1. The Code is uploaded at **https://github.com/vikram94/DemoProject.git**   
2. Checkout or download the code from the repository. Structure has 2 POJO classes - User and Expression, One Controller class, One Repository interface, User Service class - for user logic, One EvaluationService that calculates the value of expression.

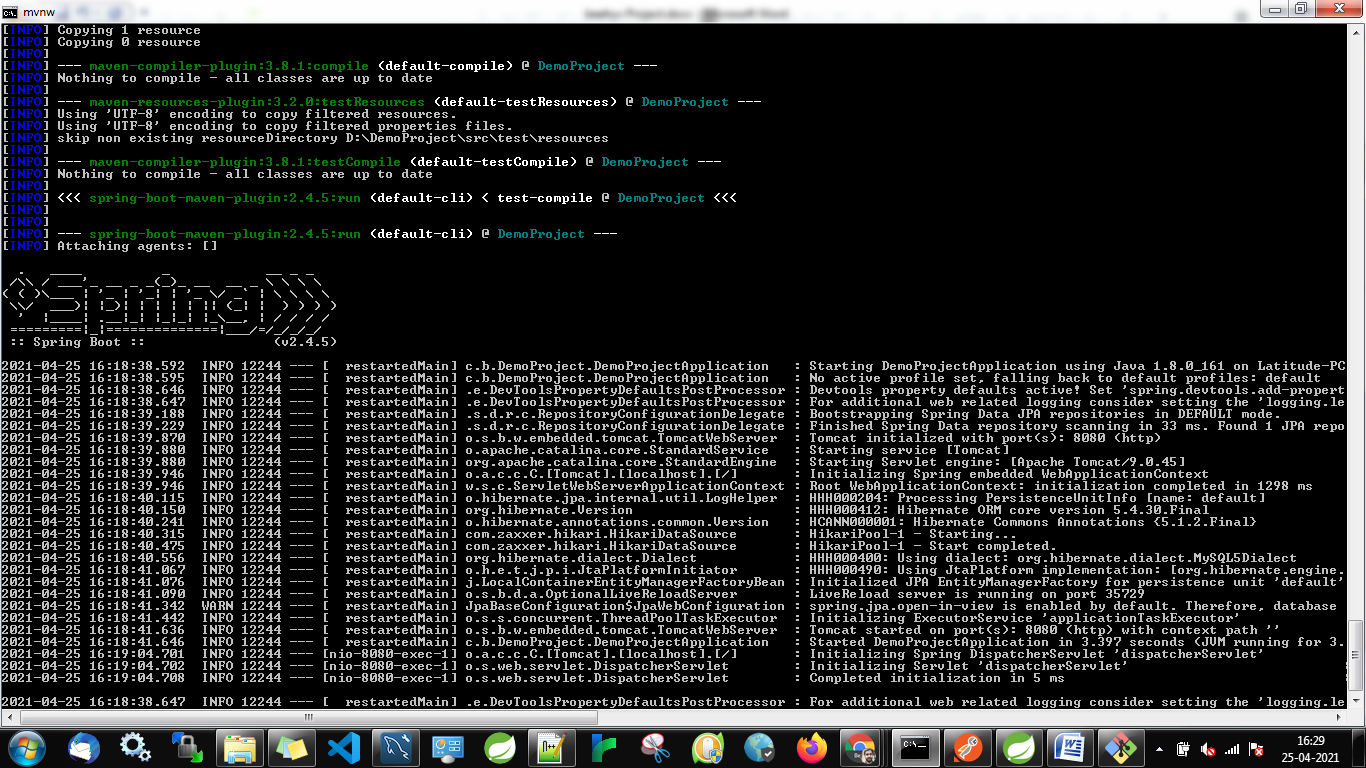
3. Now if your code is in the local then update spring.datasource.password in src/main/resources application.properties. For schema initialization.

4. Requirements - MySQL version 5.6 or better, JDK 1.8, Maven 3.2+, STS or Eclipse, POSTMAN

5. In the projects directory open command line and type and press enter - **mvnw spring-boot:run**

6. You should see your code building in the command line. If you find error in red related to .m2\repository. You can go to that location and delete the folder in .m2/repository giving error while build. and again run **mvnw spring-boot:run** . This will download latest dependencies. Continue this step again if there are errors.

8. If Successful then check MYSQL if you have a user table with 5 columns.  
7. Successful error free tomcat up and running looks like the image below.



8. If everything works fine you can open Postman. There are 2 apis.

9. **http://localhost:8080/user/evaluate/{id}.** This is a **POST** request with id = any integer(upto - 10digit) and request body -

{

"expression": "3\*(2-5)"

}

Requesting this api will give the result and store the count of operators that are used in the expression. Will update if it is an existing user. else will insert a new row with count of each operator. If the expression is invalid then will output an error message.

10. **http://localhost:8080/user/mostused/{id}**  is a **GET** request. Will output the max operator for id(user). And if the user does not exist will throw an error. If there are more than one maximum operators then it will show all of them space separated.  
11. For Enhancement, Since we have used only one table there can be performance issue if there are millions of requests. To avoid that we can use sharding user table. We can have 10 table and on the basis of userids initial digit we can store the row in the corresponding table. eg - 1234 in table user\_1, 5678 in table user\_5