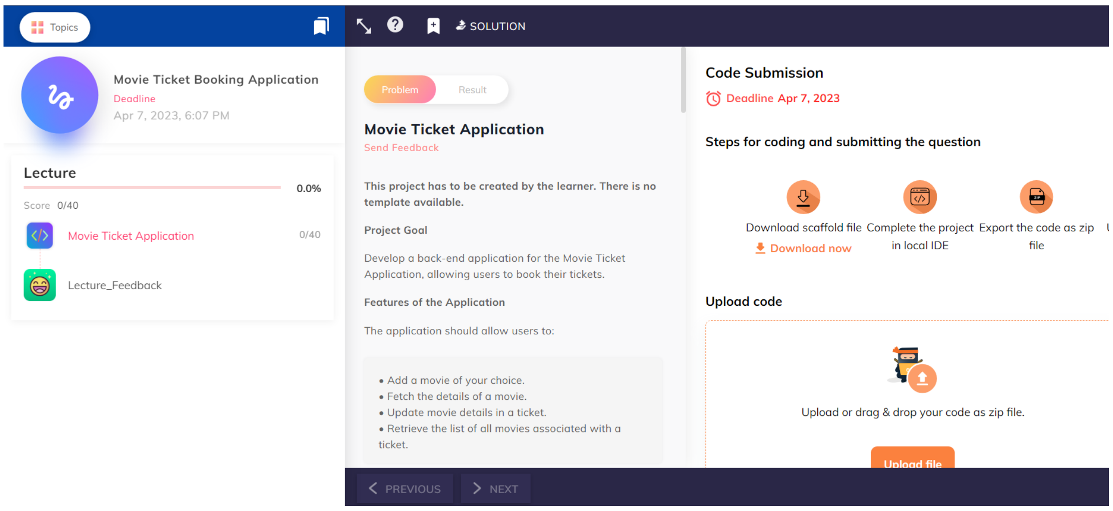
How To Attempt Mini Project?

This document aims to help learners attempt mini-projects. Learners are advised to read the document before attempting any mini-projects. To assist learners, a step-by-step guide is provided.

Guide on Attempting Mini Project:

1. Below is a mini-project from the course, and we will see how to attempt it.



Topics

SOLUTION

Movie Ticket Booking Application
Deadline
Apr 7, 2023, 6:07 PM

Lecture
0.0%
Score 0/40
Movie Ticket Application 0/40
Lecture_Feedback

Problem Result

Movie Ticket Application
Send Feedback

This project has to be created by the learner. There is no template available.

Project Goal

Develop a back-end application for the Movie Ticket Application, allowing users to book their tickets.

Features of the Application

The application should allow users to:

- Add a movie of your choice.
- Fetch the details of a movie.
- Update movie details in a ticket.
- Retrieve the list of all movies associated with a ticket.

Code Submission
Deadline Apr 7, 2023

Steps for coding and submitting the question

Download scaffold file
Download now
Complete the project in local IDE
Export the code as zip file

Upload code

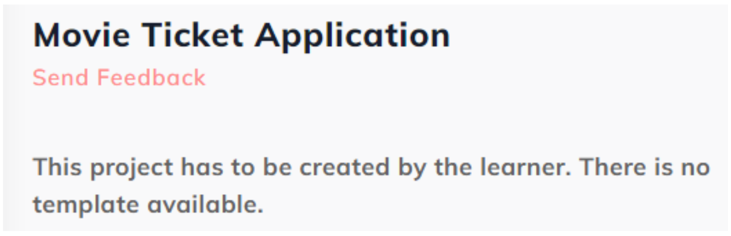
Upload or drag & drop your code as zip file.

Upload file

PREVIOUS NEXT

2. Mini-projects are like coding problems but with a few differences.

3. The learner must create a mini-project from scratch, as no template is available.



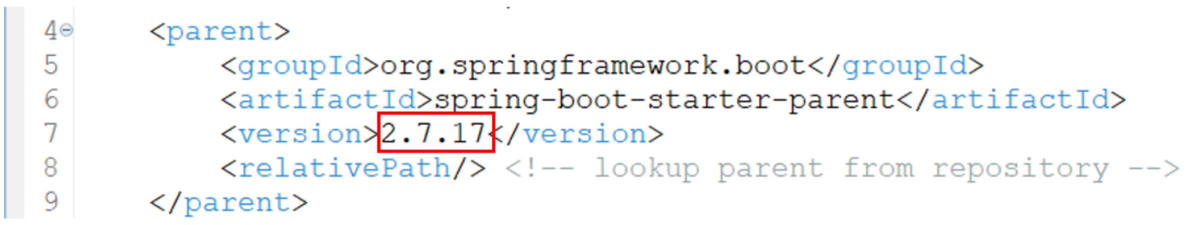
Movie Ticket Application
Send Feedback

This project has to be created by the learner. There is no template available.

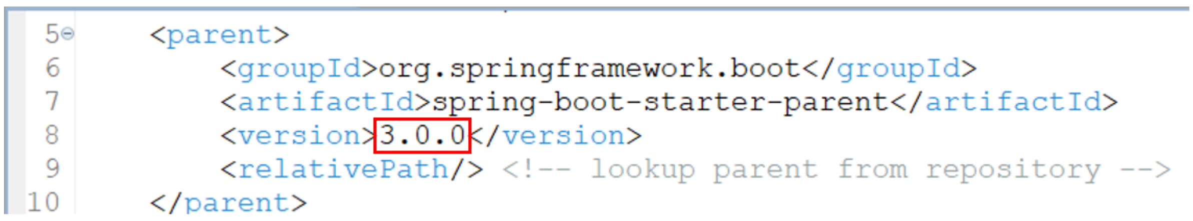
4. However, there are some configurations you are supposed to do to submit your project successfully.

5. The version for Springboot should be **3.0.0,** and for Java, it should be **17**.

6. If you cannot find the mentioned version for spring, then simply change the version to **3.0.0** in your **pom.xml** and update the maven project as shown below.

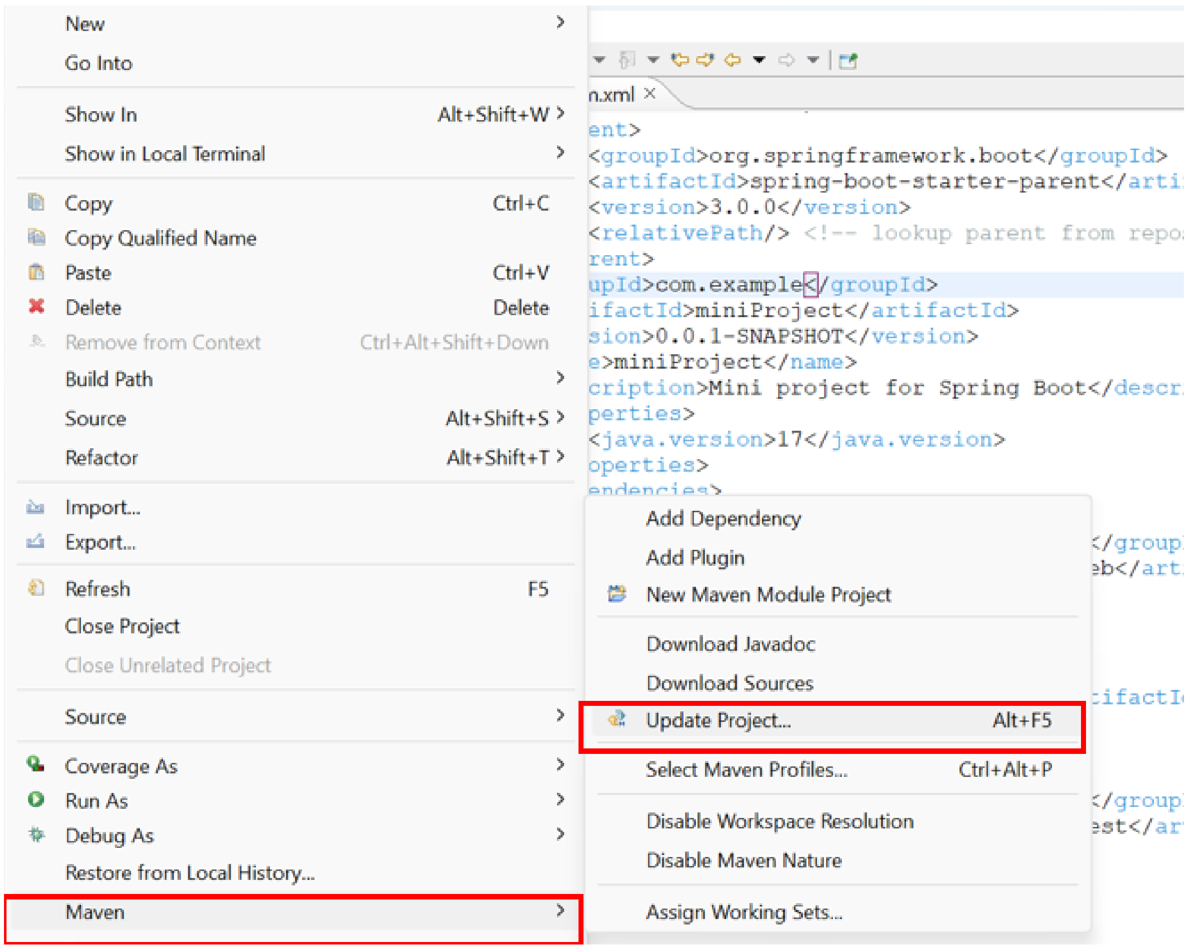


```
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.17</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
```



```
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.0.0</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
```

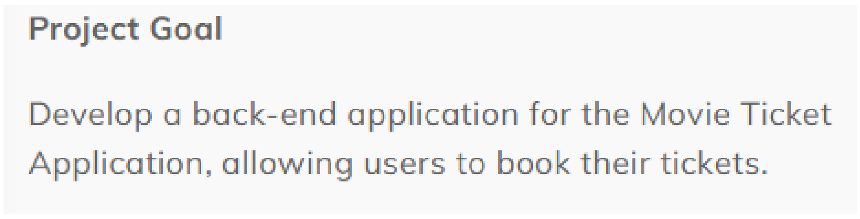
7. We are updating the maven project for the new spring version.



```
ent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</arti
<version>3.0.0</version>
<relativePath/> <!-- lookup parent from repo
rent>
upId>com.example</groupId>
ifactId>miniProject</artifactId>
sion>0.0.1-SNAPSHOT</version>
e>miniProject</name>
cription>Mini project for Spring Boot</descr
perties>
<java.version>17</java.version>
operties>
endencies>
```

8. It consists of the following segments as mentioned below:

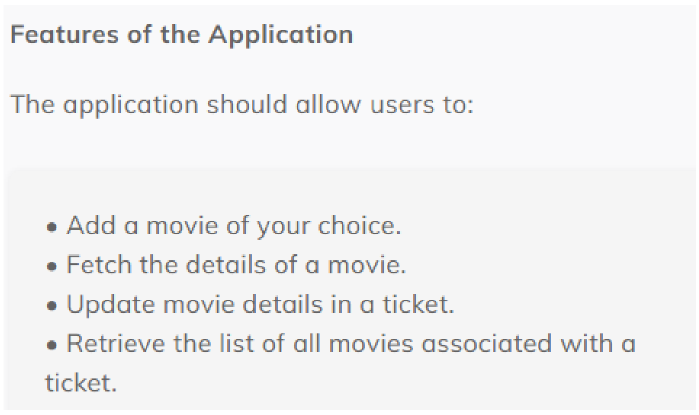
- **Project Goal: It includes the primary motive of what we will do.**



Project Goal

Develop a back-end application for the Movie Ticket Application, allowing users to book their tickets.

- **Features of the Application: It gives a general description of the features the application will have**

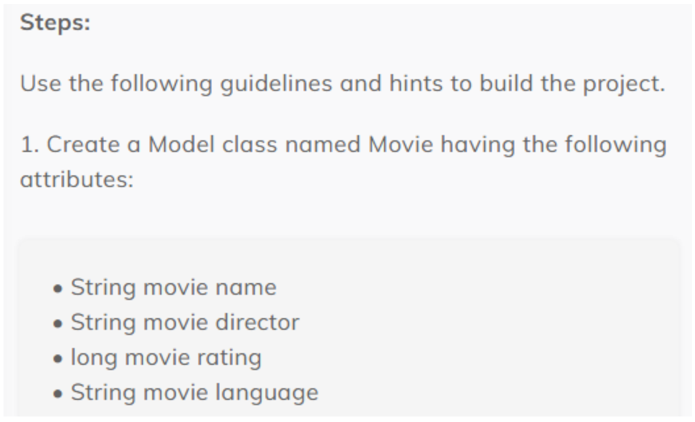


Features of the Application

The application should allow users to:

- Add a movie of your choice.
- Fetch the details of a movie.
- Update movie details in a ticket.
- Retrieve the list of all movies associated with a ticket.

- **Steps: To solve a mini project successfully, we have broken them down into smaller steps, and you must complete them accordingly.**



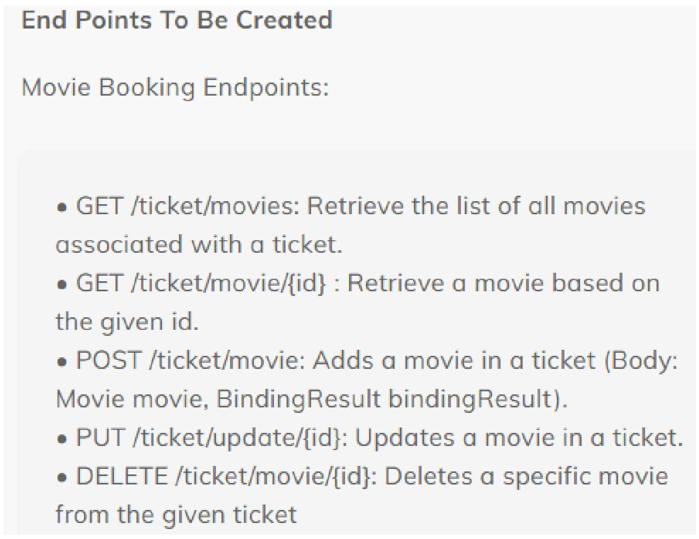
Steps:

Use the following guidelines and hints to build the project.

1. Create a Model class named Movie having the following attributes:

- String movie name
- String movie director
- long movie rating
- String movie language

- **End Points To Be Created: It contains the information regarding the APIs you are supposed to design.**

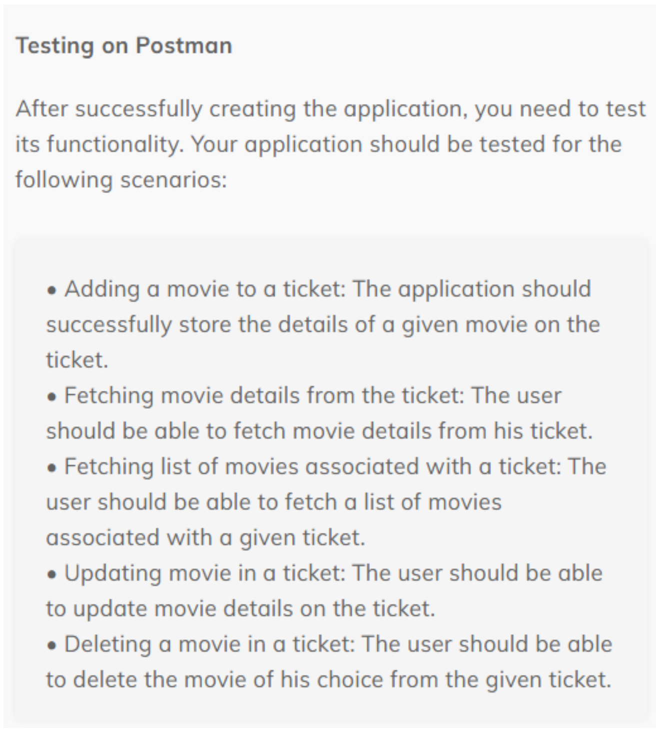


End Points To Be Created

Movie Booking Endpoints:

- GET /ticket/movies: Retrieve the list of all movies associated with a ticket.
- GET /ticket/movie/{id} : Retrieve a movie based on the given id.
- POST /ticket/movie: Adds a movie in a ticket (Body: Movie movie, BindingResult bindingResult).
- PUT /ticket/update/{id}: Updates a movie in a ticket.
- DELETE /ticket/movie/{id}: Deletes a specific movie from the given ticket

- **Testing on Postman: This section briefs about testing your application through Postman.**

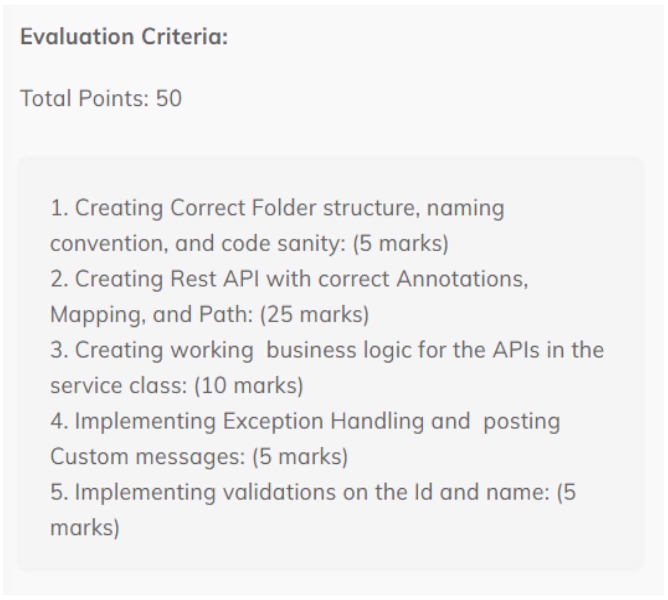


Testing on Postman

After successfully creating the application, you need to test its functionality. Your application should be tested for the following scenarios:

- Adding a movie to a ticket: The application should successfully store the details of a given movie on the ticket.
- Fetching movie details from the ticket: The user should be able to fetch movie details from his ticket.
- Fetching list of movies associated with a ticket: The user should be able to fetch a list of movies associated with a given ticket.
- Updating movie in a ticket: The user should be able to update movie details on the ticket.
- Deleting a movie in a ticket: The user should be able to delete the movie of his choice from the given ticket.

- **Evaluation Criteria: It includes the factor your mini project will be tested.**

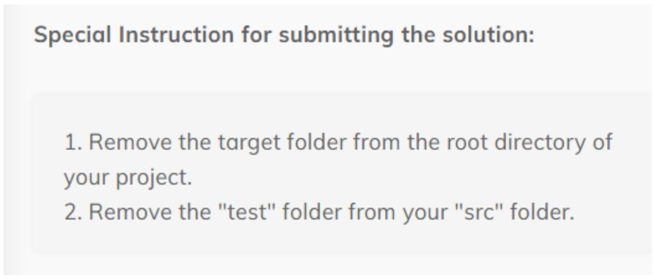


Evaluation Criteria:

Total Points: 50

1. Creating Correct Folder structure, naming convention, and code sanity: (5 marks)
2. Creating Rest API with correct Annotations, Mapping, and Path: (25 marks)
3. Creating working business logic for the APIs in the service class: (10 marks)
4. Implementing Exception Handling and posting Custom messages: (5 marks)
5. Implementing validations on the Id and name: (5 marks)

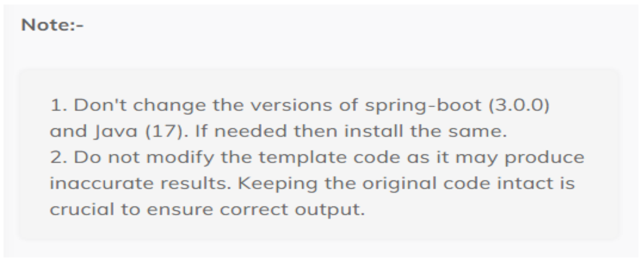
9. Special Instructions have guidelines for submitting a solution that needs to be followed before submission.



Special Instruction for submitting the solution:

1. Remove the target folder from the root directory of your project.
2. Remove the "test" folder from your "src" folder.

10. The Note section contains the do's and don'ts, which are the basic requirements for the project.



Note:-

1. Don't change the versions of spring-boot (3.0.0) and Java (17). If needed then install the same.
2. Do not modify the template code as it may produce inaccurate results. Keeping the original code intact is crucial to ensure correct output.

Configuration for Enabling Auto Testing in the Project

1. You need to add the following dependencies to your **pom.xml** as it enables the testing feature for your mini project. **(This Is required to ensure the testing is working correctly).**

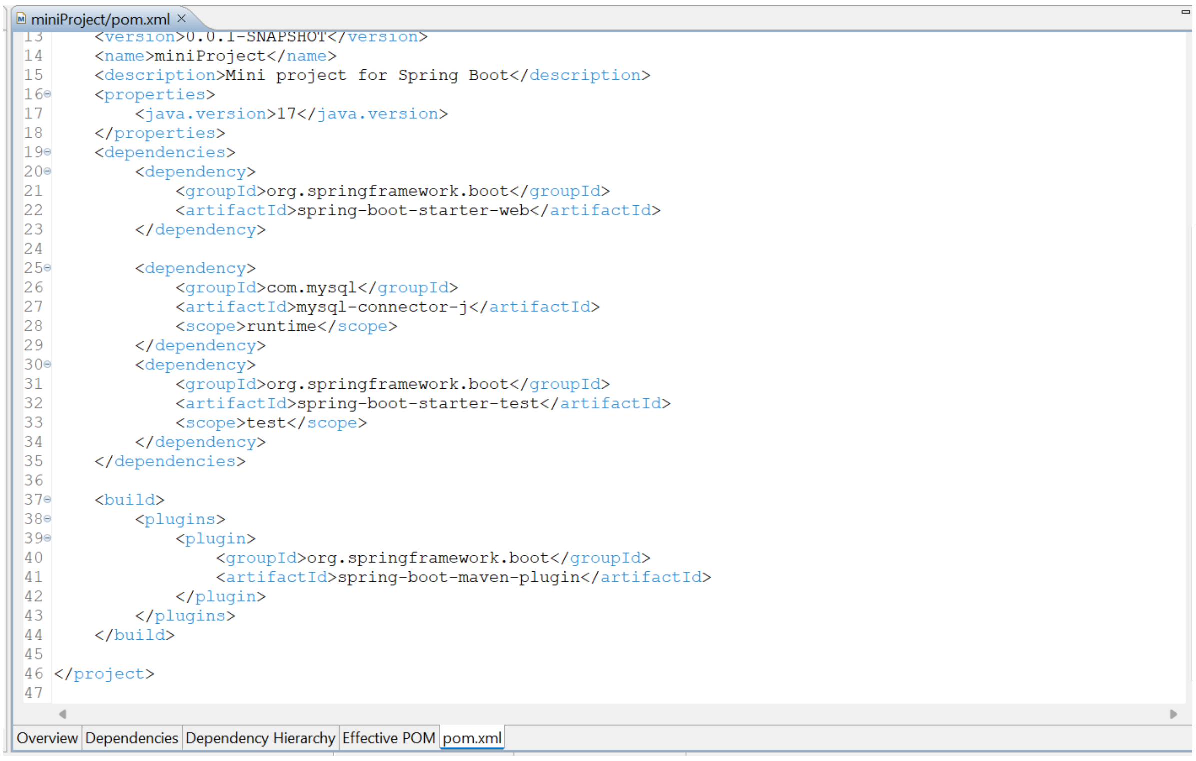
   Link:- pluginfile Link

```
<plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>3.1.2</version>
        <configuration>
          <statelessTestsetReporter
implementation="org.apache.maven.plugin.surefire.extensions.junit5.JUnit5Xml30State
lessReporter">
            <disable>false</disable>
            <version>3.0</version>
            <usePhrasedFileName>false</usePhrasedFileName>
            <usePhrasedTestSuiteClassName>true</usePhrasedTestSuiteClassName>
            <usePhrasedTestCaseClassName>true</usePhrasedTestCaseClassName>
            <usePhrasedTestCaseMethodName>true</usePhrasedTestCaseMethodName>
          </statelessTestsetReporter>
          <consoleOutputReporter
implementation="org.apache.maven.plugin.surefire.extensions.junit5.JUnit5ConsoleOut
putReporter">
            <disable>false</disable>
            <encoding>UTF-8</encoding>
            <usePhrasedFileName>false</usePhrasedFileName>
          </consoleOutputReporter>
          <statelessTestsetInfoReporter
implementation="org.apache.maven.plugin.surefire.extensions.junit5.JUnit5StatelessT
estsetInfoReporter">
            <disable>false</disable>
            <usePhrasedFileName>false</usePhrasedFileName>
            <usePhrasedClassNameInRunning>true</usePhrasedClassNameInRunning>

<usePhrasedClassNameInTestCaseSummary>true</usePhrasedClassNameInTestCaseSummary>
          </statelessTestsetInfoReporter>

<reportsDirectory>${project.build.directory}/custom-reports</reportsDirectory>
        </configuration>
      </plugin>
```

2. After creating the project from Spring Initializr, your pom.xml will look like below.

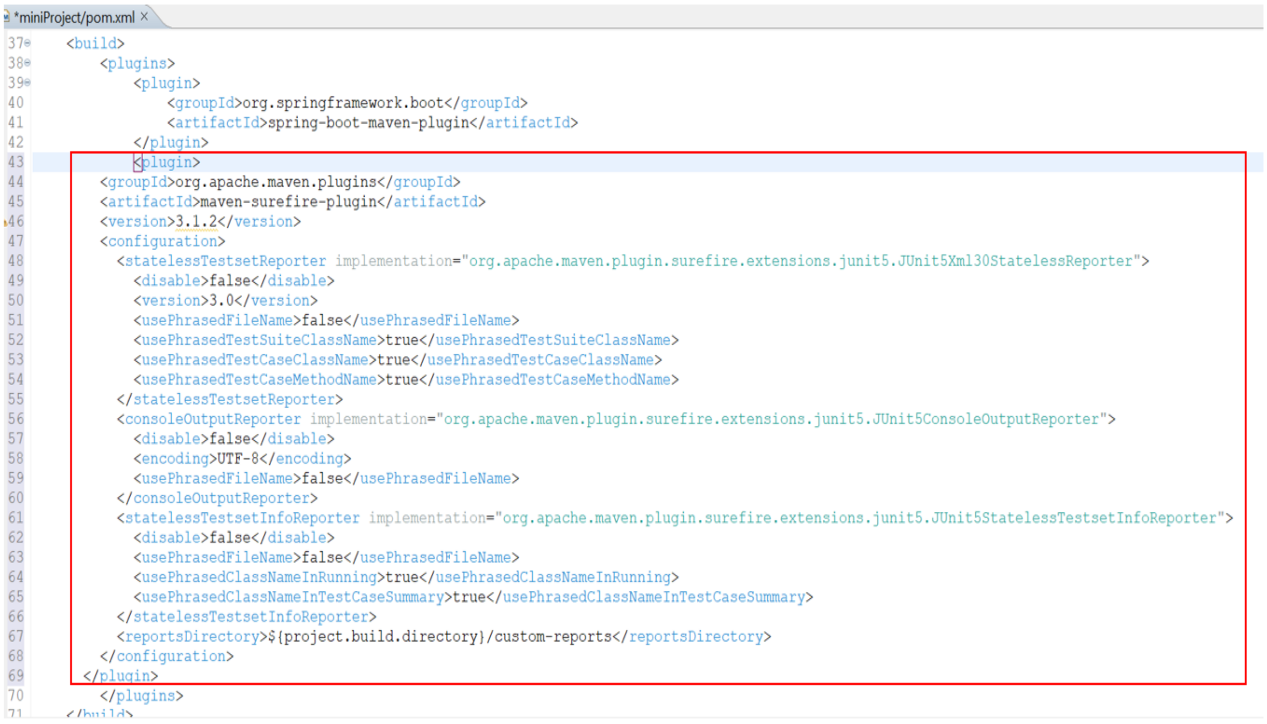


miniProject/pom.xml

```
13      <version>0.0.1-SNAPSHOT</version>
14      <name>miniProject</name>
15      <description>Mini project for Spring Boot</description>
16      <properties>
17          <java.version>17</java.version>
18      </properties>
19      <dependencies>
20          <dependency>
21              <groupId>org.springframework.boot</groupId>
22              <artifactId>spring-boot-starter-web</artifactId>
23          </dependency>
24
25          <dependency>
26              <groupId>com.mysql</groupId>
27              <artifactId>mysql-connector-j</artifactId>
28              <scope>runtime</scope>
29          </dependency>
30          <dependency>
31              <groupId>org.springframework.boot</groupId>
32              <artifactId>spring-boot-starter-test</artifactId>
33              <scope>test</scope>
34          </dependency>
35      </dependencies>
36
37      <build>
38          <plugins>
39              <plugin>
40                  <groupId>org.springframework.boot</groupId>
41                  <artifactId>spring-boot-maven-plugin</artifactId>
42              </plugin>
43          </plugins>
44      </build>
45
46  </project>
47
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

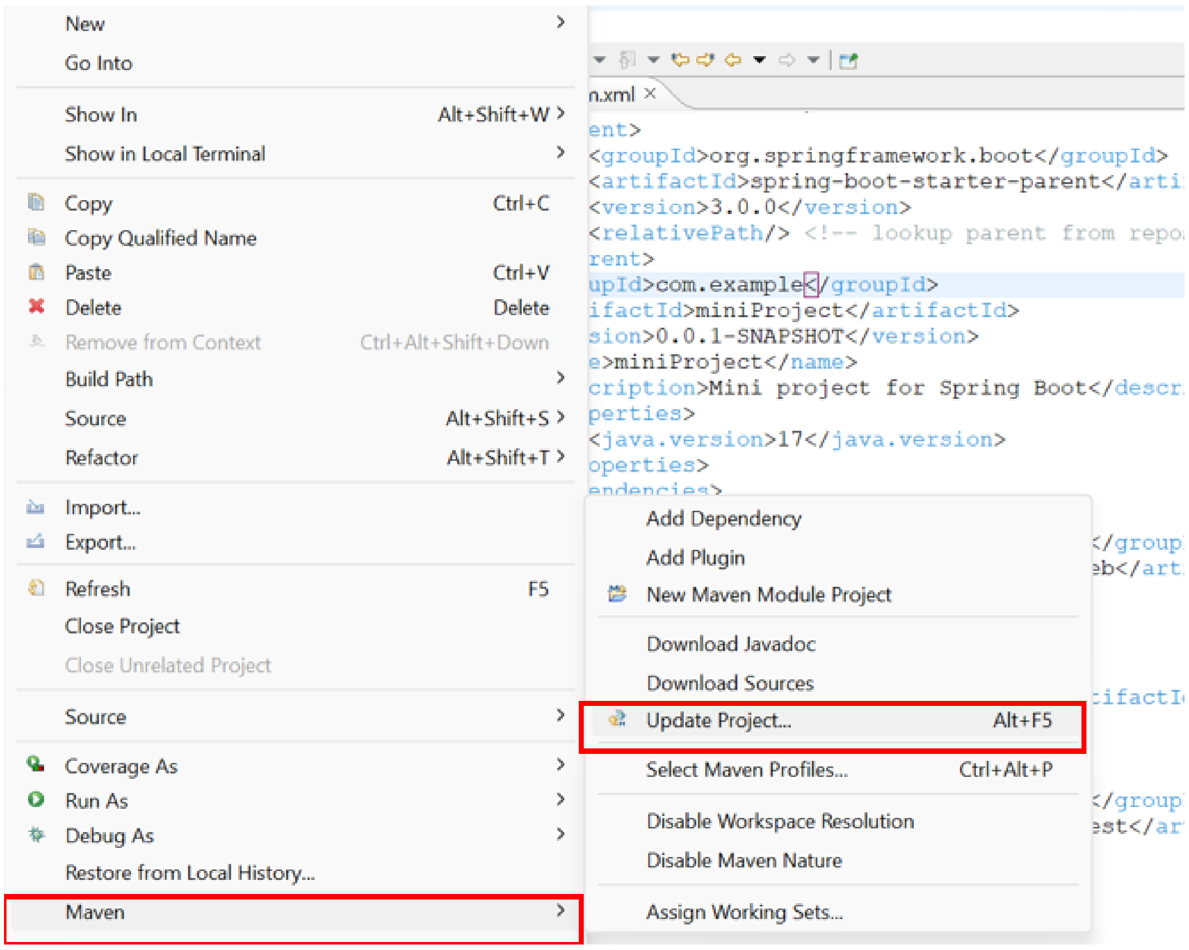
3. At the bottom, you have a section called ***<plugins>***; we will add the above-provided plugin just after the ***<plugin>*** tag is closed, after line **42**, as shown below.



*miniProject/pom.xml

```
37      <build>
38          <plugins>
39              <plugin>
40                  <groupId>org.springframework.boot</groupId>
41                  <artifactId>spring-boot-maven-plugin</artifactId>
42              </plugin>
43              <plugin>
44          <groupId>org.apache.maven.plugins</groupId>
45          <artifactId>maven-surefire-plugin</artifactId>
46          <version>3.1.2</version>
47          <configuration>
48            <statelessTestsetReporter implementation="org.apache.maven.plugin.surefire.extensions.junit5.JUnit5Xml30StatelessReporter">
49              <disable>false</disable>
50              <version>3.0</version>
51              <usePhrasedFileName>false</usePhrasedFileName>
52              <usePhrasedTestSuiteClassName>true</usePhrasedTestSuiteClassName>
53              <usePhrasedTestCaseClassName>true</usePhrasedTestCaseClassName>
54              <usePhrasedTestCaseMethodName>true</usePhrasedTestCaseMethodName>
55            </statelessTestsetReporter>
56            <consoleOutputReporter implementation="org.apache.maven.plugin.surefire.extensions.junit5.JUnit5ConsoleOutputReporter">
57              <disable>false</disable>
58              <encoding>UTF-8</encoding>
59              <usePhrasedFileName>false</usePhrasedFileName>
60            </consoleOutputReporter>
61            <statelessTestsetInfoReporter implementation="org.apache.maven.plugin.surefire.extensions.junit5.JUnit5StatelessTestsetInfoReporter">
62              <disable>false</disable>
63              <usePhrasedFileName>false</usePhrasedFileName>
64              <usePhrasedClassNameInRunning>true</usePhrasedClassNameInRunning>
65              <usePhrasedClassNameInTestCaseSummary>true</usePhrasedClassNameInTestCaseSummary>
66            </statelessTestsetInfoReporter>
67            <reportsDirectory>${project.build.directory}/custom-reports</reportsDirectory>
68          </configuration>
69        </plugin>
70          </plugins>
71      </build>
```

4. Lastly, you have to update the maven project as shown below for installing the newly added plugin.



```
ent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</arti
<version>3.0.0</version>
<relativePath/> <!-- lookup parent from repo
rent>
upId>com.example</groupId>
ifactId>miniProject</artifactId>
sion>0.0.1-SNAPSHOT</version>
e>miniProject</name>
cription>Mini project for Spring Boot</descr
perties>
<java.version>17</java.version>
operties>
endencies>
</group
eb</art
cifactI
</group
est</ar
```