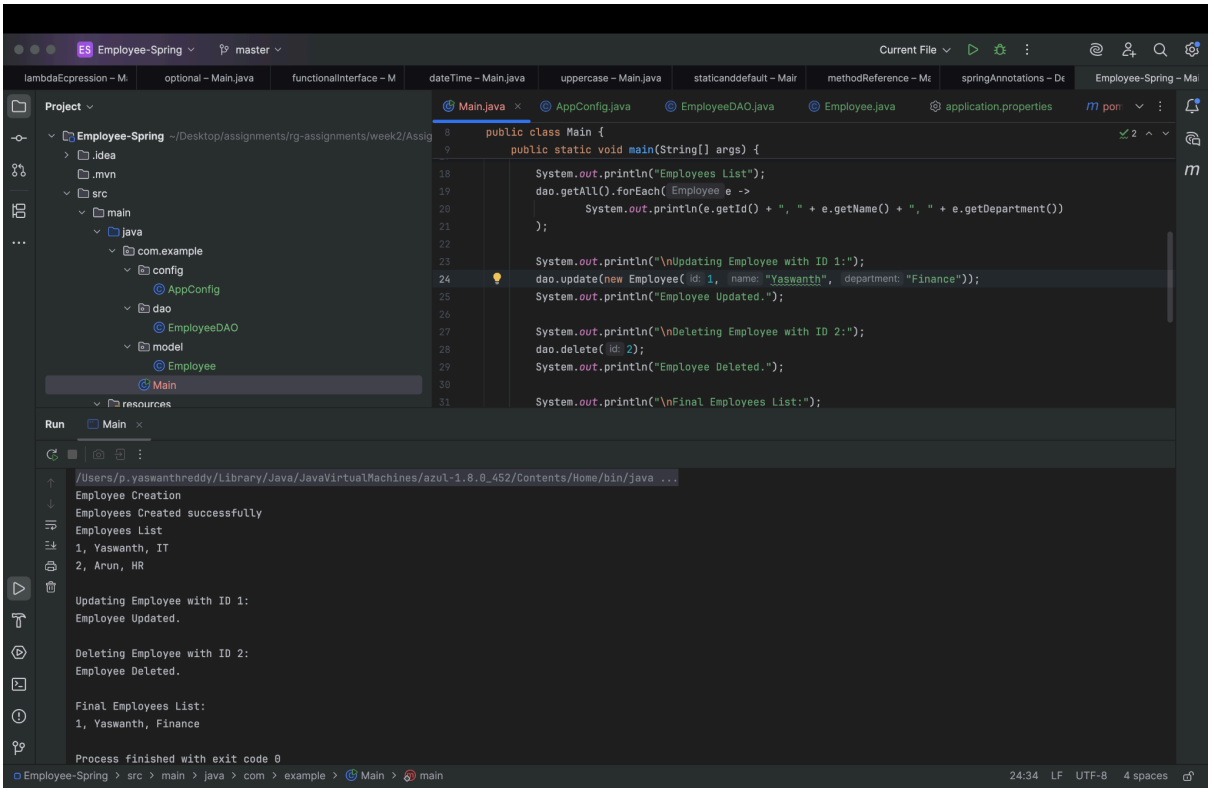
```
    public void printAppName() {
        System.out.println("Application Name: " + appName);
    }
}
```

9. The @SpringBootApplication is a convenience annotation in Spring Boot that combines the following three annotations:
    - @Configuration
    - @EnableAutoConfiguration
    - @ComponentScan

    It tells Spring Boot to start auto-configuring your app, scan for @Component, @Service, @Repository, and @Controller classes and treat the class as a configuration class.

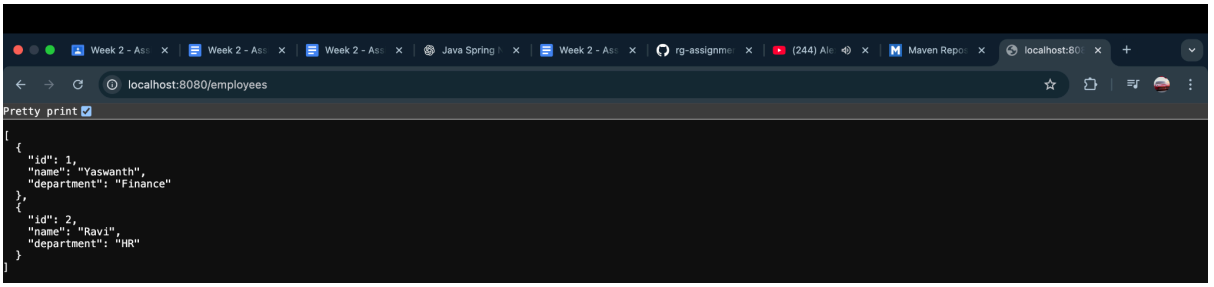
10. Maven command to start spring boot application - **mvn spring-boot:run**

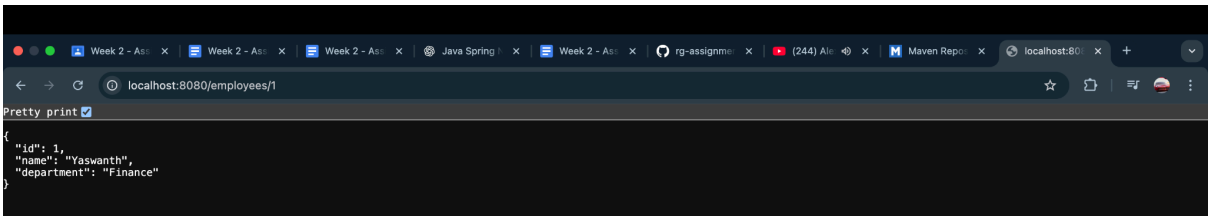
11. Spring Task Output Screenshots



Github link:
https://github.com/PathireddyYaswanthReddy/rg-assignments/blob/master/week2/Assignment%201%20Spring/Employee-Spring/src/main/java/com/example/Main.java

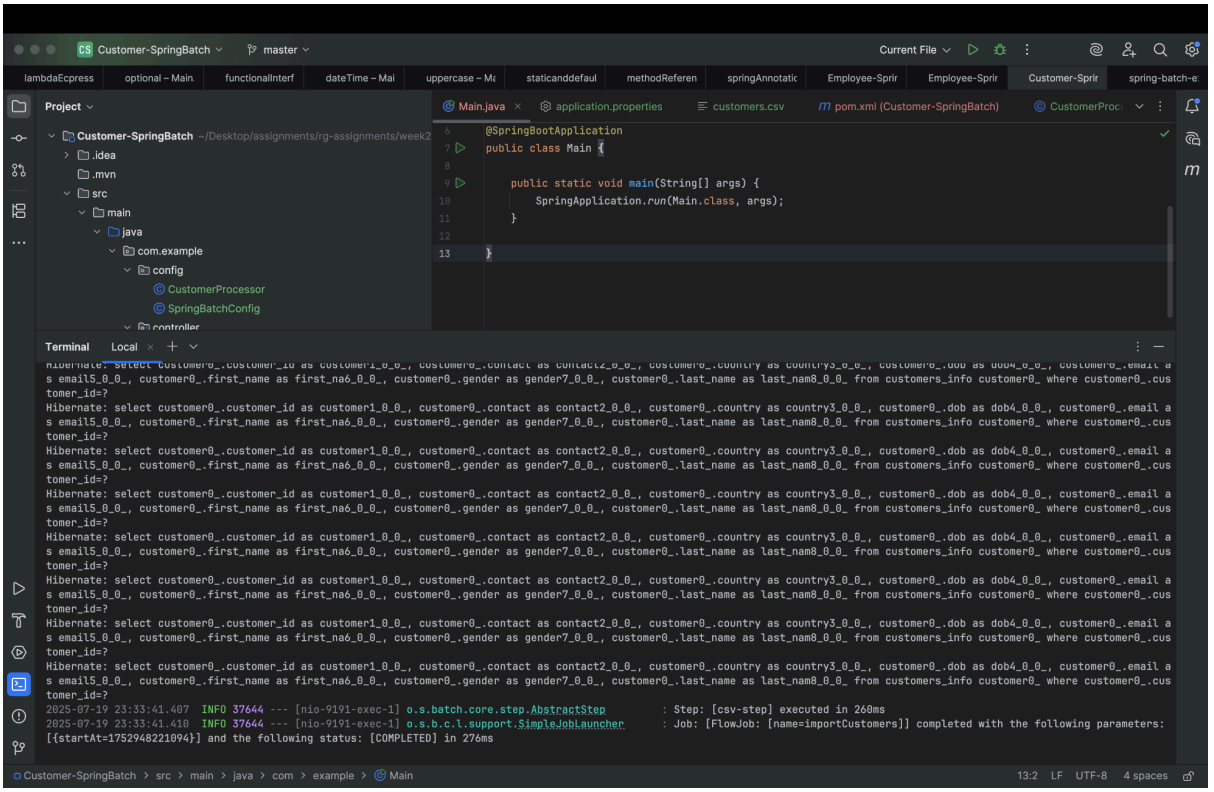
12. Spring Boot and Spring Data JPA output screenshots





Github link:
https://github.com/PathireddyYaswanthReddy/rg-assignments/blob/master/week2/Assignment%201%20Spring/Employee-SpringBoot-JPA/src/main/java/com/example/Main.java

13. Spring Batch output screenshots



```
(base) p.yaswanthreddy@PYASWANTHs-MacBook-Pro ~ % curl -X POST http://localhost:
9191/jobs/importCustomers
(base) p.yaswanthreddy@PYASWANTHs-MacBook-Pro ~ %
```

```
mysql> show tables;
| Tables_in_testdb |
+------------------+
| Employee         |
+------------------+
1 row in set (0.004 sec)

mysql> show tables;
+------------------------------+
| Tables_in_testdb             |
+------------------------------+
| BATCH_JOB_EXECUTION          |
| BATCH_JOB_EXECUTION_CONTEXT  |
| BATCH_JOB_EXECUTION_PARAMS   |
| BATCH_JOB_EXECUTION_SEQ      |
| BATCH_JOB_INSTANCE           |
| BATCH_JOB_SEQ                |
| BATCH_STEP_EXECUTION         |
| BATCH_STEP_EXECUTION_CONTEXT |
| BATCH_STEP_EXECUTION_SEQ     |
| customers_info               |
| Employee                     |
+------------------------------+
11 rows in set (0.006 sec)

mysql> select * from customers_info limit 5;
+-------------+--------------+-------------+------------+------------------------+------------+--------+-----------+
| customer_id | contact      | country     | dob        | email                  | first_name | gender | last_name |
+-------------+--------------+-------------+------------+------------------------+------------+--------+-----------+
|           3 | 997-395-9270 | France      | 12-05-1994 | sgaw2@mit.edu          | Sallie     | Female | Gaw       |
|           6 | 615-444-4318 | Brazil      | 24-05-1990 | wdouberday5@mozilla.org | West       | Male   | Douberday |
|          10 | 654-596-4163 | Netherlands | 04-08-1999 | borridge9@exblog.jp    | Babs       | Female | Orridge   |
|          13 | 627-566-6711 | China       | 16-11-2012 | thoodlassc@ucsd.edu    | Thelma     | Female | Hoodlass  |
|          18 | 641-445-0261 | Brazil      | 22-02-2015 | nbaggallyh@discuz.net  | Noelani    | Female | Baggally  |
+-------------+--------------+-------------+------------+------------------------+------------+--------+-----------+
5 rows in set (0.004 sec)

mysql> select count(*) from customers_info;
+----------+
| count(*) |
+----------+
|     1000 |
+----------+
1 row in set (0.001 sec)

mysql>
```

Github link:
https://github.com/PathireddyYaswanthReddy/rg-assignments/blob/master/week2/Assignment%201%20Spring/Customer-SpringBatch/src/main/java/com/example/Main.java