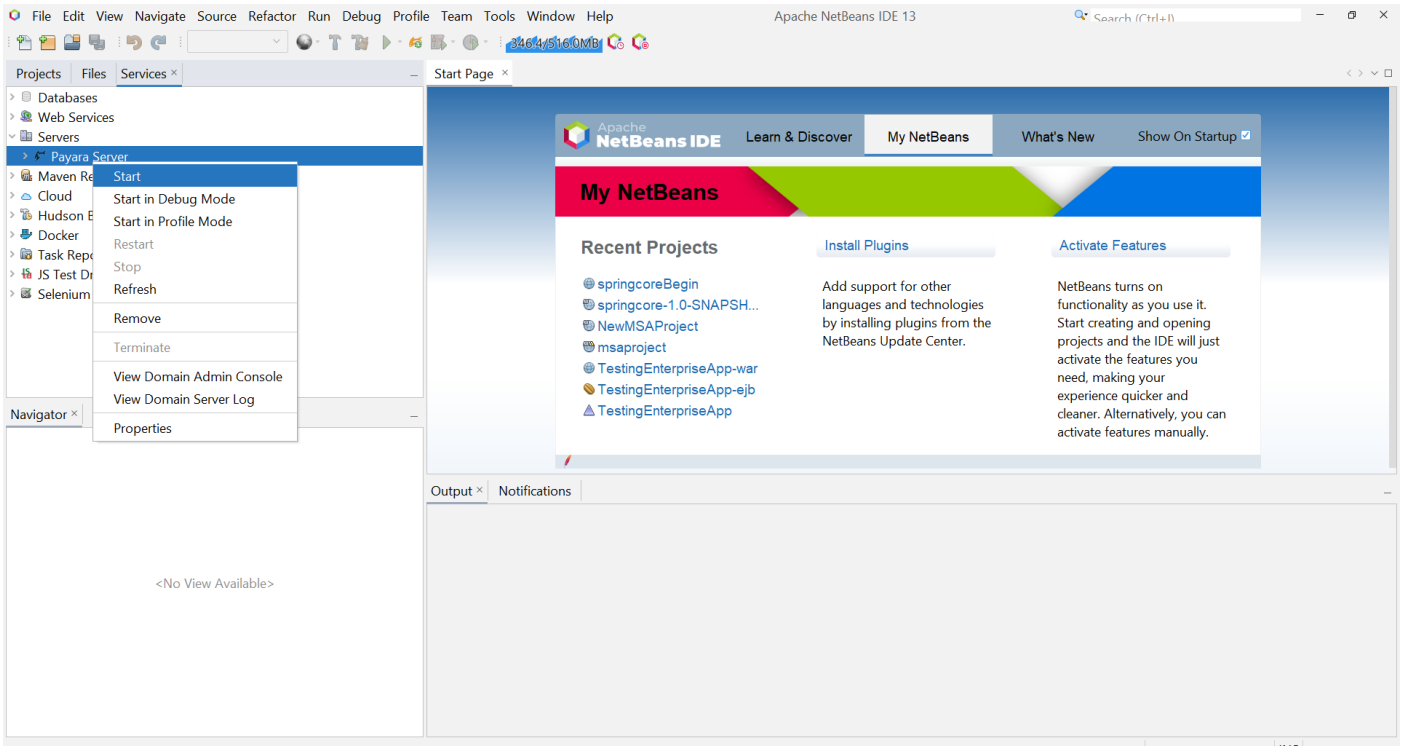
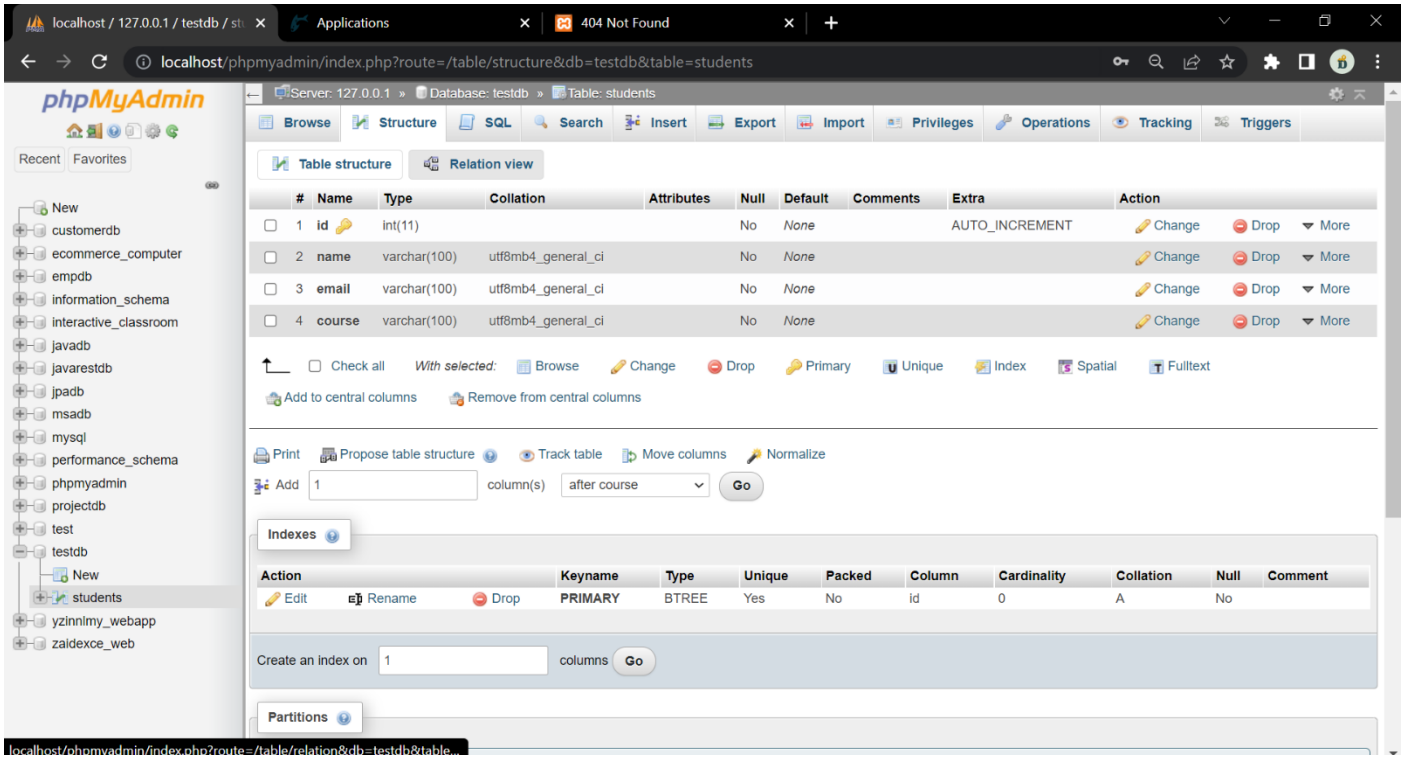
Today I'll show you How to run 2 Microservice application in Payara microprofile. Before moving on I want to you to have following files in your machine.

1) mysql-connector-java-5.1.23-bin or mysql-connector-java-5.1.30.jar
2) payara-micro-5.2022.2.jar
3) domain.xml
4) jwtenizr.jar

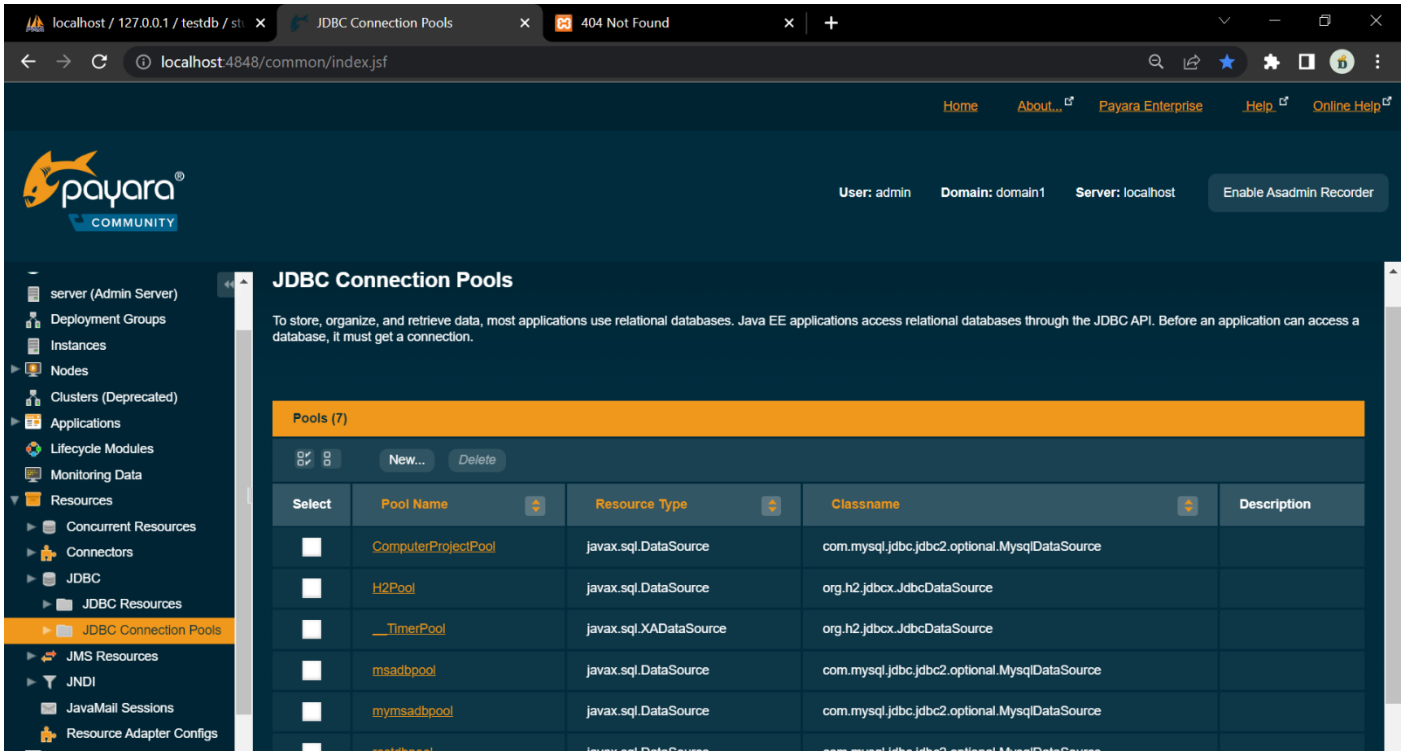
1) Open Netbeans 13 or later and start payara server



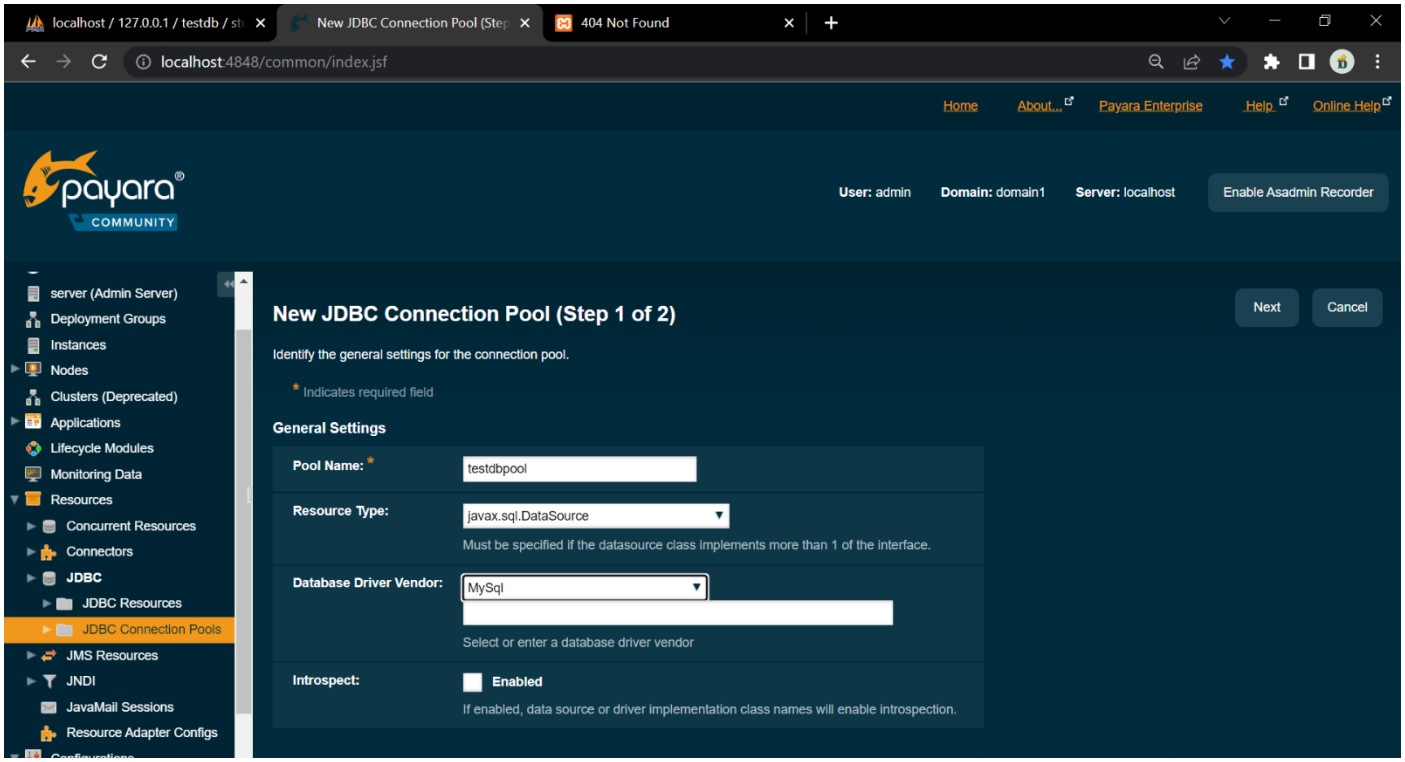
2) Turn on your xampp/wamp server and create one database with tables in http://localhost/phpmyadmin/index.php



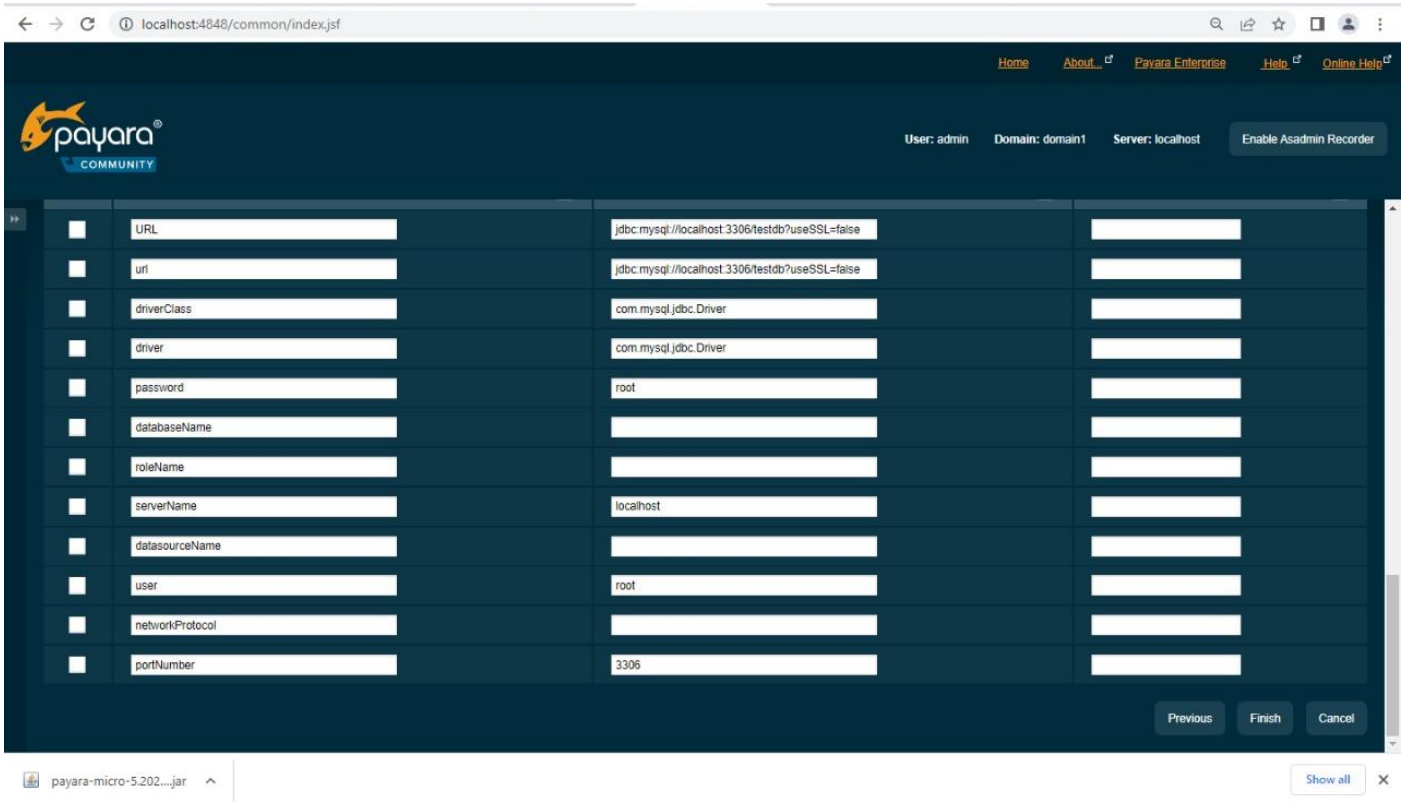
3) I've created database with named testdb and created one table named students. Now add Url http://localhost:4848/common/index.jsf in your browser and Go to jdbc Connection Pool



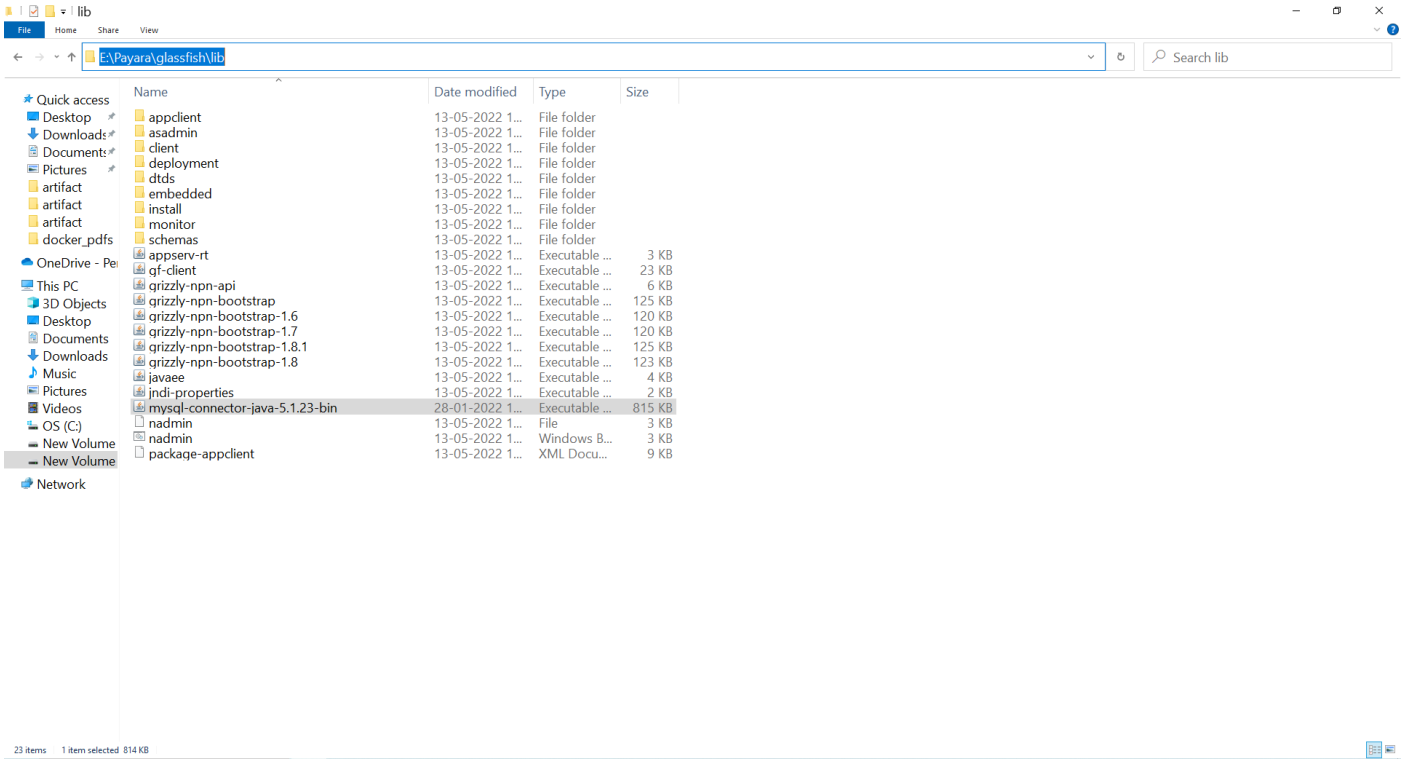
4) Create new connection pool, Here in my case I've given the pool name testdbpool and select the following options mentioned in ss and then click next.



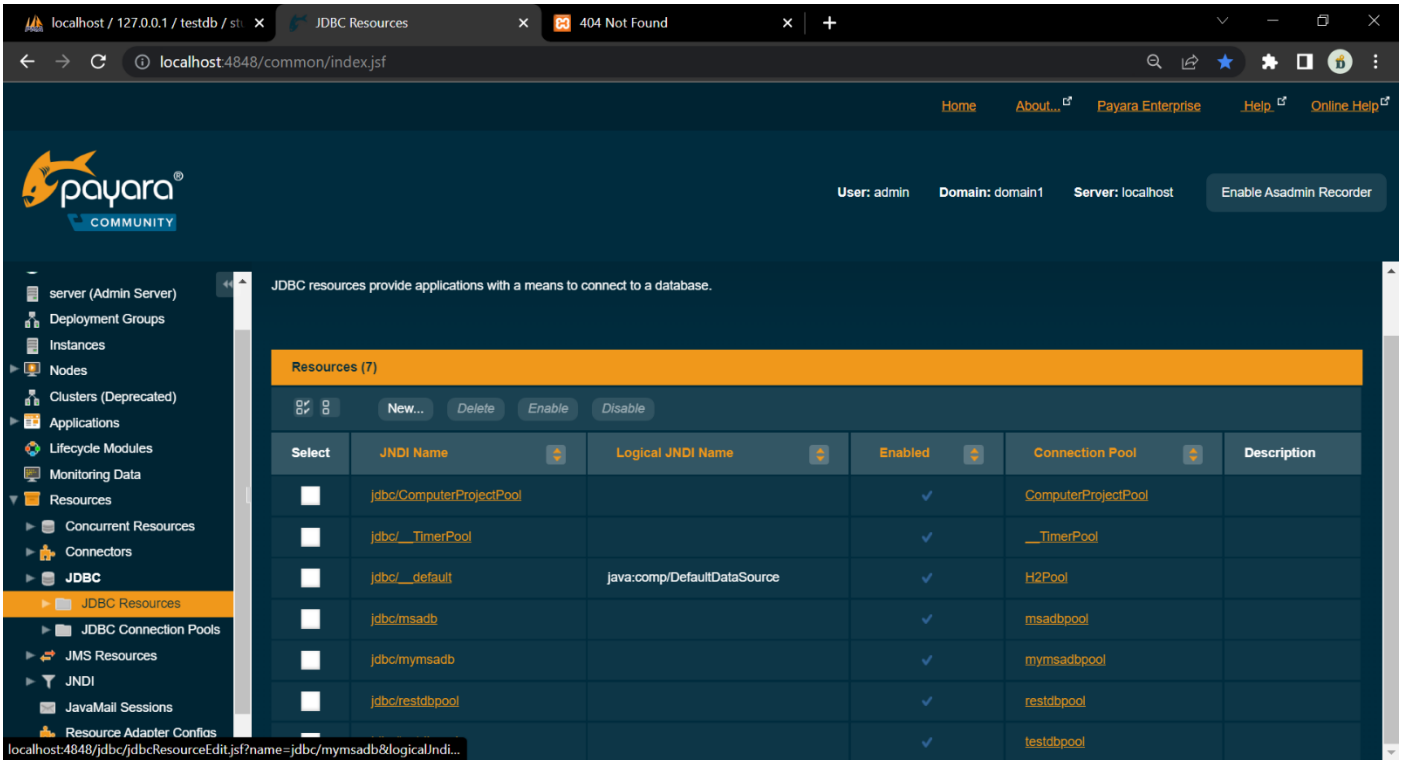
5) Scroll down little bit and add the following properties, Your machine might contained more properties than properties mentioned in following ss. Search or Add the properties and Leave extra properties as it is and click next



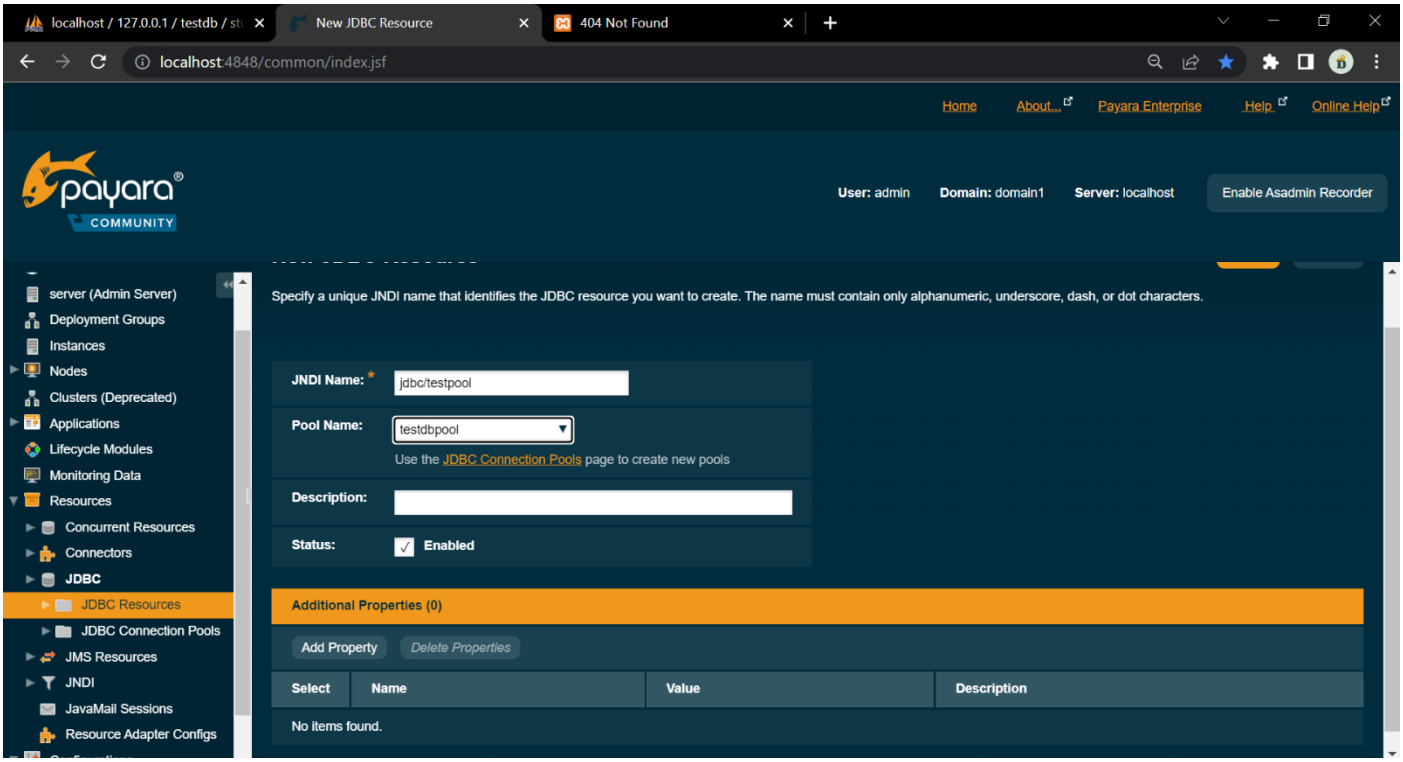
6) Before moving on I want you check the weather mysql-connector-java-5.1.23-bin or mysql-connector-java-5.1.30.jar is existed in your payara server location which is mentioned in below ss. If not then add over there



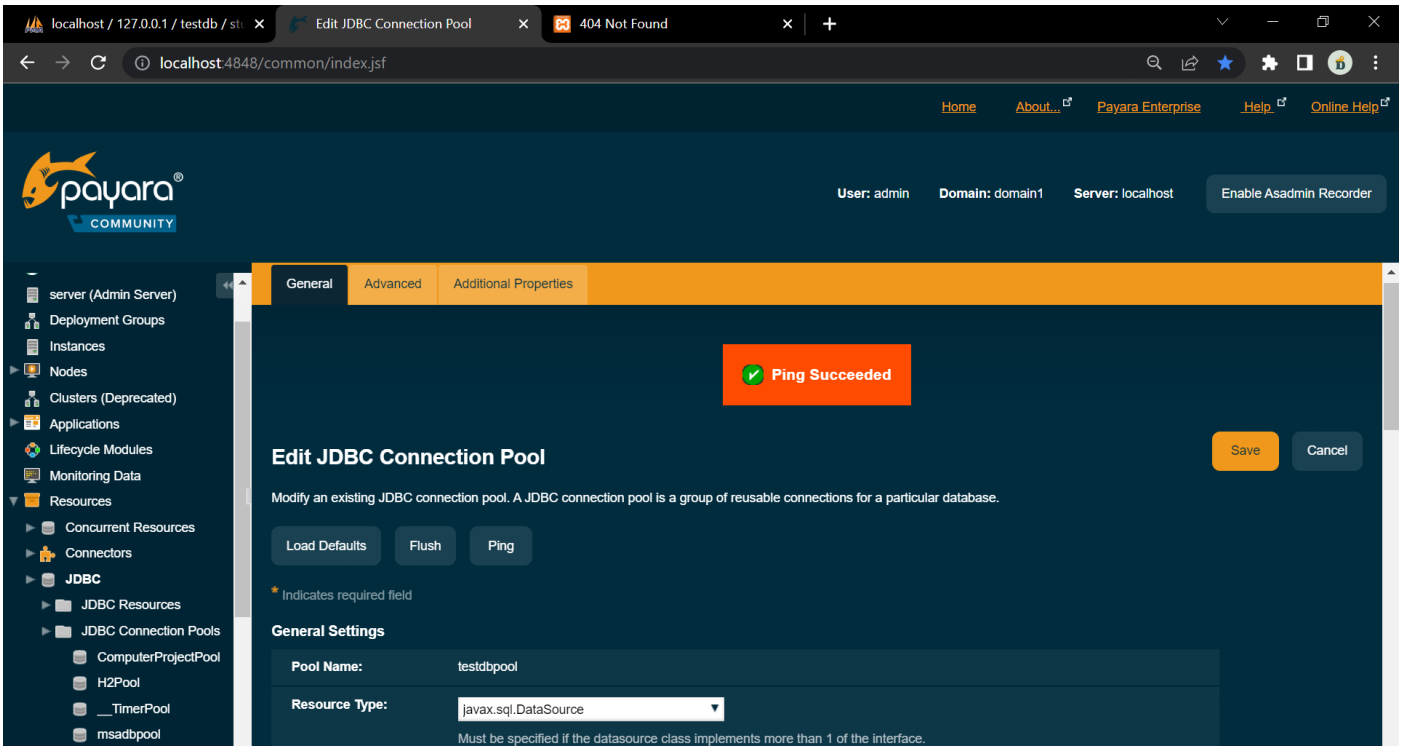
7) Back to the admin console of payara in your browser and click on JDBC Resources click new add new one



8) Click on New button and new jdbc resource and then click on ok button, See the below ss.

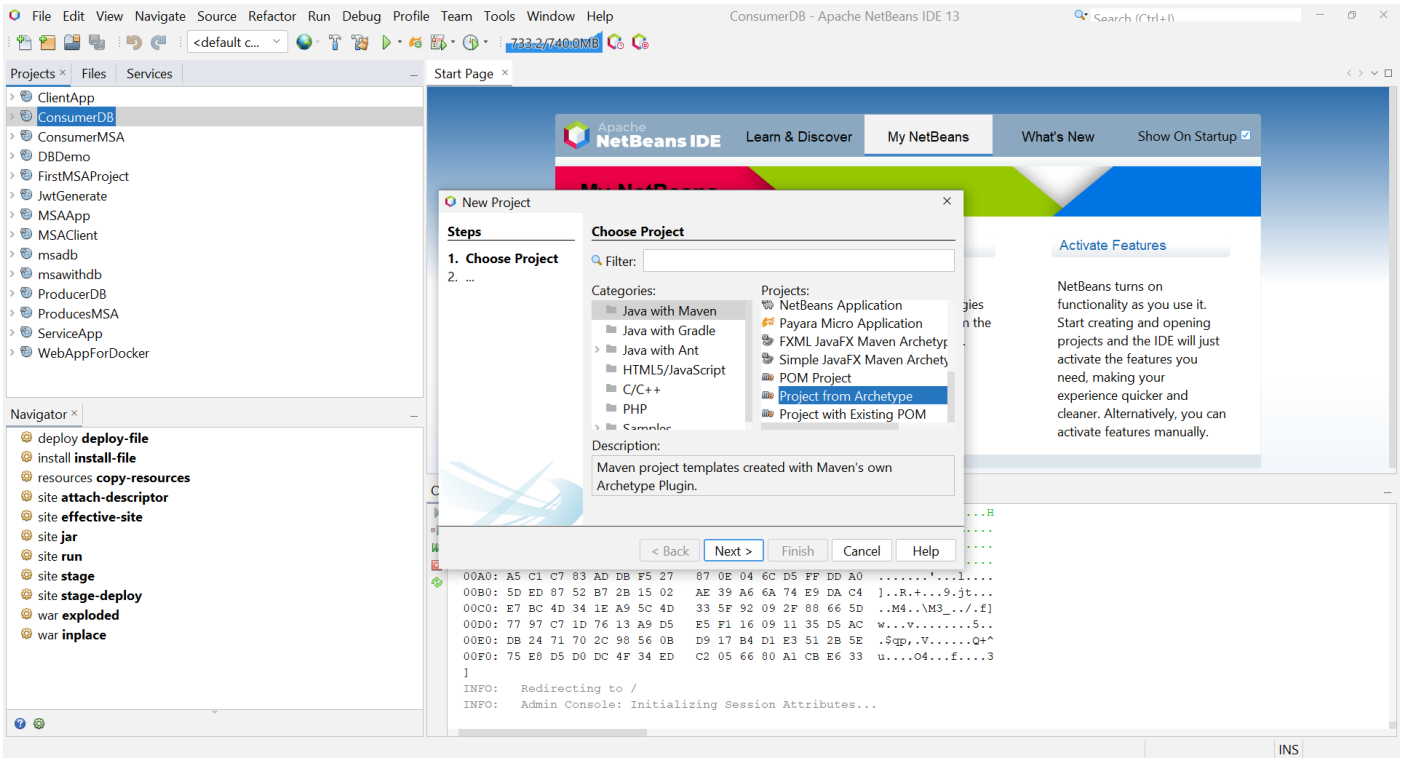


9) Again to Go to JDBC Connection pools and select the connection pool that you've created recently and click on ping button , You must get ping successed message.



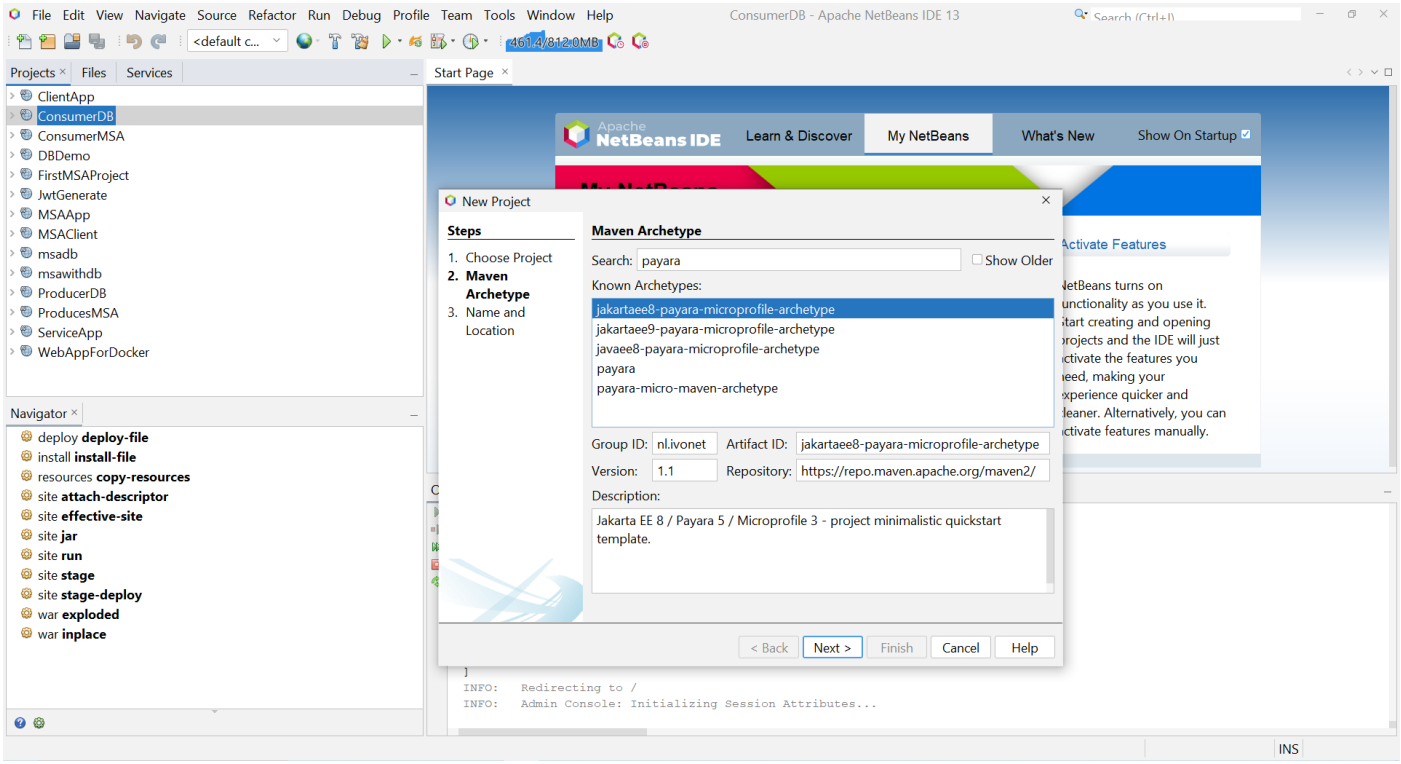
Alright Your database part is done, Now move on Application creation part

10) Create new maven project with type Project from Archetype and click next

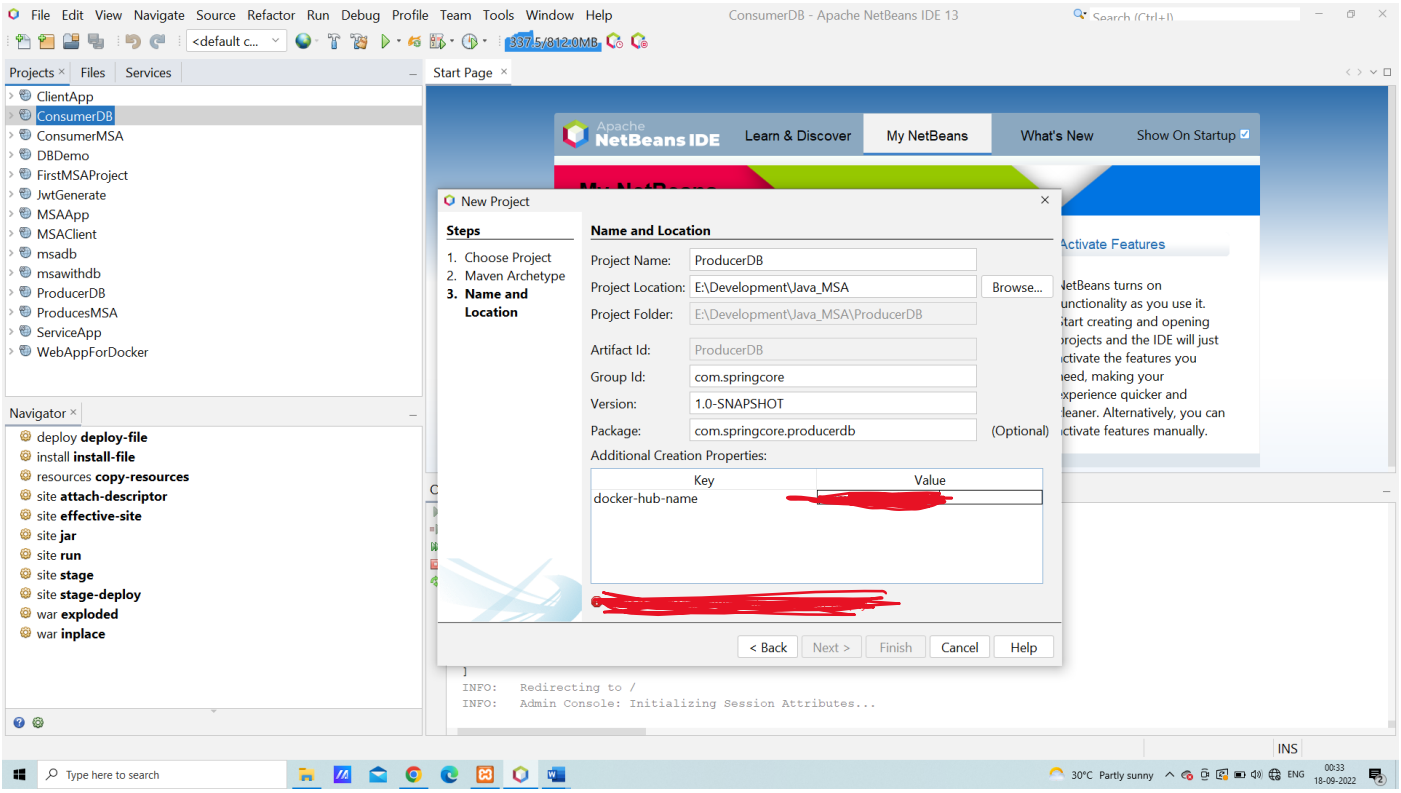


11) Search jakartaee8-payara-microprofile-archetype and click next

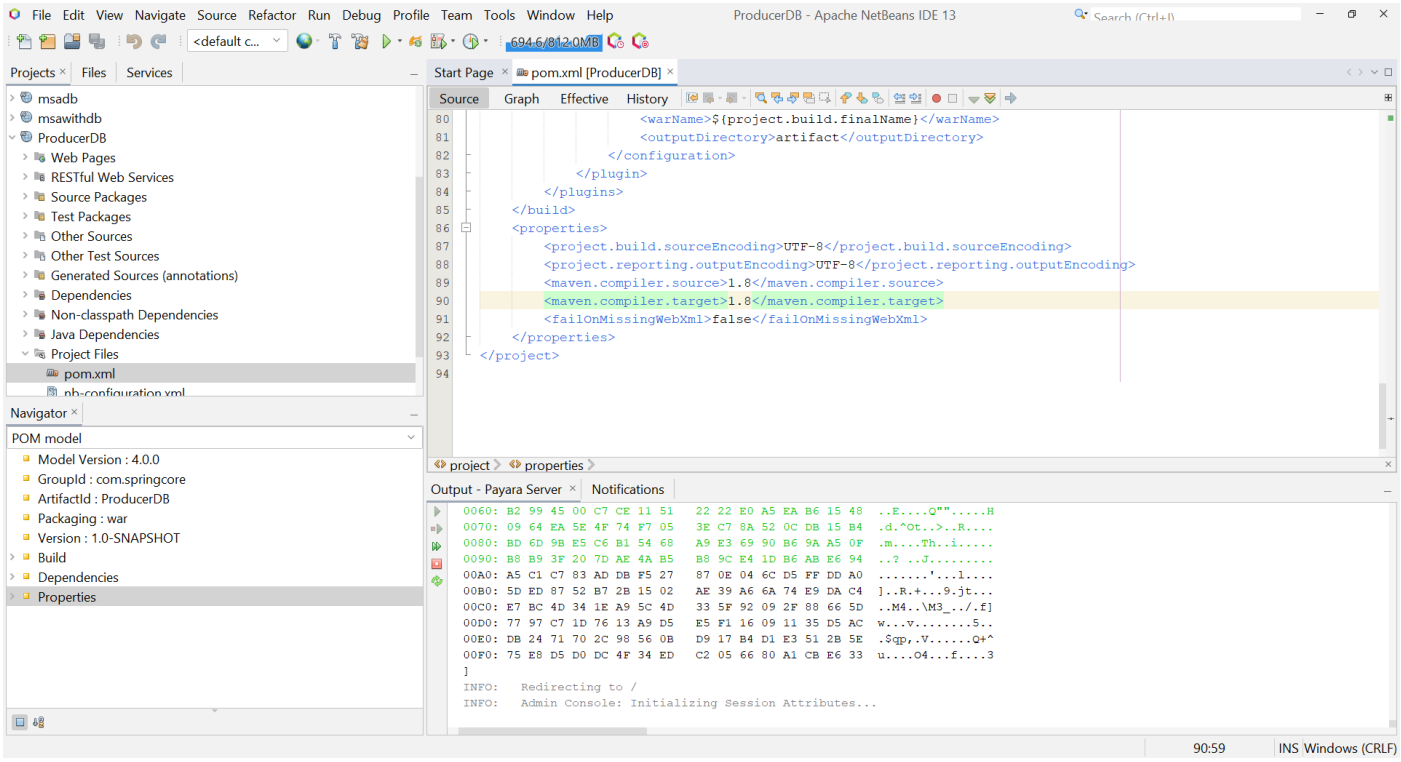
(Note : Currently we are creating Producer App)



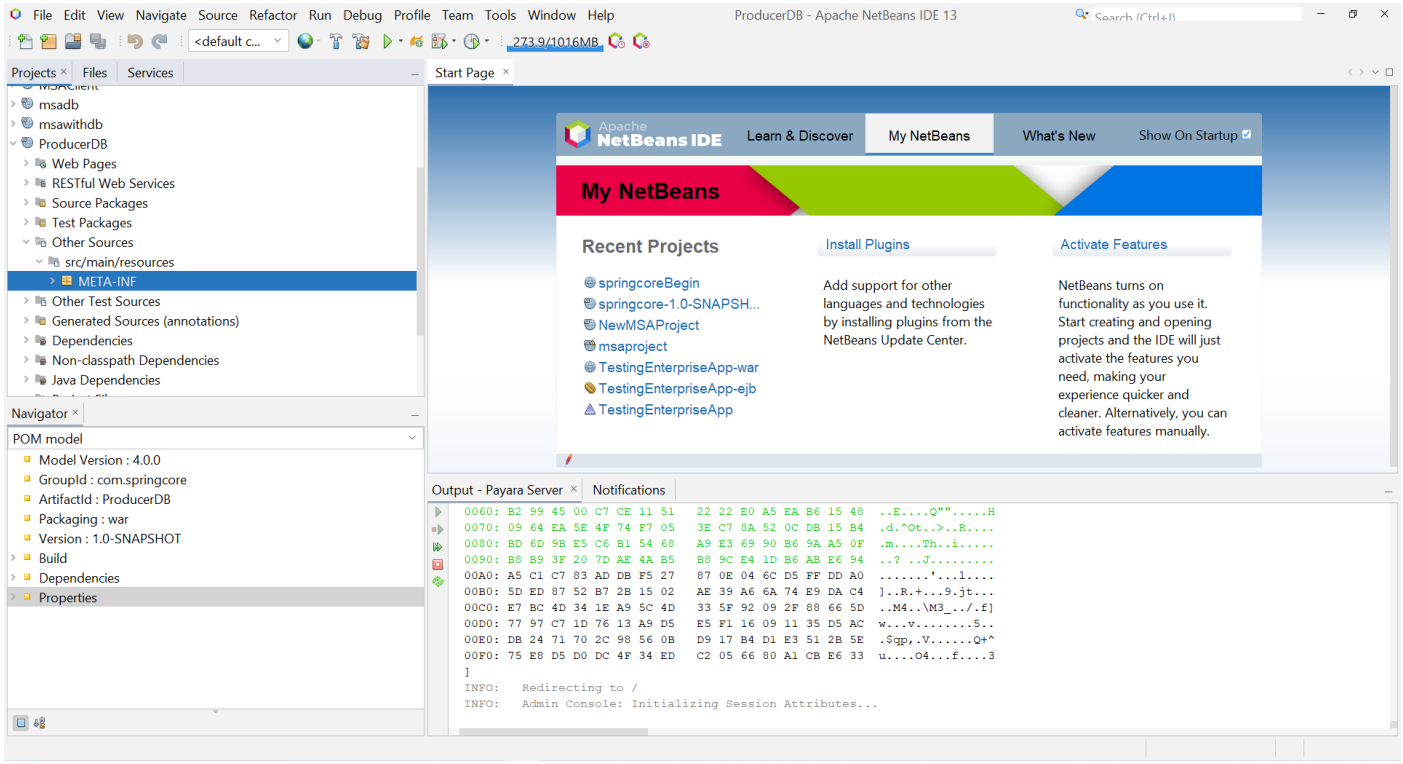
12) Enter the project Name and enter the docker-hub-name (optional) , Here in my case I've given ProducerDB and click next



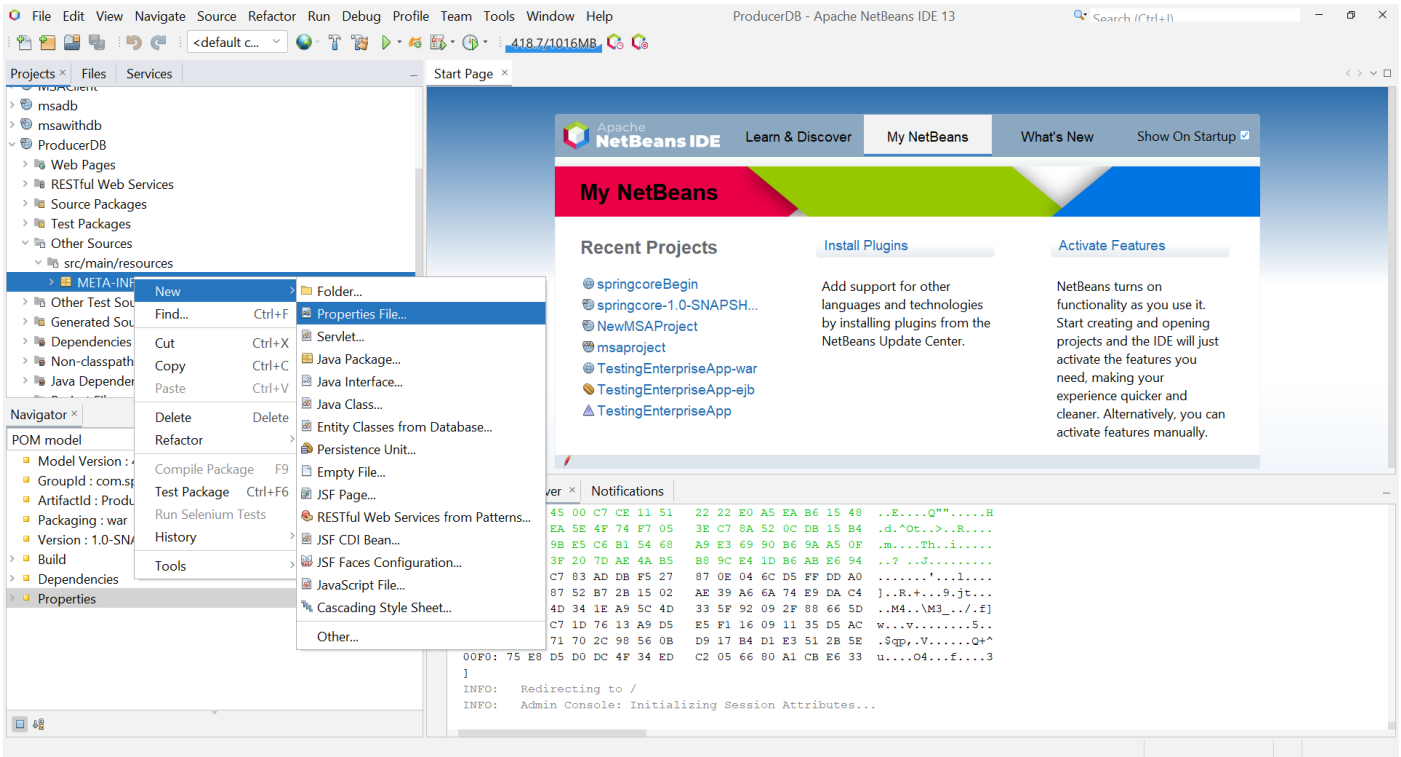
13) Go to Project and select Project Files and Select pom.xml and make some changes mentioned in ss.



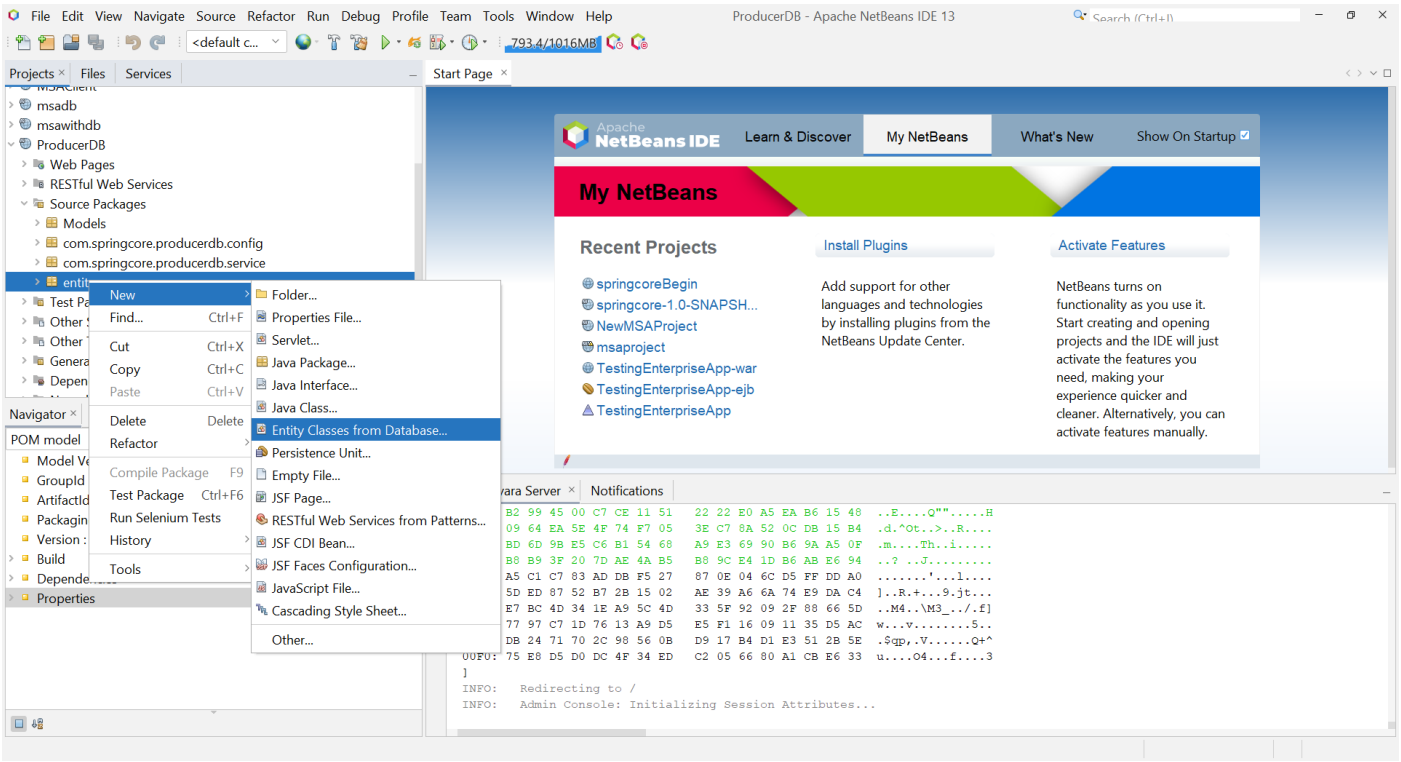
14) Go to Other Sources -> src/main/resources and create folder named META-INF (name is mandatory)



15) Right click on that META-INF and create one properties file named micrprofile-config

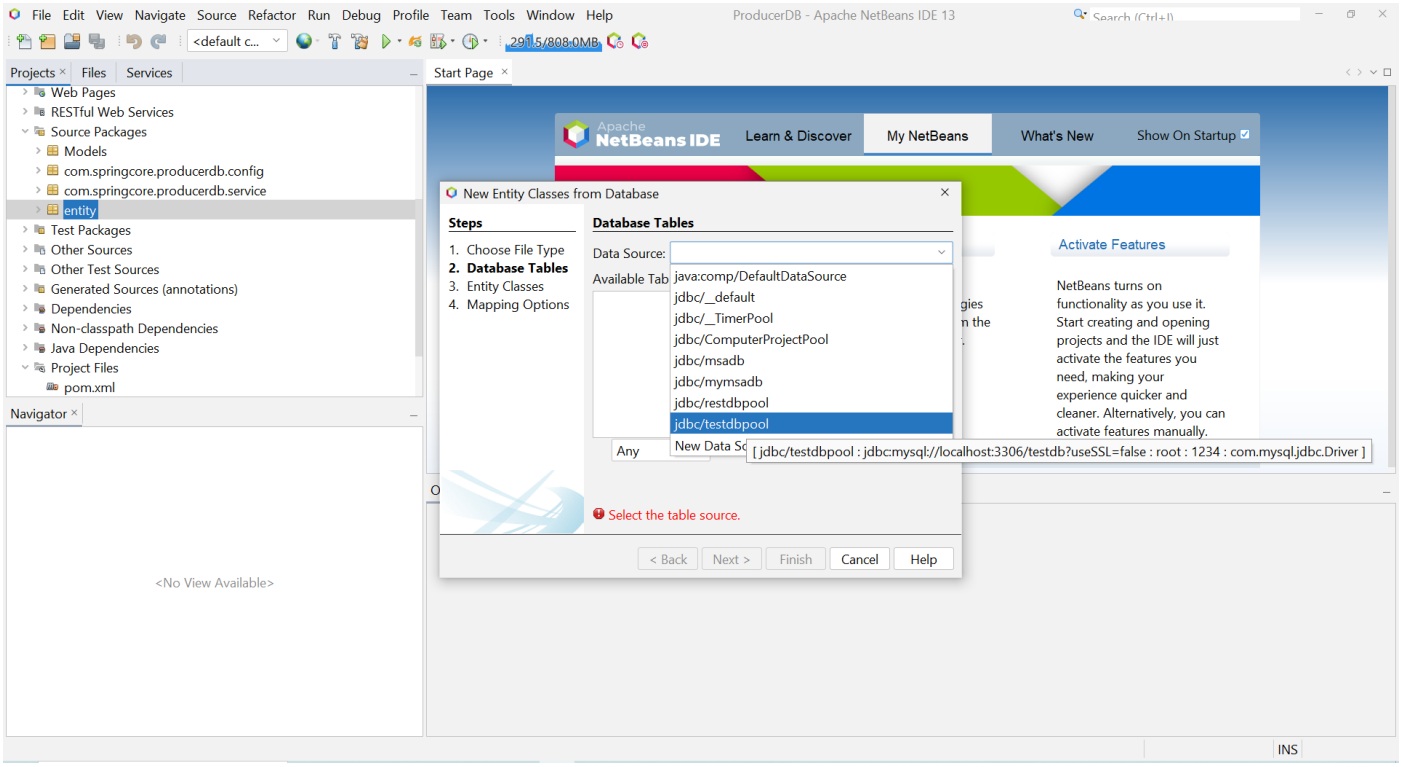


16) Go to source packages and create one package called entity, Right click on that package and select new->entity classes from database

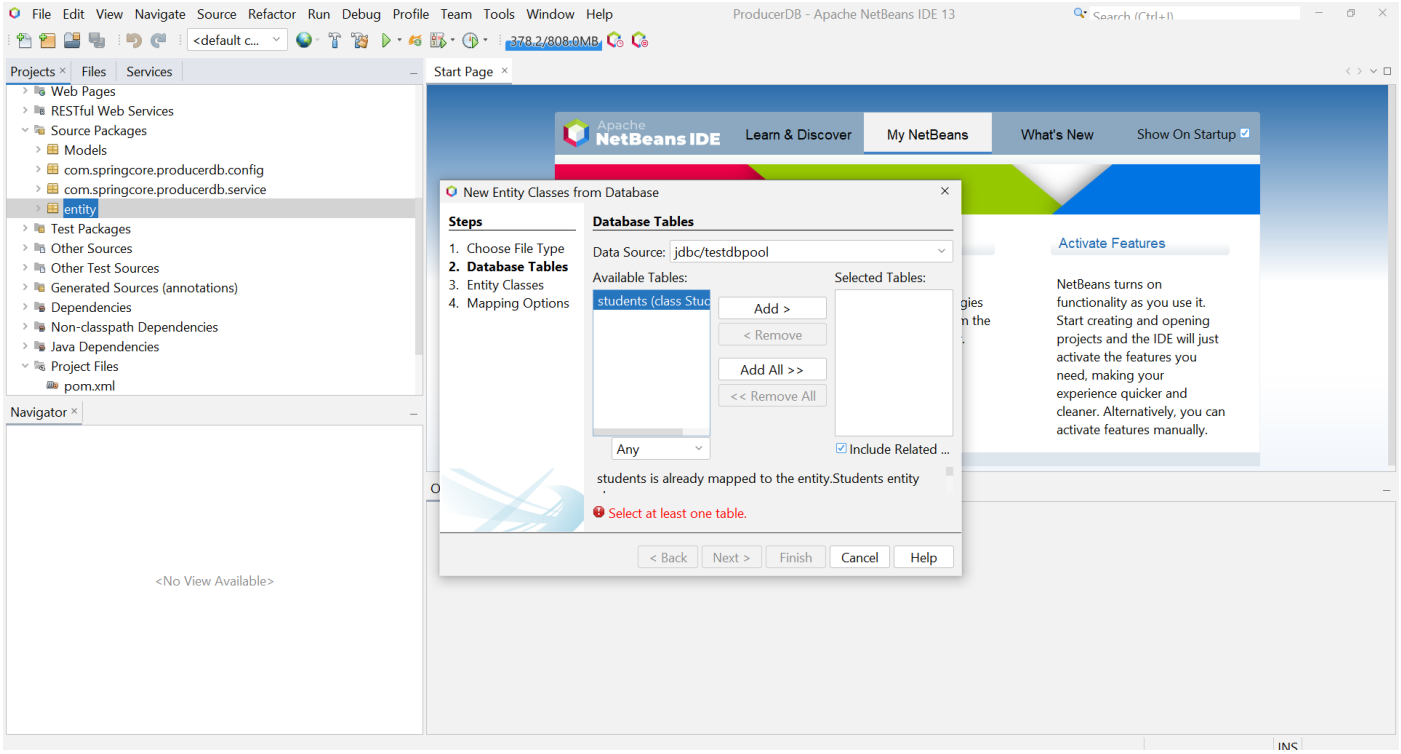


17) Fetch from server , Choose your server and then click next.

18) Select your jdbc resource , Here in my it would be jdbc/testpooldb and click next.
(If next dialogue window ask you password then enter mysql password)

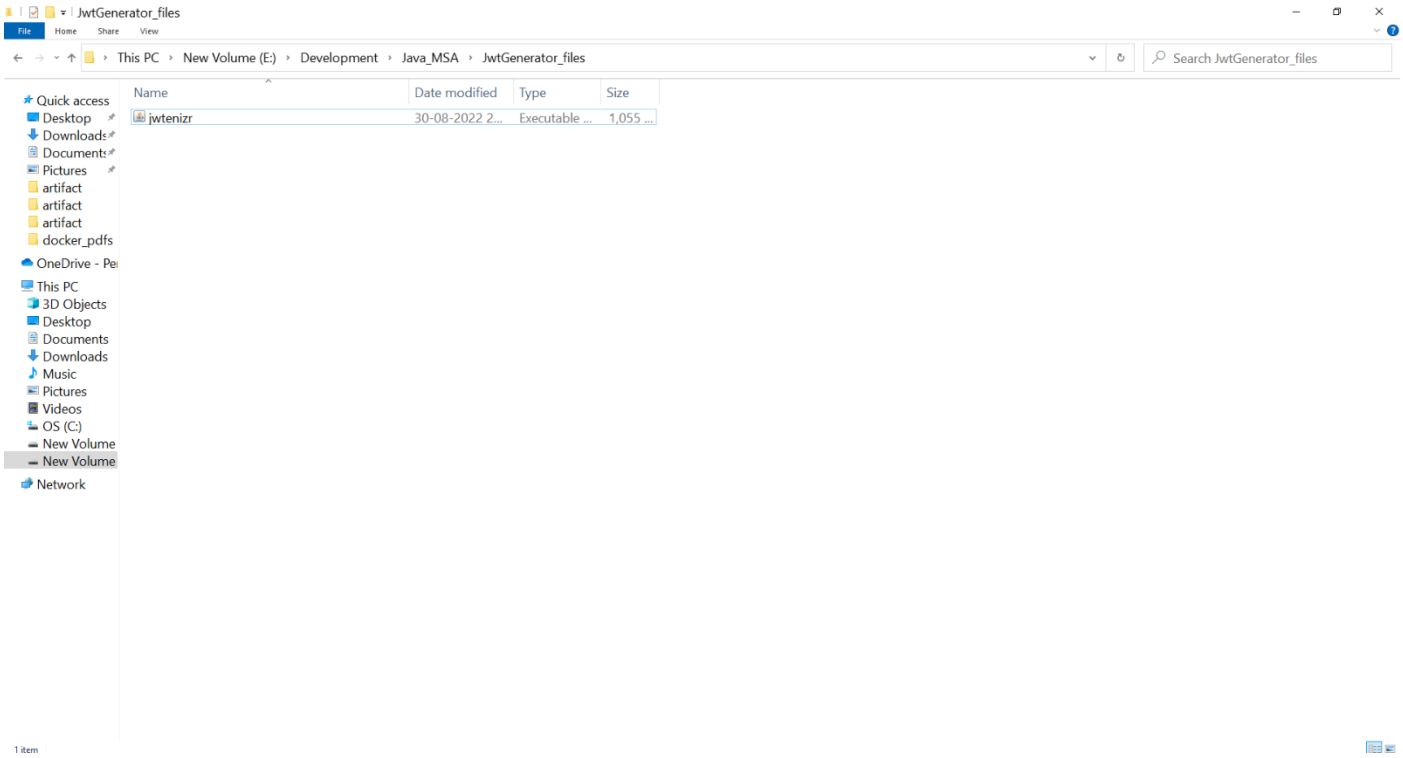


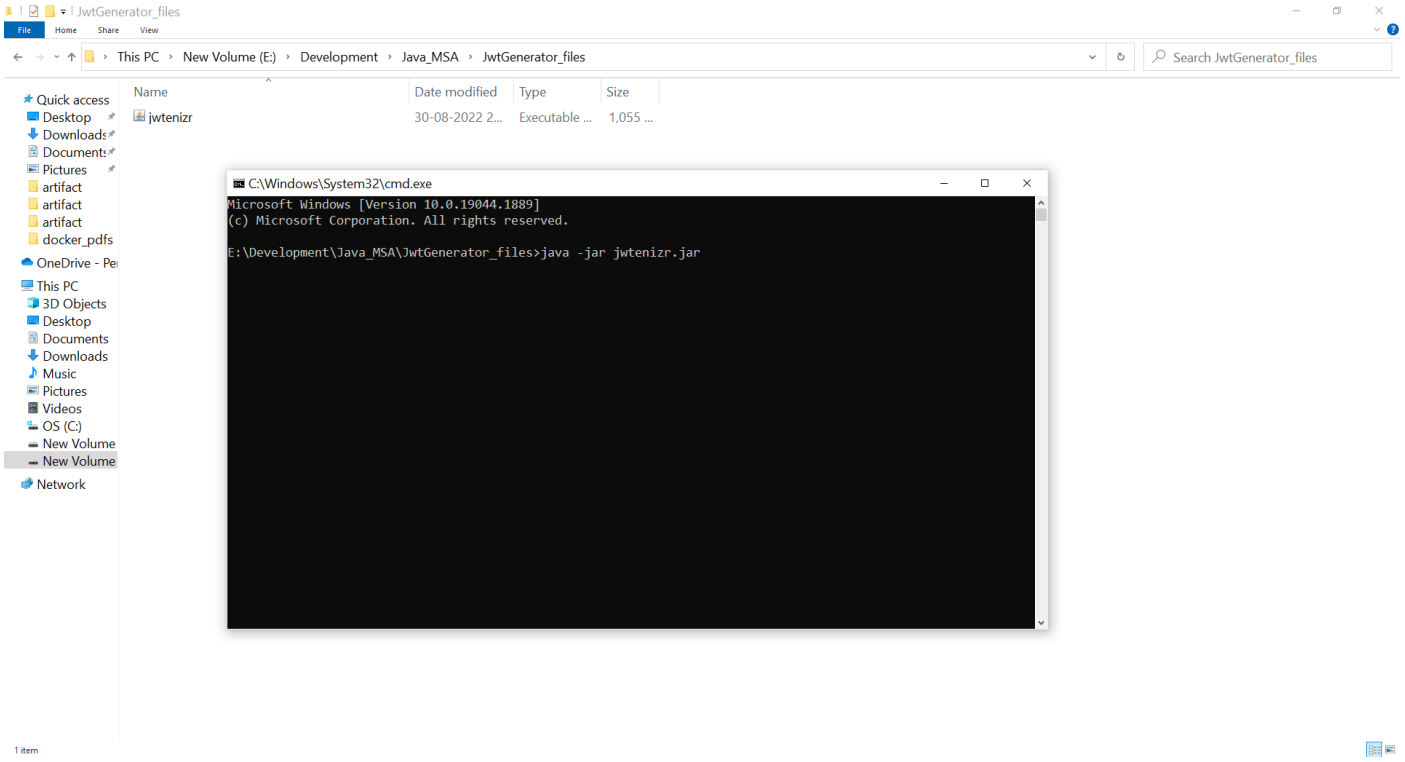
19) Add the tables and click next, next till finish don't make any changes



Pefect! Let's create JWT token and issuer , For that you need jwt jar file, Put jwt jar file in one blank folder

20) Open command prompt/terminal over there and add the command

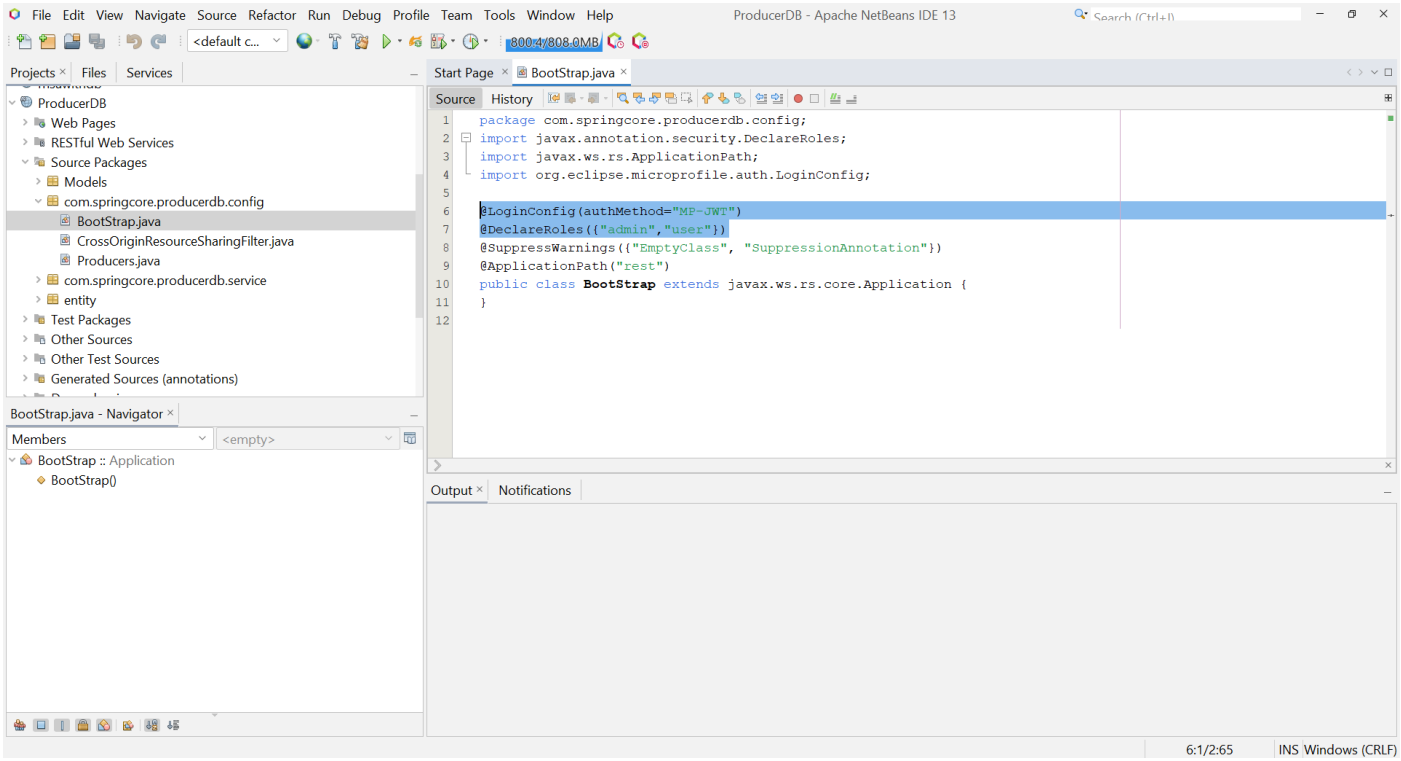




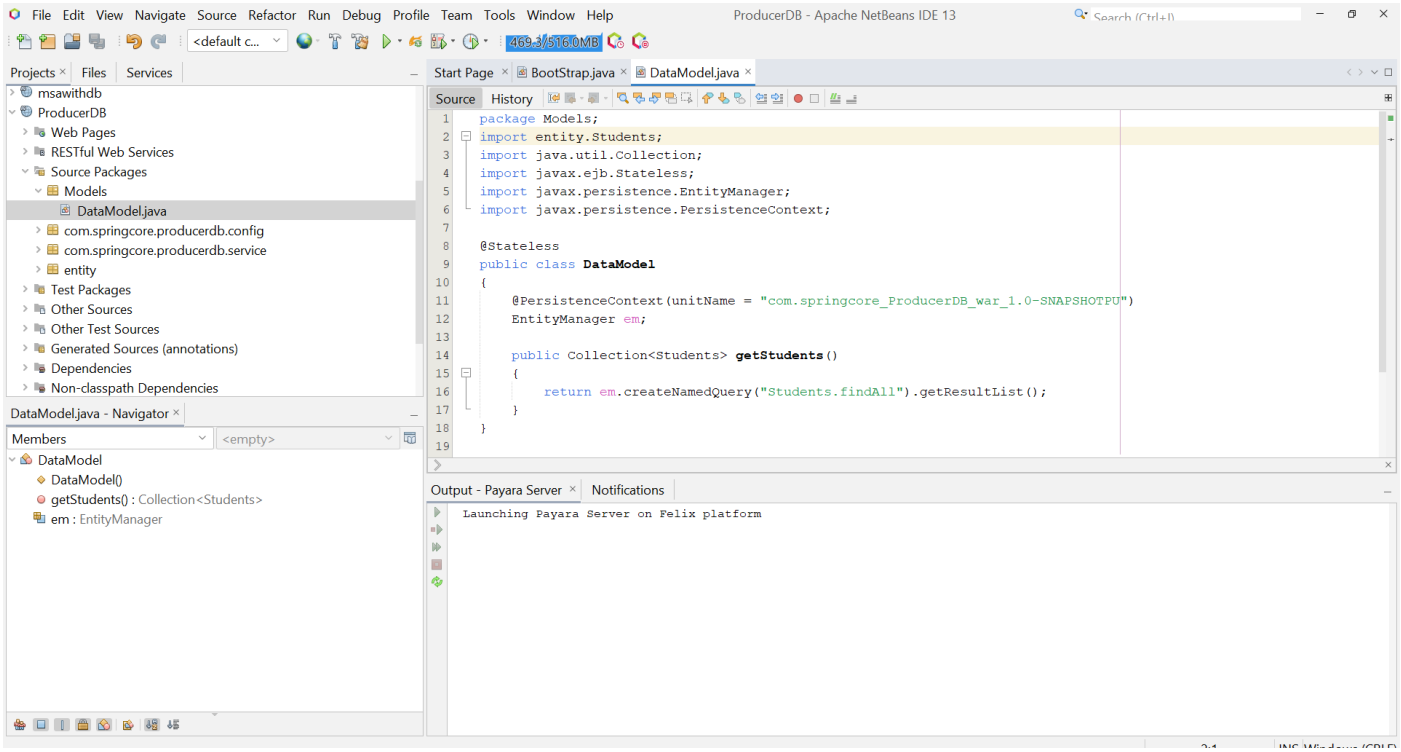
21) Some file would be generated and then open jwt-token file in your favourite text editor

and make some changes like groups set to admin and user and then save that file , again fire the command using command prompt or terminal . See that command in step no 20. Don't forget this step. Otherwise your app will give unexpected behaviour

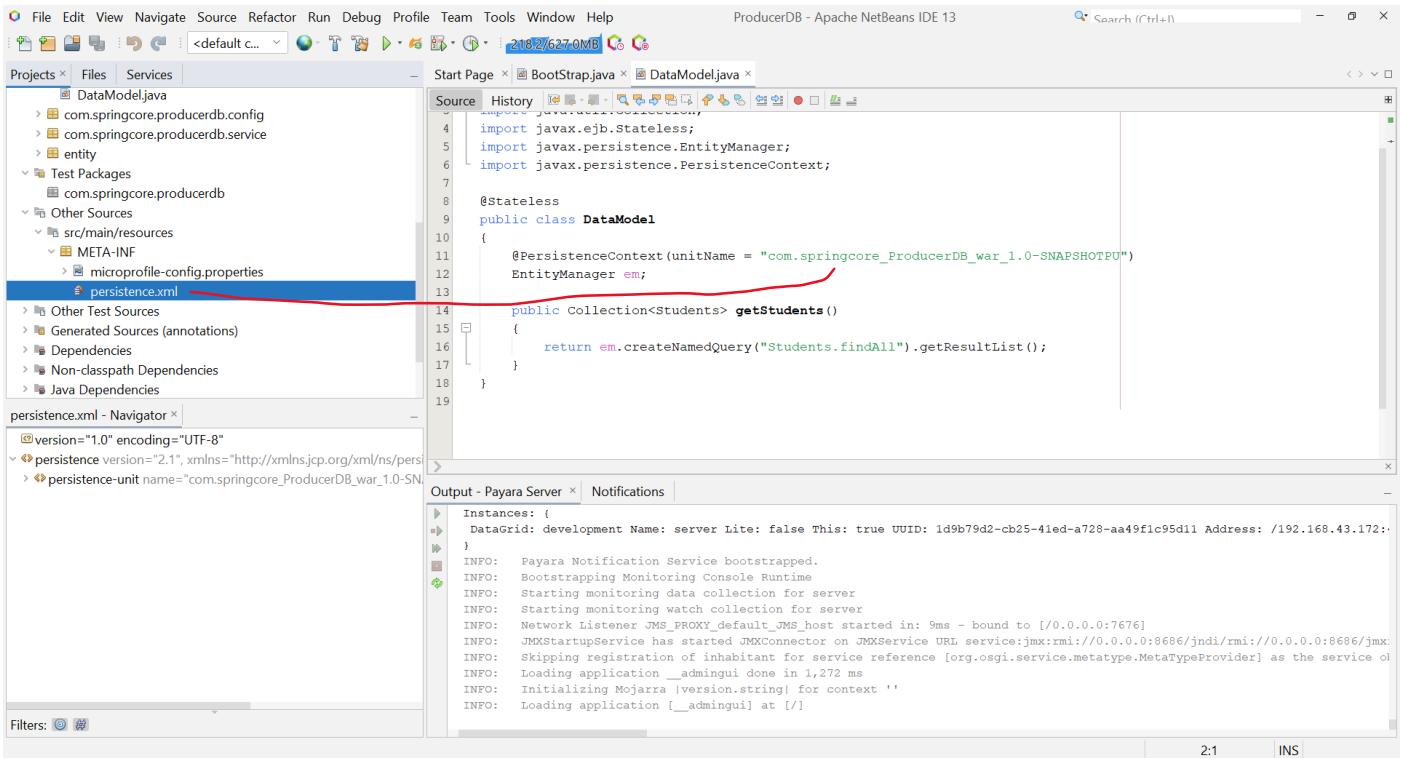
22) Go to your project go inside the package com.springcore.projectname.config and open Bootstrap.java and add some annotations highlighted in ss.



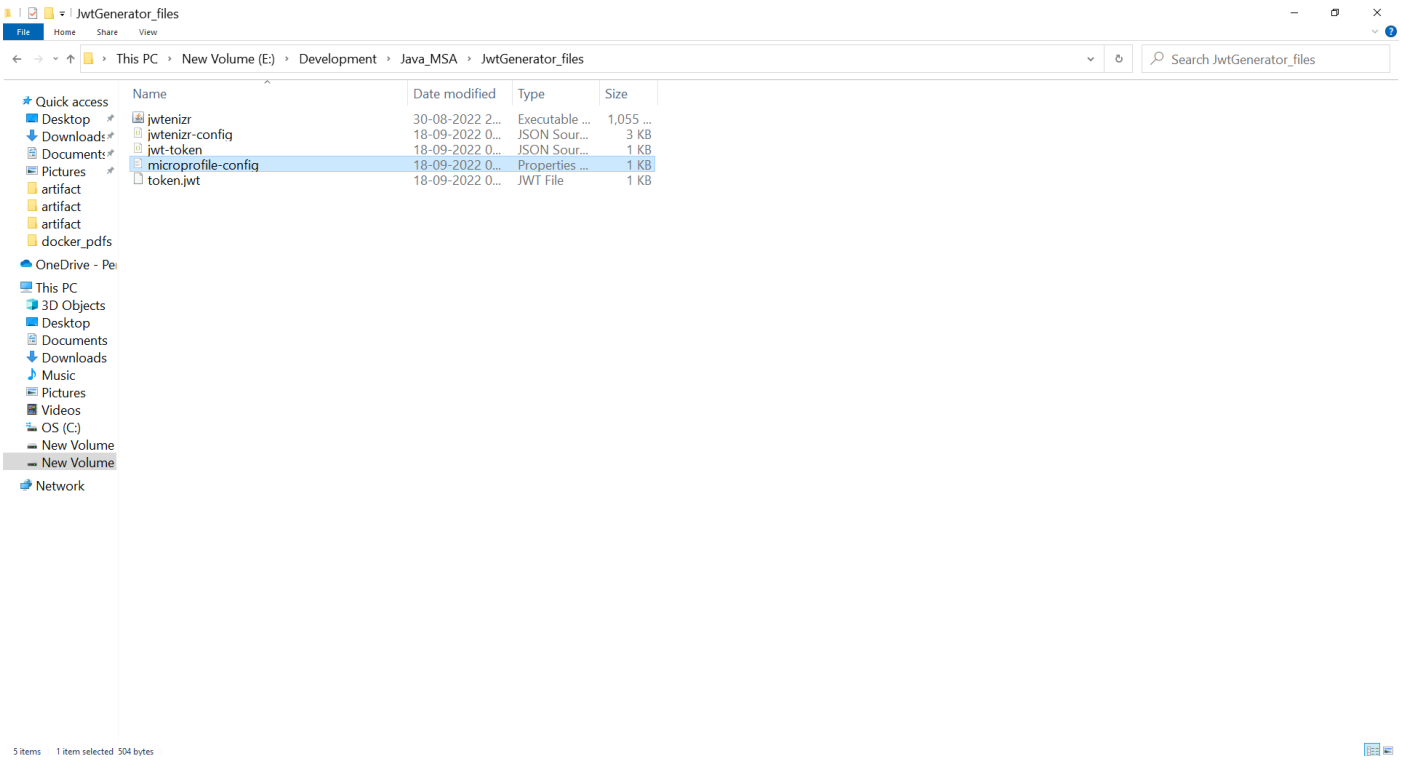
23) Create new package Model and create new Java class file , In my case I've created Models package and DataModel.java file in that package and add the code



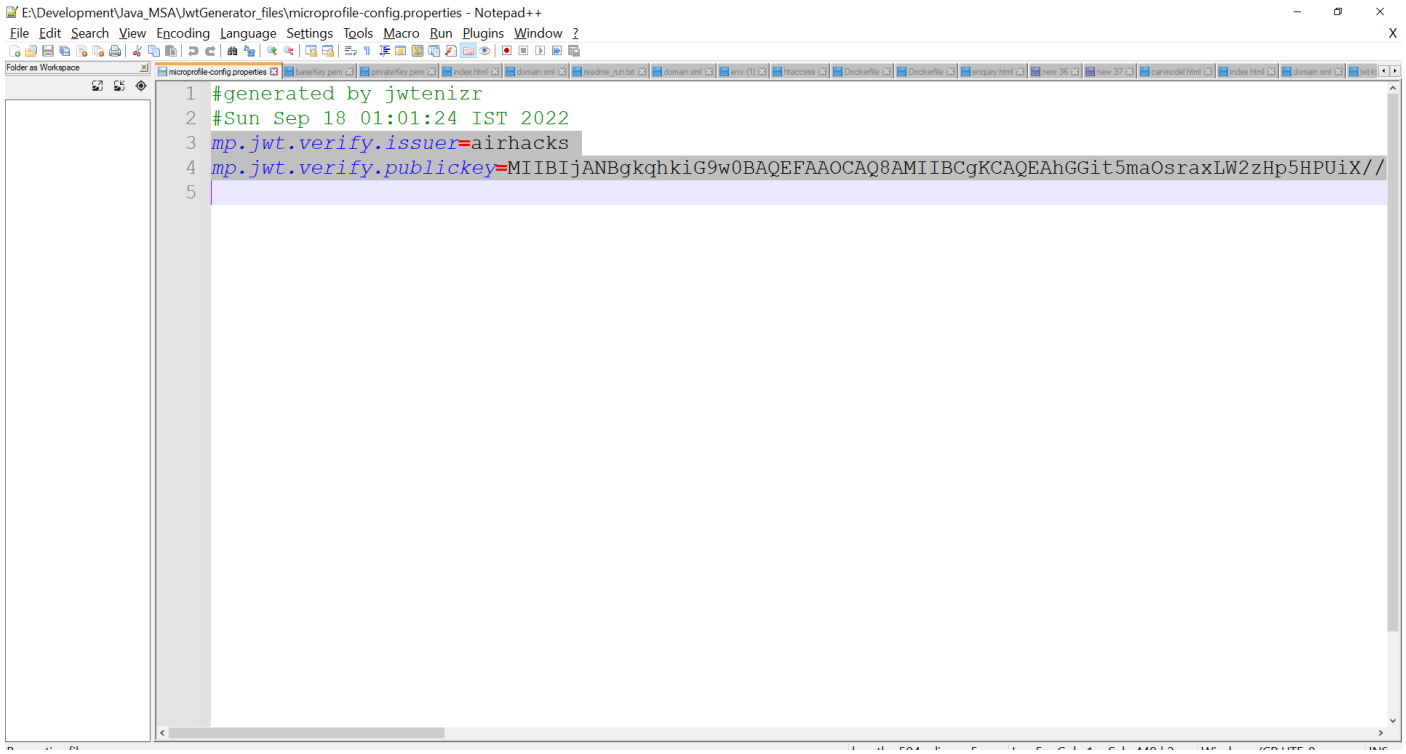
24) Replace unitName with your unitName available on persistence.xml file , (Other sources->src/main/resources/META-INF)

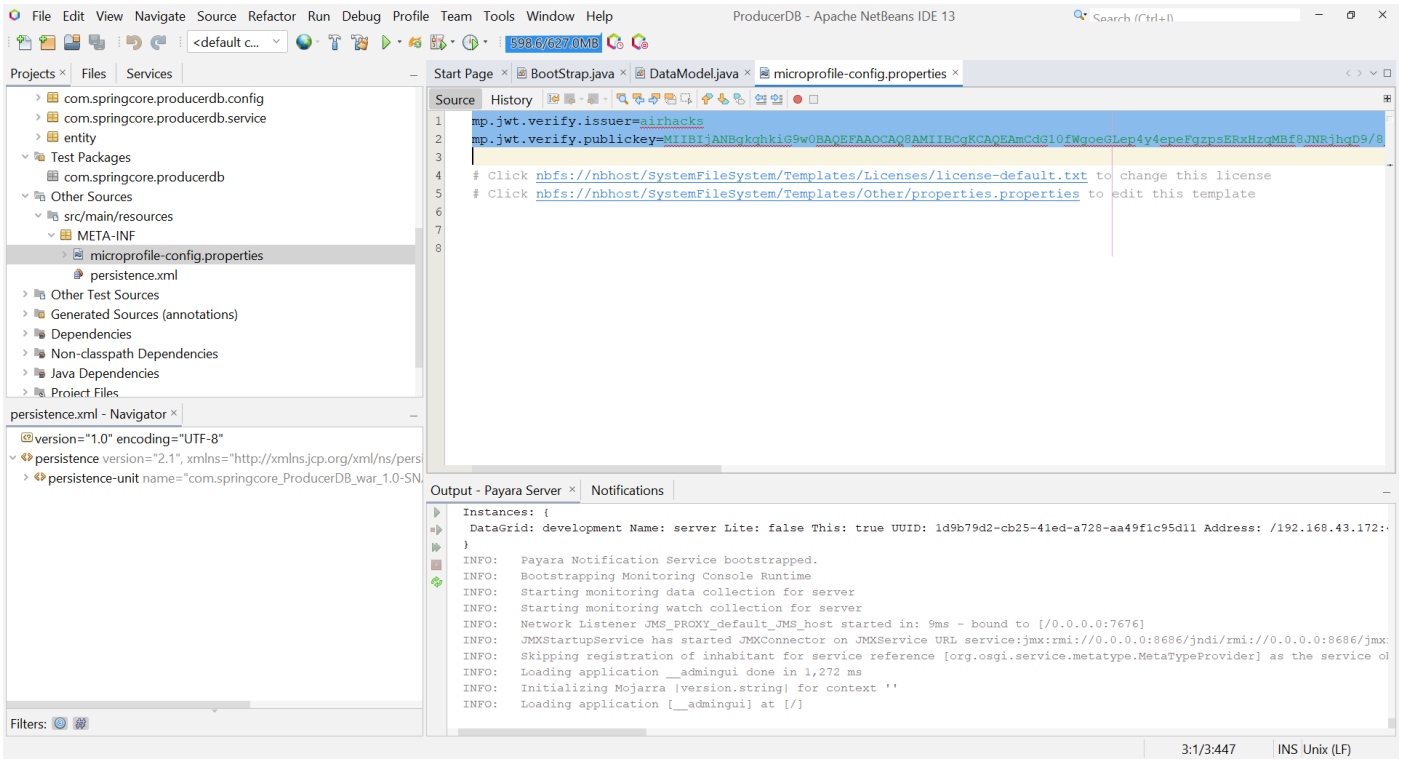


25) Open microprofile-config file that you've generated in your favourite text editor

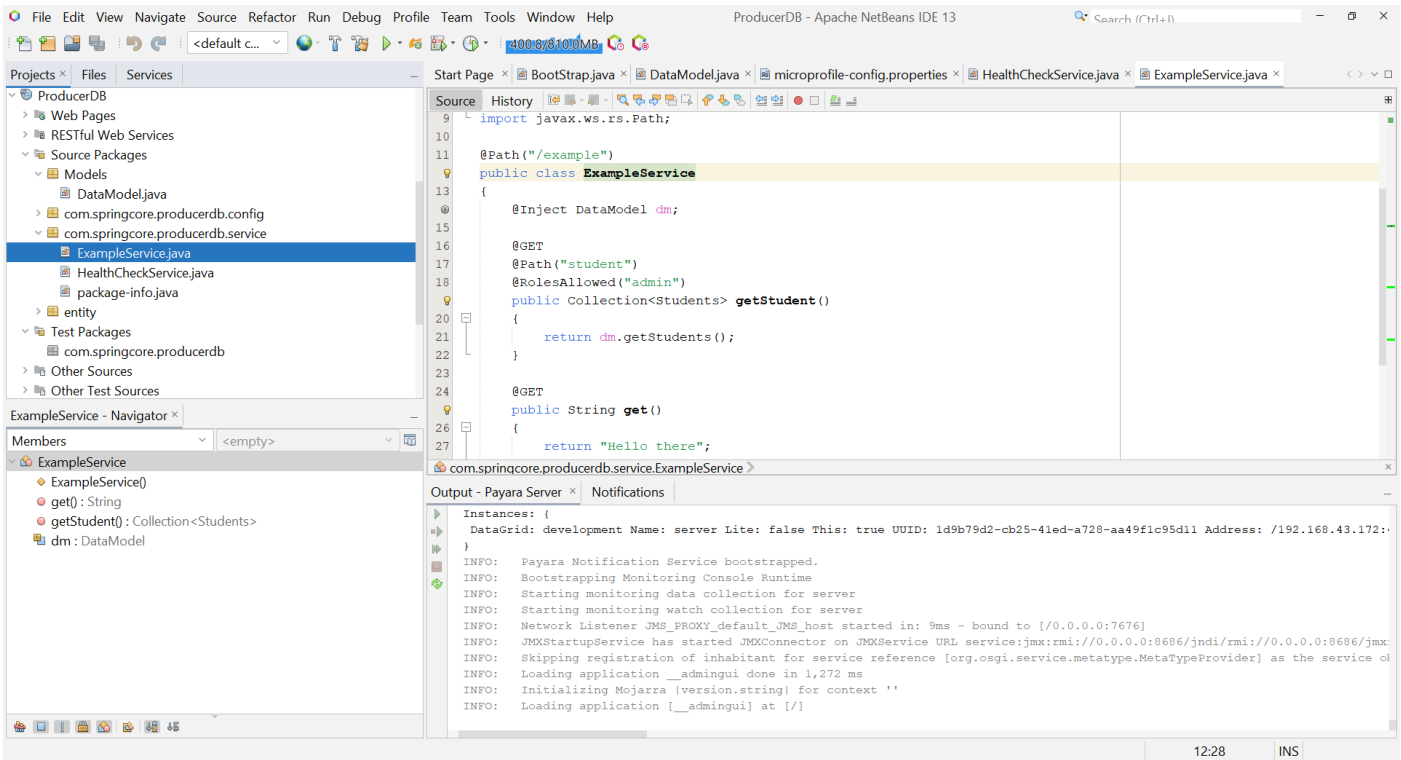


26) Copy the the below content and paste on your microprofile-config properties file

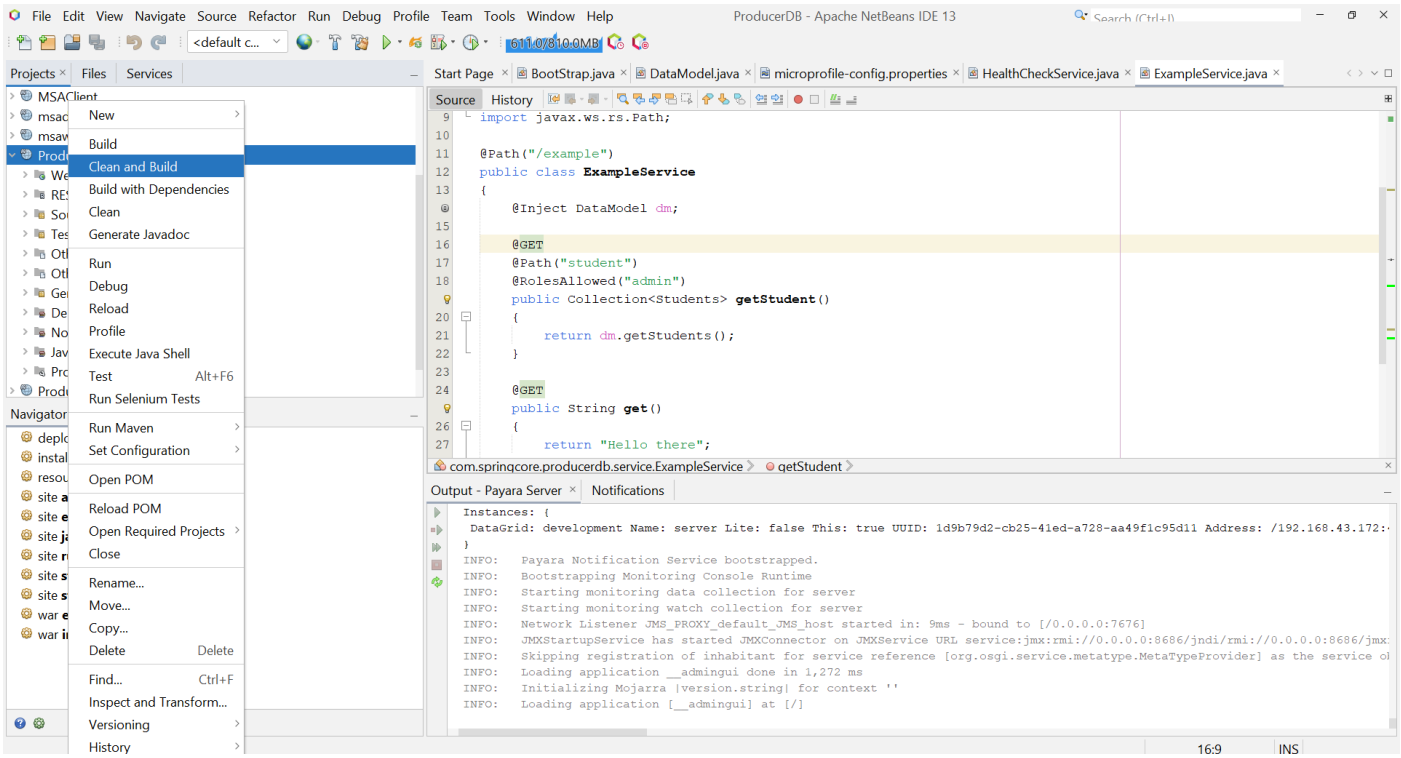




27) Open ExampleService.java (source package/com.springcore.projectname.service / ExampleService.java) and add the code

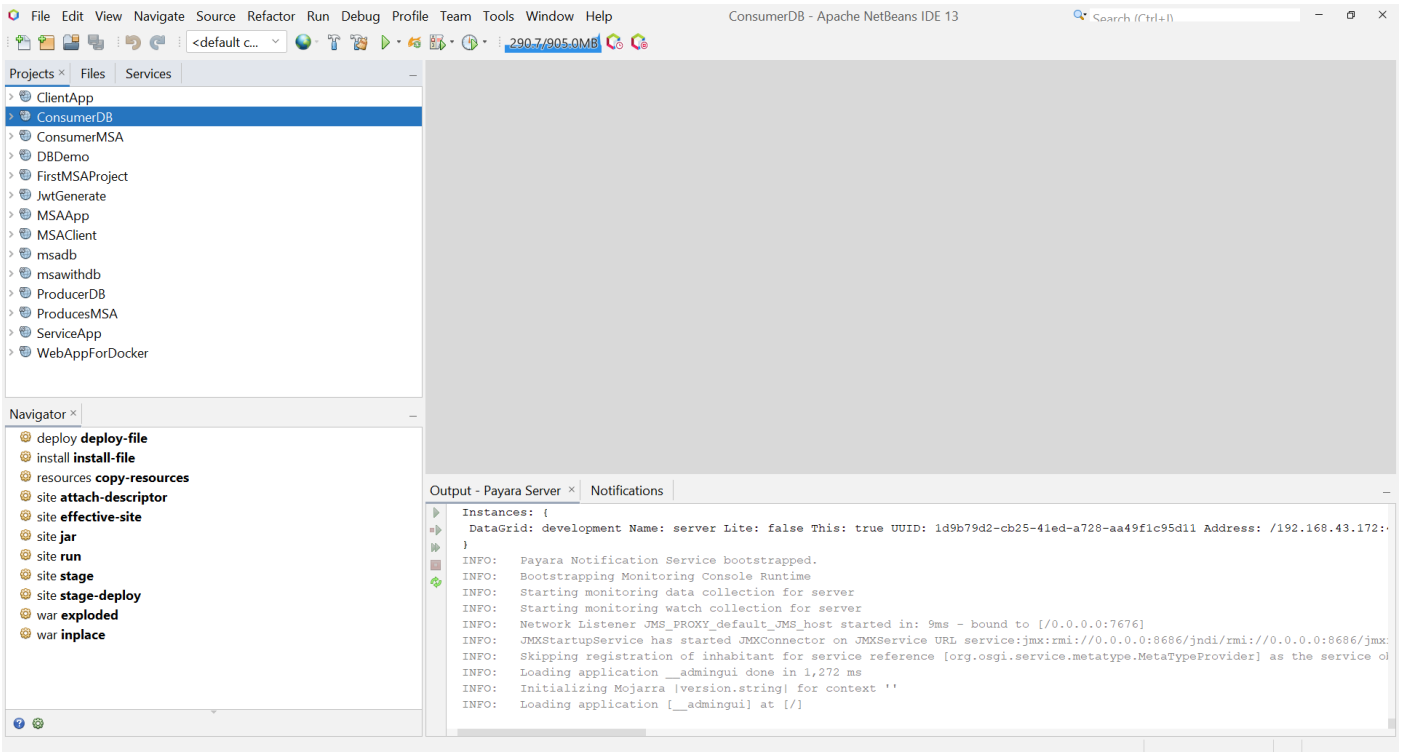


28) Right click on the project and click on clean and build

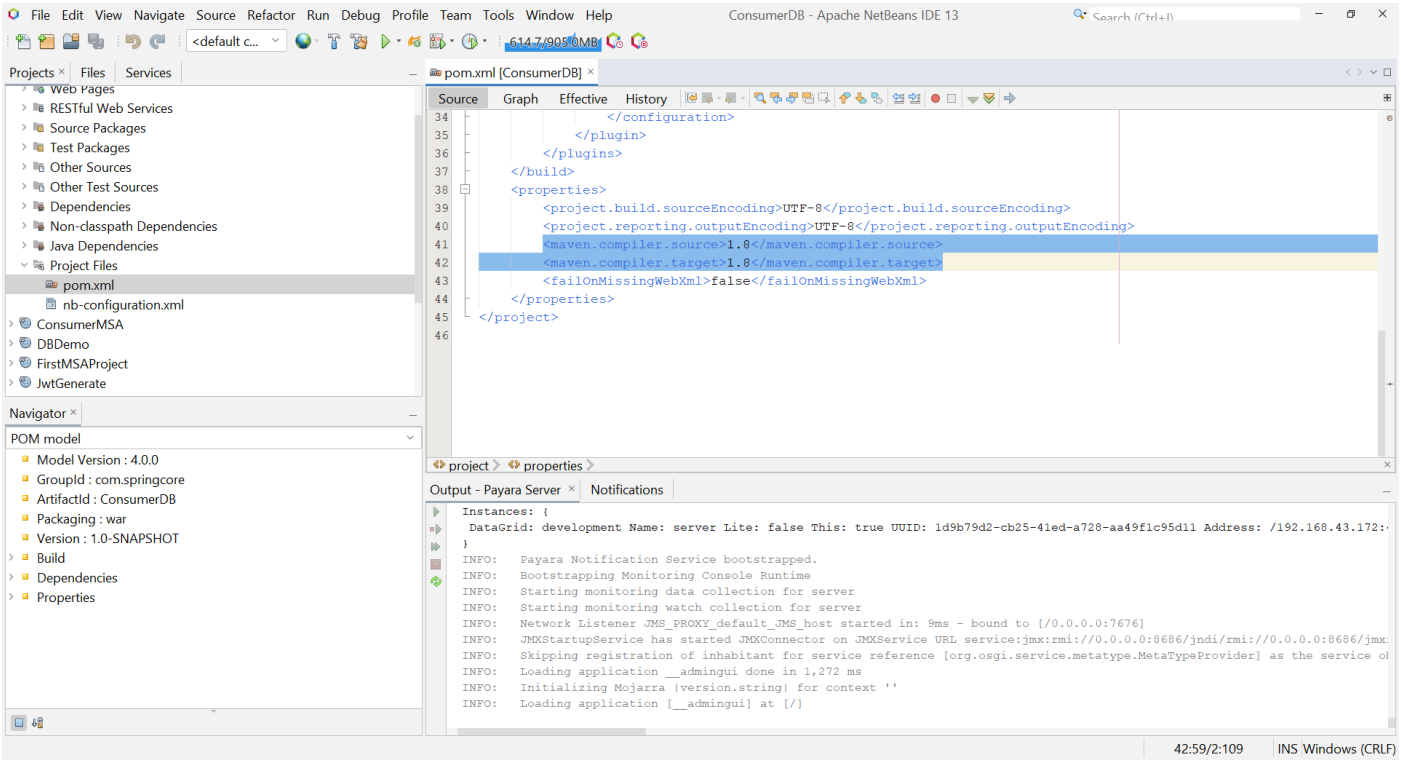


Alright we have done till upto Producer App Let's move on to create Consumer App which fetch the data from Producer App.

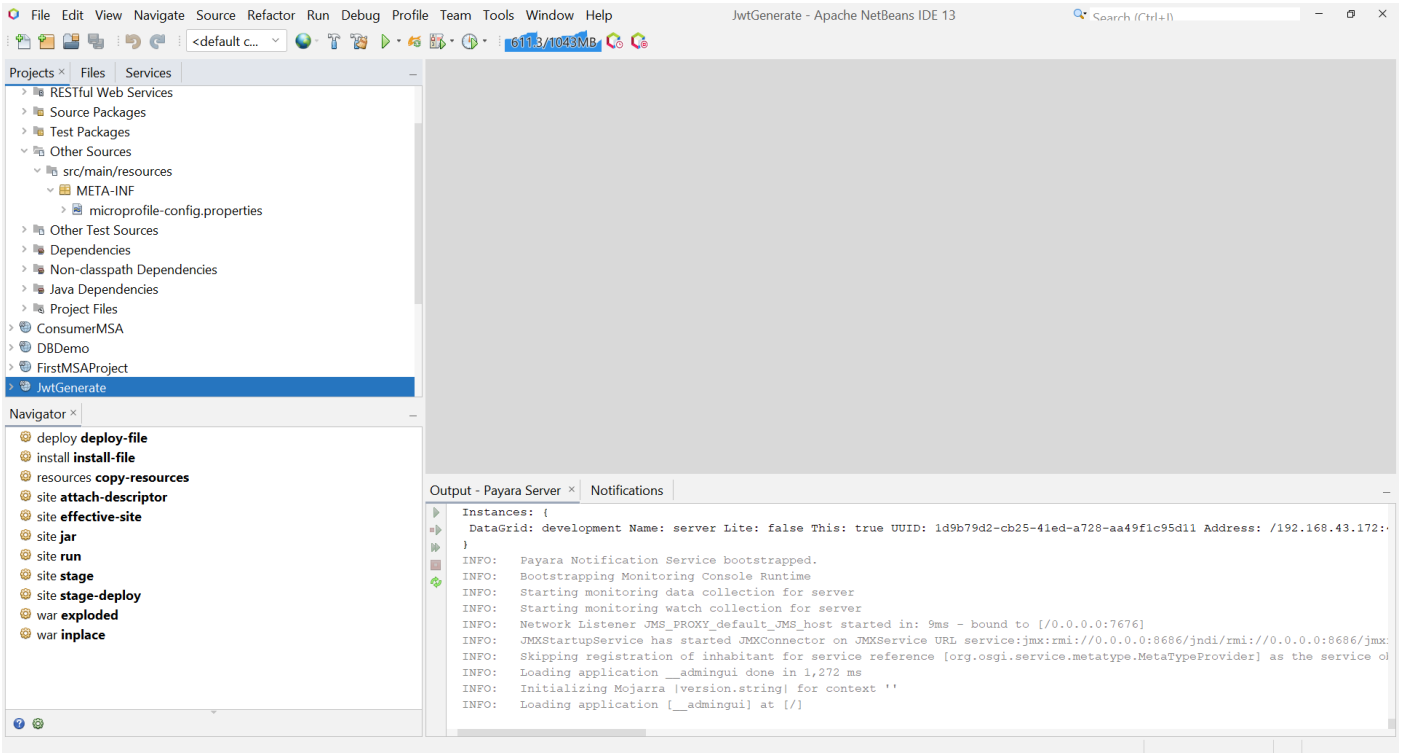
29) Create new project same as earlier , In my case I've given the project name ConsumerDB.



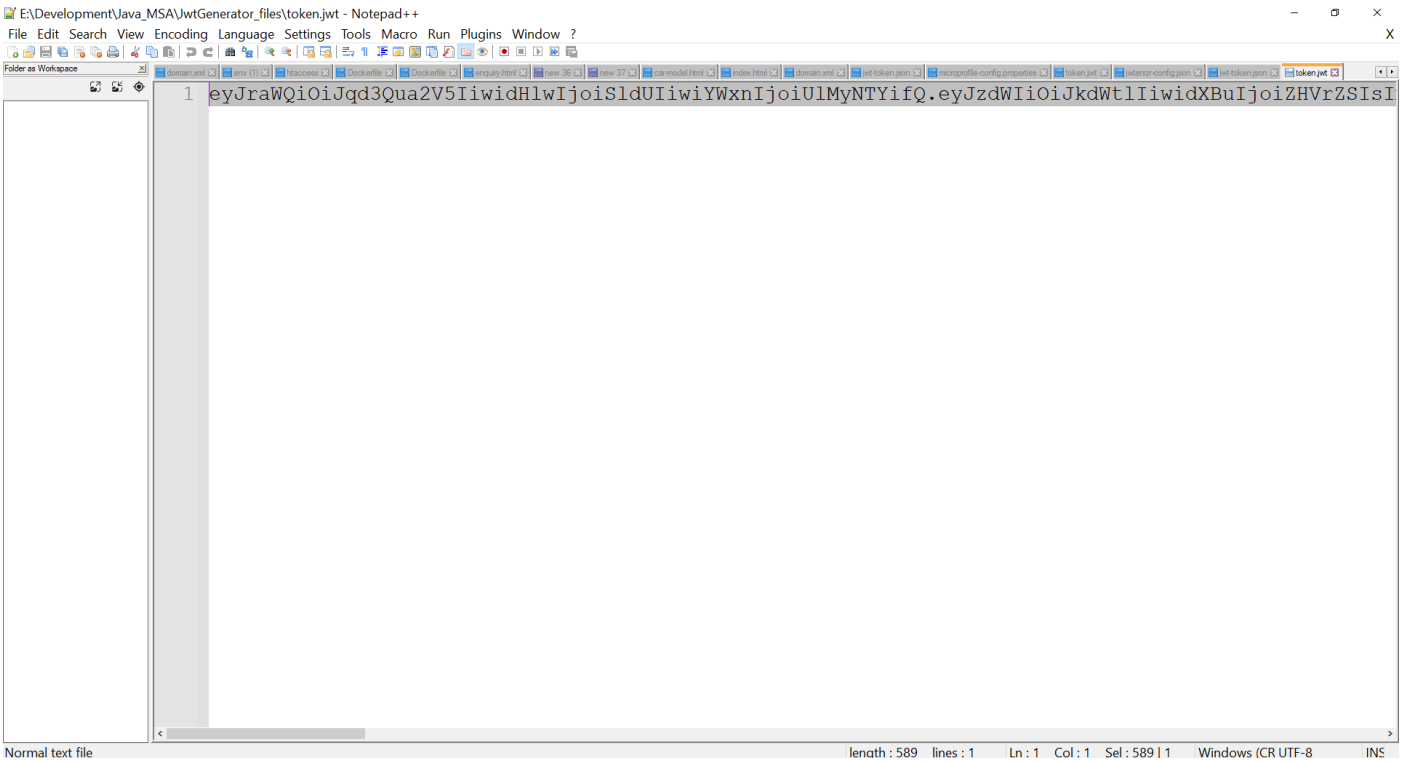
30) As we did earlier make some changes in pom.xml file



31) Go to Other Sources->src/main/resources and create new folder (Not the package) and name should be META-INF and then create one property file name should microprofile-config in Consumer App as we did same as earlier.

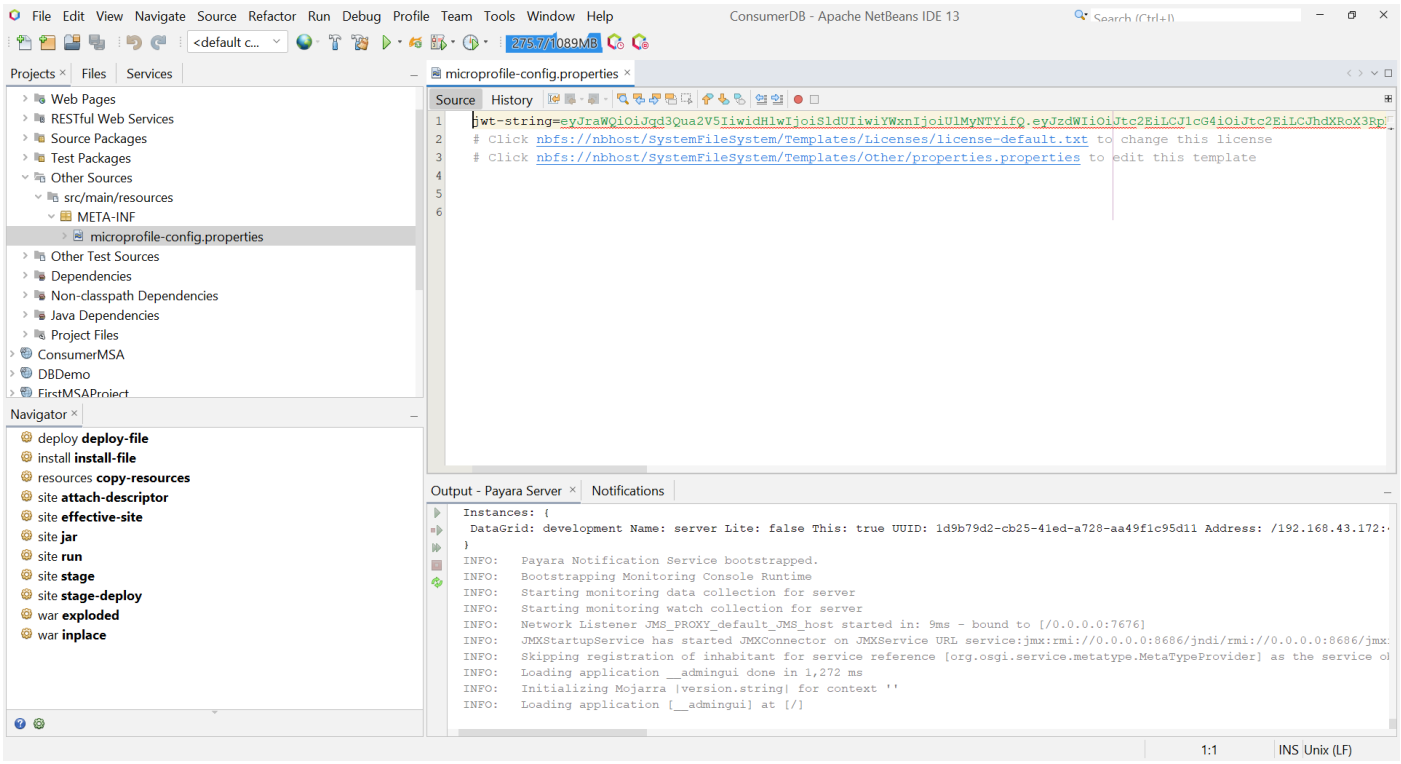


32) Now Open jwt.token file that you've generated in your favourite text editor copy the whole string.



33) Paste that string in your microprofile-config property file of Consumer App

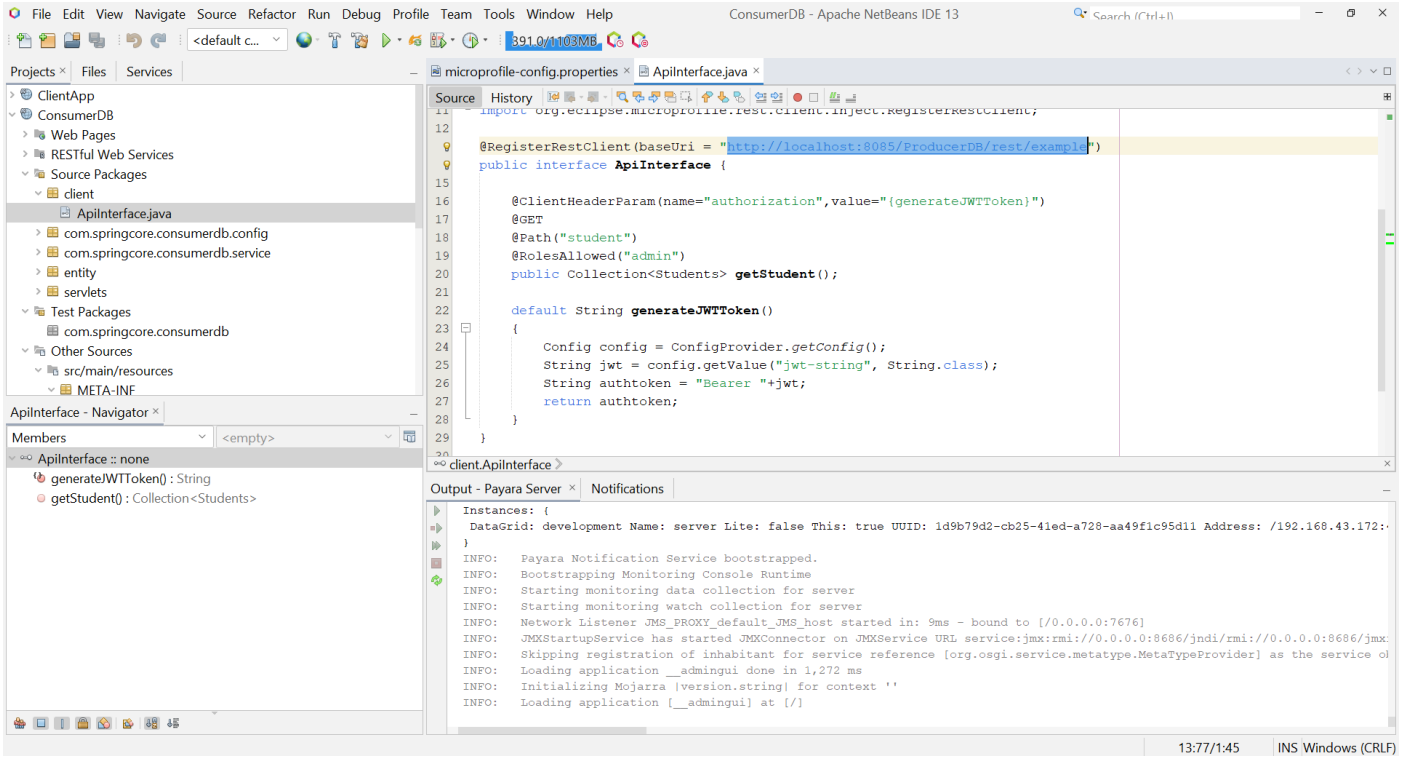
Format should be **jwt-string=paste_your_jwt_string**



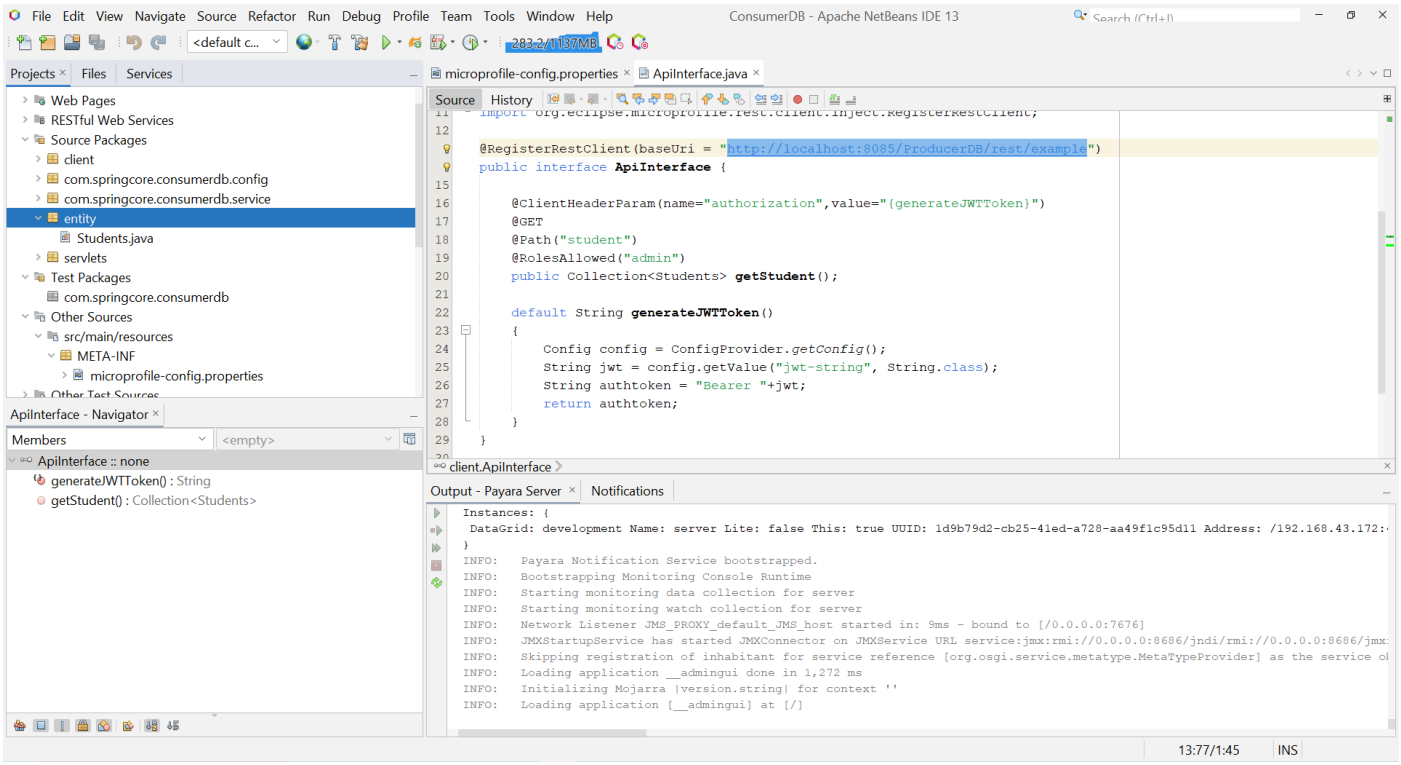
34) Create one package and in that package create one Interface. In my Case I've Created ApiInterface.java in client package and add the code

(Note : your base url should be your Producer app Url along with port number in which you gonna run your app in payara micro server)

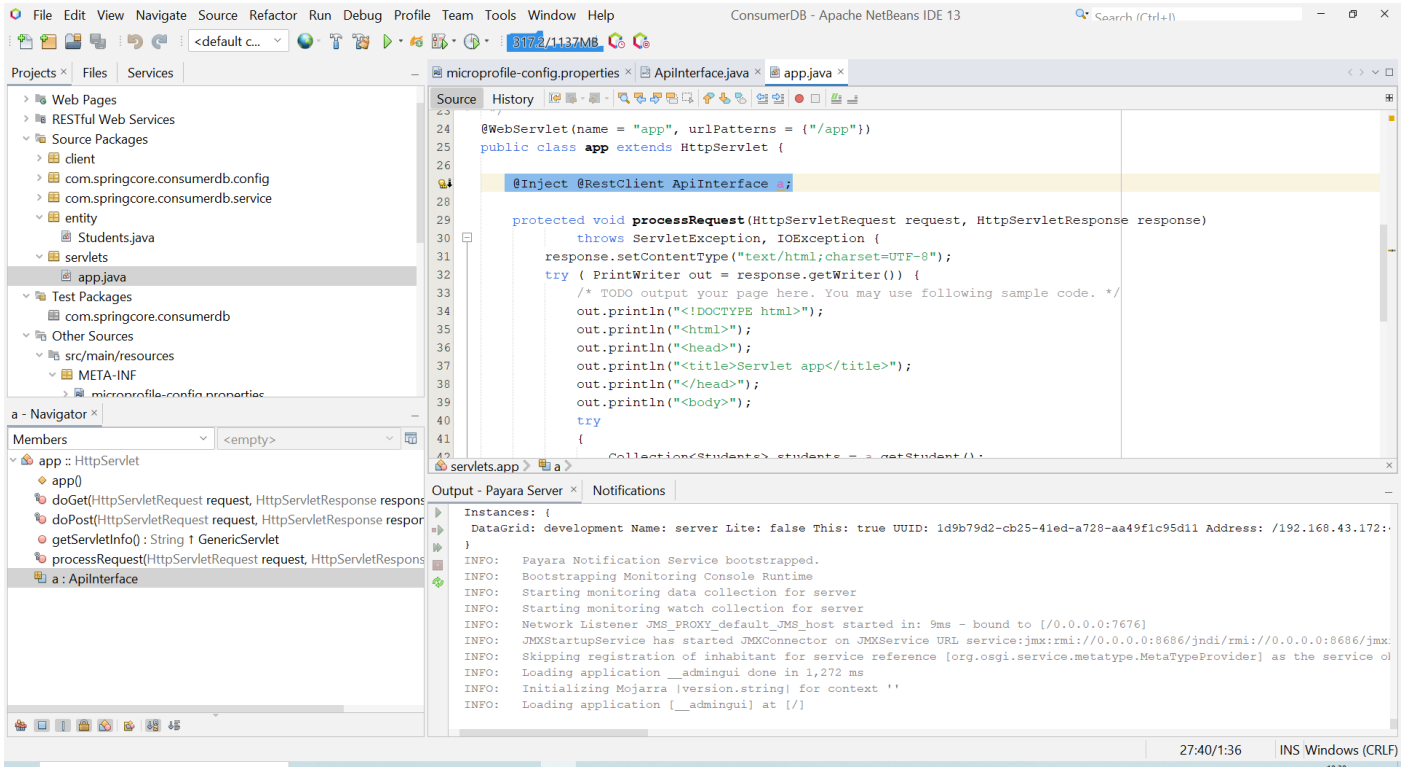
In my case url : http://localhost:8085/ProducerDB/rest/example

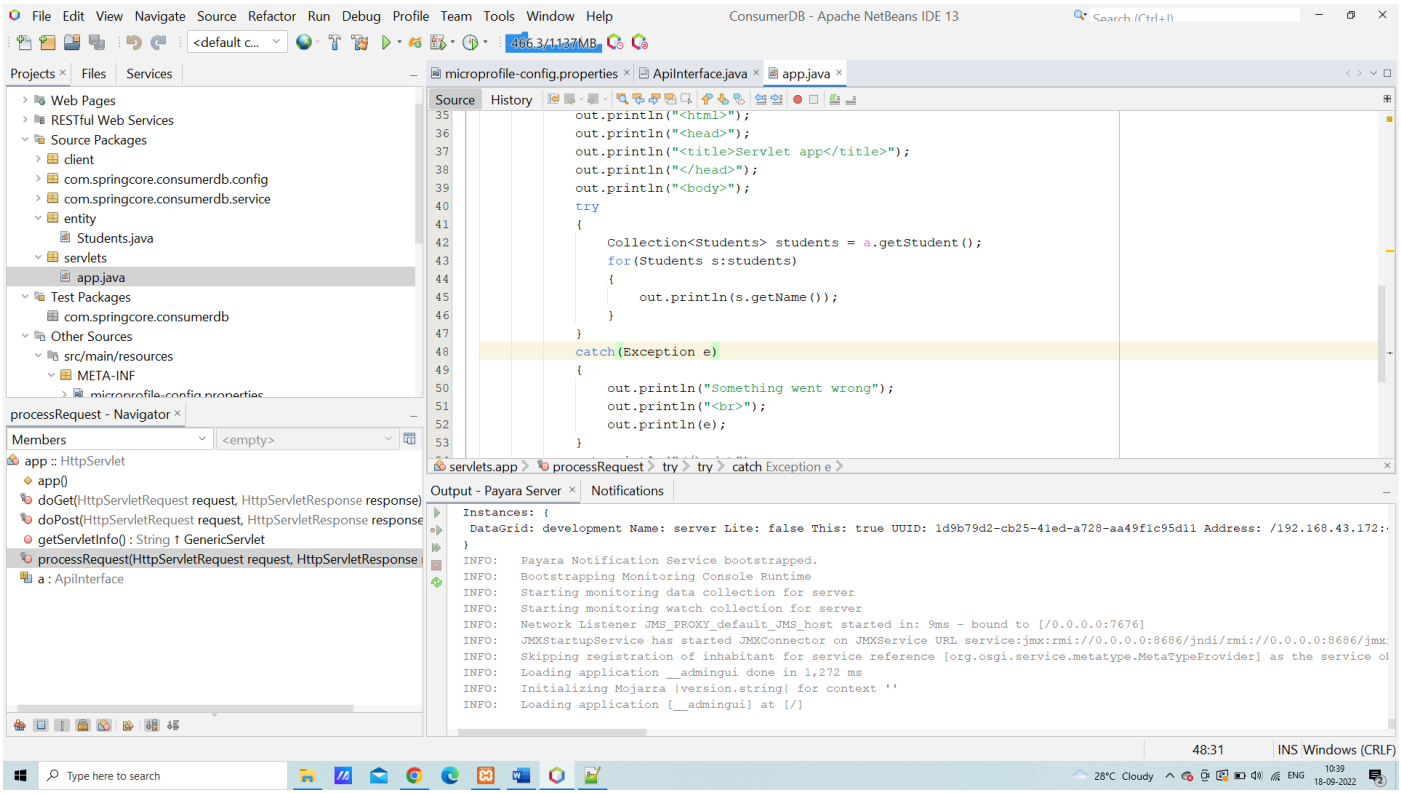


35) Copy the entity package from Producer App and paste that package in your source package of Consumer App.

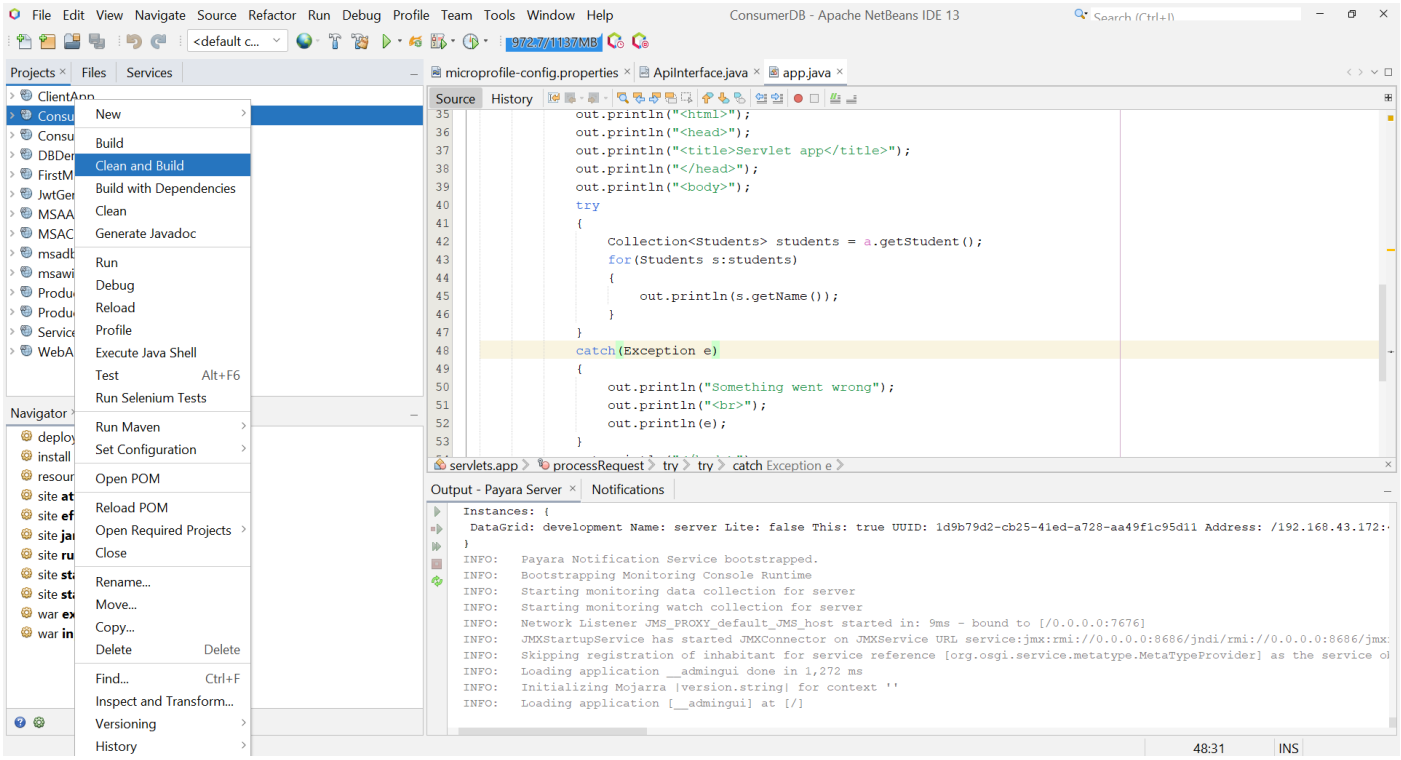


36) Create another package and add one Servlet. Write the Code mentioned in the below ss.
In my case , I've create app.java servlet in servlets package.



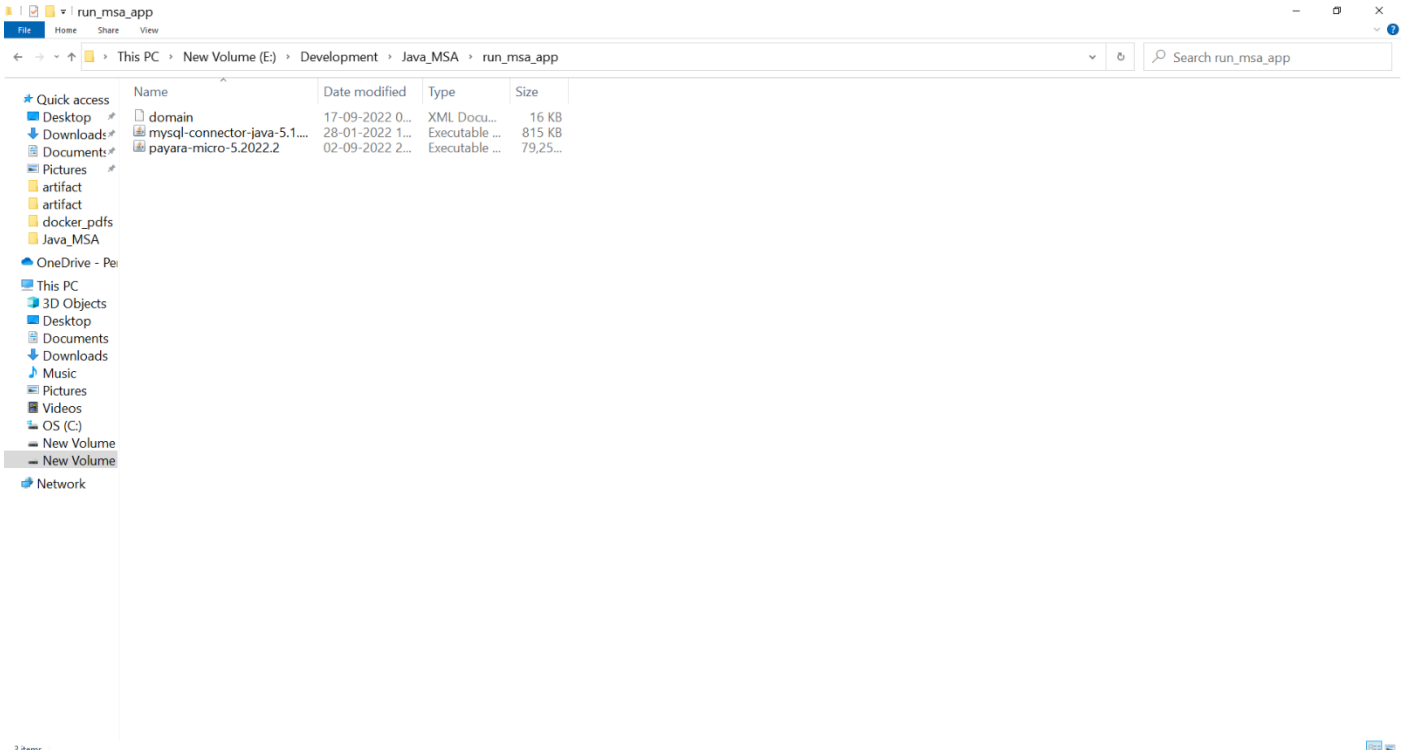


37) Right click on that project click on Clean and Build

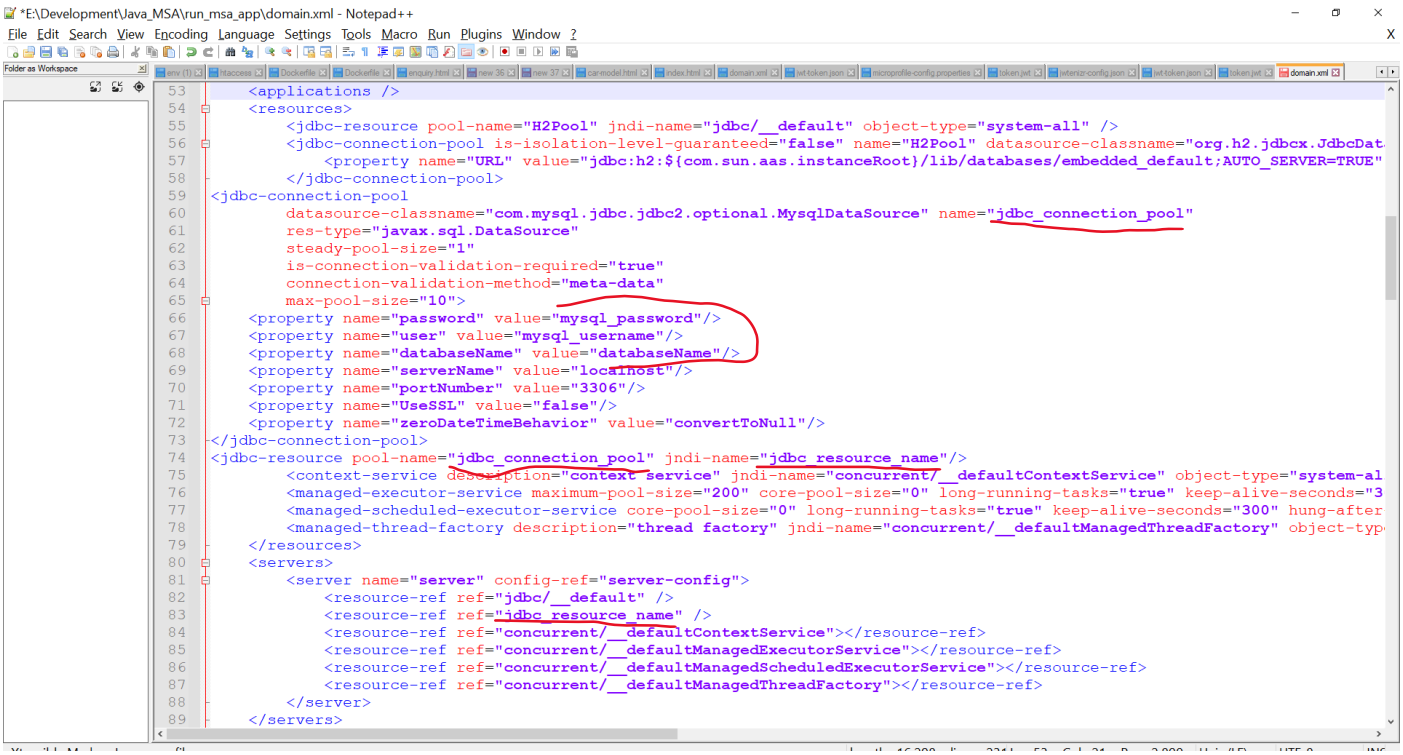


Alright Fine! Let's move on deployment part, but before moving on deployment I want you to make some changes in domain.xml according to your jdbc connection

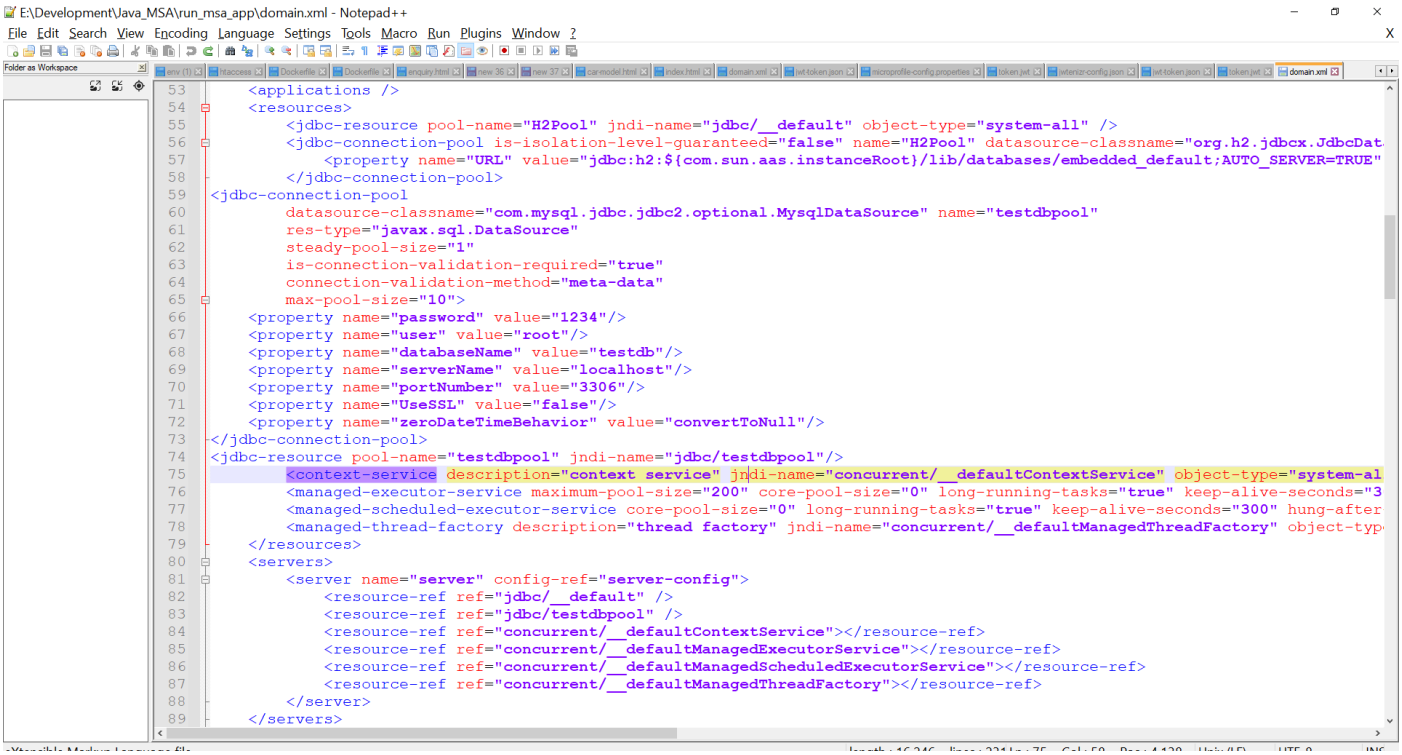
38) Create one blank folder and put payara-microprofile.jar file and mysql-connector jar file along with domain.xml file



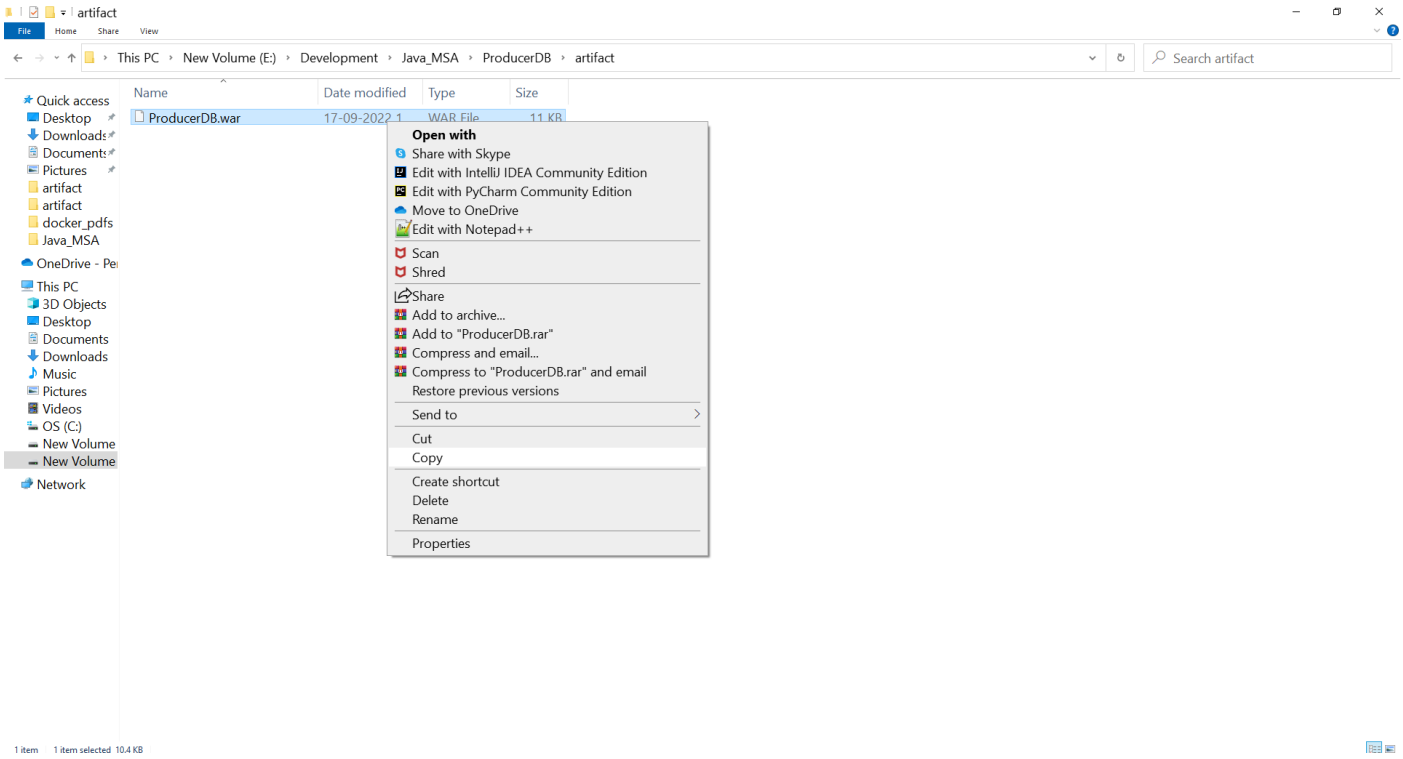
39) Open domain.xml file and make some changes denotated by red underline

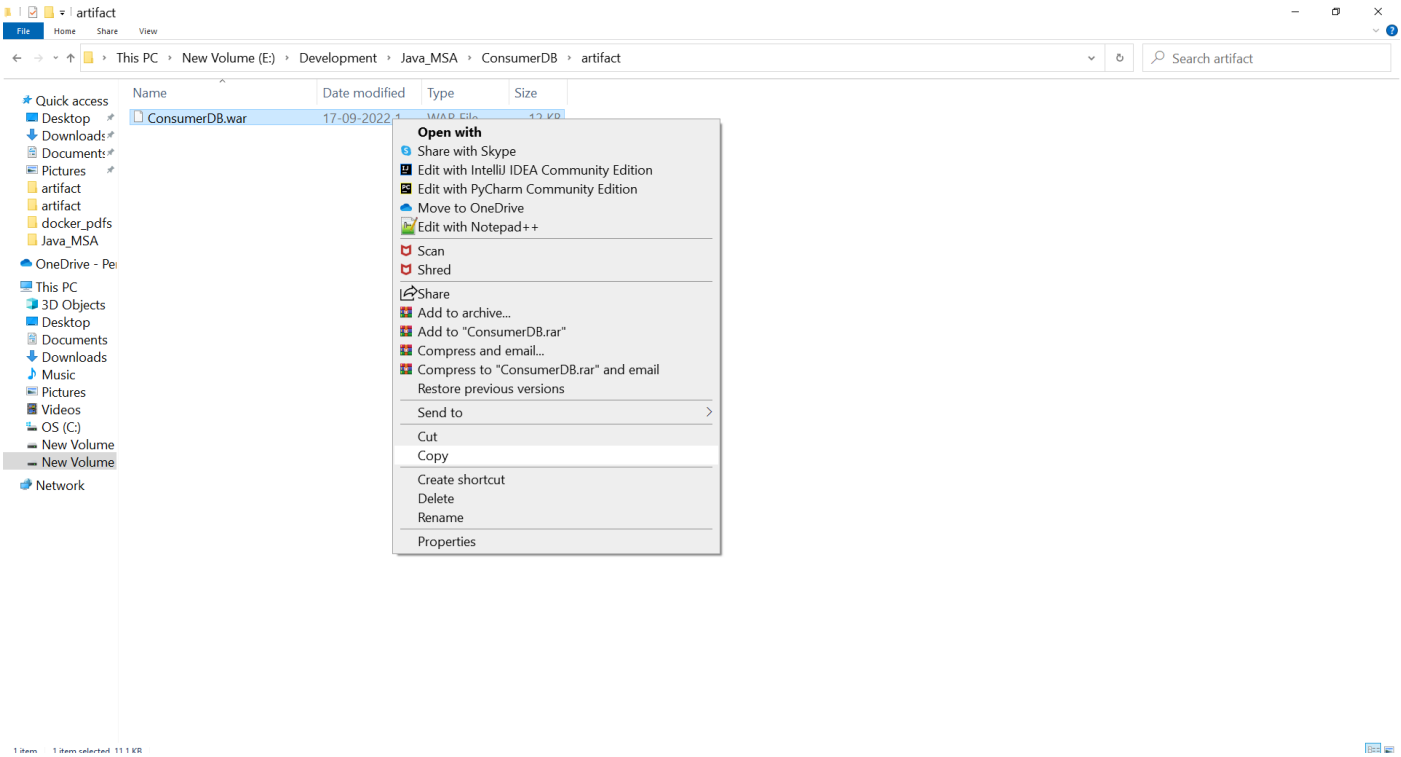


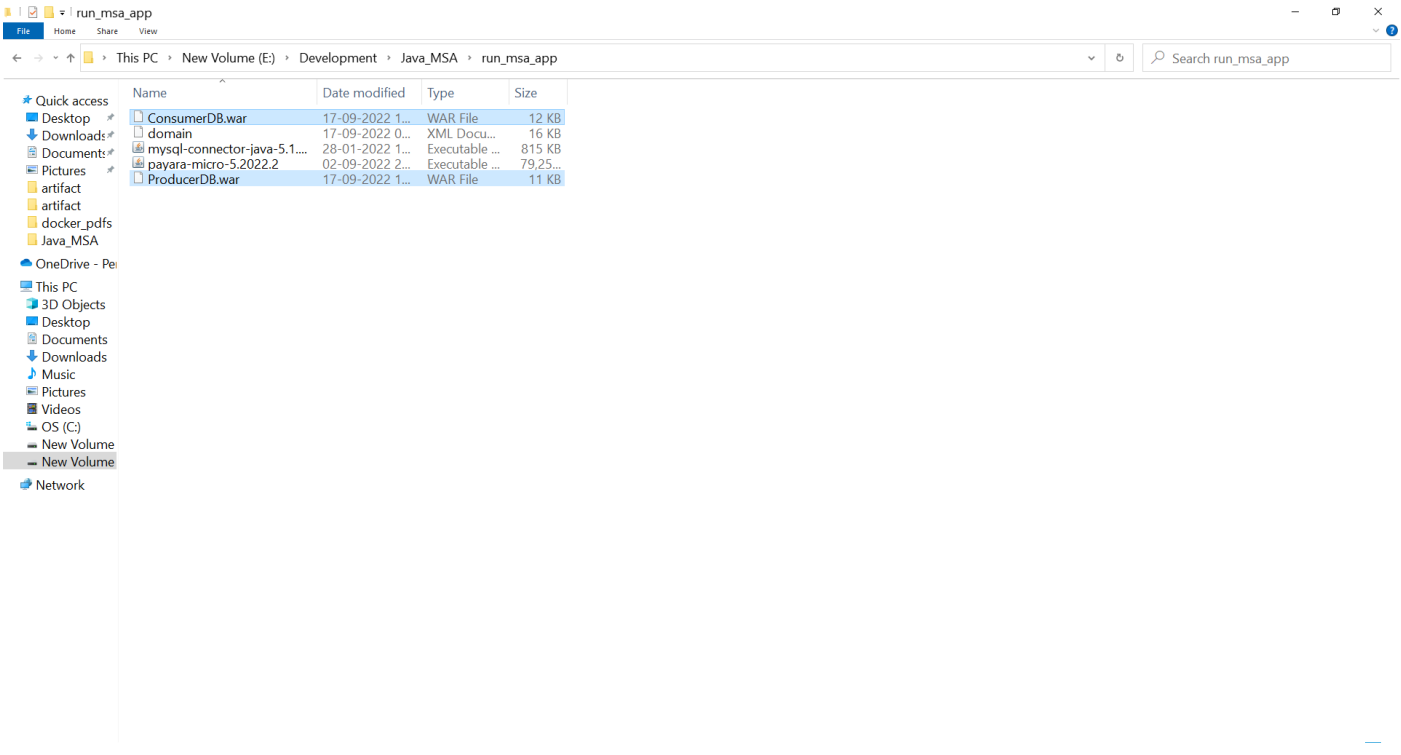
In my case output would be



40) Go to your Producer App Directory->artifact and copy war file and paste that war file in the blank folder. Perform same for Consumer App copy war file and paste that war file in the blank folder. Blank folder that you've created where you've domain.xml, mysql-connector and payara-microprofile jar file.




artifact
File Home Share View
This PC › New Volume (E:) › Development › Java_MSA › ConsumerDB › artifact
Search artifact
Quick access
Desktop
Downloads
Documents
Pictures
artifact
artifact
docker_pdfs
Java_MSA
OneDrive - Pe
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
New Volume
New Volume
Network
Name
Date modified
Type
Size
ConsumerDB.war
17-09-2022 1...
Open with
Share with Skype
Edit with IntelliJ IDEA Community Edition
Edit with PyCharm Community Edition
Move to OneDrive
Edit with Notepad++
Scan
Shred
Share
Add to archive...
Add to "ConsumerDB.rar"
Compress and email...
Compress to "ConsumerDB.rar" and email
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties

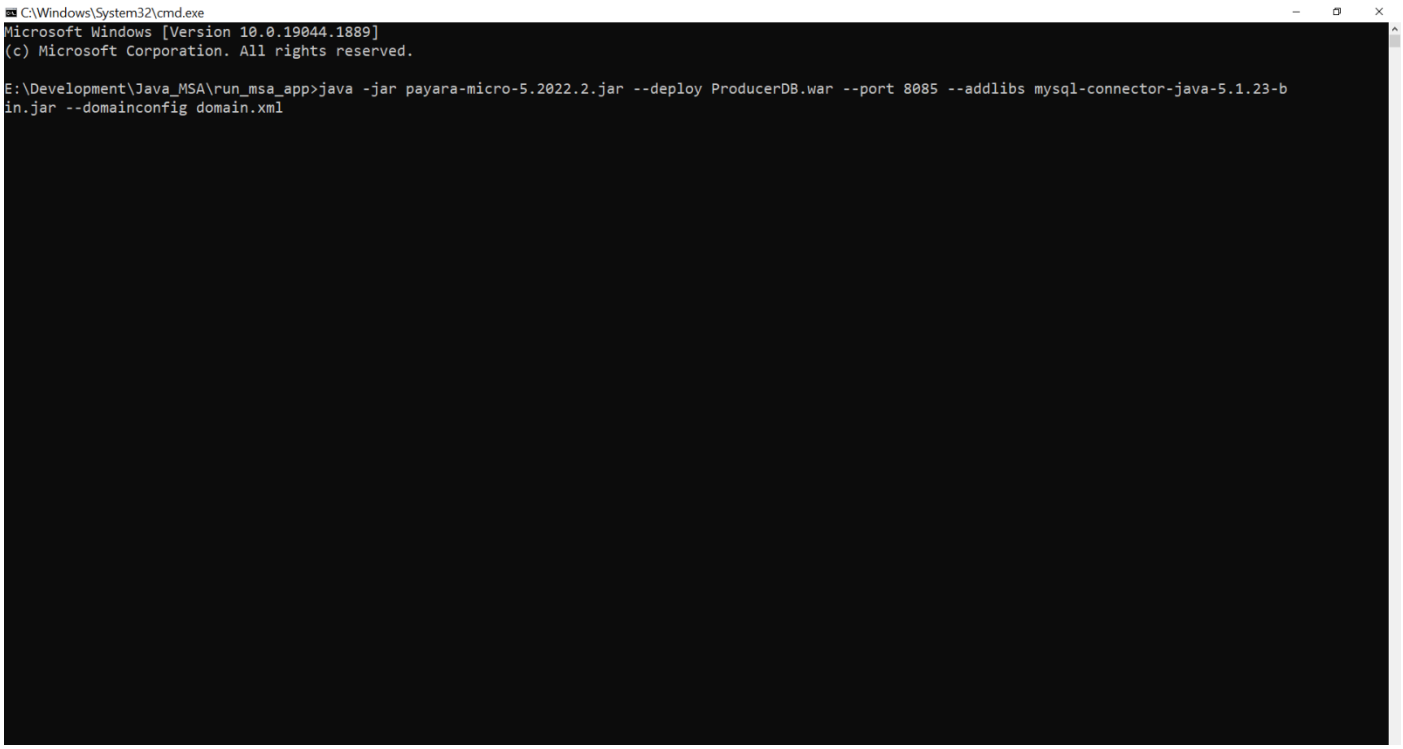

run_msa_app
File Home Share View
This PC › New Volume (E:) › Development › Java_MSA › run_msa_app
Search run_msa_app
Quick access
Desktop
Downloads
Documents
Pictures
artifact
artifact
docker_pdfs
Java_MSA
OneDrive - Pe
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
New Volume
New Volume
Network
Name
Date modified
Type
Size
ConsumerDB.war
17-09-2022 1...
WAR File
12 KB
domain
17-09-2022 0...
XML Docu...
16 KB
mysql-connector-java-5.1....
28-01-2022 1...
Executable ...
815 KB
payara-micro-5.2022.2
02-09-2022 2...
Executable ...
79,25...
ProducerDB.war
17-09-2022 1...
WAR File
11 KB

41) Open the terminal/Command Prompt on that location and Let's deploy the Producer App first. Before careful while writing the command.

In My case command :

java -jar payara-micro-5.2022.2.jar --deploy ***ProducerDB.war*** --port ***8085*** --addlibs mysql-connector-java-5.1.23-bin.jar --domainconfig domain.xml

I gonna run this Producer app on Port Number 8085, You have to run the Producer App on that port number which you mentioned in step no. 34

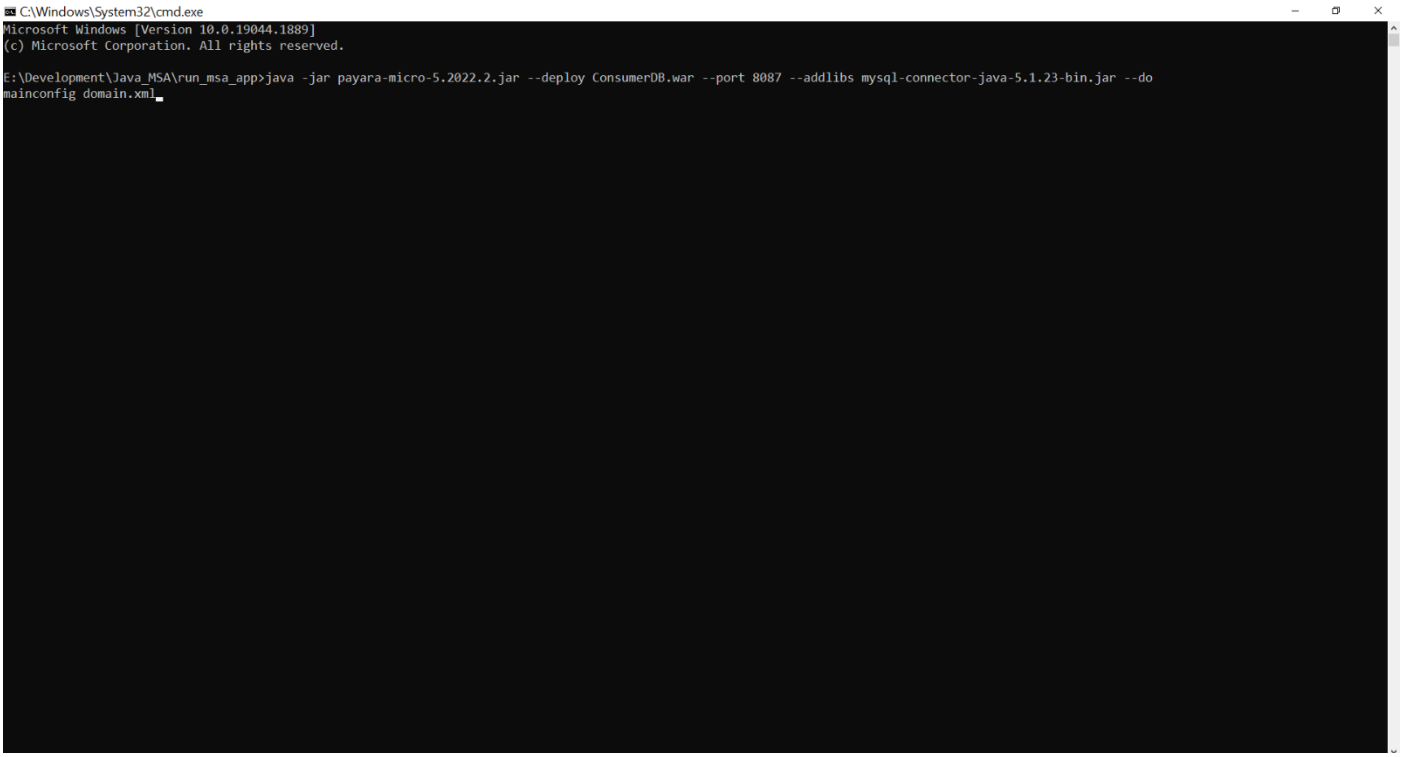


42) Wait and Watch, Open another terminal/Command prompt don't close earlier terminal/Command Prompt. And fire the same command again for consumer app

In my case Command would

java -jar payara-micro-5.2022.2.jar --deploy **ConsumerDB.war** --port **8087** --addlibs mysql-connector-java-5.1.23-bin.jar --domainconfig domain.xml

Here You can select any port number for Consumer App



Alright Both terminal should give some script as output when it is done Open the browser and Enter the following urls

For Producer App my URL would : http://localhost:8085/ProducerDB/rest/example/student

It should give Unauthorized status

URL : http://localhost:port_number/your_project_name/rest/example/your_path

For consumer Enter the URL : http://localhost:8087/ConsumerDB/app

URL : http://localhost:port_number/project_name/url_of_your_servlet

Happy Coding!