WEB TECHNOLOGIES LAB

Name: M. Sri Varsha
Roll no.: 19131A05D3
Class: CSE-3

WEEK-6

Aim:

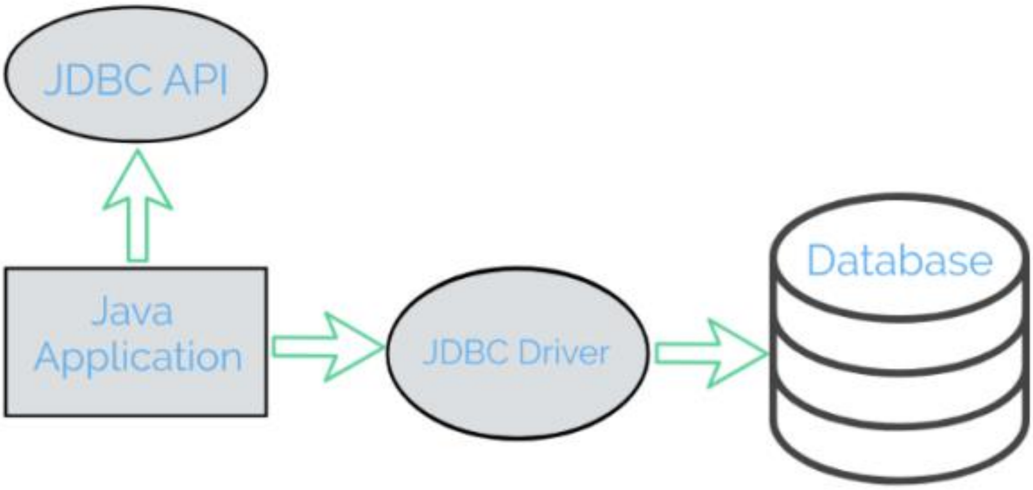
Create necessary tables for the application chosen using JDBC and establish database connectivity.

Scenario 1: Establish the connectivity using JDBC drivers.
Scenario 2: Use create and select statements.
Scenario 3: Use insert, update and delete queries.

Java Database Connectivity (JDBC):

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is part of JavaSE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database. There are four types of JDBC drivers.

i) JDBC-ODBC Bridge Driver
ii) Native Driver
iii) Network Protocol Driver
iv) Thin Driver



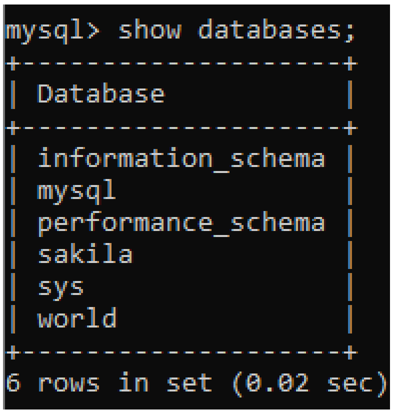
6.1 Java Database Connectivity with MySQL:

To connect Java application with the MySQL database, we need to follow 4 following steps.

1. **Driver class:** The driver class for the MySQL database is **com.mysql.jdbc.Driver**.
2. **Connection URL:** The connection URL for the mysql database is **jdbc:mysql://localhost:3306/sonoo** where **jdbc** is the **API**, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and sonoo is the database name.
3. **Username**: The default username for the mysql database is **root**
4. **Password**: It is the password given by the user at the time of installing the mysql database.

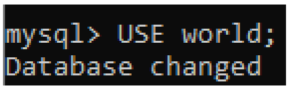
After the installation of MySQL, we have to select a database in which we will work.

The list of databases can be displayed using the **show database;** command.



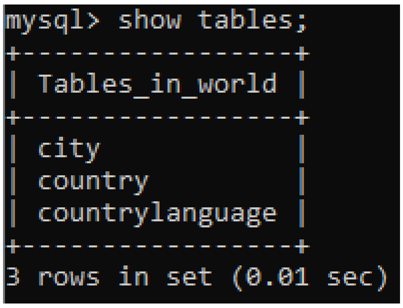
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sakila             |
| sys                |
| world              |
+--------------------+
6 rows in set (0.02 sec)
```

To select a database use the **use databasename;** command. Here world database is in use.



```
mysql> USE world;
Database changed
```

To view the list of tables in the world database use the command **show tables;**



```
mysql> show tables;
+-----------------+
| Tables_in_world |
+-----------------+
| city            |
| country         |
| countrylanguage |
+-----------------+
3 rows in set (0.01 sec)
```

6.2 Setup for JDBC using VS code:

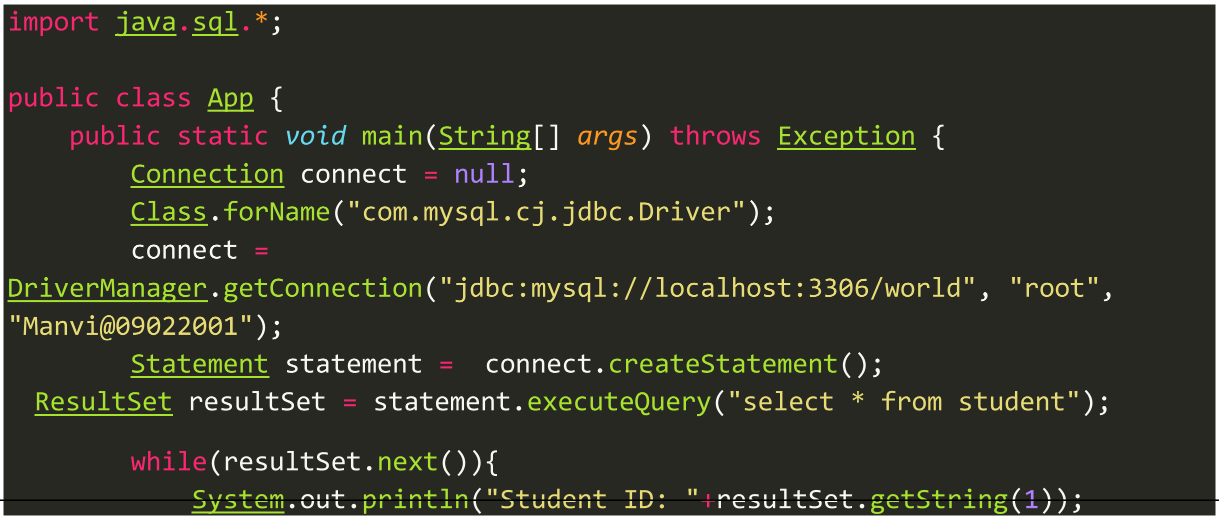
Step-1: First we need to check whether the JDK is installed or not if not, we need to install.

Step-2: After installing JDK open the visual studio code and create the new java project by following steps

- Open command palette and search for java project
- Select No Build tools and choose the location where you have to save the application

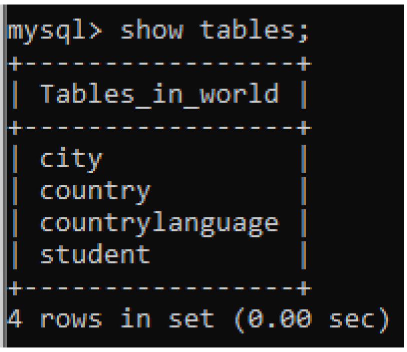
Step-3: After creating the Java project add **mysqlconnector.jar** file to the reference library (that you can find out at the Java project that was appear at the left bottom of the vs code).

i) PROGRAM for creating a table using create and select statement.



```
import java.sql.*;

public class App {
    public static void main(String[] args) throws Exception {
        Connection connect = null;
        Class.forName("com.mysql.cj.jdbc.Driver");
        connect = 
DriverManager.getConnection("jdbc:mysql://localhost:3306/world", "root", 
"Manvi@09022001");
        Statement statement =  connect.createStatement();
  ResultSet resultSet = statement.executeQuery("select * from student");

        while(resultSet.next()){
            System.out.println("Student ID: "+resultSet.getString(1));
            System.out.println("Student Name: "+resultSet.getString(2));
            System.out.println("Student Department: "+resultSet.getString(3));
            System.out.println();
        }
        resultSet.close();
        statement.close();
        connect.close();
    }
}
```

OUTPUT:

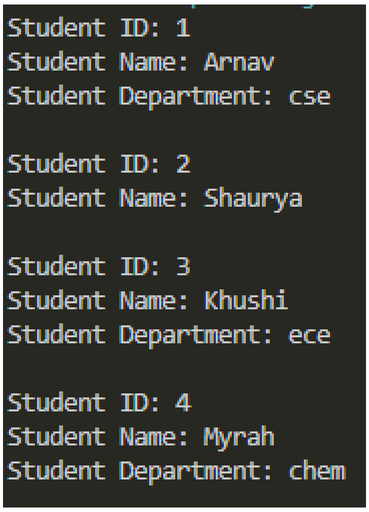


```
mysql> show tables;
+-----------------+
| Tables_in_world |
+-----------------+
| city            |
| country         |
| countrylanguage |
| student         |
+-----------------+
4 rows in set (0.00 sec)
```

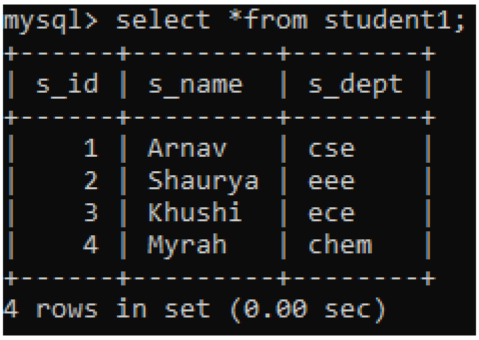
ii) PROGRAM using insert statement.

```
import java.sql.*;

public class App {
    public static void main(String[] args) throws Exception {
        Connection connect = null;
        Class.forName("com.mysql.cj.jdbc.Driver");
        connect =
DriverManager.getConnection("jdbc:mysql://localhost:3306/world", "root",
"Manvi@09022001");
        Statement statement =  connect.createStatement();
        // insert
        statement.execute("create table student1 (s_id integer, s_name
varchar(20), s_dept varchar(5) )");
        statement.execute("insert into student1 values(01,'Arnav','cse')");
        statement.execute("insert into student1 values(02,'Shaurya','eee')");
        statement.execute("insert into student1 values(03,'Khushi','ece')");
        statement.execute("insert into student1 values(04,'Myrah','chem')");
ResultSet resultSet = statement.executeQuery("select * from student1");
        while(resultSet.next()){
            System.out.println("Student ID: "+resultSet.getString(1));
            System.out.println("Student Name: "+resultSet.getString(2));
            System.out.println("Student Department: "+resultSet.getString(3));
            System.out.println();
        }
        resultSet.close();
        statement.close();
        connect.close();
    }
}
```

OUTPUT:

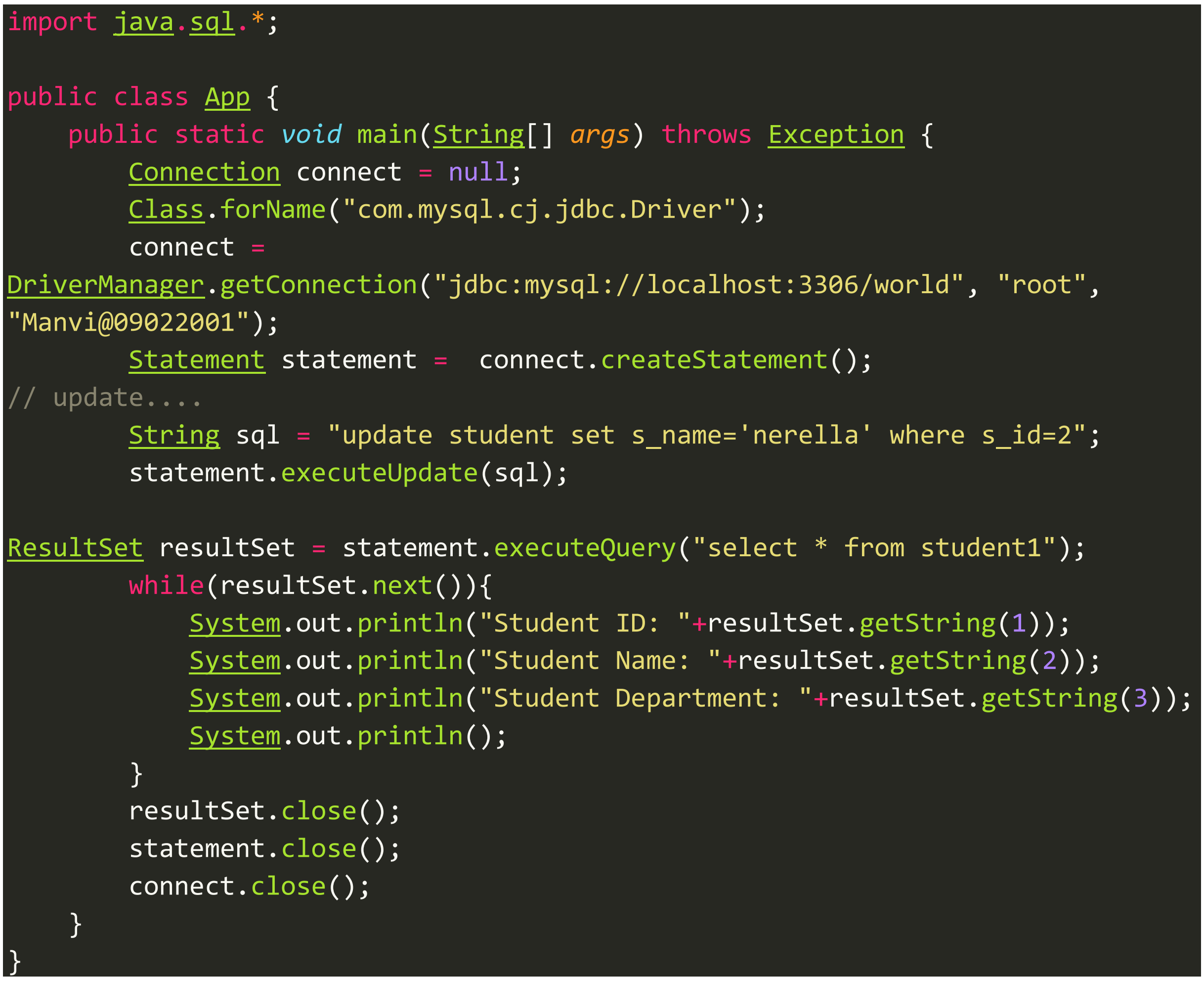


```
Student ID: 1
Student Name: Arnav
Student Department: cse

Student ID: 2
Student Name: Shaurya

Student ID: 3
Student Name: Khushi
Student Department: ece

Student ID: 4
Student Name: Myrah
Student Department: chem
```



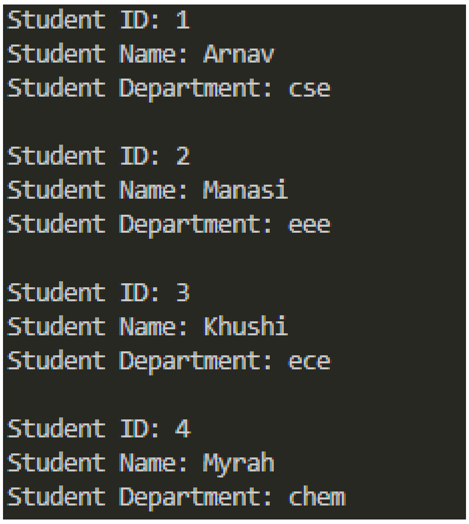
```
mysql> select *from student1;
+------+---------+--------+
| s_id | s_name  | s_dept |
+------+---------+--------+
|    1 | Arnav   | cse    |
|    2 | Shaurya | eee    |
|    3 | Khushi  | ece    |
|    4 | Myrah   | chem   |
+------+---------+--------+
4 rows in set (0.00 sec)
```

iii) PROGRAM using update statement.

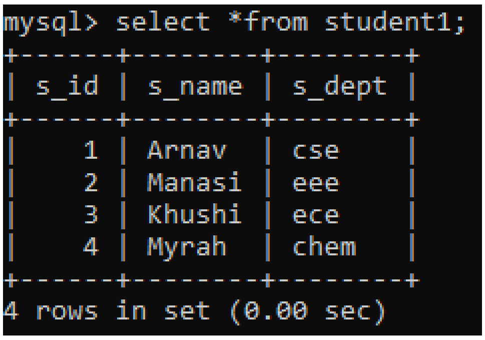


```java
import java.sql.*;

public class App {
    public static void main(String[] args) throws Exception {
        Connection connect = null;
        Class.forName("com.mysql.cj.jdbc.Driver");
        connect =
DriverManager.getConnection("jdbc:mysql://localhost:3306/world", "root",
"Manvi@09022001");
        Statement statement =  connect.createStatement();
// update....
        String sql = "update student set s_name='nerella' where s_id=2";
        statement.executeUpdate(sql);

ResultSet resultSet = statement.executeQuery("select * from student1");
        while(resultSet.next()){
            System.out.println("Student ID: "+resultSet.getString(1));
            System.out.println("Student Name: "+resultSet.getString(2));
            System.out.println("Student Department: "+resultSet.getString(3));
            System.out.println();
        }
        resultSet.close();
        statement.close();
        connect.close();
    }
}
```

OUTPUT:

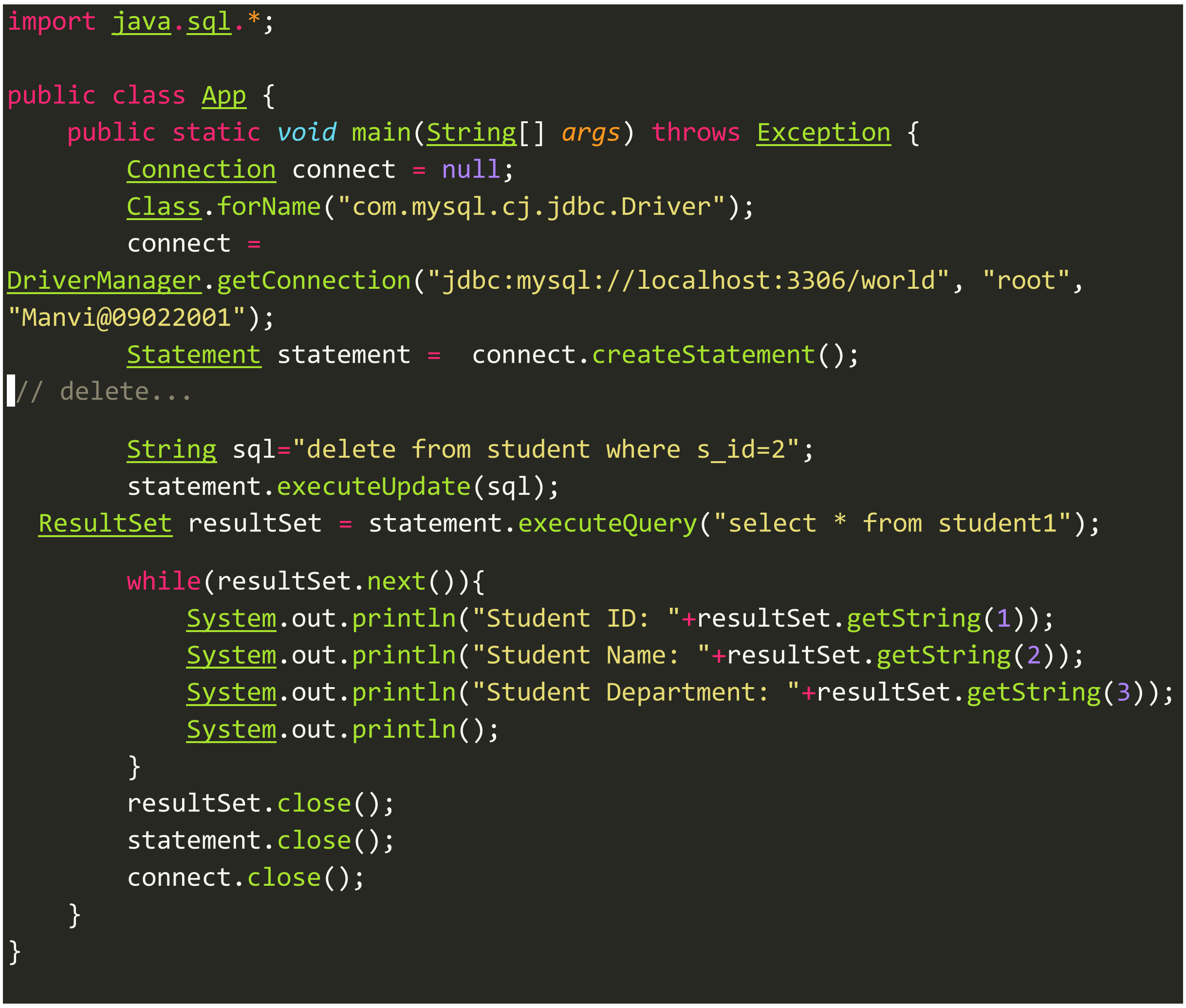


```
Student ID: 1
Student Name: Arnav
Student Department: cse

Student ID: 2
Student Name: Manasi
Student Department: eee

Student ID: 3
Student Name: Khushi
Student Department: ece

Student ID: 4
Student Name: Myrah
Student Department: chem
```



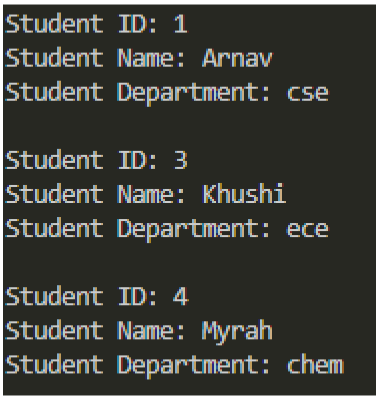
```
mysql> select *from student1;
+------+--------+--------+
| s_id | s_name | s_dept |
+------+--------+--------+
|    1 | Arnav  | cse    |
|    2 | Manasi | eee    |
|    3 | Khushi | ece    |
|    4 | Myrah  | chem   |
+------+--------+--------+
4 rows in set (0.00 sec)
```

d) PROGRAM using delete statement

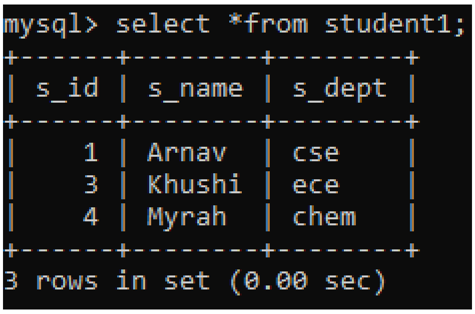


```
import java.sql.*;

public class App {
    public static void main(String[] args) throws Exception {
        Connection connect = null;
        Class.forName("com.mysql.cj.jdbc.Driver");
        connect =
DriverManager.getConnection("jdbc:mysql://localhost:3306/world", "root",
"Manvi@09022001");
        Statement statement =  connect.createStatement();
// delete...

        String sql="delete from student where s_id=2";
        statement.executeUpdate(sql);
  ResultSet resultSet = statement.executeQuery("select * from student1");

        while(resultSet.next()){
            System.out.println("Student ID: "+resultSet.getString(1));
            System.out.println("Student Name: "+resultSet.getString(2));
            System.out.println("Student Department: "+resultSet.getString(3));
            System.out.println();
        }
        resultSet.close();
        statement.close();
        connect.close();
    }
}
```

OUTPUT:



```
Student ID: 1
Student Name: Arnav
Student Department: cse

Student ID: 3
Student Name: Khushi
Student Department: ece

Student ID: 4
Student Name: Myrah
Student Department: chem
```



```
mysql> select *from student1;
+------+--------+--------+
| s_id | s_name | s_dept |
+------+--------+--------+
|    1 | Arnav  | cse    |
|    3 | Khushi | ece    |
|    4 | Myrah  | chem   |
+------+--------+--------+
3 rows in set (0.00 sec)
```

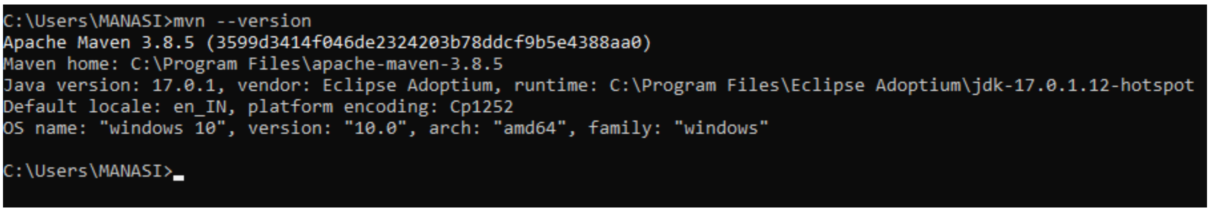
WEEK-7&8

Create the necessary servlets for the application chosen

- Check the authenticity of the login details with the information available in the database. If he is a valid user it must redirect to site resources otherwise it should stay in same page with an invalid username/password message.
- Insert the details of the registration page into the database. If registration is successful it must display “Registration is successful”.
- Update the password field in the database.

Step-1: Download Maven

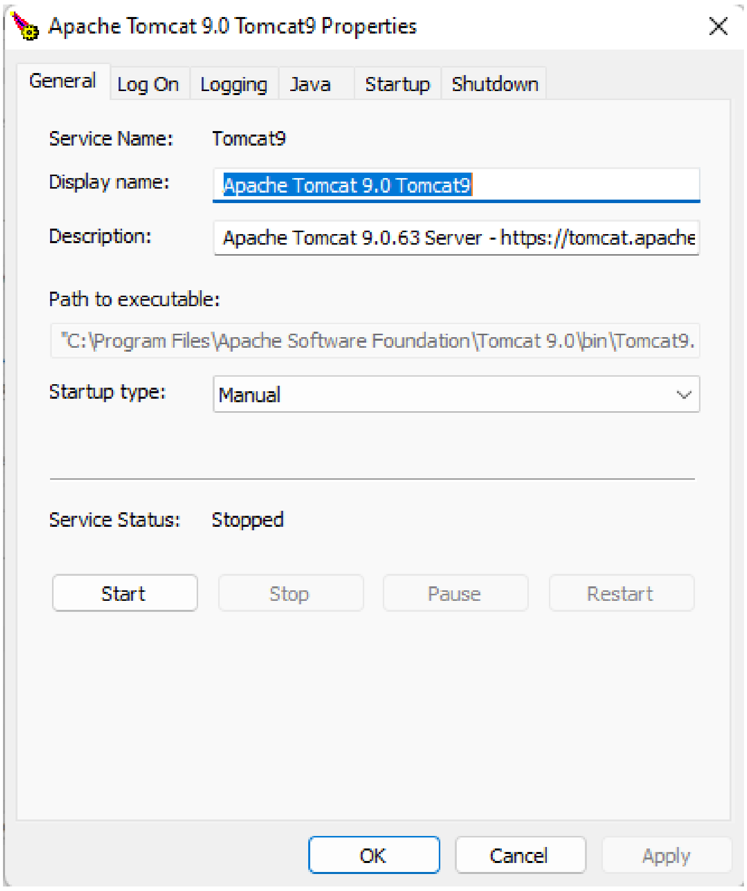
- Download and install the Maven from https://maven.apache.org/download.cgi
- After installing set the **bin** path in the Environment variables.
- To check whether Maven is installed successful or not open command prompt type command as **mvn –version**



```
C:\Users\MANASI>mvn --version
Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)
Maven home: C:\Program Files\apache-maven-3.8.5
Java version: 17.0.1, vendor: Eclipse Adoptium, runtime: C:\Program Files\Eclipse Adoptium\jdk-17.0.1.12-hotspot
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\MANASI>_
```

Step-2: Download Apache Tomcat

- Download and install the Apache Tomcat from https://tomcat.apache.org/download-90.cgi and open the application



Apache Tomcat 9.0 Tomcat9 Properties

Service Name: Tomcat9

Display name: Apache Tomcat 9.0 Tomcat9

Description: Apache Tomcat 9.0.63 Server - https://tomcat.apache

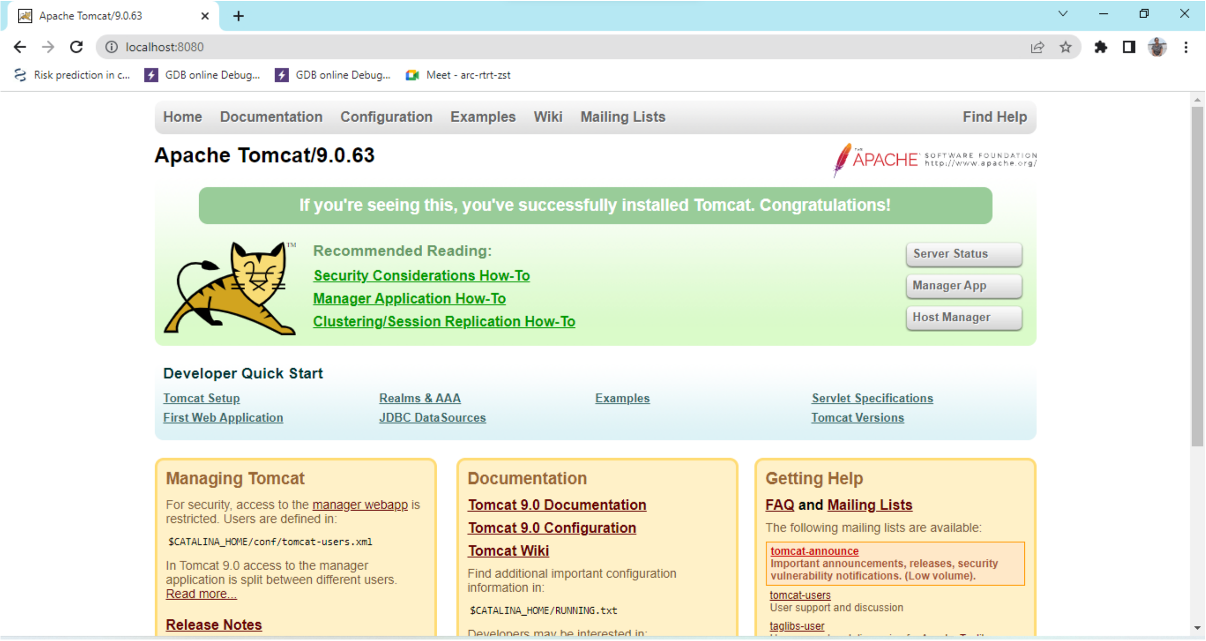
Path to executable:

"C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin\Tomcat9.

Startup type: Manual

Service Status: Stopped

- Start the server and should remember Startup type should be Manual. Now open the browser and type **localhost:8080**



Apache Tomcat/9.0.63

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

Security Considerations How-To

Manager Application How-To

Clustering/Session Replication How-To

Developer Quick Start

Tomcat Setup | Realms & AAA | Examples | Servlet Specifications

First Web Application | JDBC DataSources | Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes

Documentation

Tomcat 9.0 Documentation

Tomcat 9.0 Configuration

Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Getting Help

FAQ and Mailing Lists

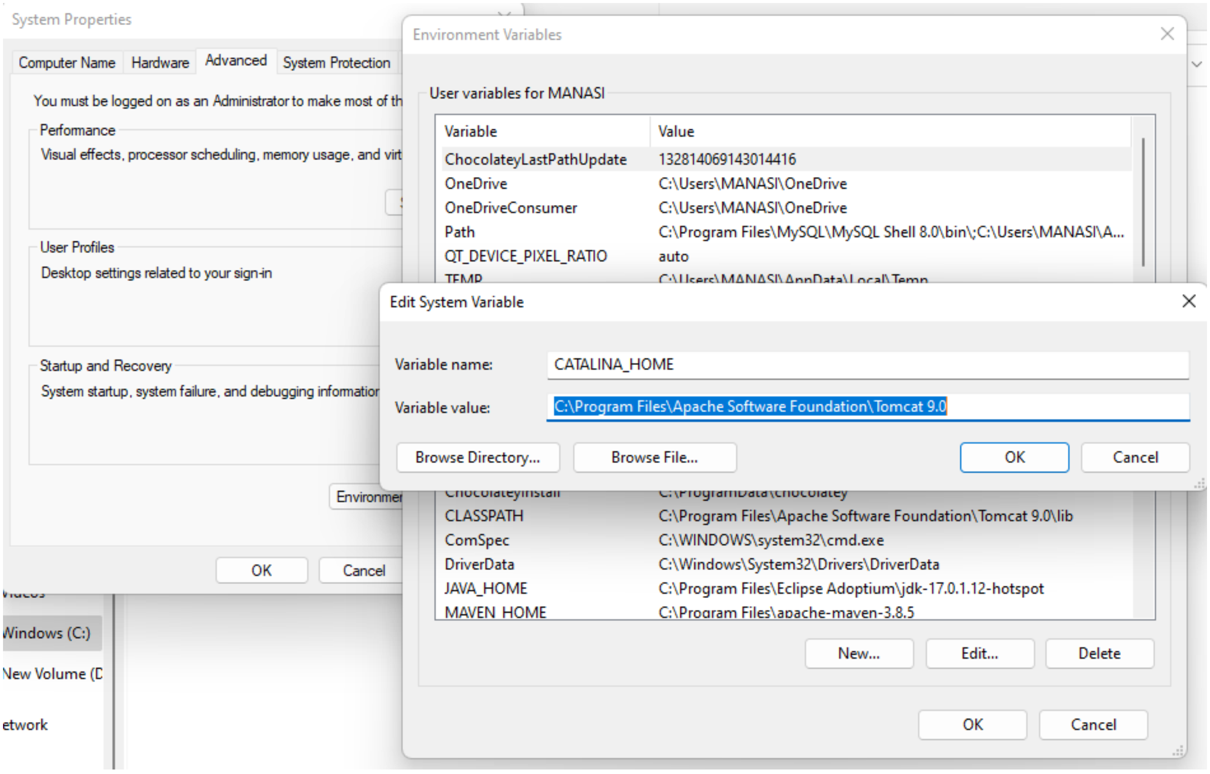
The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

taglibs-user

- Next copy the Tomcat location where you have installed it. Now open Environment Variables and create New System Variable where variable name should be **CATALINA_HOME** and variable value is Tomcat location that you already copied.



You must be logged on as an Administrator to make most of th

Performance

Visual effects, processor scheduling, memory usage, and virt

User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

User variables for MANASI

Variable	Value
ChocolateyLastPathUpdate	132814069143014416
OneDrive	C:\Users\MANASI\OneDrive
OneDriveConsumer	C:\Users\MANASI\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\MANASI\A...
QT_DEVICE_PIXEL_RATIO	auto
TEMP	C:\Users\MANASI\AppData\Local\Temp

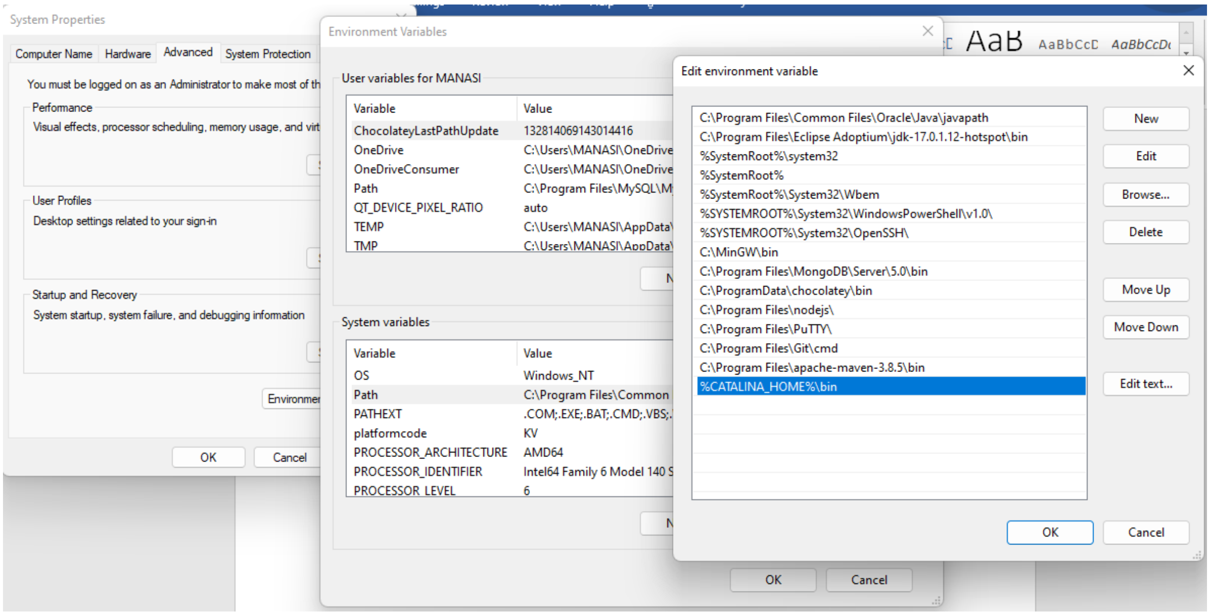
Edit System Variable

Variable name: CATALINA_HOME

Variable value: C:\Program Files\Apache Software Foundation\Tomcat 9.0

Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
CLASSPATH	C:\Program Files\Apache Software Foundation\Tomcat 9.0\lib
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Eclipse Adoptium\jdk-17.0.1.12-hotspot
MAVEN_HOME	C:\Program Files\apache-maven-3.8.5

- Now System variable click on **path** and type **%CATALINA_HOME%\bin**



You must be logged on as an Administrator to make most of th

Performance

Visual effects, processor scheduling, memory usage, and virt

User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

User variables for MANASI

Variable	Value
ChocolateyLastPathUpdate	132814069143014416
OneDrive	C:\Users\MANASI\OneDrive
OneDriveConsumer	C:\Users\MANASI\OneDrive
Path	C:\Program Files\MySQL\M
QT_DEVICE_PIXEL_RATIO	auto
TEMP	C:\Users\MANASI\AppData\
TMP	C:\Users\MANASI\AppData\

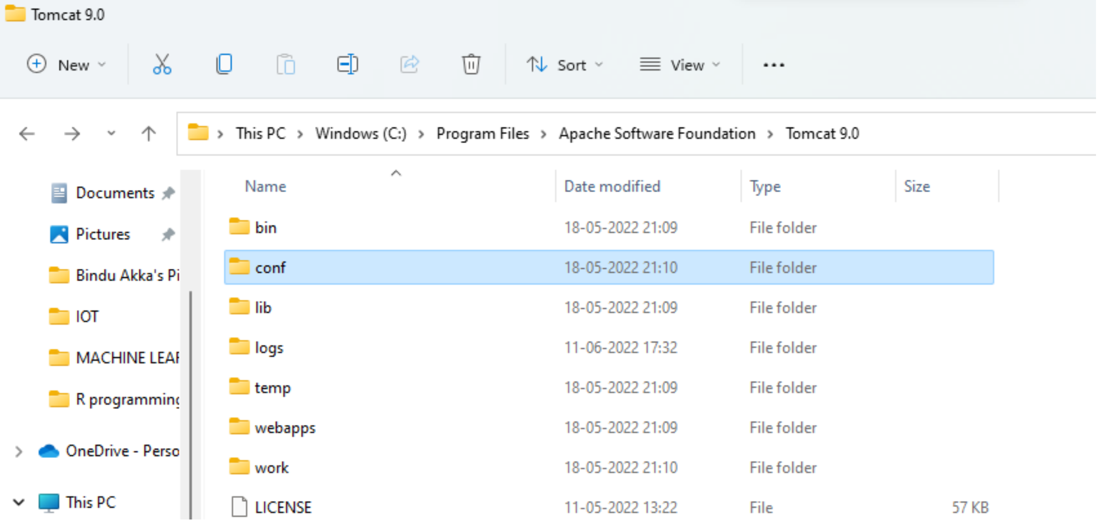
System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files\Common
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.
platformcode	KV
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 140 S
PROCESSOR_LEVEL	6

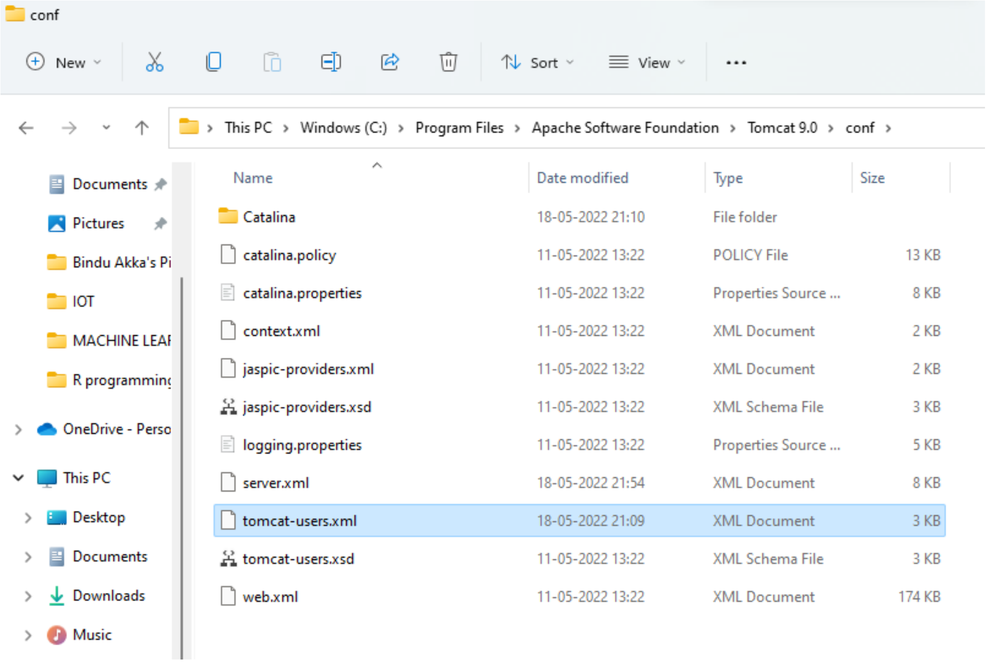
Edit environment variable

```
C:\Program Files\Common Files\Oracle\Java\javapath
C:\Program Files\Eclipse Adoptium\jdk-17.0.1.12-hotspot\bin
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\MinGW\bin
C:\Program Files\MongoDB\Server\5.0\bin
C:\ProgramData\chocolatey\bin
C:\Program Files\nodejs\
C:\Program Files\PuTTY\
C:\Program Files\Git\cmd
C:\Program Files\apache-maven-3.8.5\bin
%CATALINA_HOME%\bin
```

- Now you have to open the **conf** folder that was down to the **bin** folder and then open the **tomcat-users.xml** file that was present inside the **conf** folder.

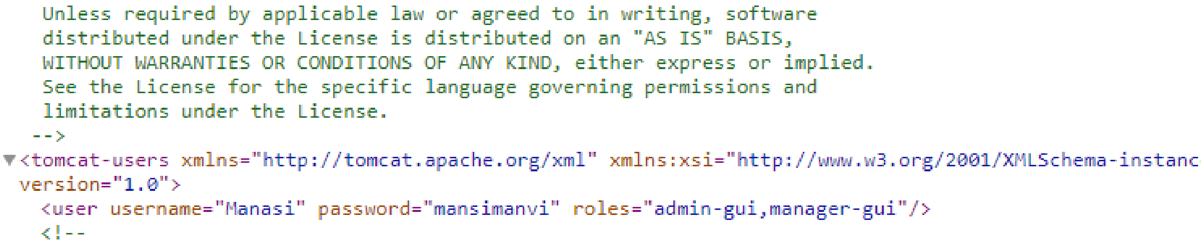


Name	Date modified	Type	Size
bin	18-05-2022 21:09	File folder	
conf	18-05-2022 21:10	File folder	
lib	18-05-2022 21:09	File folder	
logs	11-06-2022 17:32	File folder	
temp	18-05-2022 21:09	File folder	
webapps	18-05-2022 21:09	File folder	
work	18-05-2022 21:10	File folder	
LICENSE	11-05-2022 13:22	File	57 KB



Name	Date modified	Type	Size
Catalina	18-05-2022 21:10	File folder	
catalina.policy	11-05-2022 13:22	POLICY File	13 KB
catalina.properties	11-05-2022 13:22	Properties Source ...	8 KB
context.xml	11-05-2022 13:22	XML Document	2 KB
jaspic-providers.xml	11-05-2022 13:22	XML Document	2 KB
jaspic-providers.xsd	11-05-2022 13:22	XML Schema File	3 KB
logging.properties	11-05-2022 13:22	Properties Source ...	5 KB
server.xml	18-05-2022 21:54	XML Document	8 KB
tomcat-users.xml	18-05-2022 21:09	XML Document	3 KB
tomcat-users.xsd	11-05-2022 13:22	XML Schema File	3 KB
web.xml	11-05-2022 13:22	XML Document	174 KB

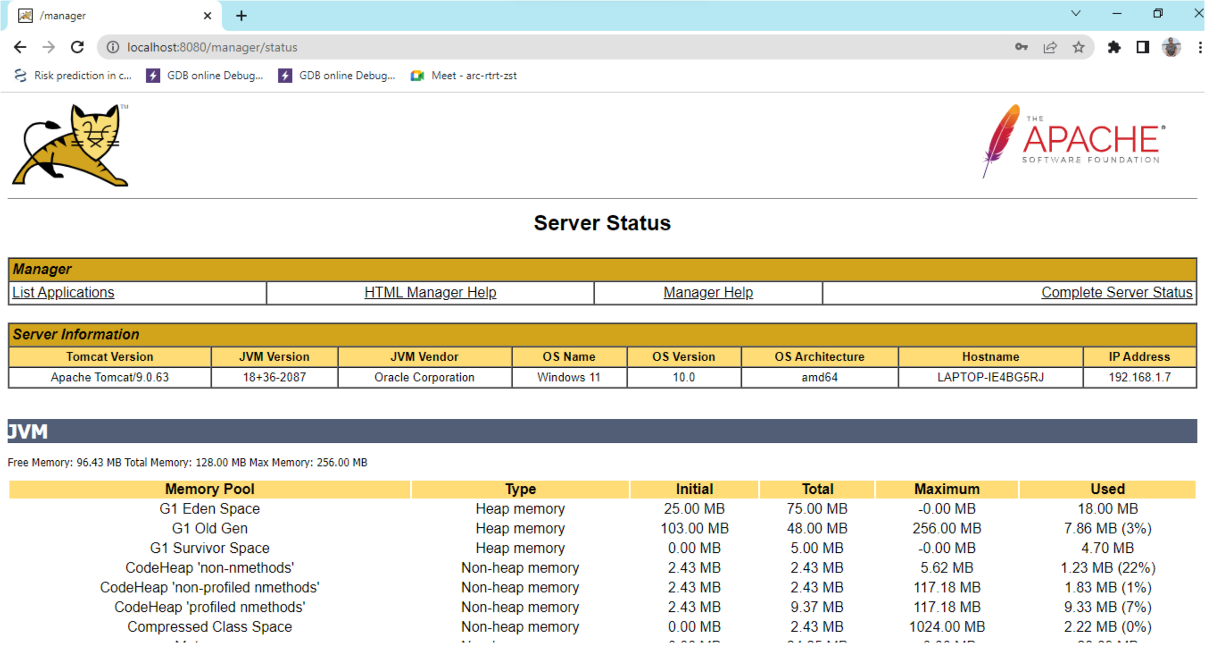
Now you have set the **username** and **password** by typing the code as below in the tomcat-users.xml file.



```
    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
  -->
▼<tomcat-users xmlns="http://tomcat.apache.org/xml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instanc
  version="1.0">
    <user username="Manasi" password="mansimanvi" roles="admin-gui,manager-gui"/>
    <!--
```

You have to add this code at the end of the file.

- Now open the Tomcat server in the browser you can identify the **server status** at top right side of the web page then click on it and give your **username** and **password**.



Server Status

Manager			
List Applications	HTML Manager Help	Manager Help	Complete Server Status

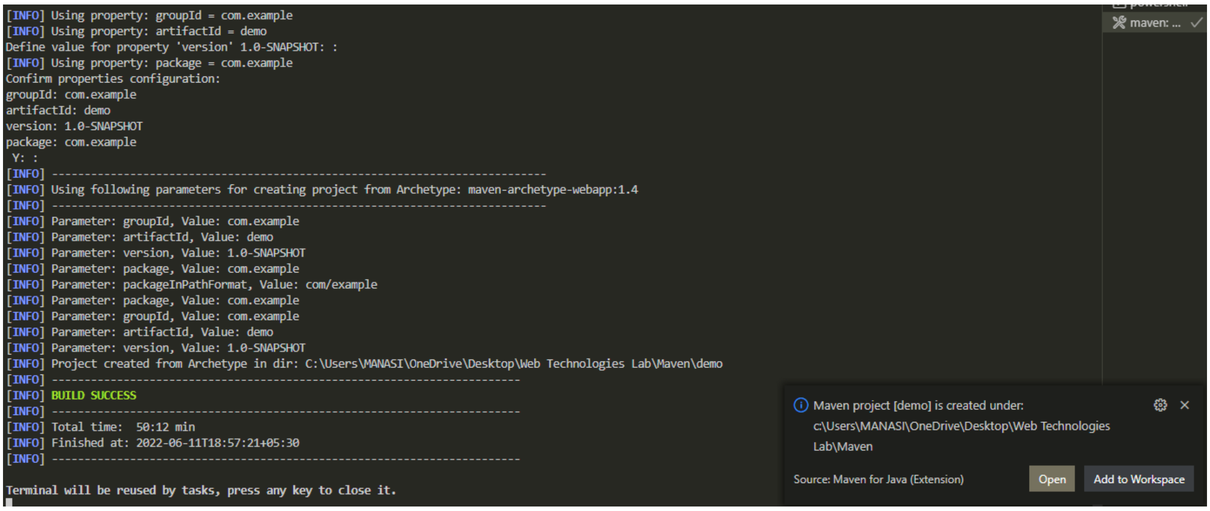
Server Information							
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/9.0.63	18+36-2087	Oracle Corporation	Windows 11	10.0	amd64	LAPTOP-IE4BG5RJ	192.168.1.7

JVM

Free Memory: 96.43 MB Total Memory: 128.00 MB Max Memory: 256.00 MB

Memory Pool	Type	Initial	Total	Maximum	Used
G1 Eden Space	Heap memory	25.00 MB	75.00 MB	-0.00 MB	18.00 MB
G1 Old Gen	Heap memory	103.00 MB	48.00 MB	256.00 MB	7.86 MB (3%)
G1 Survivor Space	Heap memory	0.00 MB	5.00 MB	-0.00 MB	4.70 MB
CodeHeap 'non-nmethods'	Non-heap memory	2.43 MB	2.43 MB	5.62 MB	1.23 MB (22%)
CodeHeap 'non-profiled nmethods'	Non-heap memory	2.43 MB	2.43 MB	117.18 MB	1.83 MB (1%)
CodeHeap 'profiled nmethods'	Non-heap memory	2.43 MB	9.37 MB	117.18 MB	9.33 MB (7%)
Compressed Class Space	Non-heap memory	0.00 MB	2.43 MB	1024.00 MB	2.22 MB (0%)

After successfully signing in you may get this. This is all about configuring the Tomcat server.



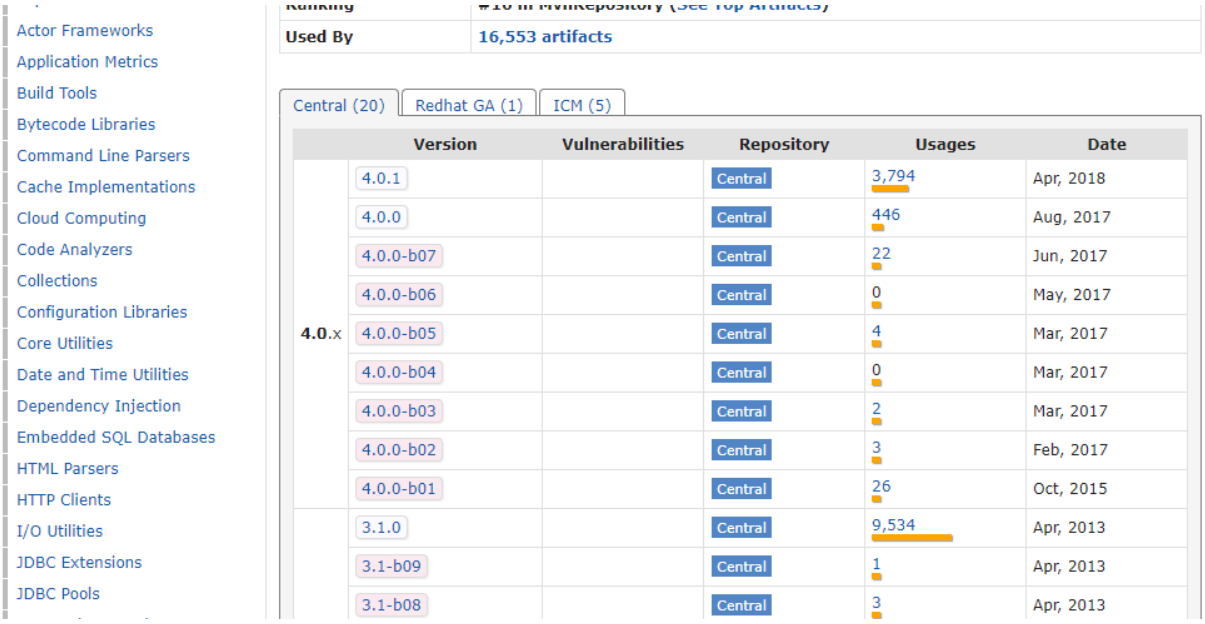
```
[INFO] Using property: groupId = com.example
[INFO] Using property: artifactId = demo
Define value for property 'version' 1.0-SNAPSHOT: :
[INFO] Using property: package = com.example
Confirm properties configuration:
groupId: com.example
artifactId: demo
version: 1.0-SNAPSHOT
package: com.example
 Y: :
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Archetype: maven-archetype-webapp:1.4
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: com.example
[INFO] Parameter: artifactId, Value: demo
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.example
[INFO] Parameter: packageInPathFormat, Value: com/example
[INFO] Parameter: package, Value: com.example
[INFO] Parameter: groupId, Value: com.example
[INFO] Parameter: artifactId, Value: demo
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\Users\MANASI\OneDrive\Desktop\Web Technologies Lab\Maven\demo
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  50:12 min
[INFO] Finished at: 2022-06-11T18:57:21+05:30
[INFO] ------------------------------------------------------------------------

Terminal will be reused by tasks, press any key to close it.
```

Maven project [demo] is created under: c:\Users\MANASI\OneDrive\Desktop\Web Technologies Lab\Maven

Source: Maven for Java (Extension)

Step-4: Servlet API

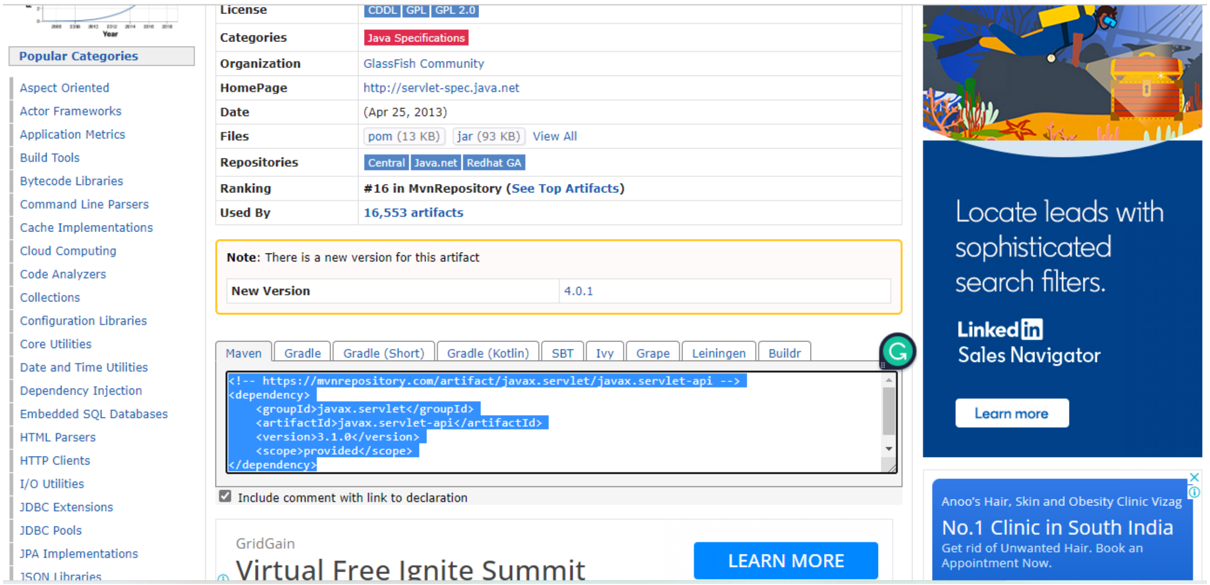
Open the browser and use this link to get the java servlet API https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api and then choose the dependency **3.1.0**



Used By	16,553 artifacts

	Version	Vulnerabilities	Repository	Usages	Date
4.0.x	4.0.1		Central	3,794	Apr, 2018
	4.0.0		Central	446	Aug, 2017
	4.0.0-b07		Central	22	Jun, 2017
	4.0.0-b06		Central	0	May, 2017
	4.0.0-b05		Central	4	Mar, 2017
	4.0.0-b04		Central	0	Mar, 2017
	4.0.0-b03		Central	2	Mar, 2017
	4.0.0-b02		Central	3	Feb, 2017
	4.0.0-b01		Central	26	Oct, 2015
	3.1.0		Central	9,534	Apr, 2013
	3.1-b09		Central	1	Apr, 2013
	3.1-b08		Central	3	Apr, 2013

After opening the 3.1.0 dependency copy the code.



License	CDDL GPL GPL 2.0
Categories	Java Specifications
Organization	GlassFish Community
HomePage	http://servlet-spec.java.net
Date	(Apr 25, 2013)
Files	pom (13 KB) jar (93 KB) View All
Repositories	Central Java.net Redhat GA
Ranking	#16 in MvnRepository (See Top Artifacts)
Used By	16,553 artifacts

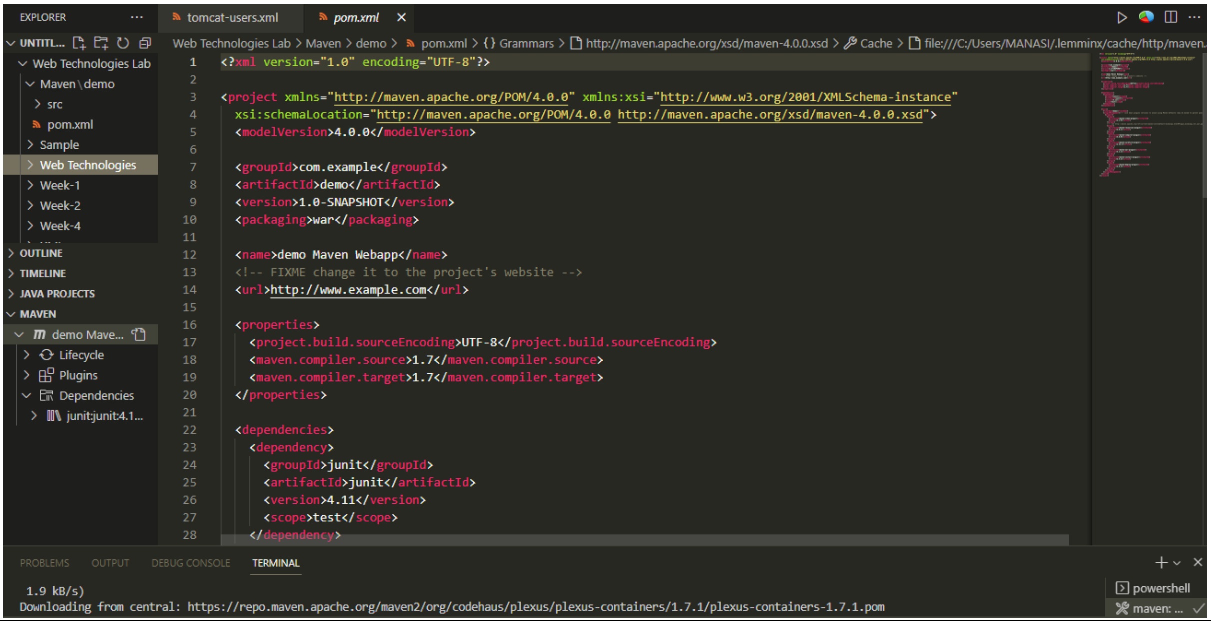
Note: There is a new version for this artifact

New Version	4.0.1

```
<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
    <scope>provided</scope>
</dependency>
```

☑ Include comment with link to declaration

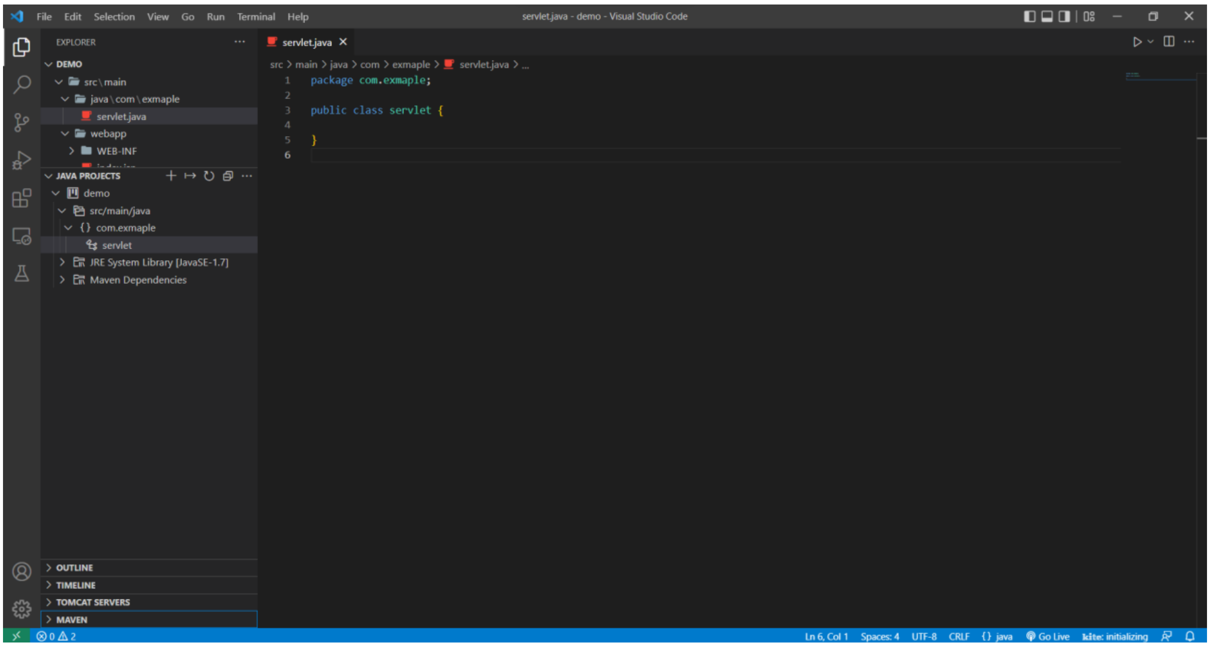
Now come to vs code open the project that was created earlier you may get like this.



```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>demo</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>demo Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
```

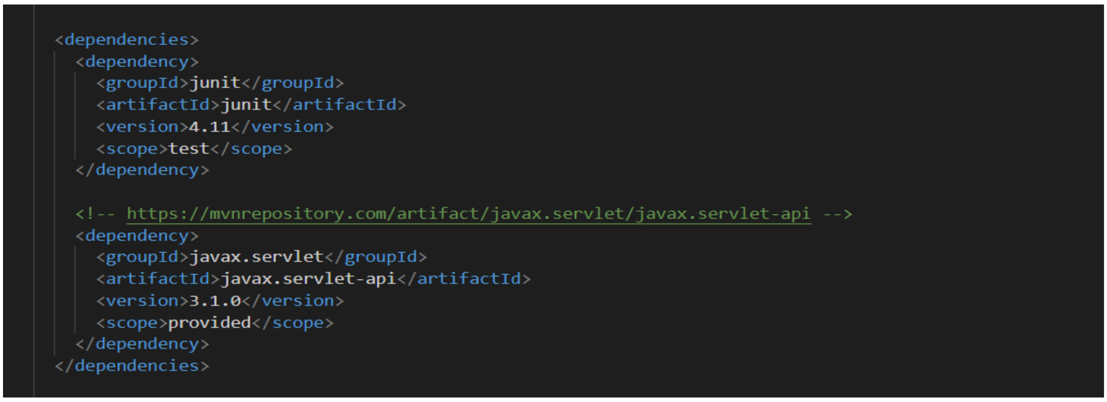
```
 1.9 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-containers/1.7.1/plexus-containers-1.7.1.pom
```

Now we have create **java** folder in the main folder that was in src folder then click on java projects in left panel and then right click on src/main/java then click new package



```
package com.exmaple;

public class servlet {

}
```

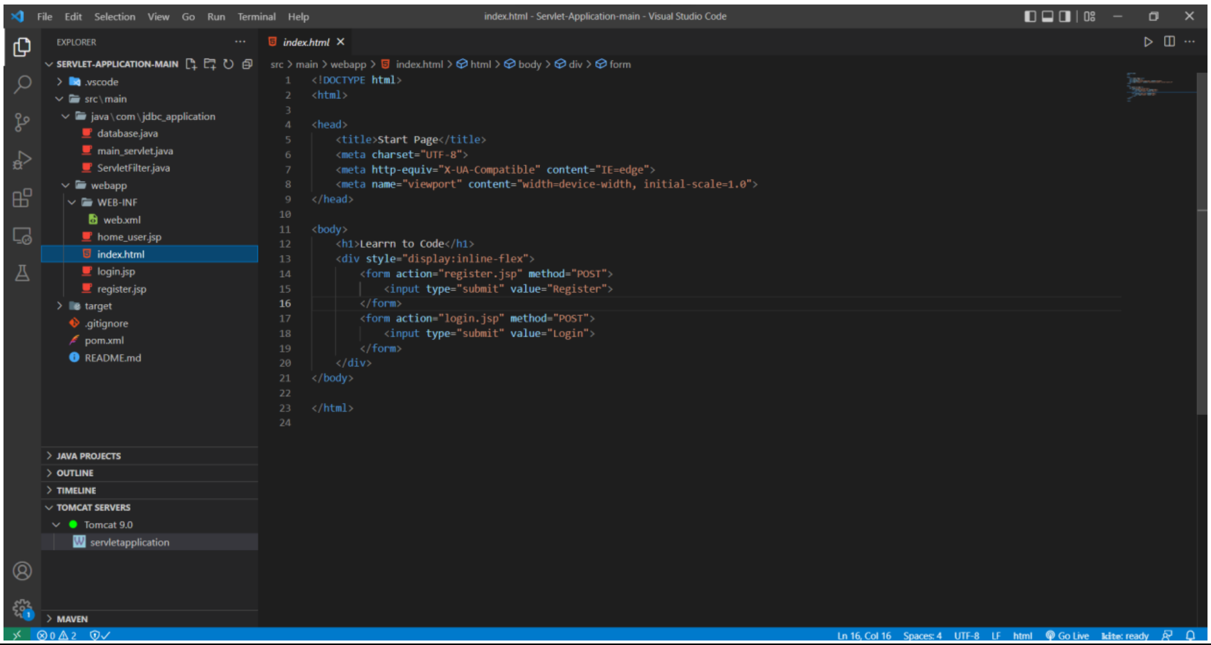
Now you have to paste the maven repository dependency code into the dependency section in the pom.xml file



```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>

  <!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
  <dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```

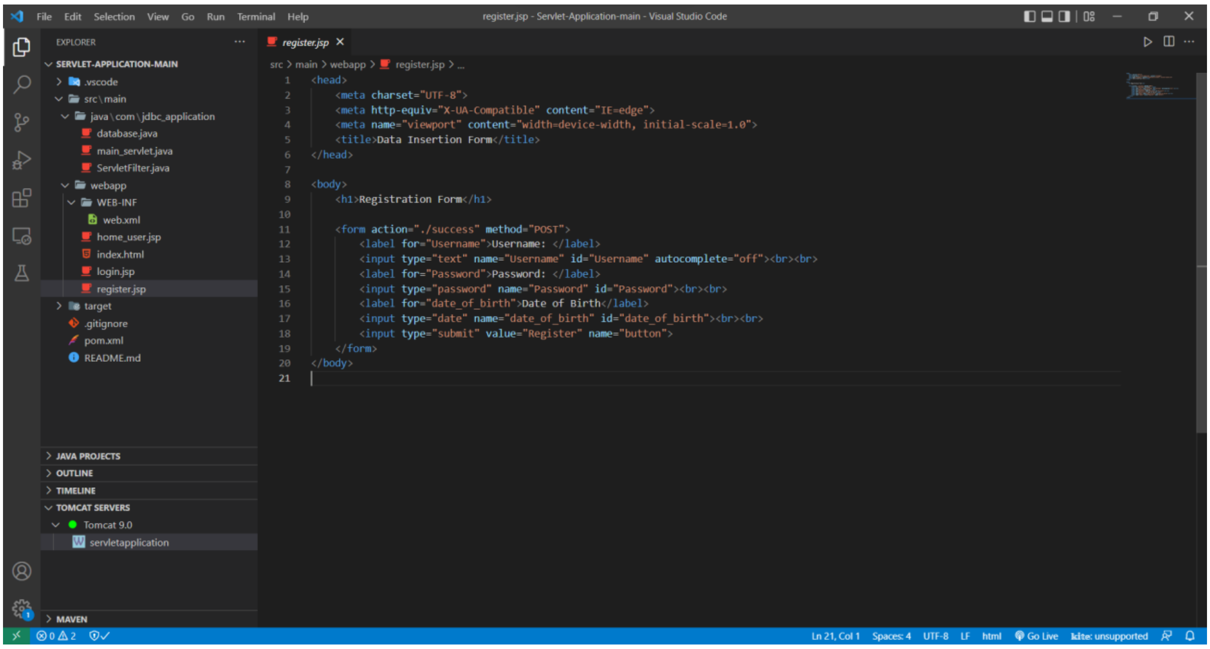
PROGRAM:

i. Index.html



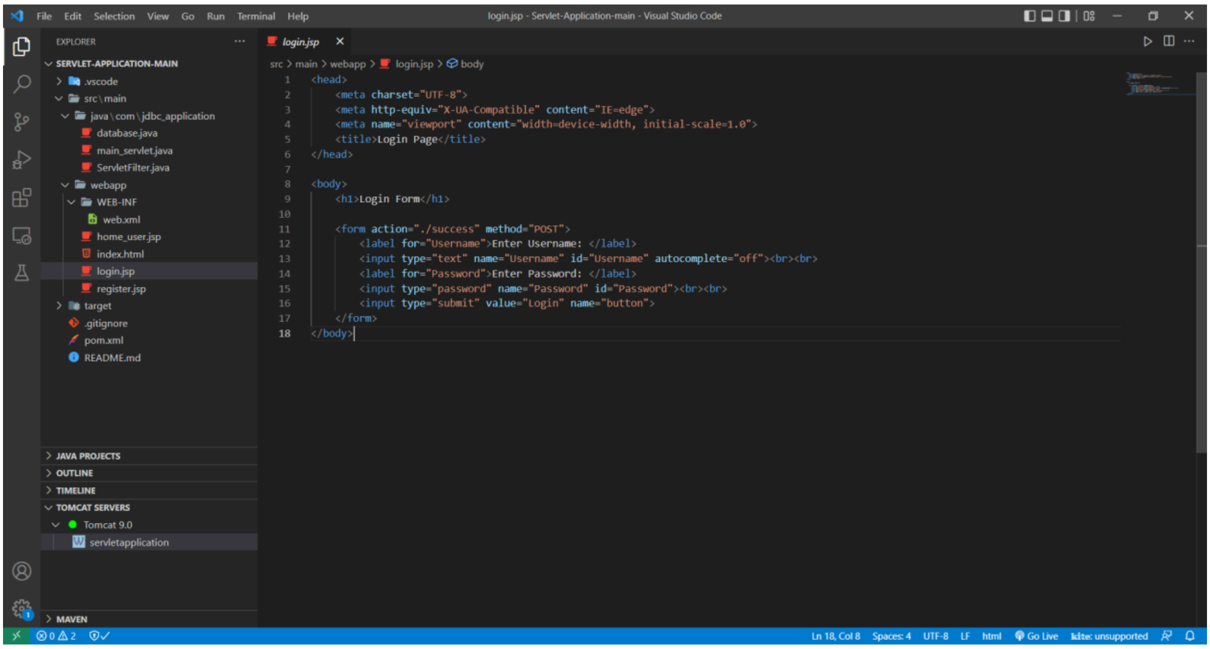
```
<!DOCTYPE html>
<html>

<head>
    <title>Start Page</title>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>

<body>
    <h1>Learrn to Code</h1>
    <div style="display:inline-flex">
        <form action="register.jsp" method="POST">
            <input type="submit" value="Register">
        </form>
        <form action="login.jsp" method="POST">
            <input type="submit" value="Login">
        </form>
    </div>
</body>

</html>
```

ii. register.jsp



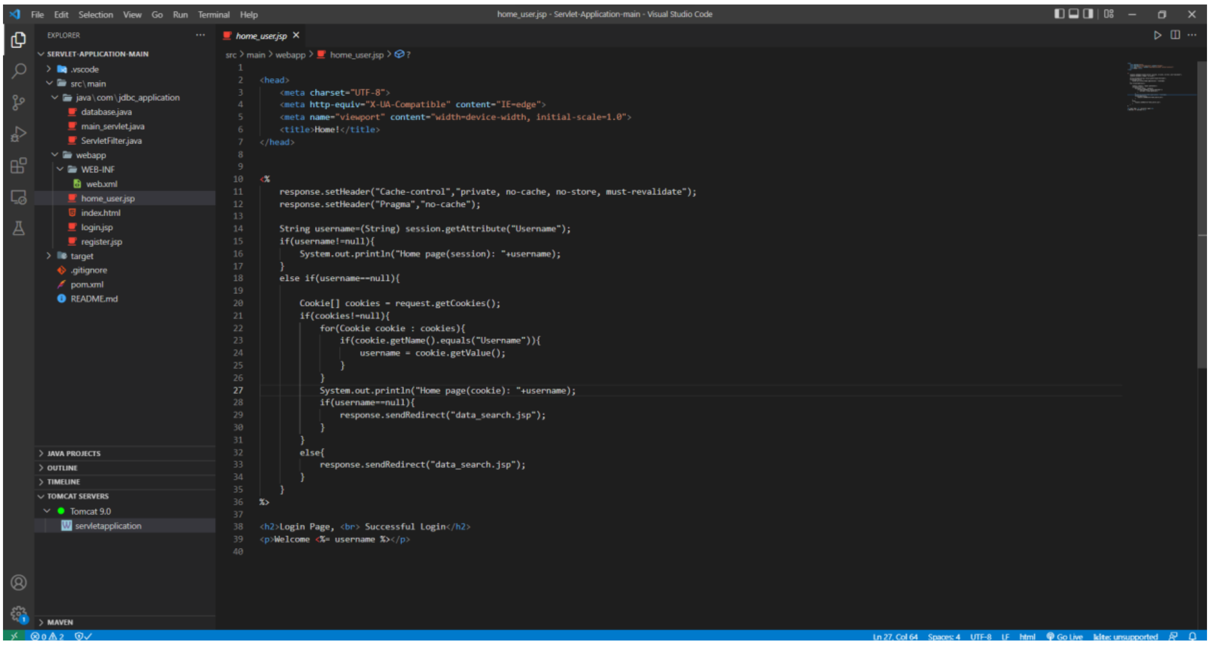
```html
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Data Insertion Form</title>
</head>

<body>
    <h1>Registration Form</h1>

    <form action="./success" method="POST">
        <label for="Username">Username: </label>
        <input type="text" name="Username" id="Username" autocomplete="off"><br><br>
        <label for="Password">Password: </label>
        <input type="password" name="Password" id="Password"><br><br>
        <label for="date_of_birth">Date of Birth</label>
        <input type="date" name="date_of_birth" id="date_of_birth"><br><br>
        <input type="submit" value="Register" name="button">
    </form>
</body>

```

iii. login.jsp



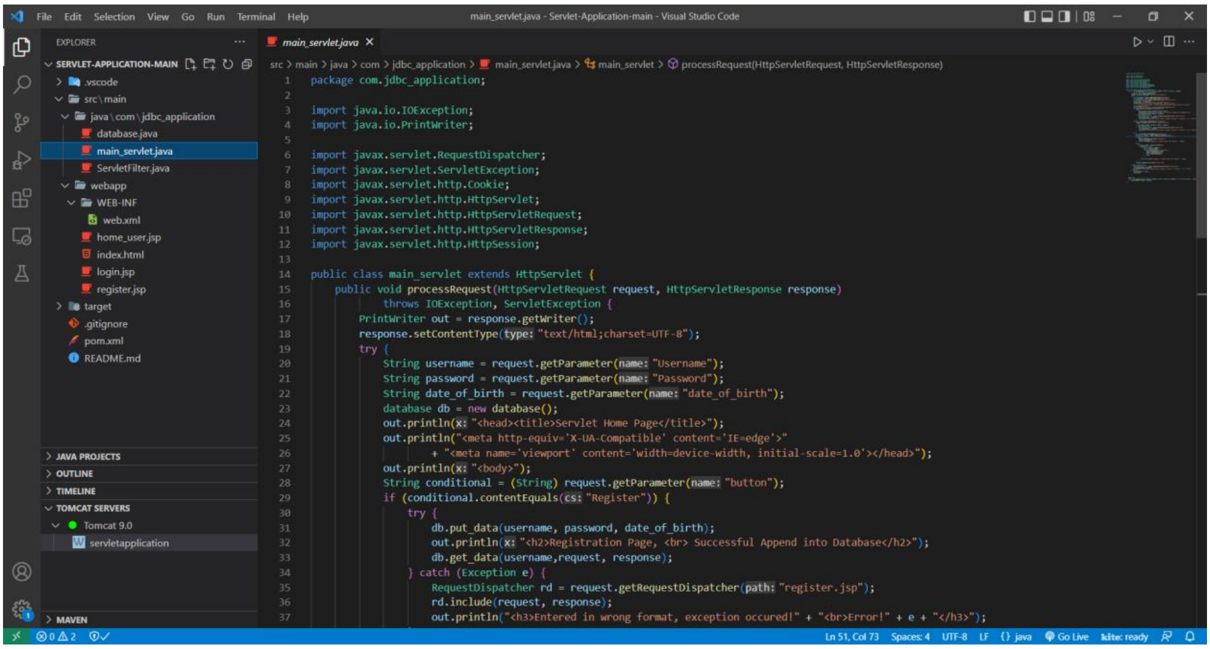
```html
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
</head>

<body>
    <h1>Login Form</h1>

    <form action="./success" method="POST">
        <label for="Username">Enter Username: </label>
        <input type="text" name="Username" id="Username" autocomplete="off"><br><br>
        <label for="Password">Enter Password: </label>
        <input type="password" name="Password" id="Password"><br><br>
        <input type="submit" value="Login" name="button">
    </form>
</body>
```

iv. home_user.jsp

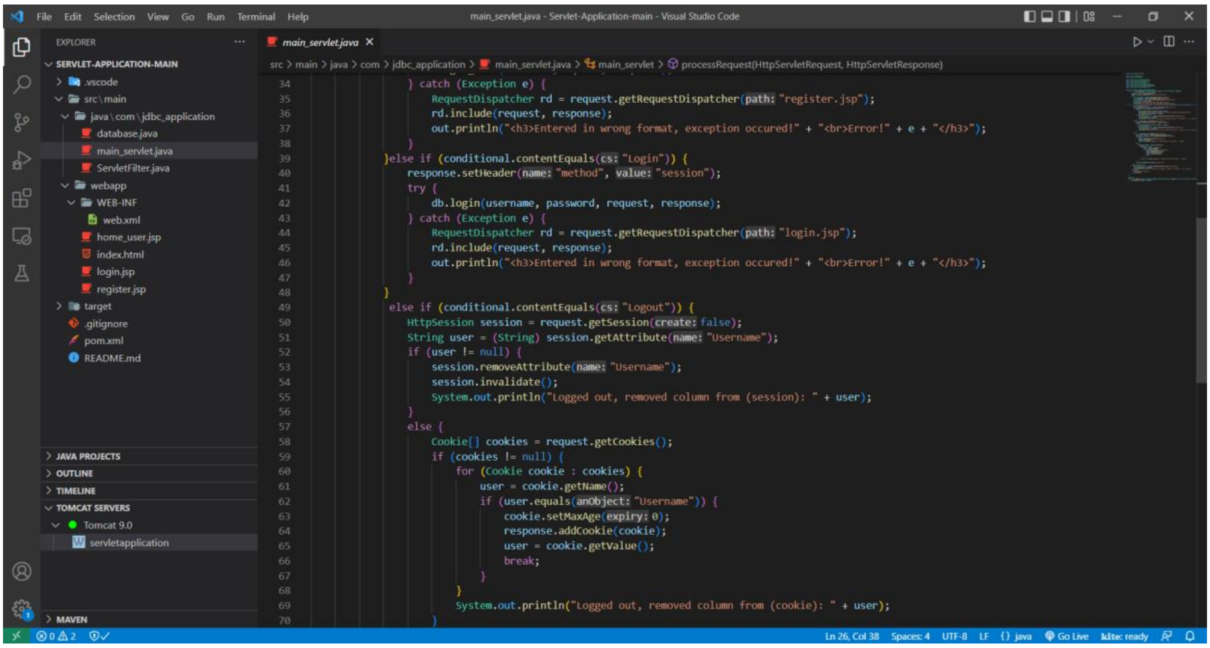


```

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home!</title>
</head>


<%
    response.setHeader("Cache-control","private, no-cache, no-store, must-revalidate");
    response.setHeader("Pragma","no-cache");

    String username=(String) session.getAttribute("Username");
    if(username!=null){
        System.out.println("Home page(session): "+username);
    }
    else if(username==null){

        Cookie[] cookies = request.getCookies();
        if(cookies!=null){
            for(Cookie cookie : cookies){
                if(cookie.getName().equals("Username")){
                    username = cookie.getValue();
                }
            }
            System.out.println("Home page(cookie): "+username);
            if(username==null){
                response.sendRedirect("data_search.jsp");
            }
        }
        else{
            response.sendRedirect("data_search.jsp");
        }
    }
%>

<h2>Login Page, <br> Successful Login</h2>
<p>Welcome <%= username %></p>

```

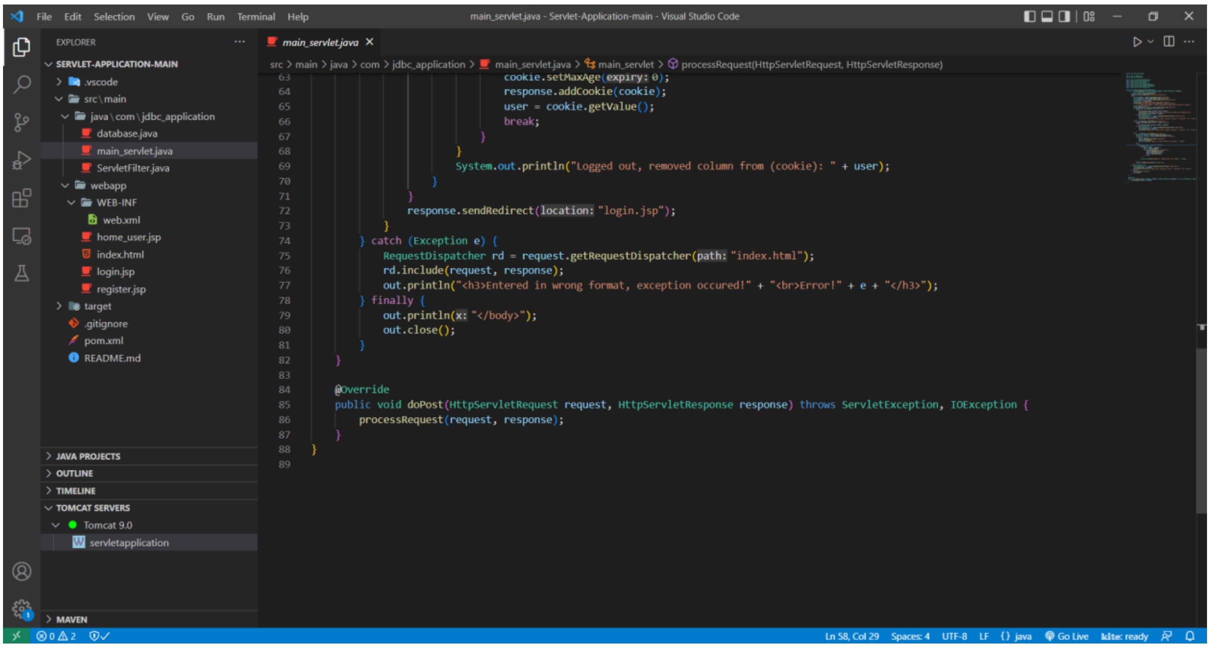
main_servlet.java


main_servlet.java - Servlet-Application-main - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
SERVLET-APPLICATION-MAIN
.vscode
src\main
java\com\jdbc_application
database.java
main_servlet.java
ServletFilter.java
webapp
WEB-INF
web.xml
home_user.jsp
index.html
login.jsp
register.jsp
target
.gitignore
pom.xml
README.md
JAVA PROJECTS
OUTLINE
TIMELINE
TOMCAT SERVERS
Tomcat 9.0
servletapplication
MAVEN
main_servlet.java
src > main > java > com > jdbc_application > main_servlet.java > main_servlet > processRequest(HttpServletRequest, HttpServletResponse)
Ln 51, Col 73 Spaces: 4 UTF-8 LF {} java Go Live kite: ready

```
package com.jdbc_application;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class main_servlet extends HttpServlet {
    public void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws IOException, ServletException {
        PrintWriter out = response.getWriter();
        response.setContentType(type: "text/html;charset=UTF-8");
        try {
            String username = request.getParameter(name: "Username");
            String password = request.getParameter(name: "Password");
            String date_of_birth = request.getParameter(name: "date_of_birth");
            database db = new database();
            out.println(x: "<head><title>Servlet Home Page</title>");
            out.println("<meta http-equiv='X-UA-Compatible' content='IE=edge'>"
                    + "<meta name='viewport' content='width=device-width, initial-scale=1.0'></head>");
            out.println(x: "<body>");
            String conditional = (String) request.getParameter(name: "button");
            if (conditional.contentEquals(cs: "Register")) {
                try {
                    db.put_data(username, password, date_of_birth);
                    out.println(x: "<h2>Registration Page, <br> Successful Append into Database</h2>");
                    db.get_data(username,request, response);
                } catch (Exception e) {
                    RequestDispatcher rd = request.getRequestDispatcher(path: "register.jsp");
                    rd.include(request, response);
                    out.println("<h3>Entered in wrong format, exception occured!" + "<br>Error!" + e + "</h3>");
```

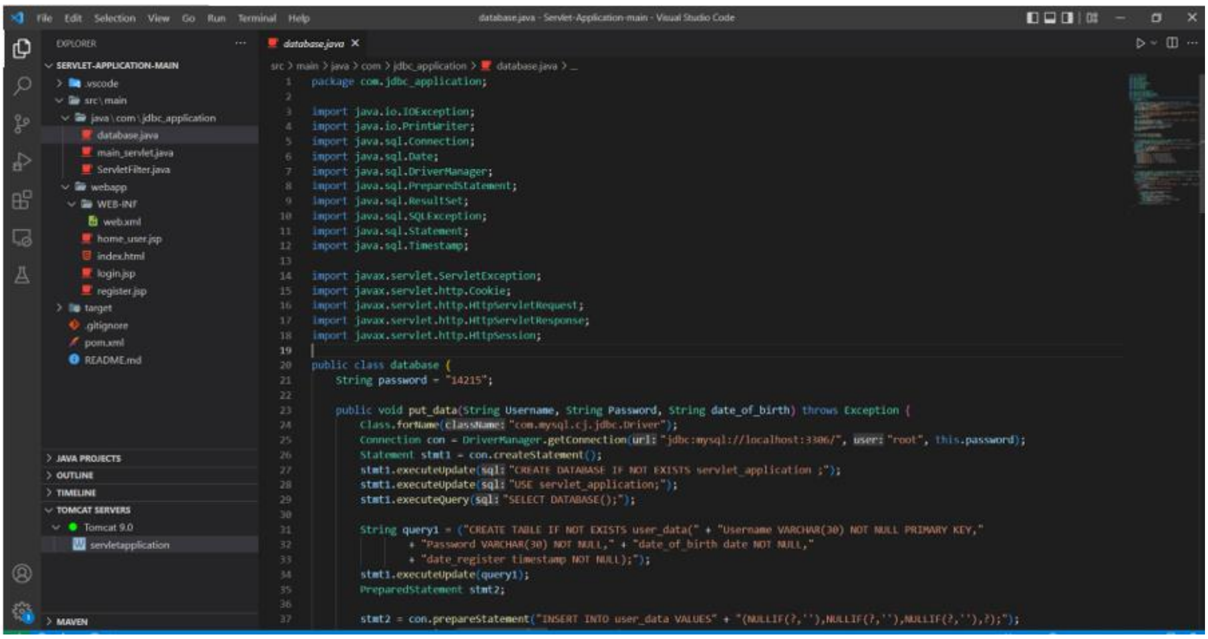


```
                } catch (Exception e) {
                    RequestDispatcher rd = request.getRequestDispatcher(path: "register.jsp");
                    rd.include(request, response);
                    out.println("<h3>Entered in wrong format, exception occured!" + "<br>Error!" + e + "</h3>");
                }
            }else if (conditional.contentEquals(cs: "Login")) {
                response.setHeader(name: "method", value: "session");
                try {
                    db.login(username, password, request, response);
                } catch (Exception e) {
                    RequestDispatcher rd = request.getRequestDispatcher(path: "login.jsp");
                    rd.include(request, response);
                    out.println("<h3>Entered in wrong format, exception occured!" + "<br>Error!" + e + "</h3>");
                }
            }
             else if (conditional.contentEquals(cs: "Logout")) {
                HttpSession session = request.getSession(create: false);
                String user = (String) session.getAttribute(name: "Username");
                if (user != null) {
                    session.removeAttribute(name: "Username");
                    session.invalidate();
                    System.out.println("Logged out, removed column from (session): " + user);
                }
                else {
                    Cookie[] cookies = request.getCookies();
                    if (cookies != null) {
                        for (Cookie cookie : cookies) {
                            user = cookie.getName();
                            if (user.equals(anObject: "Username")) {
                                cookie.setMaxAge(expiry: 0);
                                response.addCookie(cookie);
                                user = cookie.getValue();
                                break;
                            }
                        }
                        System.out.println("Logged out, removed column from (cookie): " + user);
                    }
```

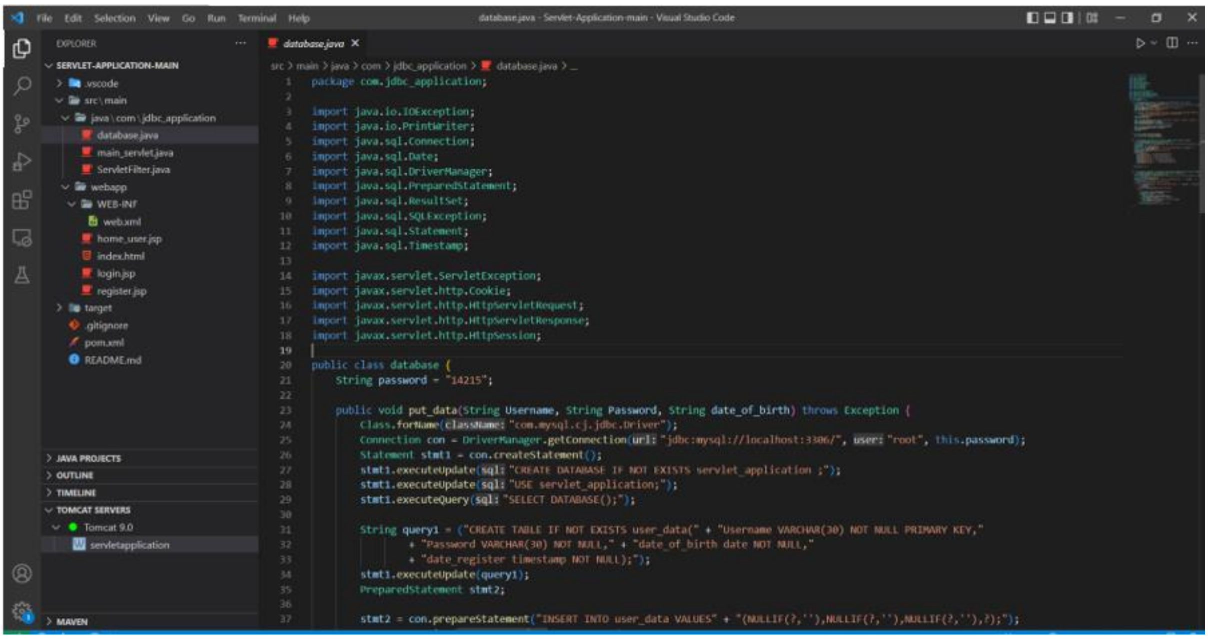


```
                                cookie.setMaxAge(expiry: 0);
                                response.addCookie(cookie);
                                user = cookie.getValue();
                                break;
                            }
                        }
                        System.out.println("Logged out, removed column from (cookie): " + user);
                    }
                }
                response.sendRedirect(location: "login.jsp");
            }
        } catch (Exception e) {
            RequestDispatcher rd = request.getRequestDispatcher(path: "index.html");
            rd.include(request, response);
            out.println("<h3>Entered in wrong format, exception occured!" + "<br>Error!" + e + "</h3>");
        } finally {
            out.println(x: "</body>");
            out.close();
        }
    }

    @Override
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        processRequest(request, response);
    }
}

```

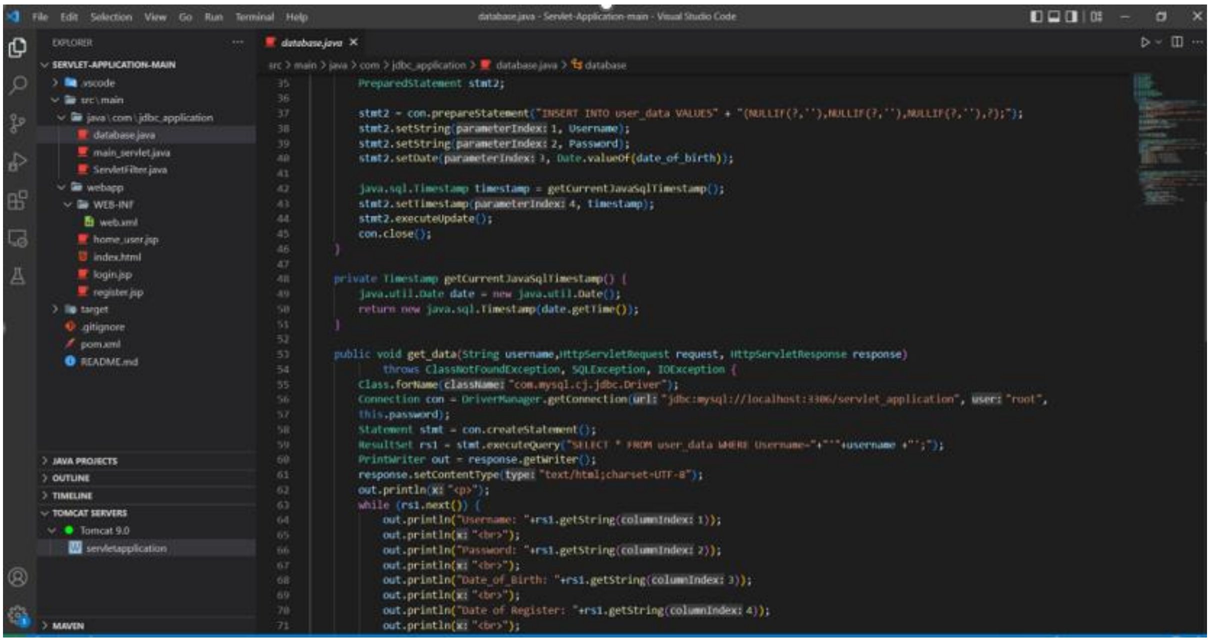
database.java



```
package com.jdbc_application;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Timestamp;

import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class database {
    String password = "14215";

    public void put_data(String Username, String Password, String date_of_birth) throws Exception {
        Class.forName(className: "com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(url: "jdbc:mysql://localhost:3306/", user: "root", this.password);
        Statement stmt1 = con.createStatement();
        stmt1.executeUpdate(sql: "CREATE DATABASE IF NOT EXISTS servlet_application ;");
        stmt1.executeUpdate(sql: "USE servlet_application;");
        stmt1.executeQuery(sql: "SELECT DATABASE();");

        String query1 = ("CREATE TABLE IF NOT EXISTS user_data(" + "Username VARCHAR(30) NOT NULL PRIMARY KEY,"
                + "Password VARCHAR(30) NOT NULL," + "date_of_birth date NOT NULL,"
                + "date_register timestamp NOT NULL);");
        stmt1.executeUpdate(query1);
        PreparedStatement stmt2;

        stmt2 = con.prepareStatement("INSERT INTO user_data VALUES" + "(NULLIF(?,''),NULLIF(?,''),NULLIF(?,''),?);");
```

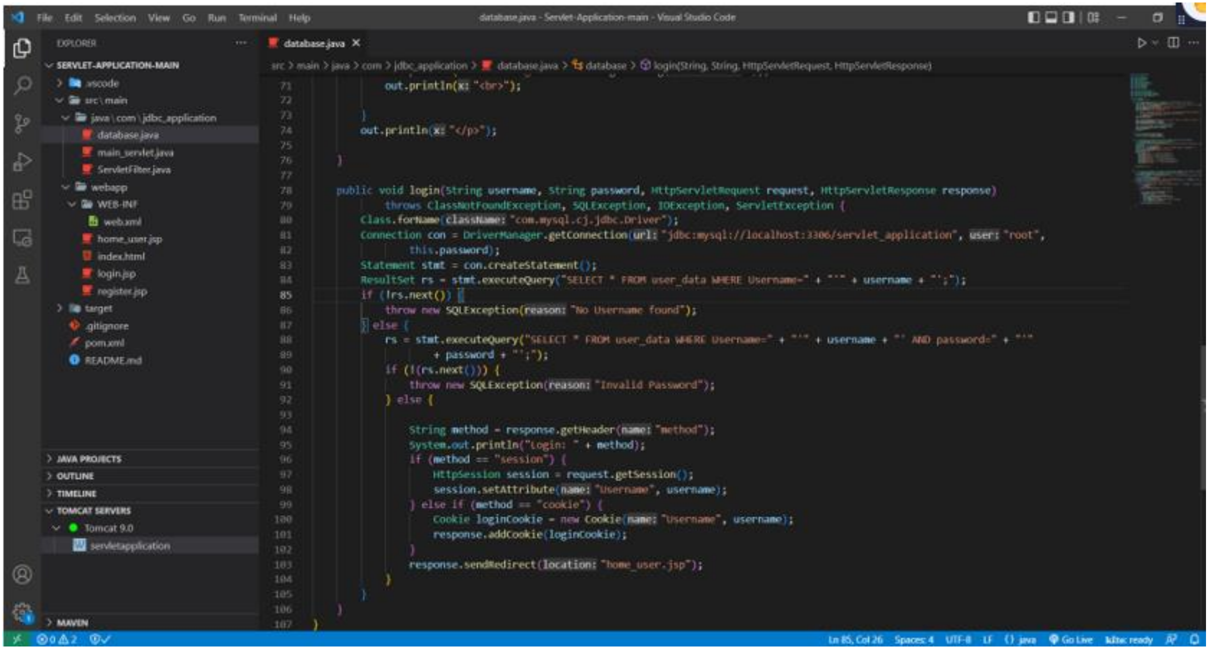


```java
package com.jdbc_application;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Timestamp;

import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class database {
    String password = "14215";

    public void put_data(String Username, String Password, String date_of_birth) throws Exception {
        Class.forName(className: "com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(url: "jdbc:mysql://localhost:3306/", user: "root", this.password);
        Statement stmt1 = con.createStatement();
        stmt1.executeUpdate(sql: "CREATE DATABASE IF NOT EXISTS servlet_application ;");
        stmt1.executeUpdate(sql: "USE servlet_application;");
        stmt1.executeQuery(sql: "SELECT DATABASE();");

        String query1 = ("CREATE TABLE IF NOT EXISTS user_data(" + "Username VARCHAR(30) NOT NULL PRIMARY KEY,"
                + "Password VARCHAR(30) NOT NULL," + "date_of_birth date NOT NULL,"
                + "date_register timestamp NOT NULL);");
        stmt1.executeUpdate(query1);
        PreparedStatement stmt2;

        stmt2 = con.prepareStatement("INSERT INTO user_data VALUES" + "(NULLIF(?,''),NULLIF(?,''),NULLIF(?,''),?);");
```



File Edit Selection View Go Run Terminal Help
database.java - Servlet-Application-main - Visual Studio Code
EXPLORER
SERVLET-APPLICATION-MAIN
database.java
src > main > java > com > jdbc_application > database.java > database
TOMCAT SERVERS
Tomcat 9.0
servletapplication
MAVEN

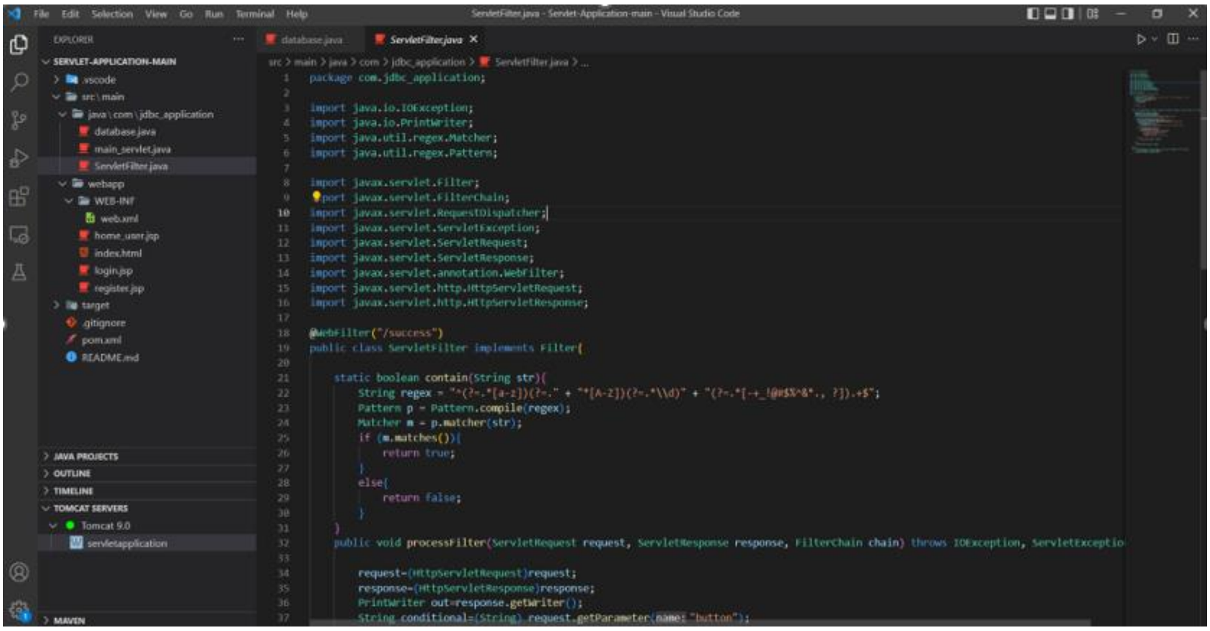
```java
        PreparedStatement stmt2;

        stmt2 = con.prepareStatement("INSERT INTO user_data VALUES" + "(NULLIF(?,''),NULLIF(?,''),NULLIF(?,''),?);");
        stmt2.setString(parameterIndex: 1, Username);
        stmt2.setString(parameterIndex: 2, Password);
        stmt2.setDate(parameterIndex: 3, Date.valueOf(date_of_birth));

        java.sql.Timestamp timestamp = getCurrentJavaSqlTimestamp();
        stmt2.setTimestamp(parameterIndex: 4, timestamp);
        stmt2.executeUpdate();
        con.close();
    }

    private Timestamp getCurrentJavaSqlTimestamp() {
        java.util.Date date = new java.util.Date();
        return new java.sql.Timestamp(date.getTime());
    }

    public void get_data(String username,HttpServletRequest request, HttpServletResponse response)
            throws ClassNotFoundException, SQLException, IOException {
        Class.forName(className: "com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(url: "jdbc:mysql://localhost:3306/servlet_application", user: "root",
        this.password);
        Statement stmt = con.createStatement();
        ResultSet rs1 = stmt.executeQuery("SELECT * FROM user_data WHERE Username="+"'"+username +"';");
        PrintWriter out = response.getWriter();
        response.setContentType(type: "text/html;charset=UTF-8");
        out.println(x: "<p>");
        while (rs1.next()) {
            out.println("Username: "+rs1.getString(columnIndex: 1));
            out.println(x: "<br>");
            out.println("Password: "+rs1.getString(columnIndex: 2));
            out.println(x: "<br>");
            out.println("Date_of_Birth: "+rs1.getString(columnIndex: 3));
            out.println(x: "<br>");
            out.println("Date of Register: "+rs1.getString(columnIndex: 4));
            out.println(x: "<br>");
```



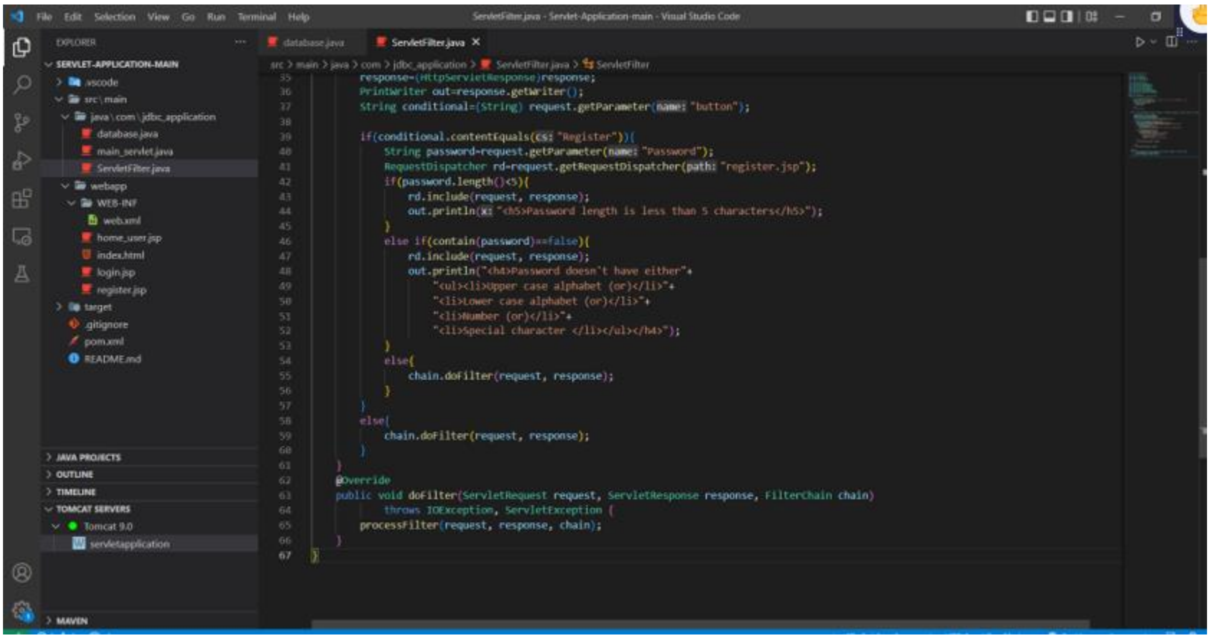
File Edit Selection View Go Run Terminal Help
database.java - Servlet-Application-main - Visual Studio Code
EXPLORER
SERVLET-APPLICATION-MAIN
database.java
src > main > java > com > jdbc_application > database.java > database > login(String, String, HttpServletRequest, HttpServletResponse)
TOMCAT SERVERS
Tomcat 9.0
servletapplication
MAVEN
Ln 85, Col 26 Spaces: 4 UTF-8 LF {} java Go Live

```java
            out.println(x: "<br>");

        }
        out.println(x: "</p>");

    }

    public void login(String username, String password, HttpServletRequest request, HttpServletResponse response)
            throws ClassNotFoundException, SQLException, IOException, ServletException {
        Class.forName(className: "com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection(url: "jdbc:mysql://localhost:3306/servlet_application", user: "root",
                this.password);
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM user_data WHERE Username=" + "'" + username + "';");
        if (!rs.next()) {
            throw new SQLException(reason: "No Username found");
        } else {
            rs = stmt.executeQuery("SELECT * FROM user_data WHERE Username=" + "'" + username + "' AND password=" + "'"
                    + password + "';");
            if (!(rs.next())) {
                throw new SQLException(reason: "Invalid Password");
            } else {

                String method = response.getHeader(name: "method");
                System.out.println("Login: " + method);
                if (method == "session") {
                    HttpSession session = request.getSession();
                    session.setAttribute(name: "Username", username);
                } else if (method == "cookie") {
                    Cookie loginCookie = new Cookie(name: "Username", username);
                    response.addCookie(loginCookie);
                }
                response.sendRedirect(location: "home_user.jsp");
            }
        }
    }
}
```

ServletFilter.java

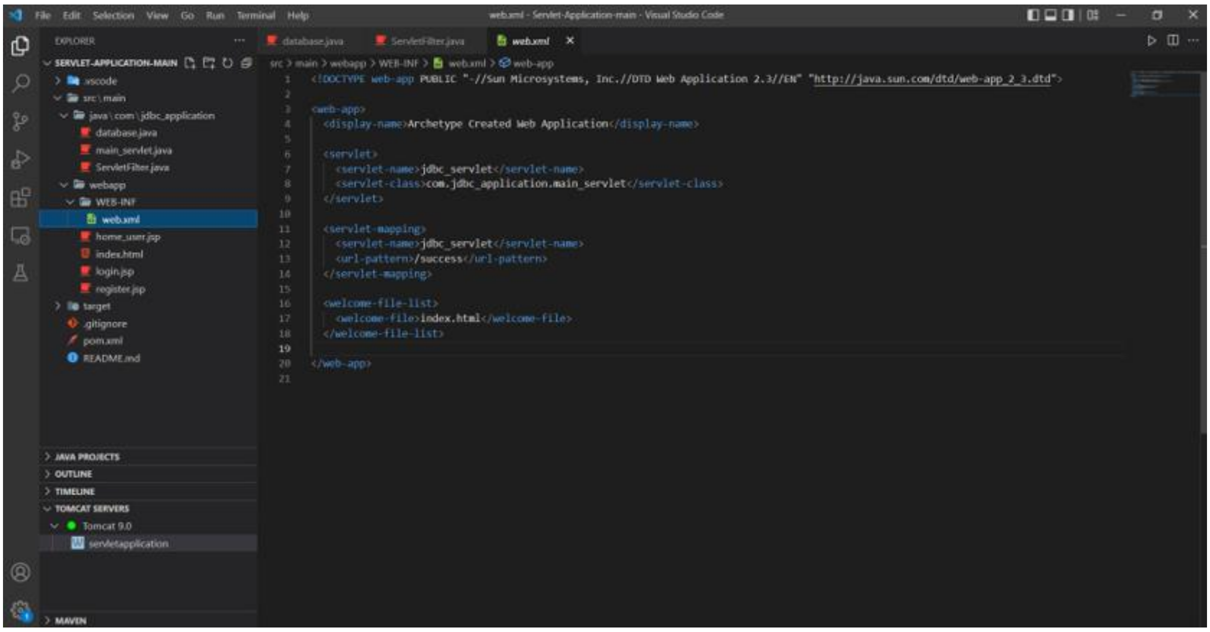


```java
package com.jdbc_application;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.annotation.WebFilter;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebFilter("/success")
public class ServletFilter implements Filter{

    static boolean contain(String str){
        String regex = "^(?=.*[a-z])(?=." + "*[A-Z])(?=.*\\d)" + "(?=.*[-+_!@#$%^&*., ?]).+$";
        Pattern p = Pattern.compile(regex);
        Matcher m = p.matcher(str);
        if (m.matches()){
            return true;
        }
        else{
            return false;
        }
    }
    public void processFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletExceptio

        request=(HttpServletRequest)request;
        response=(HttpServletResponse)response;
        PrintWriter out=response.getWriter();
        String conditional=(String) request.getParameter(name: "button");
```



```java
        response=(HttpServletResponse)response;
        PrintWriter out=response.getWriter();
        String conditional=(String) request.getParameter(name: "button");

        if(conditional.contentEquals(cs: "Register")){
            String password=request.getParameter(name: "Password");
            RequestDispatcher rd=request.getRequestDispatcher(path: "register.jsp");
            if(password.length()<5){
                rd.include(request, response);
                out.println(x: "<h5>Password length is less than 5 characters</h5>");
            }
            else if(contain(password)==false){
                rd.include(request, response);
                out.println("<h4>Password doesn't have either"+
                    "<ul><li>Upper case alphabet (or)</li>"+
                    "<li>Lower case alphabet (or)</li>"+
                    "<li>Number (or)</li>"+
                    "<li>Special character </li></ul></h4>");
            }
            else{
                chain.doFilter(request, response);
            }
        }
        else{
            chain.doFilter(request, response);
        }
    }
    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
            throws IOException, ServletException {
        processFilter(request, response, chain);
    }
}
```

web.xml

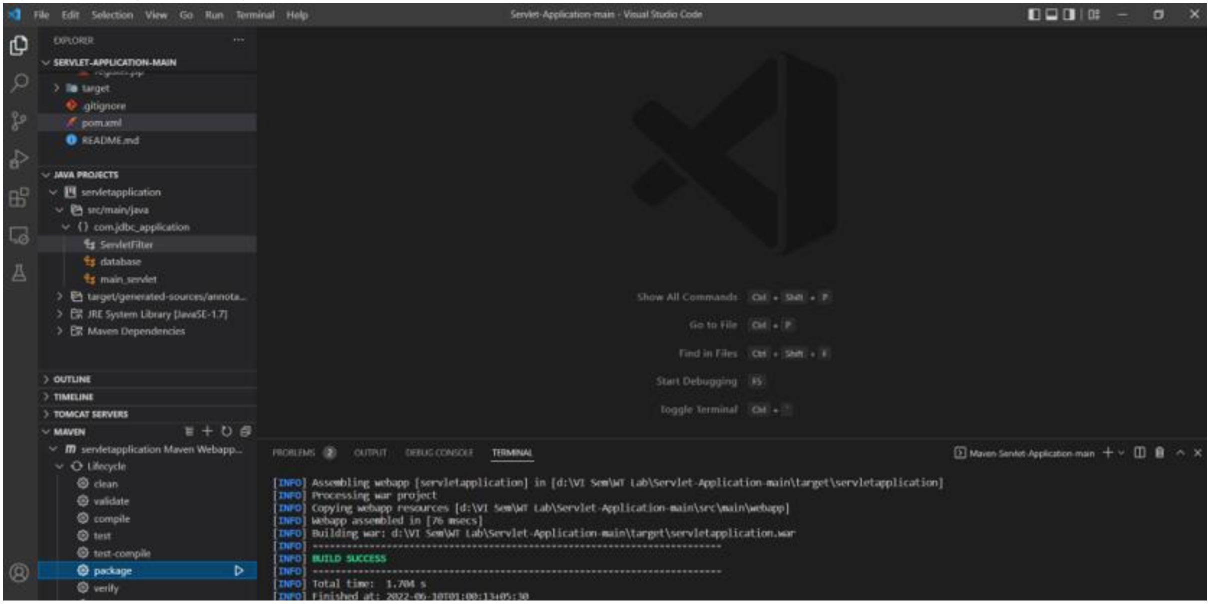


```xml
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <display-name>Archetype Created Web Application</display-name>

  <servlet>
    <servlet-name>jdbc_servlet</servlet-name>
    <servlet-class>com.jdbc_application.main_servlet</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>jdbc_servlet</servlet-name>
    <url-pattern>/success</url-pattern>
  </servlet-mapping>

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
  </welcome-file-list>

</web-app>

```

Deploying into Tomcat Server:

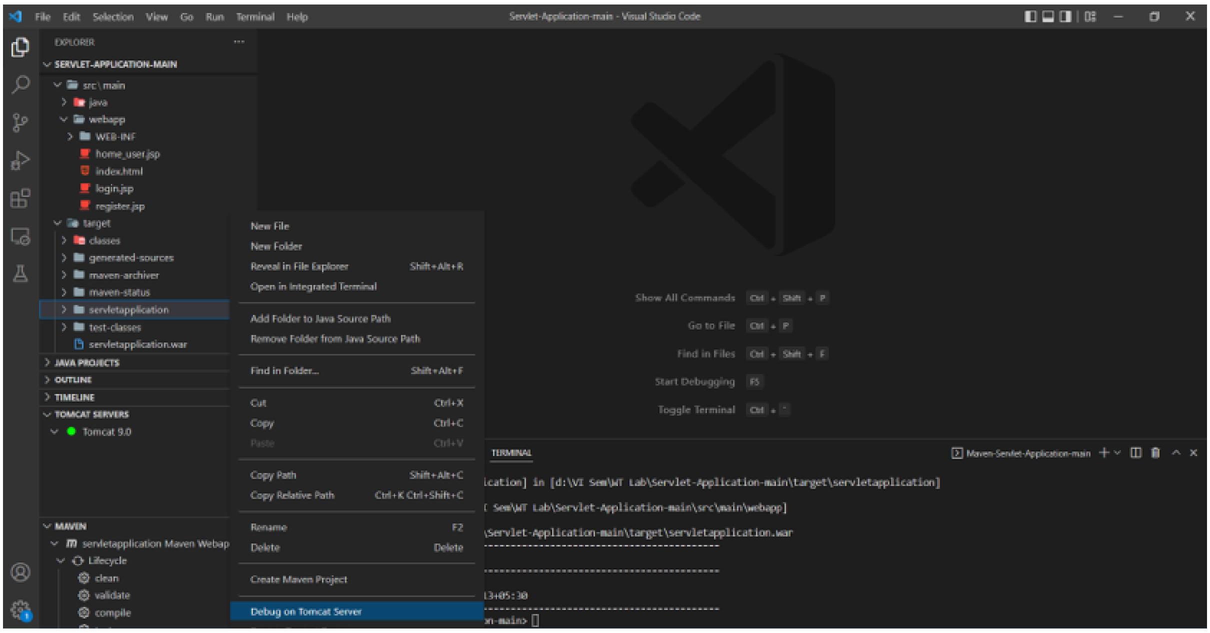
First install the **Tomcat for Java** Extension in vs code and then add your already installed Tomcat server (programs files→Apache Software Foundation→Tomcat 9.0).

To deploy into server (MAVEN→Maven Webapp→Lifecycle→packages (run the packages by clicking the play button)).

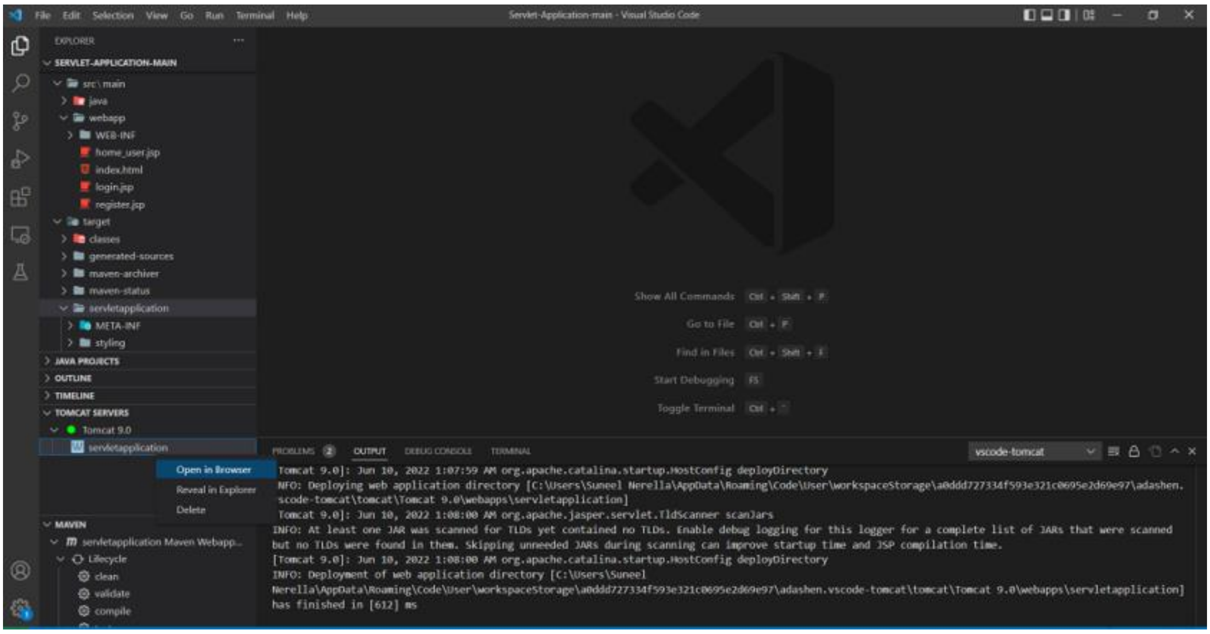


```
[INFO] Assembling webapp [servletapplication] in [d:\VI Sem\WT Lab\Servlet-Application-main\target\servletapplication]
[INFO] Processing war project
[INFO] Copying webapp resources [d:\VI Sem\WT Lab\Servlet-Application-main\src\main\webapp]
[INFO] Webapp assembled in [76 msecs]
[INFO] Building war: d:\VI Sem\WT Lab\Servlet-Application-main\target\servletapplication.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.704 s
[INFO] Finished at: 2022-06-10T01:00:13+05:30
```

Next Under **target folder** double click **Name of your project** floder and then debug in tomcat server.

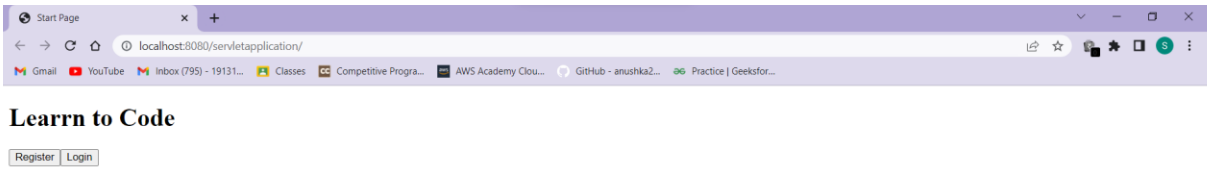


```
ication] in [d:\VI Sem\WT Lab\Servlet-Application-main\target\servletapplication]
[ Sem\WT Lab\Servlet-Application-main\src\main\webapp]
\Servlet-Application-main\target\servletapplication.war
--------------------------------------------------------

--------------------------------------------------------

13+05:30
--------------------------------------------------------
on-main> 
```



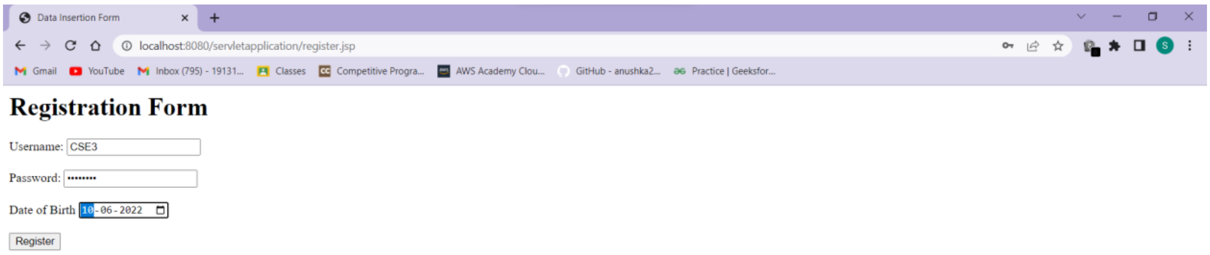
```
Tomcat 9.0]: Jun 10, 2022 1:07:59 AM org.apache.catalina.startup.HostConfig deployDirectory
NFO: Deploying web application directory [C:\Users\Suneel Nerella\AppData\Roaming\Code\User\workspaceStorage\a0ddd727334f593e321c0695e2d69e97\adashen.
scode-tomcat\tomcat\Tomcat 9.0\webapps\servletapplication]
Tomcat 9.0]: Jun 10, 2022 1:08:00 AM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned
but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.
[Tomcat 9.0]: Jun 10, 2022 1:08:00 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory [C:\Users\Suneel
Nerella\AppData\Roaming\Code\User\workspaceStorage\a0ddd727334f593e321c0695e2d69e97\adashen.vscode-tomcat\tomcat\Tomcat 9.0\webapps\servletapplication]
has finished in [612] ms
```

Output



Learrn to Code

Register | Login



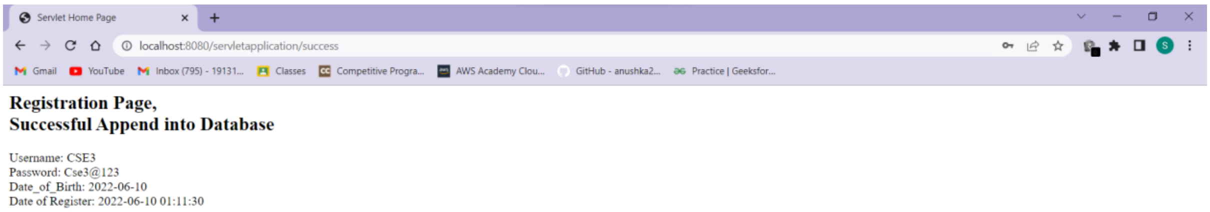
Registration Form

Username: CSE3

Password: ••••••••

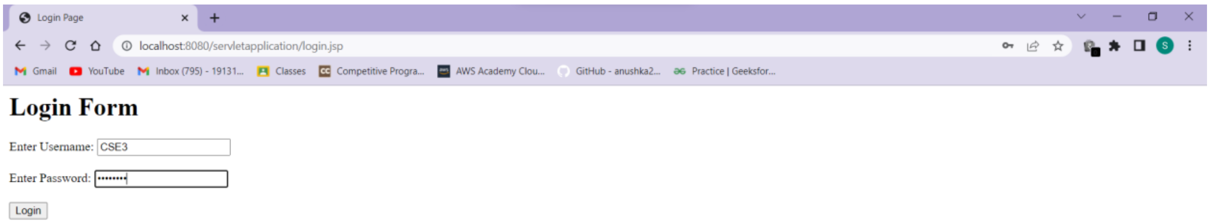
Date of Birth 10-06-2022

Register



Registration Page,
Successful Append into Database

Username: CSE3
Password: Cse3@123
Date_of_Birth: 2022-06-10
Date of Register: 2022-06-10 01:11:30

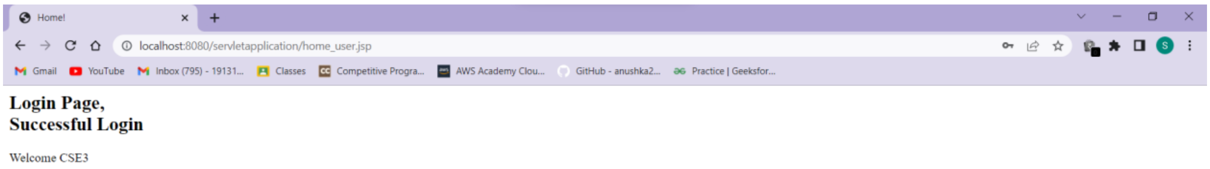


Login Form

Enter Username: CSE3

Enter Password: ••••••••

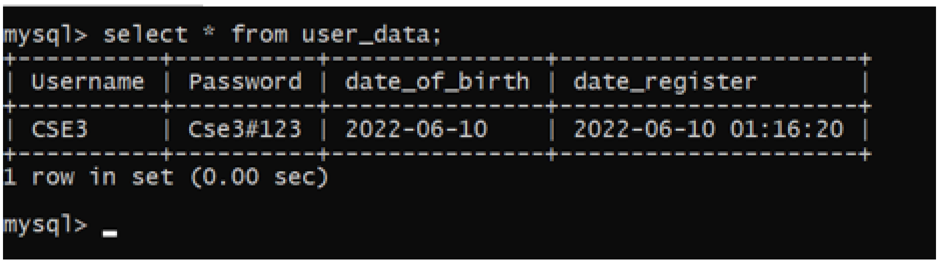
Login



Login Page,
Successful Login

Welcome CSE3

Database



```
mysql> select * from user_data;
+----------+----------+---------------+---------------------+
| Username | Password | date_of_birth | date_register       |
+----------+----------+---------------+---------------------+
| CSE3     | Cse3#123 | 2022-06-10    | 2022-06-10 01:16:20 |
+----------+----------+---------------+---------------------+
1 row in set (0.00 sec)

mysql> _
```