PIXOGRAM Assessment

process to create ASSESSMENT (FSA) for pixogram

Contents

[1 Important Instructions 2](#_Toc34155680)

[2 Business-Requirement: 2](#_Toc34155681)

[3 User-Service layer 3](#_Toc34155682)

[3.1 UserService Module 3](#_Toc34155683)

[4 Data Layer 5](#_Toc34155684)

[5 Testing module 5](#_Toc34155685)

[6 Project structure 7](#_Toc34155686)

[7 Instructions to execute the program 8](#_Toc34155687)

# Important Instructions

1. As of now the product supports creating full stack assignments that doesn’t involve UI. Idea is to come up with a document that translates in to interfaces and telling the end user to code the implementation to these interfaces.
2. Please make sure that your code does not have any compilation errors while submitting your case study solution.
3. The final solution **should be deployable in Apache Tomcat**
4. **Deploying the solution in Apache Tomcat should run the Spring MVC project** at <http://localhost:portnumber> or <http://ipaddress:portnumber>.
5. **Use MySQL database where database functionality is required.**
6. Implement the code using best design standards for:
   1. Variable declarations
   2. Class names
   3. Package names
   4. Code Refactoring

# Business-Requirement:

**Pixogram**: **Single Page Picture Sharing Application**

1. Register as a user
2. Login as a user
3. Retrieve password
4. Manage your user account
5. Login/Logout to/from your account on PixoGram
6. Add Content
   1. Upload single/multiple pictures, caption and description
   2. Upload single/multiple videos, caption and description
7. Manage Content
   1. Organize Picture in Gallery
   2. Organize Videos in Playlists
   3. Rename Pictures and Videos
   4. Edit Caption, Description, Comment
8. Social Features
   1. Use emoji’s in comment
   2. Like or Unlike comment, pictures and videos of other users
   3. Follow/Unfollow other users
9. Edit Pictures
   1. Apply effects to pictures (sepia, grayscale, etc.)
10. Hide Pictures/Videos
11. Activity/Newsfeed
    1. View activity log of user-activity on the PixoGram
12. Offline Functionality:
    1. Certain parts of the application should be available in absence of connectivity.
    2. Relevant areas on the screen should display “Connectivity Not Available”
13. BONUS REWARDS/SCORE Feature:
    1. To implement offline image upload functionality such that user can upload content when offline. It will sync with backend when connected.

# 3 User-Service layer

a) As of now the product supports creating full stack assignments that doesn’t involve UI. Idea is to come up with a document that translates in to interfaces and telling the end user to code the implementation to these interfaces.

b) Entire code structure will be available. User need to write only the implementations and Service logics.

c) The template code base should comprise of following-

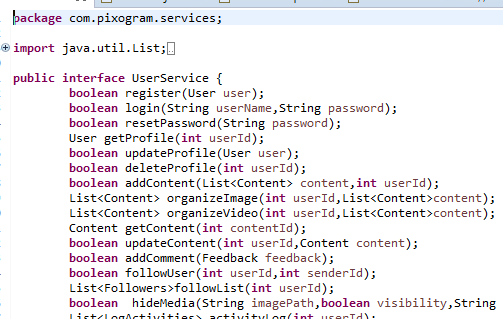
1) Interfaces

2) Implementation classes with only declared methods (will be updated by end user)

## 3.1 UserService Module

1. **Services**:

**In this project there is only a single service interface i**.e. UserService interface. The below screenshot has only the declared methods in the interface, the user implementing the below interface has to write all the implementation to execute the program.

**UserService**

We would like to highlight all the methods used in the service method.

***Register***

***login***

***resetPassword***

***getProfile***

***updateProfile***

***deleteProfile***

***addContent***

***organizeImage***

***organizeVideo***

***getContent***

***updateContent***

***addComment***

***followUser***

***followList***

***hideMedia***

***activityLog***

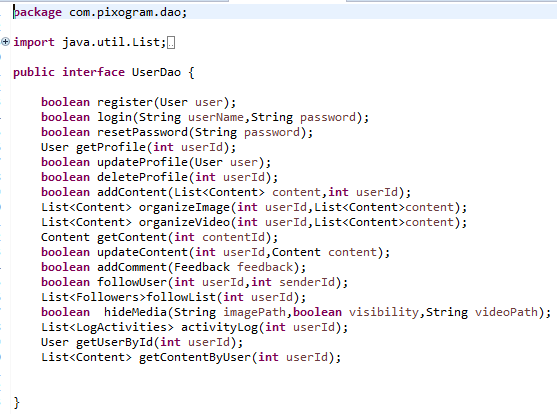
***getUserById***

***getContentByUser***

# Data Layer

**Dao Interface:**

We have defined a structure in order to access the database, UserDao is an interface that has all the methods associated to access the database. All the transactions can be implemented in the Dao layer. User needs to go through the structure before implementing the code.

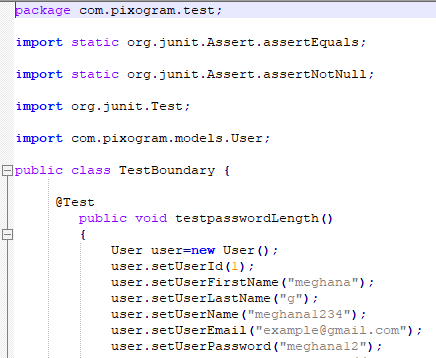


# Testing module

Since the content writer also shares the input files, if any, as needed for the assignment, he need to share a file called Problem.txt with operations team that lists the mapping between test case and supposed expected output. E.g. below

* + 1. Testcase1=5
    2. Testcase1=1
    3. Testcase3=5

Above means there are test cases in the assignment and the values are the expected output. Need to execute the test cases. A sample structure is shown below.



### 

**The candidates are expected to carry out following core java operations by referring to the problem statement. User can use the above structure to complete the exercise.** Make sure the following points should be considered while exercising your program

**Core Java 1.7, Web services, Restful API**

1. Make sure to follow the Naming conventions in java
2. Need to add the required instance variables
3. Make sure to use the for loop statement
4. Use the conditional operations
5. Need to create an object and set their respective properties and behaviour
6. Check if you can go for overloading the methods
7. Use the inheritance concept in the program
8. Check where can we use the private keyword in the program
9. Perform threading activity.(Optional)
10. Make sure to add a collection list to your program
11. Lambda Expressions in Java
12. Check if Comparison operators can be used.
13. Make sure to add a collection list to your program
14. Lambda Expressions in Java
15. Check if Comparison operators can be used.
16. Querying the Database
17. Inserting rows of data into the database
18. Using Encapsulation and data hiding
19. Need to perform CRUD operations
20. Session object need to be used in the program
21. Need to handle exceptions.
22. Need to use collection classes to handle objects.

# Project structure

* + 1. Business Layer - Which contains all Interfaces and Implementation classes.

Interfaces – UserService

Services –UserServiceImpl

* + 1. Data Layer -Which contains Hibernate connection to Database.

Mapping Classes

Hibernate Connection classes

* + 1. Entity Layer - Which contains all needed entity classes.

Pojo’s classes- Content, Feedback, Follows, LogActivities, User

* + 1. Web Layer- Controllers, Views and Main method.

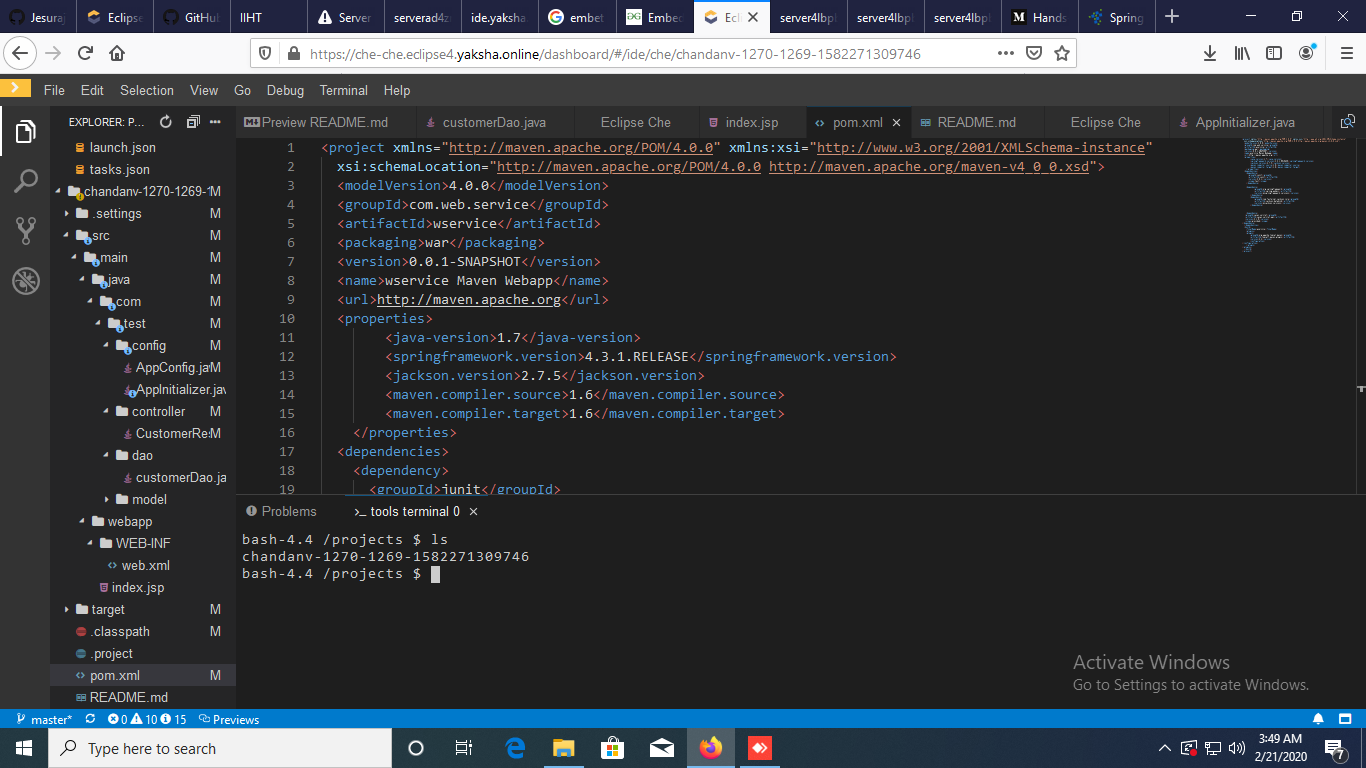
# 7 Instructions to execute the program

Below are the instructions to run or execute the program after completing the logic:

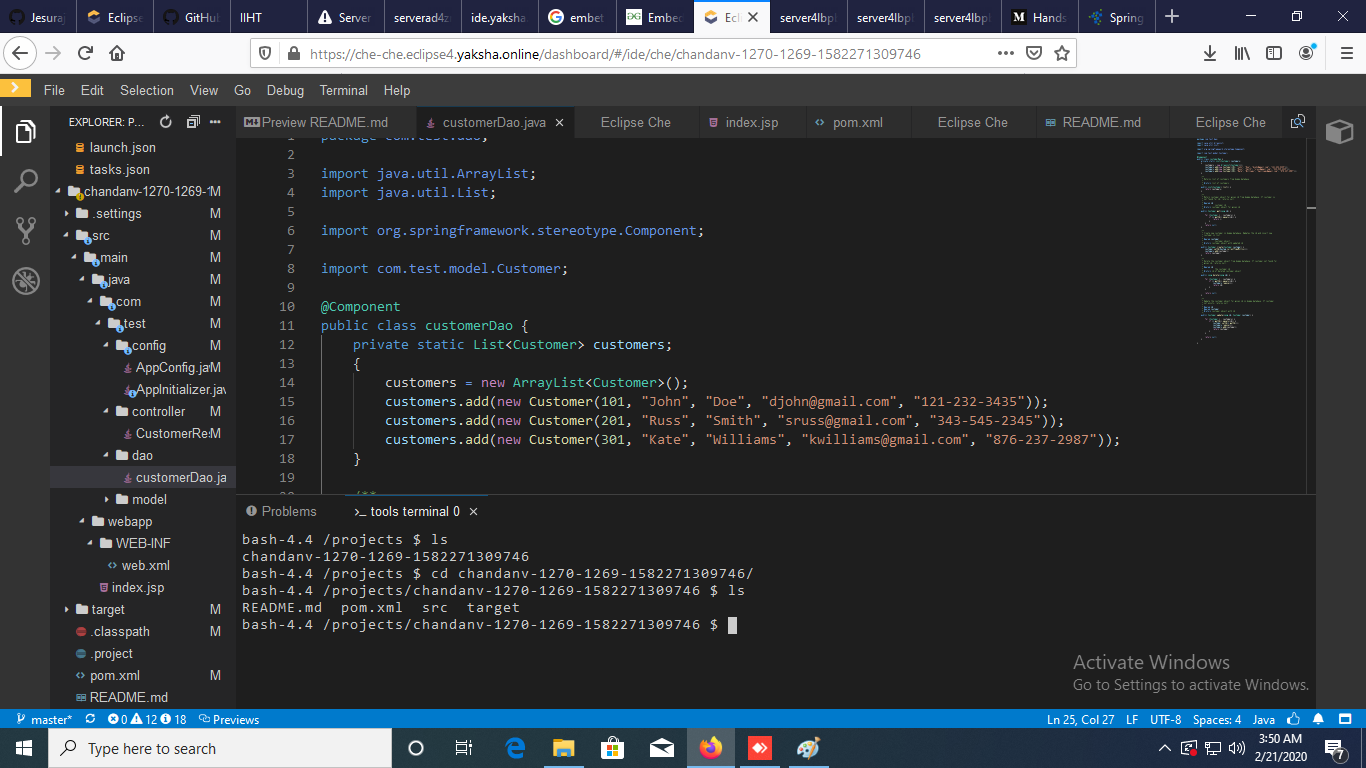
**For Java program without server environment:**

1. Open eclipse che terminal

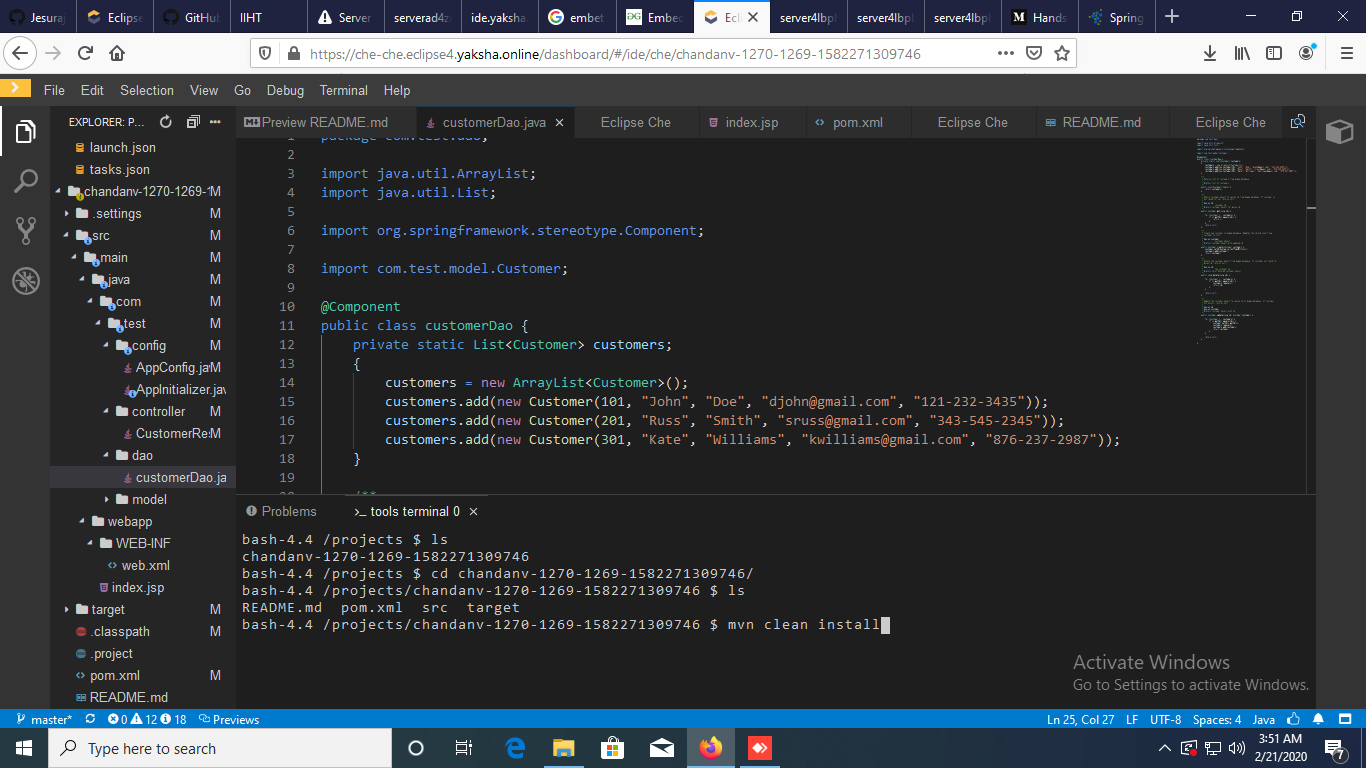
2. "ls" to display the folder

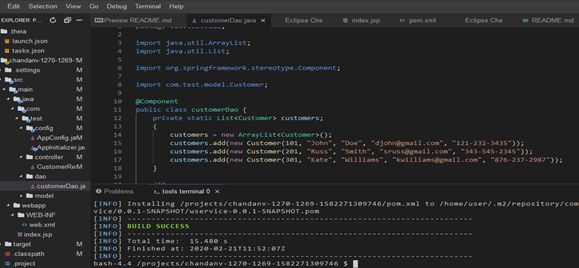


1. Change to working directory "cd your-directoryname"

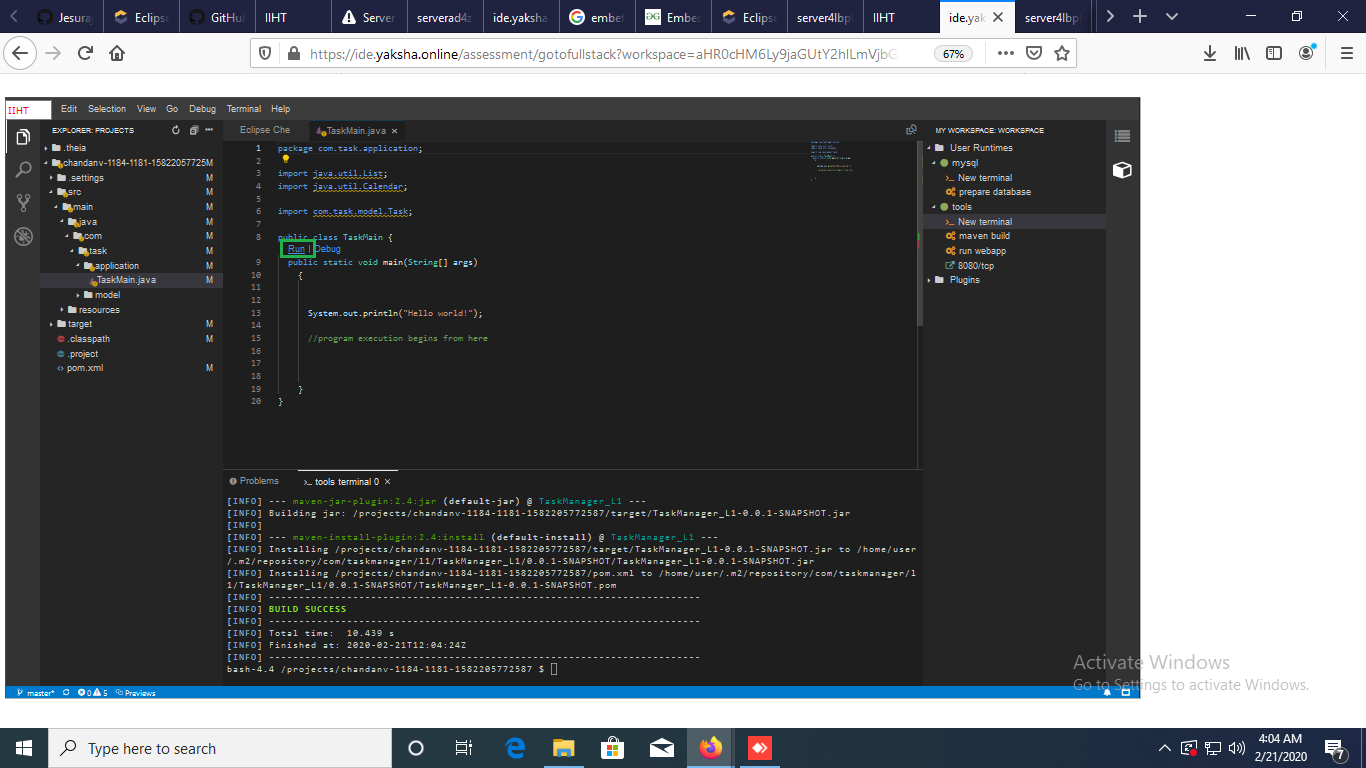


1. Type command: "mvn clean install"





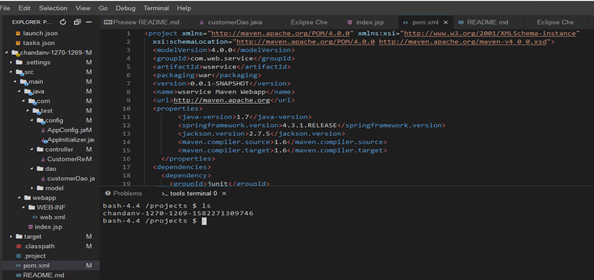
1. Select your Java main file and open it from respective folder "by default: src/main/java/" and. Run option will appear on it "Run | Debug"



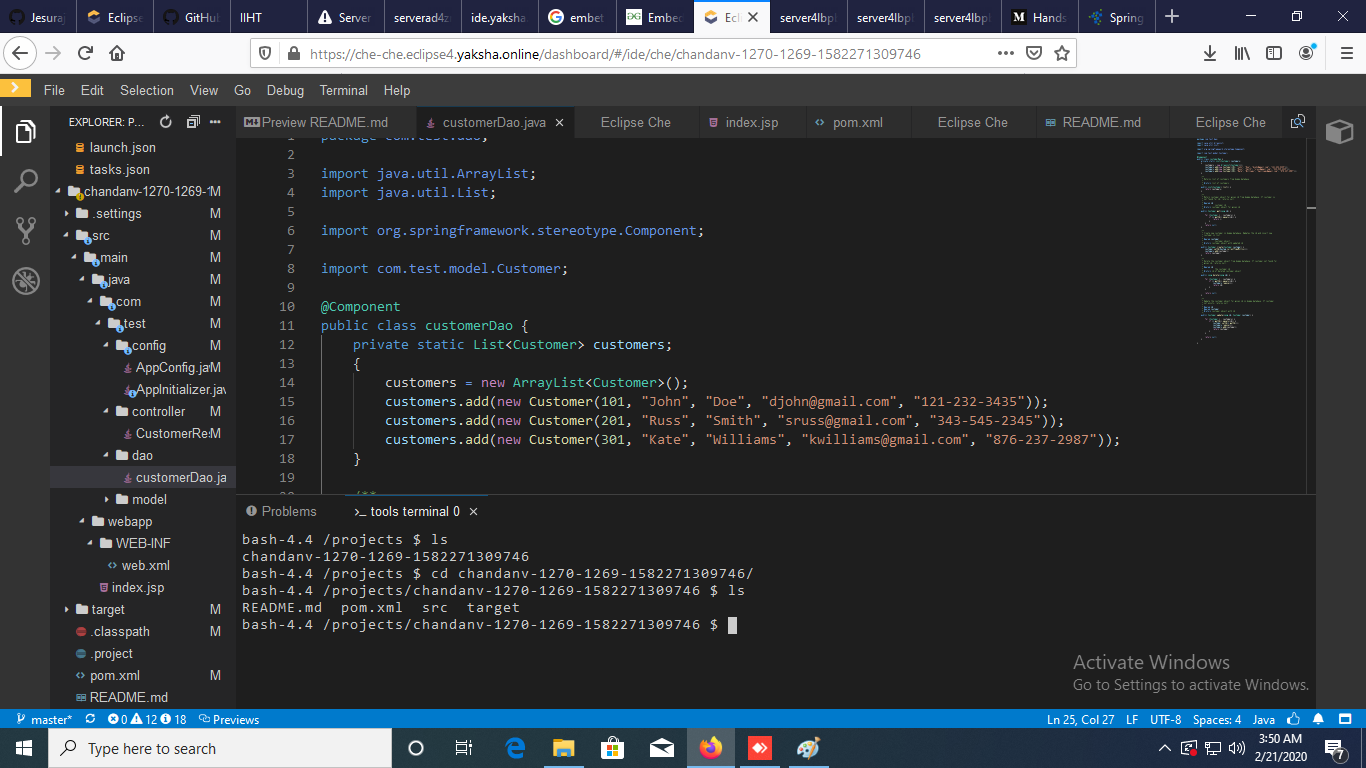
**For Java program with server environment:**

1. Open eclipse che terminal

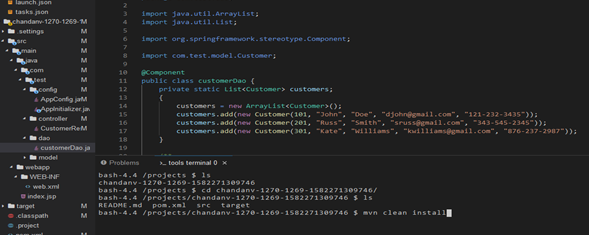
2. "ls" to display the folder

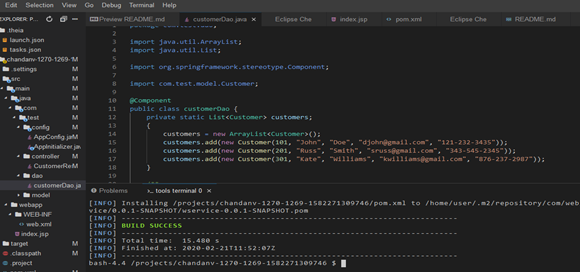


3. Change to working directory "cd your-directoryname"

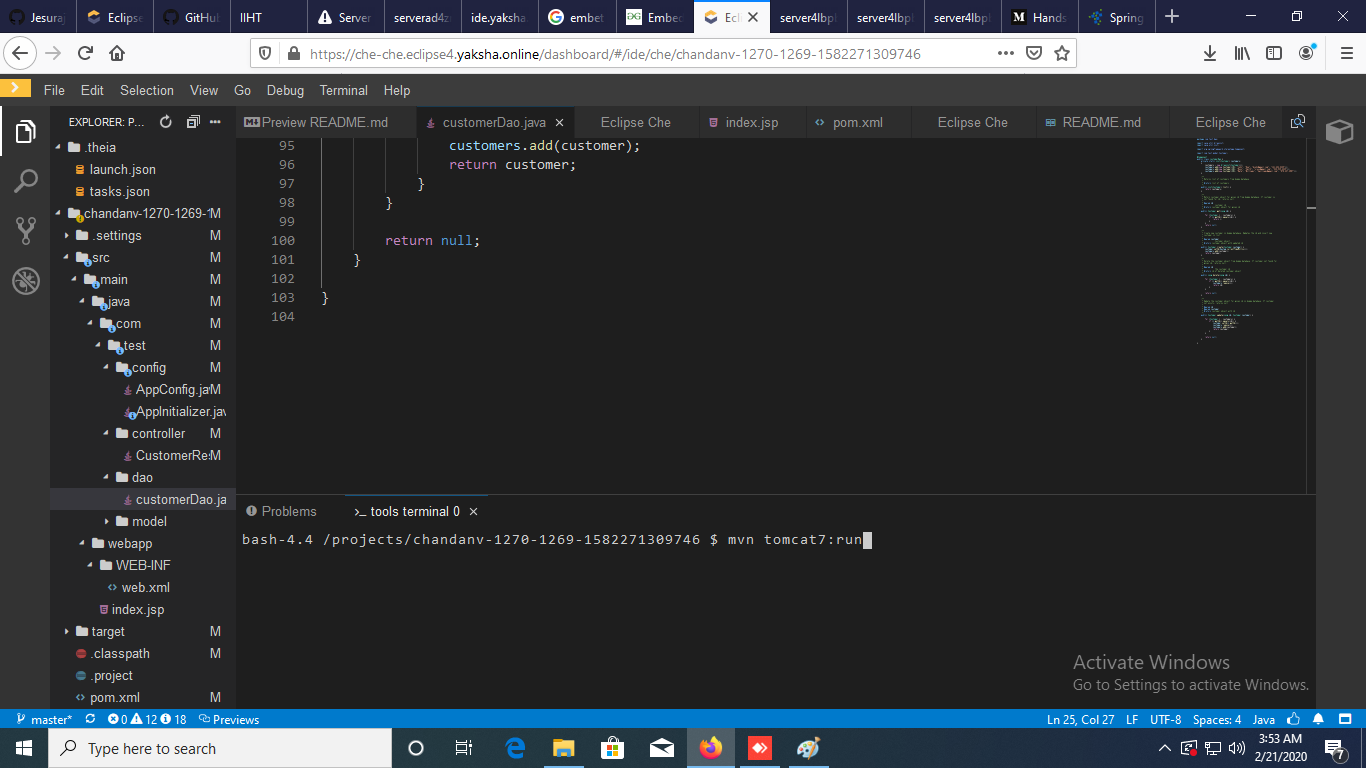


4. Type command: "mvn clean install"





5. After successful build type the command "mvn tomcat7:run"



1. Upon successful build, an option will be provided to open the link in new table.

