Pimpri Chinchwad Education Trust's

Pimpri Chinchwad College of Engineering

Department of Computer Engineering

STQA Mini Project 2

Quiz Web App

December 03, 2020

Members :

1. Aishwarya Patil (BECOB228)
2. Swapnajit Patil(BECOB229)
3. Sachin Rokade(BECOB240)
4. Vijay Sharma (BECOB249)

Guide:

Prof. Shrikant Kokate

Index

Sr. No.	Content	Page No.
1.	Introduction	3
2.	Functional Requirements	3
3.	Non - Functional Requirements	5
4.	Design (Block Diagram)	6
5.	Source Code / Functions	6
6.	Output Screenshots	38
7.	Manual Test Scenarios and Test Cases	41
8.	Selenium IDE Test Report - (Command - target-value) for min 15 Selenese command (Ex. asert, verify type, open, click)	47
9.	Selenium Grid Test Report - Steps to create Grid, (Java Code and Output) for Navigation USe Case, Locating HTMl Elements, Verify Title of the page, Action on HTML Elements	57
10.	Selenium Web Driver - Test Report (Java Code and Output) for Login Use case, Navigation USe Case, Locating HTMl Elements, Verify Title of the page, Action on HTML Elements	66

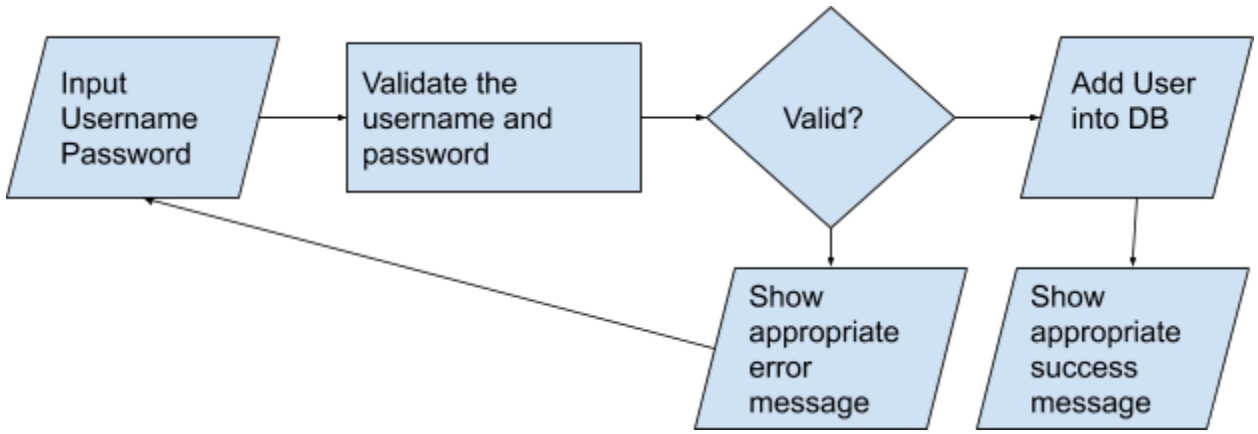
Introduction

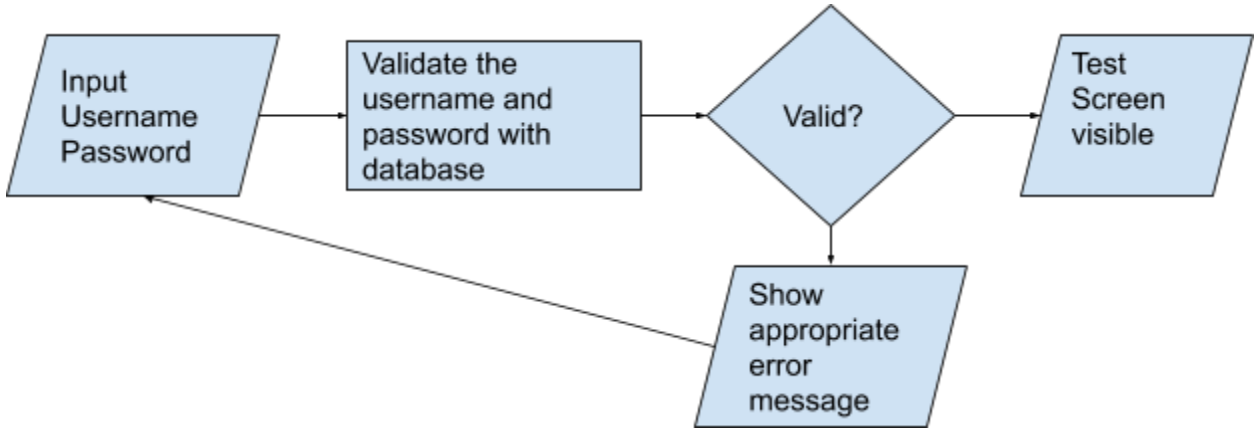
We have developed a web application to take quizzes on any topic. The application starts with a login/registration form. The participant needs to register first, after successful registration participant's details are added to the database. After this, the user can login and attempt a quiz. Once logged in with registered credentials, the quiz starts with the first question loaded with a timer. Participant selects an answer and clicks the next button. Once answered, he can't navigate back to the previous question. Participant have to answer all the questions within time. Either the user has answered all questions or the timer is over, the result screen shows the analysis of performance. User clicks the exit button to get back to the homepage. This application is useful in conducting online tests from remote places. The quiz can be given at your own convenience at any time and place.

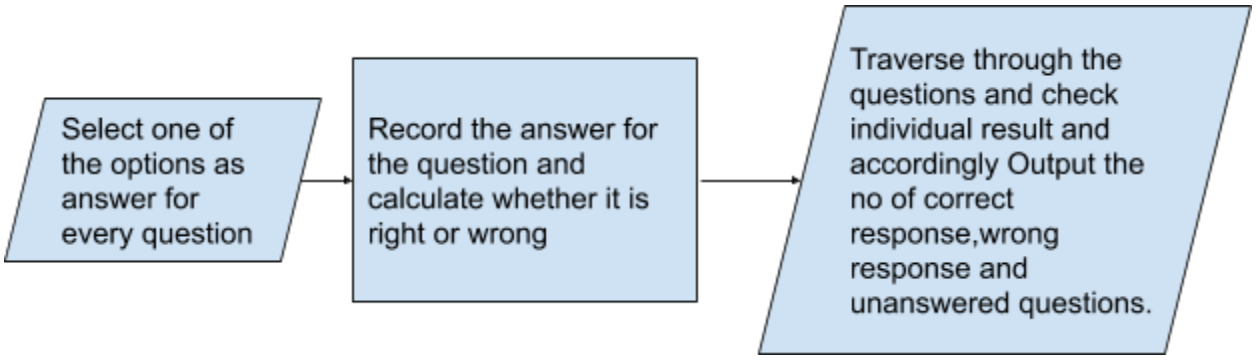
Functional Requirements

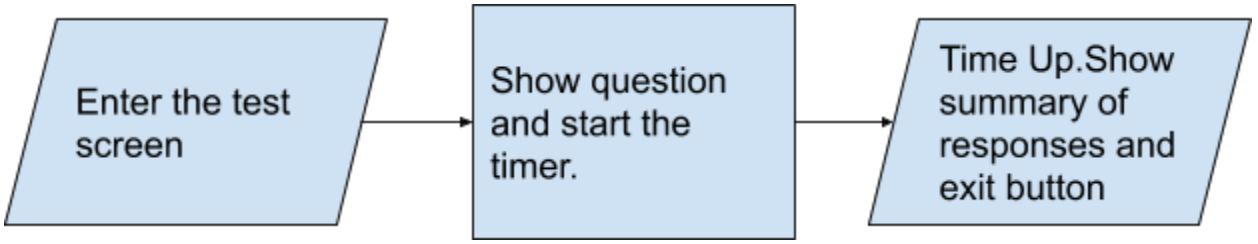
Functional requirements define the basic system behaviour. Essentially, they are what the system does or must not do, and can be thought of in terms of how the system responds to inputs.

- The main function of the application is to record participant's answers and show the correct result.
- The other functional requirements include the authorization of valid participants to take the quiz. Invalid participants should not be allowed to see the test page. Database is maintained to achieve this.
- The question should be skipped with alert if the timer is over for that question.
- The close and exit button should exit the application
- Clicking on next must take the user to the next question.


Input Username Password
Validate the username and password
Valid?
Add User into DB
Show appropriate error message
Show appropriate success message


Input Username Password
Validate the username and password with database
Valid?
Test Screen visible
Show appropriate error message


Select one of the options as answer for every question
Record the answer for the question and calculate whether it is right or wrong
Traverse through the questions and check individual result and accordingly Output the no of correct response,wrong response and unanswered questions.

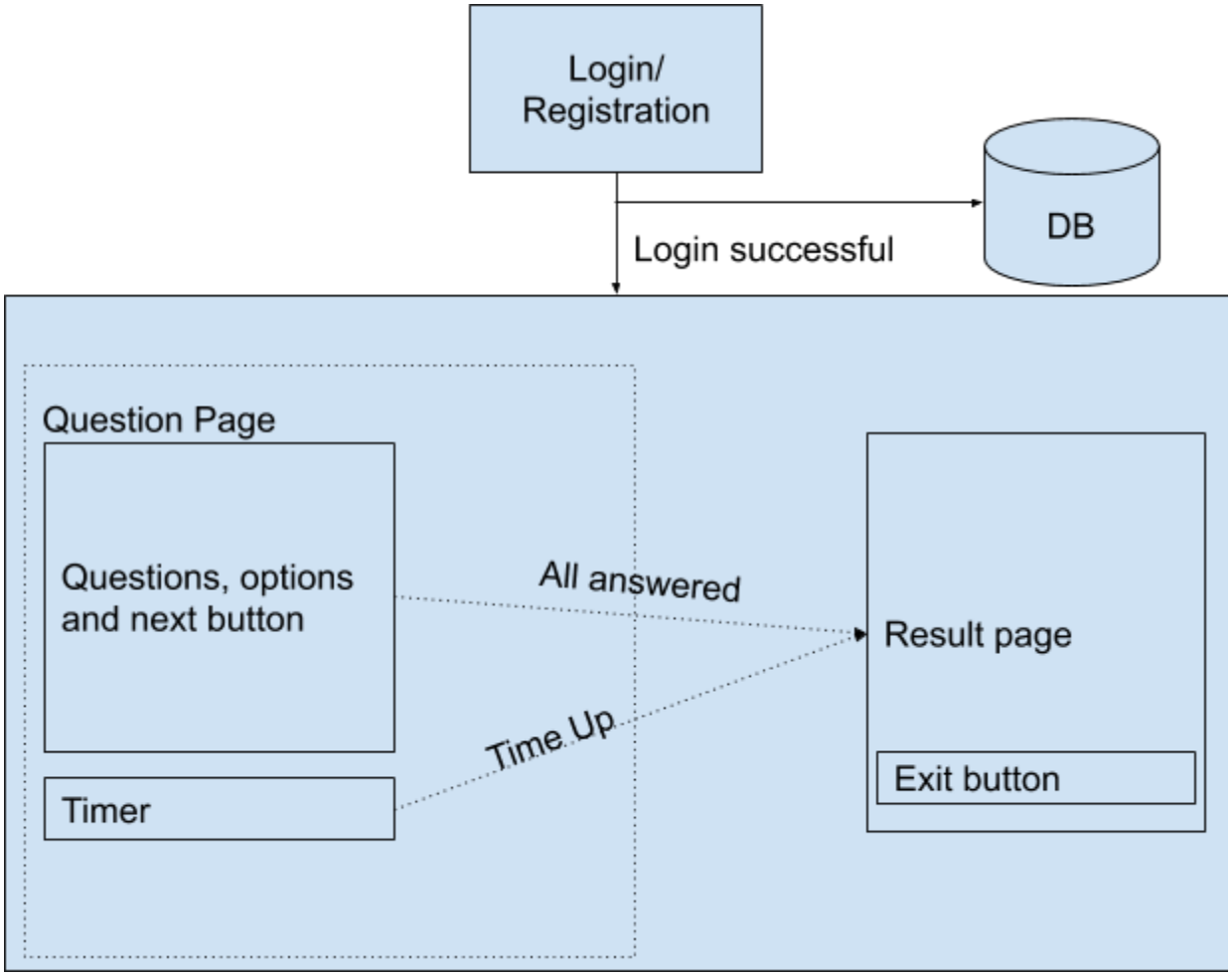

Enter the test screen
Show question and start the timer.
Time Up.Show summary of responses and exit button

Non Functional Requirements

While *functional* requirements define what the system does or must not do, *non-functional* requirements specify **how** the system should do it.

- The background of the quiz should be the same across all questions.
- Appropriate error messages or instructions should be shown, for example, enter username, enter password, invalid credentials, all fields are mandatory to fill.
- Time-up alert box should be shown if time is over.
- Close button(tab/browser) to abort the application.
- Font size,style,weight,family should be consistent across all screens.
- Timer is bright enough to be visible.
- Result and exit button should be visible immediately after the quiz. No need to search various options and menus.
- Tests are loaded quickly as soon as credentials are validated.

Design (BLOCK DIAGRAM)



Source Code

Index.html

```
<!doctype html>

<html lang="en">

<head>
```

```
<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport"

content="width=device-width, initial-scale=1, shrink-to-fit=no">


<!-- Bootstrap CSS -->

<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"


integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q
9Ifjh"

crossorigin="anonymous">


<title>HOME</title>

</head>

<body

style="background-image: url('bg.jpg'); background-color: #cccccc; background-position: center;
background-repeat: no-repeat; background-size: cover;">


<div class="container">

        <div class="row justify-content-lg-around my-5">
```

```
<div class="col-lg-3 border bg-light">
<img src="PCCOE_LOGO.jpg" class="img-fluid">
</div>

<div class=" border bg-light col-lg-7 ">

<form  class="p-3" action="DemoJSP.jsp" method="post"
name="register">
 <div class="form-group col-md-6">
  <label for="uname">Email</label>
  <input type="email" class="form-control" id="uname"name="uname">
 </div>
 <div class="form-group col-md-6">
  <label for="pass">Password</label>
  <input type="password" class="form-control" id="pass"name="pass">
 </div>

 <div class="form-group col-md-6">

 <input name="action" type="submit" class="btn btn-primary"value="Login"></input>
<input name="action" type="submit" class="btn btn-dark"value="Register"></input>
```

```
</div>
</form>
                </div>
        </div>
</div>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"

integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ
+n"
        crossorigin="anonymous"></script>
<script
        src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooA
o"
        crossorigin="anonymous"></script>
<script
        src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
```

```
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB
6"
        crossorigin="anonymous"></script>
</body>
</html>
```

home.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
import="java.sql.*,javax.servlet.http.*,java.util.Enumeration"
pageEncoding="ISO-8859-1"%>
<html lang="en">

<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
```

```
integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q
9Ifjh"

crossorigin="anonymous">

  <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4/jquery.min.js"></script>


<title>HOME</title>

<script type="text/javascript">


let i=0;

setInterval(()=>{

i++;

document.getElementById("timer").innerHTML=i;

if(i==10){

alert("timeup");window.location.replace("result.jsp");

}


}, 1000);



</script>
```

```
</head>

<body

style="background-image: url('bg.jpg'); background-color: #cccccc; background-position: center;
background-repeat: no-repeat; background-size: cover;">


<div class="container">

<div class="row justify-content-lg-around my-5">

<%

String dbUrl = "jdbc:mysql://localhost:3306/demo";

String username = "root";

String password = "";


Connection conn = null;

response.setContentType("text/html");


try {

        Class.forName("com.mysql.jdbc.Driver");


        conn = DriverManager.getConnection(dbUrl, username, password);


        if (conn != null) {
```

```
        }


PreparedStatement ps;
 Integer qno=Integer.valueOf(request.getParameter("qno"));
switch(qno){
case 1:break;
case 2:;
case 3:;
case 4:;
case 5:;
case 6:
session.setAttribute(""+(qno-1)+"",request.getParameter("answer")!=null?request.getParameter("
answer"):0);break;
}
if(qno==6){
conn.close();
%>
<jsp:forward page="result.jsp">
  <jsp:param name="qno" value="1" ></jsp:param>
  </jsp:forward>
```

```
 <%
}
ps=conn.prepareStatement("select * from questions where id=?");
ps.setInt(1,qno);
System.out.print(ps.toString());
ResultSet rs=ps.executeQuery();
rs.next();
%>
<div class="card col-lg-12 ">
<h1 id="timer">0</h1>
 <div class="card-body">
<div class="card-title"><% out.print(rs.getString("question"));%></div>

 <form  class="p-3" action="home.jsp" method="post" name="register">

<div class="form-group col-md-6">

    <input  type="radio" id="o1" name="answer" value="1"  >
     <label for="o1"><%out.print(rs.getString("option1")); %></label>
    </div>
    <div class="form-group col-md-6">
```

```
    <input type="radio" id="o2" name="answer" value="2"  >
     <label for="o2"><%out.print(rs.getString("option2")); %></label>


  </div>
   <div class="form-group col-md-6">
    <input type="radio" id="o3" name="answer" value="3"  >
     <label for="o3"><%out.print(rs.getString("option3")); %></label>


  </div> <div class="form-group col-md-6">
    <input type="radio" id="o4" name="answer" value="4"  >
     <label for="o4"><%out.print(rs.getString("option4")); %></label>


  </div>
   <input name="qno" value=<%out.print(++qno);%> type="hidden">
  <div class="form-group col-md-6">


 <input value="Next" type="submit" class="btn btn-primary">


</div>
</form>
</div>
```

```
</div>


<%
conn.close();
}catch (ClassNotFoundException e) {
		// TODO Auto-generated catch block
		%>
<div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
		<%
			out.print(e.getMessage());
		%>
		 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<%
		e.printStackTrace();
} catch (SQLException e) {
		// TODO Auto-generated catch block
%>
<div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
```

```
		<%
			out.print(e.getMessage());
		%>
		 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<%
		e.printStackTrace();
}catch (NullPointerException e) {
		// TODO Auto-generated catch block
		%>
<div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
		<%
			out.print(e.getMessage());
		%>
		 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<%
```

```
            e.printStackTrace();
}
%>
            </div>
</div>

<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"

integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ
+n"
            crossorigin="anonymous"></script>
<script
            src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooA
o"
            crossorigin="anonymous"></script>
<script
            src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
```

```
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB
6"
        crossorigin="anonymous"></script>
</body>

</html>
```

DemoJSP.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
import="java.sql.*,javax.servlet.http.*,java.util.Enumeration"
pageEncoding="ISO-8859-1"%>

<!doctype html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bootstrap CSS -->
```

```
<link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q
9Ifjh"

crossorigin="anonymous">


<title>RESULT</title>

</head>

<body

style="background-image: url('bg.jpg'); background-color: #cccccc; background-position: center;
background-repeat: no-repeat; background-size: cover;">

<div class="container">

        <div class="row justify-content-lg-around my-5">

                <%

                session.setAttribute("1", 0);

                session.setAttribute("2", 0);

                session.setAttribute("3", 0);

                session.setAttribute("4", 0);

                session.setAttribute("5", 0);


                String dbUrl = "jdbc:mysql://localhost:3306/demo";
```

```
String username = "root";
String password = "";

Connection conn = null;
response.setContentType("text/html");

try {
    Class.forName("com.mysql.jdbc.Driver");
    conn = DriverManager.getConnection(dbUrl, username, password);
    if (conn != null) {
    }
Enumeration<String> names = request.getParameterNames();
String action = request.getParameter("action");
int i;
PreparedStatement ps;
if(request.getParameter("uname").isEmpty() ||
request.getParameter("pass").isEmpty()) {
    %>
    <div class="col-lg-12 alert alert-dismissible fade show alert-primary"
role="alert" >
        <%
            out.print("Login Failed");
```

```
                                %>
                                 <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
  <span aria-hidden="true">&times;</span>
 </button>
                        </div>
                        <%


                }
                switch (action) {
                case "Register":
                        ps=conn.prepareStatement("insert into users values(?,?)");
                        i=1;
                        while(names.hasMoreElements()) {
                                if(i==3) {
                                        break;
                                }
                                ps.setString(i,request.getParameter(names.nextElement()));
                                i++;
                        }


                        ps.executeUpdate();
```

```
                        %>
                        <div class="col-lg-12 alert alert-dismissible fade show alert-primary"
role="alert" >
                                <%
                                        out.print("Registration successful");
                                %>
                                 <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
   <span aria-hidden="true">&times;</span>
 </button>
                        </div>
                        <%
                        break;



                        case "Login":
                                ps=conn.prepareStatement("select * from users where uname=?
and pass=?");
                                i=1;
                                while(names.hasMoreElements()) {
                                        if(i==3) {
                                                break;
```

```
                                        }

ps.setString(i,request.getParameter(names.nextElement()));
                                        i++;
                                    }

                                    ResultSet rs=ps.executeQuery();
                                    boolean flag=false;
                                    while(rs.next()) {

                                        flag=true;
                                    }
                                    if(flag) {

session.setAttribute("username",request.getParameter("uname"));
                    %>
                    <jsp:forward page="home.jsp">
                     <jsp:param name="qno" value="1" ></jsp:param>
                     </jsp:forward>
                    <div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
                            <%
```

```
                        out.print("Login successful");
                    %>
                     <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
  <span aria-hidden="true">&times;</span>
 </button>
                </div>
                <% }else{
                %>
                <div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
                    <%
                        out.print("User does not exist");
                    %>
                     <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
  <span aria-hidden="true">&times;</span>
 </button>
                </div>
                <%
                    }
                break;
```

```
}
conn.close();
} catch (ClassNotFoundException e) {
	// TODO Auto-generated catch block
	%>
<div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
	<%
		out.print(e.getMessage());
	%>
	 <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
  <span aria-hidden="true">&times;</span>
 </button>
</div>
<%
	e.printStackTrace();
} catch (SQLException e) {
	// TODO Auto-generated catch block
%>
<div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
	<%
		out.print(e.getMessage());
```

```
                %>
                 <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
  <span aria-hidden="true">&times;</span>
 </button>
                </div>
                <%
                        e.printStackTrace();
                } catch (NullPointerException e) {
                        // TODO Auto-generated catch block
                        %>
                <div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
                        <%
                                out.print(e.getMessage());
                        %>
                 <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
  <span aria-hidden="true">&times;</span>
 </button>
                </div>
                <%
                        e.printStackTrace();
```

```
                }
                %>
        </div>
</div>


</body>
</html>
```

result.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
import="java.sql.*,javax.servlet.http.*,java.util.Enumeration"
pageEncoding="ISO-8859-1"%>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport"
content="width=device-width, initial-scale=1, shrink-to-fit=no">


<!-- Bootstrap CSS -->
<link rel="stylesheet"
```

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"

integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q
9Ifjh"

crossorigin="anonymous">

  <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4/jquery.min.js"></script>


<title>RESULT</title>


</head>

<body

style="background-image: url('bg.jpg'); background-color: #cccccc; background-position: center;
background-repeat: no-repeat; background-size: cover;">


<div class="container">

<div class="row justify-content-lg-around my-5">

<%

String dbUrl = "jdbc:mysql://localhost:3306/demo";

String username = "root";

String password = "";


Connection conn = null;
```

```
response.setContentType("text/html");


try {
        Class.forName("com.mysql.jdbc.Driver");

        conn = DriverManager.getConnection(dbUrl, username, password);

        if (conn != null) {

        }



PreparedStatement ps;

ResultSet rs;
%>
<div class="card col-lg-12 ">
 <div class="card-body">
<div class="card-title"><% out.print("User : "+session.getAttribute("username"));%></div>
<div class="card-title"><%
```

```
ps=conn.prepareStatement("select answer from questions where id=1");
rs=ps.executeQuery();
rs.next();
if(Integer.valueOf(rs.getString("answer"))==Integer.valueOf( session.getAttribute("1").toString()))
out.print("Q1 : Correct");
else
out.print("Q1 : Wrong");
%>
</div>
<div class="card-title"><%
ps=conn.prepareStatement("select answer from questions where id=2");
rs=ps.executeQuery();
rs.next();

if(Integer.valueOf(rs.getString("answer"))==Integer.valueOf(session.getAttribute("2").toString()))
out.print("Q2 : Correct");
else
out.print("Q2 : Wrong");
%></div>
<div class="card-title"><%
ps=conn.prepareStatement("select answer from questions where id=3");
```

```
rs=ps.executeQuery();
rs.next();

if(Integer.valueOf(rs.getString("answer"))==Integer.valueOf(session.getAttribute("3").toString()))
out.print("Q3 : Correct");
else
out.print("Q3 : Wrong");
%></div>
<div class="card-title"><%
ps=conn.prepareStatement("select answer from questions where id=4");
rs=ps.executeQuery();
rs.next();

if(Integer.valueOf(rs.getString("answer"))==Integer.valueOf(session.getAttribute("4").toString()))
out.print("Q4 : Correct");
else
out.print("Q4 : Wrong");
%></div>
<div class="card-title"><%
ps=conn.prepareStatement("select answer from questions where id=5");
rs=ps.executeQuery();
```

```
rs.next();

if(Integer.valueOf(rs.getString("answer"))==Integer.valueOf(session.getAttribute("5").toString()))
out.print("Q5 : Correct");
else
out.print("Q5 : Wrong");
%></div>
<a href="index.html" >Exit</a>

</div>
</div>

<%
conn.close();

}catch (ClassNotFoundException e) {
		// TODO Auto-generated catch block
		%>
<div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
		<%
```

```
                out.print(e.getMessage());
        %>
         <button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<%
        e.printStackTrace();
} catch (SQLException e) {
        // TODO Auto-generated catch block
%>
<div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
        <%
                out.print(e.getMessage());
        %>
         <button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<%
        e.printStackTrace();
```

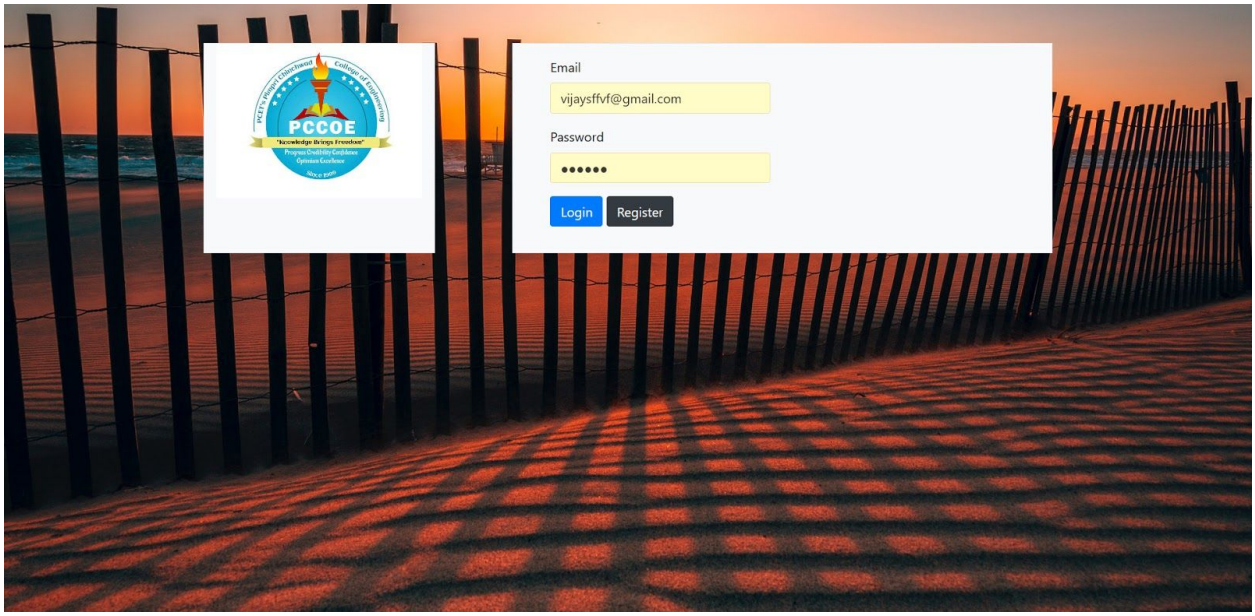
```
}catch (NullPointerException e) {
		// TODO Auto-generated catch block
		%>
<div class="col-lg-12 alert alert-dismissible fade show alert-primary" role="alert" >
		<%
			out.print(e.getMessage());
		%>
		 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<%
		e.printStackTrace();
}
%>


		</div>
</div>


<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
```

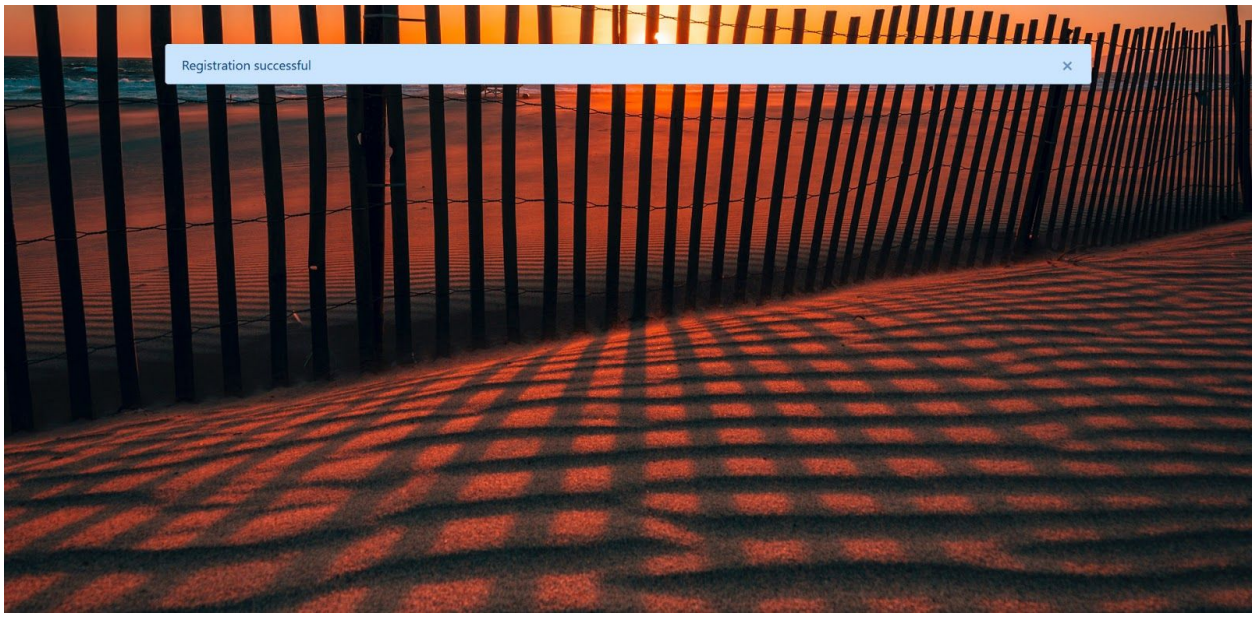
```
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"

integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ
+n"

        crossorigin="anonymous"></script>

<script

        src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooA
o"

        crossorigin="anonymous"></script>

<script

        src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"

integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB
6"

        crossorigin="anonymous"></script>

</body>


</html>
```

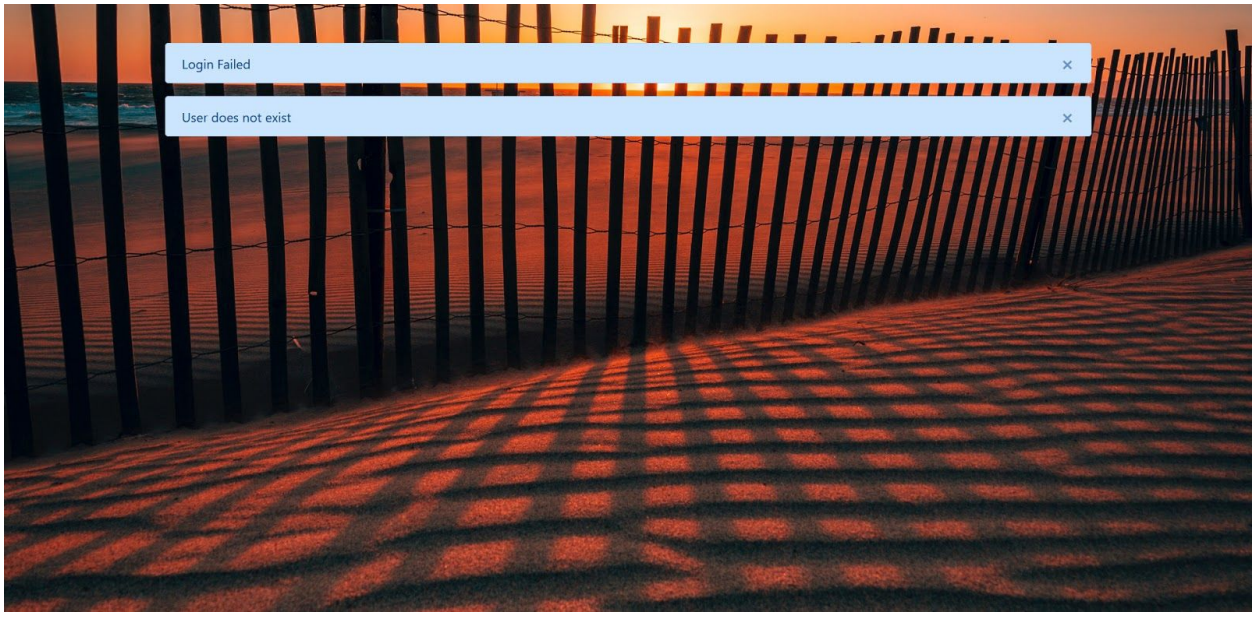
SQL Queries for Database Schema

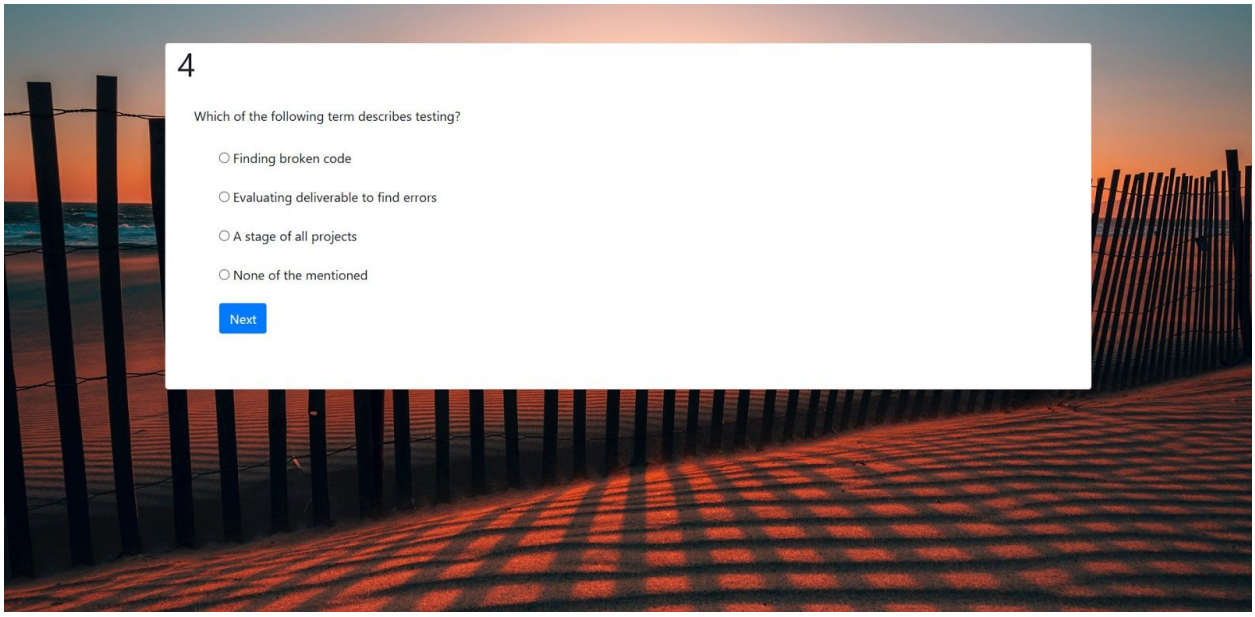
```
CREATE TABLE users(
        uname varchar(20),
        pass varchar(20),
        PRIMARY KEY(uname)
);


CREATE TABLE questions(
        id int AUTO_INCREMENT,
        question  varchar(100),
        option1 varchar(100),
        option2 varchar(100),
        option3 varchar(100),
        option4 varchar(100),
        answer varchar(100),
        PRIMARY KEY(id)
);
```

Output Screenshots






Login Failed
User does not exist


4
Which of the following term describes testing?
Finding broken code
Evaluating deliverable to find errors
A stage of all projects
None of the mentioned
Next

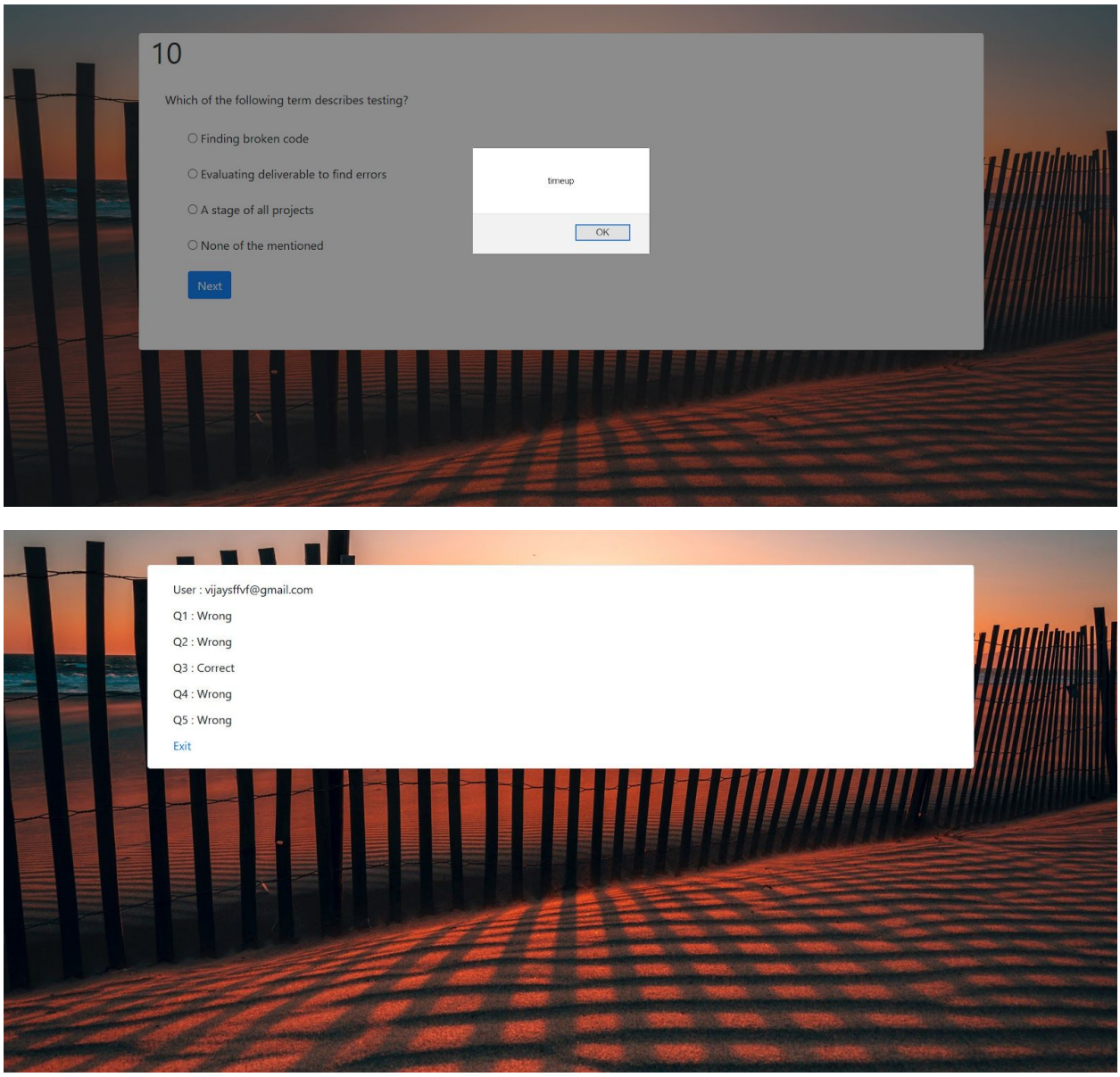


Fig. Quiz Application Screenshots

Test Scenario

Common for all scenarios :

The application can be launched by running the index file by running as “Run on Server”. A browser window will open with an index.html webpage loaded that will show the homepage of the application consisting of a login/register form with username, password fields and login register buttons. The user can click on the username field to focus it and focus can be switched to the password field using the TAB key on the keyboard. If some text is already present in the textboxes then it is automatically selected on gaining focus. Login & Register buttons are available with blue and dark grey colours respectively.

Scenario 1: Positive scenario where registration is successful

User clicks the username textfield to focus it. User enters registered username.Press TAB key. Cursor moves to the password field and the user enters the corresponding password. Click on the register button and the webpage is loaded with an alert regarding successful registration.

Scenario 2: Positive scenario where login is successful

User clicks the username textfield to focus it. User enters registered username.Press TAB key. Cursor moves to the password field and the user enters the corresponding password. Click on the login button and the webpage is loaded with the first question.

Scenario 3: Negative scenario where either username or password is empty

There can be two combinations where equivalence partitioning can be applied. One of the two fields is taken as empty each time which may represent equivalence class

Scenario 4: Negative scenario where both username and password is empty

Scenario 5: Negative scenario where either username or password is wrong

There can be two combinations where equivalence partitioning can be applied. One of the two fields is taken as wrong each time which may represent equivalence class

Scenario 6: Negative scenario both username and password is wrong

Scenario 7: Positive scenario close button(tab/browser) terminates the application(tab/browser)

Common for all scenarios :

After successful login, the test screen with preloaded questions and options is launched. The timer is also shown on the top corner side of the question. The webpage has the same background as that of the login page except the overlying div of the login window is replaced by another div that consists of question, options, next button and timer. At the top of the question there is a second-based timer with starting value 10 and for each second it is decreased by 1. Timer decreases itself every second. If the user answers a question by ticking any one option and clicks on the next button then the next question is loaded. If the user can't answer in 10 seconds then timer reaches zero and an alert is given that shows “timeup”, after clicking ok for alert, the next question is loaded. After the next button click, the next question is loaded. Either the time is up or all questions are attempted, the result page is loaded and username, summary of all answers and exit button is shown. After clicking exit, the user is taken to the login page that is the homepage of the application.

Scenario 1: Positive scenario where user selects an option and clicks the next button. Question is updated.

By default no option is selected and the user may select one of the options and click the next button. The next question is visible.

Scenario 2: Negative scenario where timer is not working

Scenario 3: Negative scenario where next button is not working

Scenario 4: Negative scenario where ALT+F4 or CTRL+W is pressed

Application is terminated.

Scenario 5: Negative scenario where selected option is not recorded

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 6: Negative scenario where correct option is selected but result is marked as wrong

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 7: Negative scenario where wrong option is selected but result is marked as correct

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 8: Positive scenario where wrong option is selected and result is marked as wrong

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 9: Positive scenario where correct option is selected and result is marked as correct

There can be four combinations where equivalence partitioning can be applied. One of the four options is selected each time which may represent equivalence class

Scenario 10: Negative scenario where repainting the result panel may overlap with existing components

Scenario 11: Negative scenario where all counts of correct,wrong and unanswered responses is wrong

There can be one combination where equivalence partitioning can be applied.

Scenario 12: Negative scenario where count of any one(correct,wrong and unanswered responses) is wrong

There can be three combinations where equivalence partitioning can be applied.

Scenario 13: Negative scenario where count of any two(correct,wrong and unanswered responses) is wrong

There can be three combinations where equivalence partitioning can be applied.

Scenario 14: Positive scenario where count of correct,wrong and unanswered responses is correct

There can be one combination where equivalence partitioning can be applied.

Scenario 15: Positive scenario where exit button from result page takes user to home page.

Test Cases

Test Case 1

- Test Precondition : application must be accessible through any web browser
- Test Sequence: SMK/MOD1/0001
- Test Scenario Traceability : TS00
- Type Of Testing : Smoke testing
- Test Case Name and Number: SMK/MOD1/0001
- Valid/invalid condition : valid condition
- Objective : to check whether the application is working or not after loading index.html file.
- Expected Results : application accessible and index page is loaded.

Test Case 2

- Test Precondition : username and password is entered correctly and clicked on the register button.
- Test Sequence: FUN/MOD1/0001
- Test Scenario Traceability : TS02
- Type Of Testing : Functional testing
- Test Case Name and Number: FUN/MOD1/0001
- Valid/invalid condition : valid condition
- Objective : user should be registered.
- Expected Results : user is added to the database and successful alert is given.

Test Case 3

- Test Precondition : username and password is entered correctly and clicked on the login button.
- Test Sequence: FUN/MOD1/0002
- Test Scenario Traceability : TS02
- Type Of Testing : Functional testing
- Test Case Name and Number: FUN/MOD1/0002
- Valid/invalid condition : valid condition
- Objective : user should be logged in and able to give a test.
- Expected Results : user is redirected to give the test and taken to the first question.

Test Case 4

- Test Precondition : Wrong details are entered on the login page and clicked on the login button.
- Test Sequence: FUN/MOD1/0003
- Test Scenario Traceability : TS02
- Type Of Testing : Functional testing
- Test Case Name and Number: FUN/MOD1/0003
- Valid/invalid condition : invalid condition
- Objective : only users with correct details can log in.
- Expected Results : user should be able to see an error message.

Test Case 5

- Test Precondition : Details(username & password) are not entered on the login page and clicked on the login/register button.
- Test Sequence: FUN/MOD1/0004
- Test Scenario Traceability : TS02
- Type Of Testing : Functional testing

- Test Case Name and Number: FUN/MOD1/0004
- Valid/invalid condition : invalid condition
- Objective : only users with correct details can log in/register (empty should be invalid).
- Expected Results : user should be able to see an error message

Test Case 6

- Test Precondition : all questions are attempted
- Test Sequence: FUN/MOD1/0005
- Test Scenario Traceability : TS02
- Type Of Testing : Functional testing
- Test Case Name and Number: FUN/MOD1/0005
- Valid/invalid condition : valid condition
- Objective : result screen is displayed.
- Expected Results : to display the result to the user.

Test Case 7

- Test Precondition : user selects an option and clicks the next button. Question is updated.
- Test Sequence: FUN/MOD1/0006
- Test Scenario Traceability : TS02
- Type Of Testing : Functional testing
- Test Case Name and Number: FUN/MOD1/0006
- Valid/invalid condition : valid condition
- Objective : to display the next question.
- Expected Results : The next question will be displayed on screen, for final question result will be displayed.

Test Case 8

- Test Precondition : user selects an option and clicks the next button. Question is updated.
- Type Of Testing : Functional testing
- Test Case Name and Number: FUN/MOD1/0007
- Test Scenario Traceability : TS02
- Valid/invalid condition : valid condition
- Objective : to display the next question.
- Expected Results : next question will be displayed on screen.

Test Case 9

- Test Precondition : ALT+F4 or close tab or close browser button is clicked.
- Test Sequence: FUN/MOD1/0008
- Test Scenario Traceability : TS02
- Type Of Testing : Functional testing
- Test Case Name and Number: FUN/MOD1/0008
- Valid/invalid condition : invalid condition
- Objective : to terminate the test
- Expected Results : Application is terminated.

Test Case 10

- Test Precondition : User clicks the exit button on result screen
- Test Sequence: FUN/MOD1/0009
- Test Scenario Traceability : TS02
- Type Of Testing : Functional testing
- Test Case Name and Number: FUN/MOD1/0009
- Valid/invalid condition : valid condition
- Objective : to get back to home page
- Expected Results : User is logged out & application is loaded with homepage.

Selenium IDE

Exit

1 open /jspDemo

2 set window size 1527x824

3 click css=html

4 click id=uname

5 type id=uname vij

6 send keys id=uname ${KEY_DOWN}

7 type id=uname vijaysffvf@gmail.com

8 click id=pass

9 type id=pass 123456

10 click name=action

11 click id=o2

12 click css=.btn

13 click css=.form-group:nth-child(4) > label

14 click css=.btn

15 click css=.form-group:nth-child(4) > label

16 click css=.btn

17 click css=.form-group:nth-child(4) > label

18 click css=.btn

19 click css=.form-group:nth-child(4) > label

20 click css=.btn

21 click linkText=Exit

Running 'exit' 17:08:16

1. open on /jspDemo OK 17:08:17

2. setWindowSize on 1527x824 OK 17:08:17

3. click on css=html OK 17:08:17

4. click on id=uname OK 17:08:18

5. type on id=uname with value vij OK 17:08:30

6. sendKeys on id=uname with value ${KEY_DOWN} OK 17:08:30

7. type on id=uname with value vijaysffvf@gmail.com OK 17:08:30

8. click on id=pass OK 17:08:31

9. type on id=pass with value 123456 OK 17:08:31

10. click on name=action OK 17:08:31

11. click on id=o2 OK 17:08:31

12. click on css=.btn OK 17:08:32

13. click on css=.form-group:nth-child(4) > label OK 17:08:32

14. click on css=.btn OK 17:08:33

15. click on css=.form-group:nth-child(4) > label OK 17:08:33

16. click on css=.btn OK 17:08:34

17. click on css=.form-group:nth-child(4) > label OK 17:08:34

18. click on css=.btn OK 17:08:35

19. click on css=.form-group:nth-child(4) > label OK 17:08:35

20. click on css=.btn OK 17:08:36

21. click on linkText=Exit OK 17:08:36

'exit' completed successfully 17:08:37

loginFail

1 open /JSPDemo/

2 set window size 1527x824

3 click id=uname

4 type id=uname vijaysffvf@gmail.com

5 click id=pass

6 click name=register

7 click name=action

Running 'loginFail' 17:10:30

1. open on /JSPDemo/ OK 17:10:31

2. setWindowSize on 1527x824 OK 17:10:31

3. click on id=uname OK 17:10:31

4. type on id=uname with value vijaysffvf@gmail.com OK 17:10:32

5. click on id=pass OK 17:10:33

6. click on name=register OK 17:10:33

7. click on name=action OK 17:10:33

'loginFail' completed successfully 17:10:33

loginSuccess

1 open /JSPDemo/

2 set window size 1527x824

3 click id=uname

4 type id=uname vija

5 send keys id=uname ${KEY_DOWN}

6 type id=uname vijaysffvf@gmail.com

7 click id=pass

8 type id=pass 123456

9 click name=action

10 click id=o2

11 click css=.btn

12 click id=o2

13 click css=.btn

14 click css=.form-group:nth-child(3)

15 click id=o3

16 click css=.btn

17 click id=o2

18 click css=.btn

19 click id=o1

20 click css=.btn

Running 'loginSuccess' 17:12:16

1. open on /JSPDemo/ OK 17:12:16

2. setWindowSize on 1527x824 OK 17:12:17

3. click on id=uname OK 17:12:17

4. type on id=uname with value vija OK 17:12:18

5. sendKeys on id=uname with value ${KEY_DOWN} OK 17:12:18

6. type on id=uname with value vijaysffvf@gmail.com OK 17:12:18

7. click on id=pass OK 17:12:18

8. type on id=pass with value 123456 OK 17:12:18

9. click on name=action OK 17:12:19

10. click on id=o2 OK 17:12:19

11. click on css=.btn OK 17:12:19

12. click on id=o2 OK 17:12:20

13. click on css=.btn OK 17:12:20

14. click on css=.form-group:nth-child(3) OK 17:12:20

15. click on id=o3 OK 17:12:21

16. click on css=.btn OK17:12:21

17. click on id=o2 OK 17:12:21

18. click on css=.btn OK 17:12:22

19. click on id=o1 OK 17:12:22

20. click on css=.btn OK 17:12:23

'loginSuccess' completed successfully 17:12:23

registerFailure

1 open /JSPDemo/

2 set window size 1527x824

3 click id=uname

4 type id=uname admin

5 click id=pass

6 type id=pass ee

7 click css=.btn-dark

8 click id=uname

9 type id=uname admin@gmail.com

10 click id=pass

11 click id=pass

12 type id=pass

13 click name=register

14 click css=.btn-dark

15 close

Running 'registerFailure' 17:12:43

1. open on /JSPDemo/ OK 17:12:44

2. setWindowSize on 1527x824 OK 17:12:44

3. click on id=uname OK 17:12:44

4. type on id=uname with value admin OK 17:12:45

5. click on id=pass OK 17:12:45

6. type on id=pass with value ee OK 17:12:46

7. click on css=.btn-dark OK 17:12:46

8. click on id=uname OK 17:12:46

9. type on id=uname with value admin@gmail.com OK 17:12:46

10. click on id=pass OK 17:12:46

11. click on id=pass OK 17:12:47

12. type on id=pass OK 17:12:47

13. click on name=register OK 17:12:47

14. click on css=.btn-dark OK 17:12:47

15. close OK 17:12:47

'registerFailure' completed successfully 17:12:47

registerSuccessful:

1 open /JSPDemo/

2 set window size 1527x824

3 click id=uname

4 type id=uname a@gmail.com

5 click id=pass

6 type id=pass 12345

7 click css=.btn-dark

8 close

Running 'registerSuccessful' 17:13:07

1. open on /JSPDemo/ OK 17:13:08

2. setWindowSize on 1527x824 OK 17:13:08

3. click on id=uname OK 17:13:08

4. type on id=uname with value a@gmail.com OK 17:13:09

5. click on id=pass OK 17:13:09

6. type on id=pass with value 12345 OK 17:13:09

7. click on css=.btn-dark OK 17:13:09

8. close OK 17:13:09

'registerSuccessful' completed successfully 17:13:09

timeup

1 open /JSPDemo/

2 set window size 1527x824

3 click id=uname

4 type id=uname v

5 send keys id=unam ${KEY_DOWN}

6 type id=uname vijaysffvf@gmail.com

7 click id=pass

8 type id=pass 123456

9 click name=action

10 click id=o2

11 click css=.btn

12 click id=o2

13 click css=.btn

14 click id=o2

15 click css=.btn

16 assert alert timeup

17 click css=.card-title:nth-child(5)

18 click linkText=Exit

Running 'timeup' 17:13:25

1. open on /JSPDemo/ OK 17:13:25

2. setWindowSize on 1527x824 OK 17:13:26

3. click on id=uname OK 17:13:26

4. type on id=uname with value v OK 17:13:27

5. sendKeys on id=uname with value ${KEY_DOWN} OK 17:13:27

6. type on id=uname with value vijaysffvf@gmail.com OK 17:13:27

7. click on id=pass OK 17:13:27

8. type on id=pass with value 123456 OK 17:13:28

9. click on name=action OK 17:13:28

10. click on id=o2 OK 17:13:28

11. click on css=.btn OK 17:13:29

12. click on id=o2 OK 17:13:29

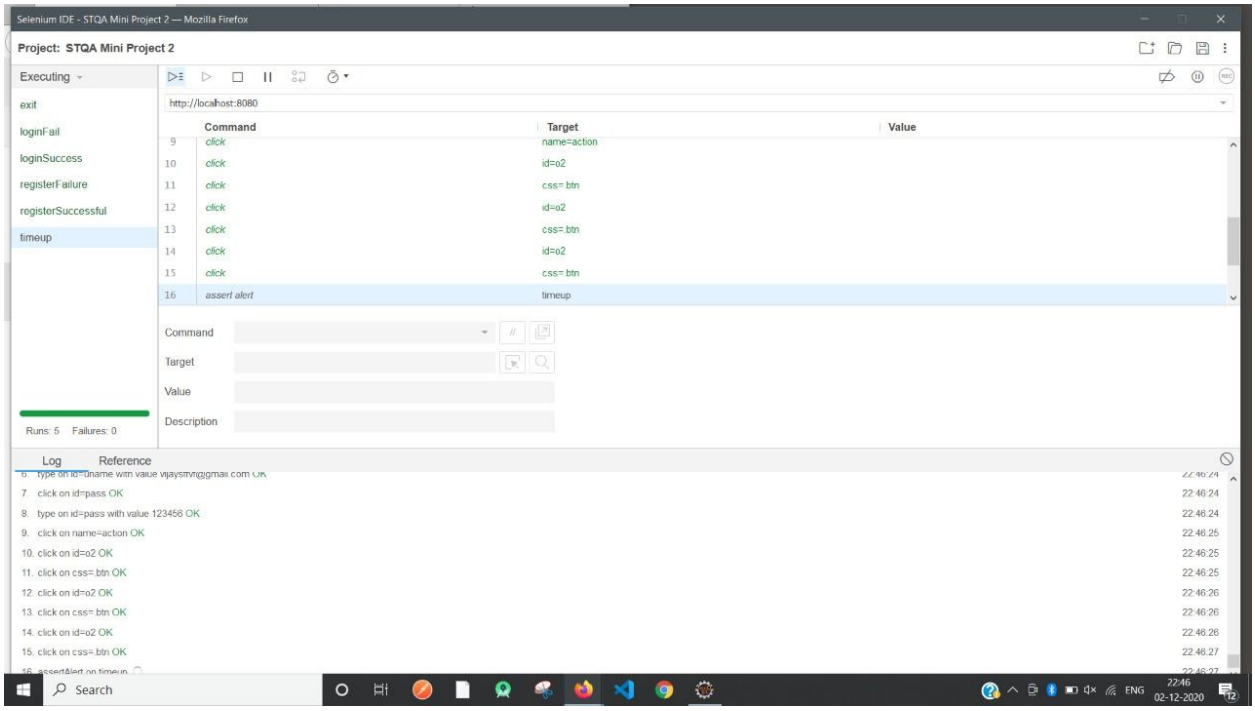
13. click on css=.btn OK 17:13:29

14. click on id=o2 OK 17:13:30

15. click on css=.btn OK 17:13:30

16. assertAlert on timeup 17:13:30

'timeup'' completed successfully 17:13:30



Selenium Grid

seleniumGrid.java

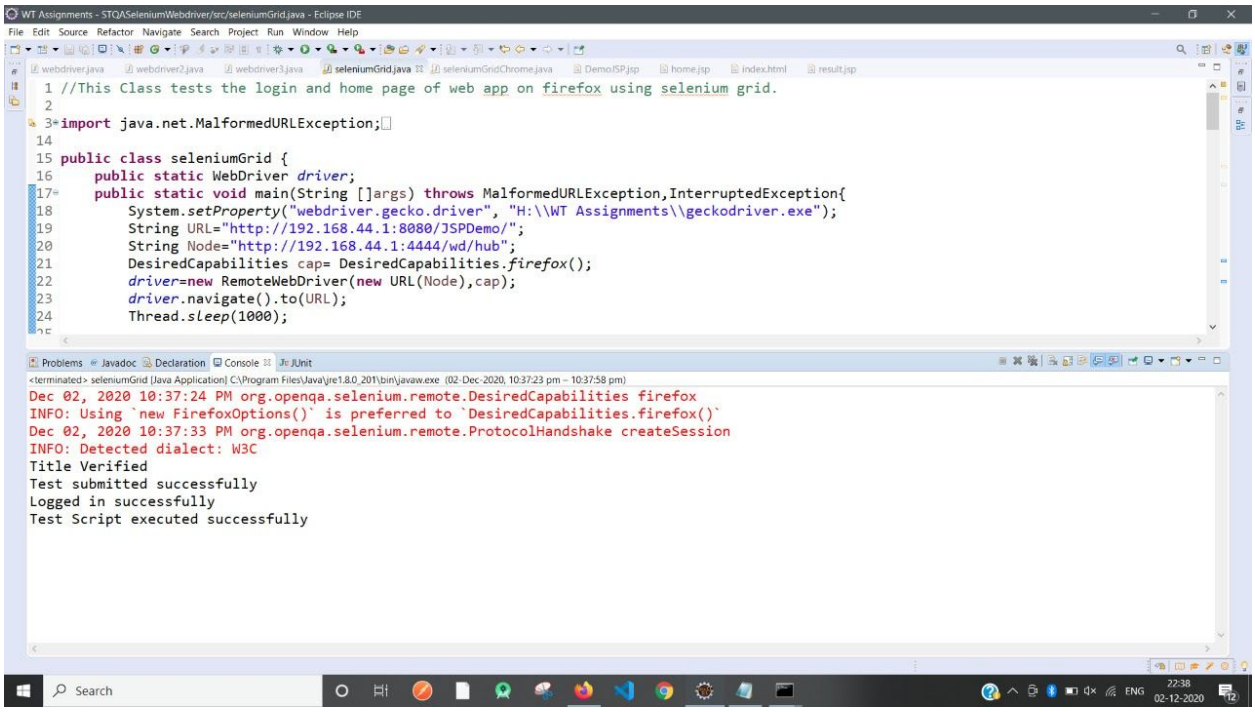
```
//This Class tests the login and home page of web app on firefox using selenium grid.

import java.net.MalformedURLException;

import java.net.URL;

import java.sql.Connection;

import java.sql.DriverManager;

import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
public class seleniumGrid {
public static WebDriver driver;
public static void main(String []args) throws MalformedURLException,InterruptedException{
        System.setProperty("webdriver.gecko.driver", "H:\\WT Assignments\\geckodriver.exe");
        String URL="http://192.168.44.1:8080/JSPDemo/";
        String Node="http://192.168.44.1:4444/wd/hub";
        DesiredCapabilities cap= DesiredCapabilities.firefox();
        driver=new RemoteWebDriver(new URL(Node),cap);
        driver.navigate().to(URL);
        Thread.sleep(1000);
        //Title check
        String actual_title =driver.getTitle();
        String title="HOME";
        if(actual_title.contentEquals(title))
        {
                System.out.println("Title Verified");
```

```
}
else
        System.out.println("Title Mismatched");
//Checking database
Connection conn = null;
String dbUrl = "jdbc:mysql://localhost:3306/demo";
String user = "root";
String pass = "";
try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(dbUrl, user, pass);
        if (conn != null) {
                System.out.print("Connected successfully to database");
        }
}catch(Exception e) {
        System.out.print("Connection Failed");
}
//Locating HTMLElements
WebElement username=driver.findElement(By.name("uname"));
username.click();
username.sendKeys("x@gmail.com");
```

```
WebElement password=driver.findElement(By.name("pass"));
password.click();
password.sendKeys("123456");
System.out.println("Test submitted successfully");
WebElement login=driver.findElement(By.id("login"));
login.click();
System.out.println("Logged in successfully");
WebElement option2=driver.findElement(By.id("o2"));
option2.click();
try {
        Thread.sleep(1000);
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
WebElement next=driver.findElement(By.id("next"));
next.click();
try {
        Thread.sleep(12000);
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
```

```
            e.printStackTrace();
        }
        driver.quit();
        System.out.println("Test Script executed successfully");
        System.exit(0);
}
}
```



seleniumGridChrome.java

//This Class tests the login and home page of web app on chrome browser using selenium grid.

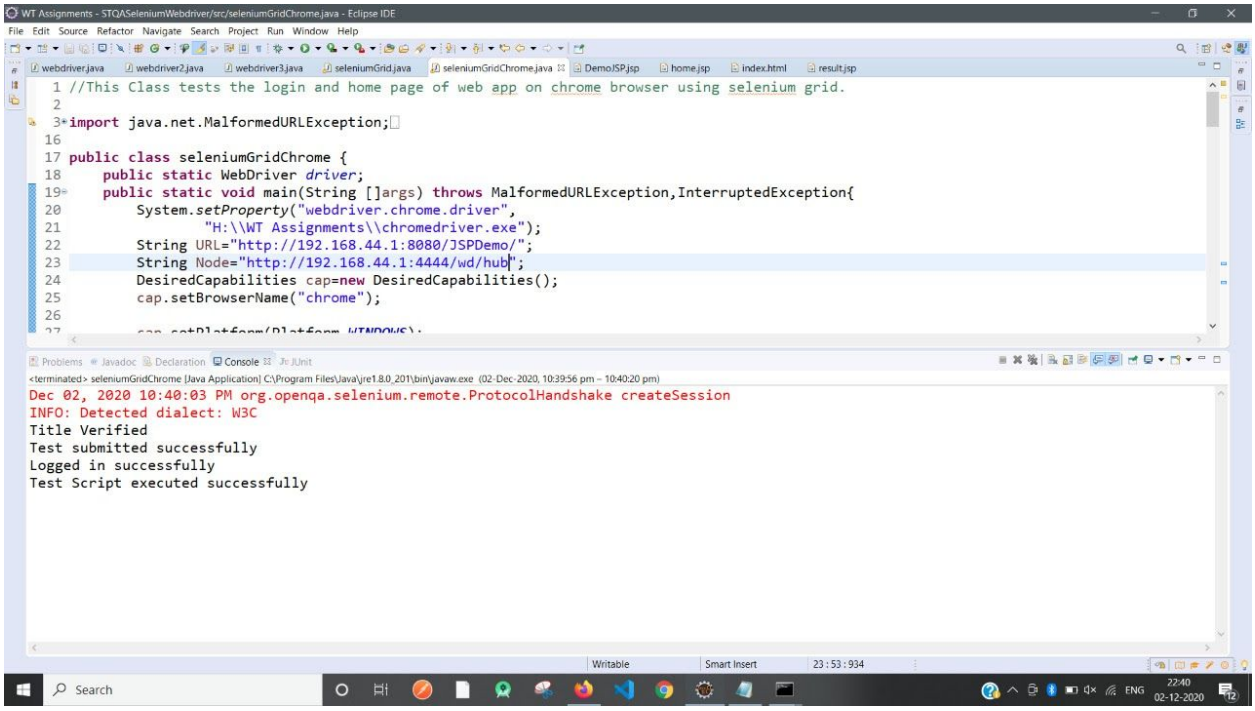
import java.net.MalformedURLException;

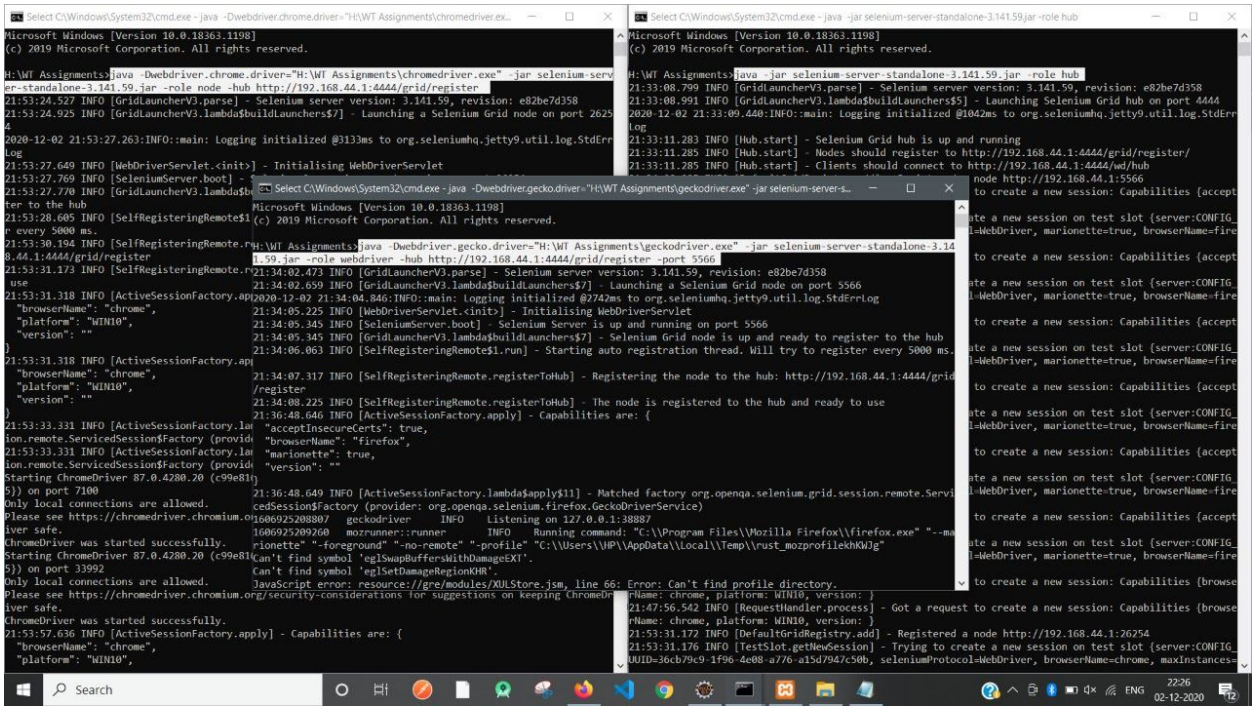
```
import java.net.URL;
import java.sql.Connection;
import java.sql.DriverManager;
import org.openqa.selenium.By;
import org.openqa.selenium.Platform;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import org.openqa.selenium.remote.RemoteWebDriver;
public class seleniumGridChrome {
public static WebDriver driver;
public static void main(String []args) throws MalformedURLException,InterruptedException{
        System.setProperty("webdriver.chrome.driver",
                        "H:\\WT Assignments\\chromedriver.exe");
        String URL="http://192.168.44.1:8080/JSPDemo/";
        String Node="http://192.168.44.1:4444/wd/hub";
        DesiredCapabilities cap=new DesiredCapabilities();
        cap.setBrowserName("chrome");
        cap.setPlatform(Platform.WINDOWS);
```

```
ChromeOptions options = new ChromeOptions();
options.setHeadless(true);
options.merge(cap);
driver=new RemoteWebDriver(new URL(Node),options);
driver.navigate().to(URL);
//Title check
String actual_title =driver.getTitle();
String title="HOME";
if(actual_title.contentEquals(title))
{
	System.out.println("Title Verified");
}
else
	System.out.println("Title Mismatched");
//		Checking database
Connection conn = null;
String dbUrl = "jdbc:mysql://localhost:3306/demo";
String user = "root";
String pass = "";
try {
	Class.forName("com.mysql.jdbc.Driver");
```

```
            conn = DriverManager.getConnection(dbUrl, user, pass);
            if (conn != null) {
                  System.out.print("Connected successfully to database");
            }
}catch(Exception e) {
            System.out.print("Connection Failed");
}
//Locating HTMLElements
WebElement username=driver.findElement(By.name("uname"));
username.click();
username.sendKeys("x@gmail.com");
WebElement password=driver.findElement(By.name("pass"));
password.click();
password.sendKeys("123456");
System.out.println("Test submitted successfully");
WebElement login=driver.findElement(By.id("login"));
login.click();
System.out.println("Logged in successfully");
WebElement option2=driver.findElement(By.id("o2"));
option2.click();
try {
```

```
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        WebElement next=driver.findElement(By.id("next"));
        next.click();
        try {
            Thread.sleep(12000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        driver.quit();
        System.out.println("Test Script executed successfully");
        System.exit(0);
}
}
```


WT Assignments - STQA\SeleniumWebdriver/src/seleniumGridChrome.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
webdriver.java
webdriver2.java
webdriver3.java
seleniumGrid.java
seleniumGridChrome.java
DemoJSP.jsp
home.jsp
index.html
result.jsp
1 //This Class tests the login and home page of web app on chrome browser using selenium grid.
2
3*import java.net.MalformedURLException;
16
17 public class seleniumGridChrome {
18 public static WebDriver driver;
19 public static void main(String []args) throws MalformedURLException,InterruptedException{
20 System.setProperty("webdriver.chrome.driver",
21 "H:\\WT Assignments\\chromedriver.exe");
22 String URL="http://192.168.44.1:8080/JSPDemo/";
23 String Node="http://192.168.44.1:4444/wd/hub";
24 DesiredCapabilities cap=new DesiredCapabilities();
25 cap.setBrowserName("chrome");
Problems Javadoc Declaration Console JUnit
<terminated> seleniumGridChrome [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (02-Dec-2020, 10:39:56 pm – 10:40:20 pm)
Dec 02, 2020 10:40:03 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Title Verified
Test submitted successfully
Logged in successfully
Test Script executed successfully
Writable
Smart Insert
23 : 53 : 934
Search
ENG
22:40
02-12-2020


Select C:\Windows\System32\cmd.exe - java -Dwebdriver.chrome.driver="H:\WT Assignments\chromedriver.ex...
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.
H:\WT Assignments>java -Dwebdriver.chrome.driver="H:\WT Assignments\chromedriver.exe" -jar selenium-server-standalone-3.141.59.jar -role node -hub http://192.168.44.1:4444/grid/register
21:53:24.527 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
21:53:24.925 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Launching a Selenium Grid node on port 2625
4
2020-12-02 21:53:27.263:INFO::main: Logging initialized @3133ms to org.seleniumhq.jetty9.util.log.StdErr
Log
21:53:27.649 INFO [WebDriverServlet.<init>] - Initialising WebDriverServlet
21:53:27.769 INFO [SeleniumServer.boot] -
21:53:27.770 INFO [GridLauncherV3.lambda$b
ter to the hub
21:53:28.605 INFO [SelfRegisteringRemote$1
r every 5000 ms.
21:53:30.194 INFO [SelfRegisteringRemote.r
8.44.1:4444/grid/register
21:53:31.173 INFO [SelfRegisteringRemote.r
use
21:53:31.318 INFO [ActiveSessionFactory.ap
"browserName": "chrome",
"platform": "WIN10",
"version": ""
}
21:53:31.318 INFO [ActiveSessionFactory.ap
"browserName": "chrome",
"platform": "WIN10",
"version": ""
}
21:53:33.331 INFO [ActiveSessionFactory.la
ion.remote.ServicedSession$Factory (provid
21:53:33.331 INFO [ActiveSessionFactory.la
ion.remote.ServicedSession$Factory (provid
Starting ChromeDriver 87.0.4280.20 (c99e81
5}) on port 7100
Only local connections are allowed.
Please see https://chromedriver.chromium.or
iver safe.
ChromeDriver was started successfully.
Starting ChromeDriver 87.0.4280.20 (c99e81
5}) on port 33992
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDr
iver safe.
ChromeDriver was started successfully.
21:53:57.636 INFO [ActiveSessionFactory.apply] - Capabilities are: {
"browserName": "chrome",
"platform": "WIN10",
Select C:\Windows\System32\cmd.exe - java -Dwebdriver.gecko.driver="H:\WT Assignments\geckodriver.exe" -jar selenium-server-s...
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.
H:\WT Assignments>java -Dwebdriver.gecko.driver="H:\WT Assignments\geckodriver.exe" -jar selenium-server-standalone-3.14
1.59.jar -role webdriver -hub http://192.168.44.1:4444/grid/register -port 5566
21:34:02.473 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
21:34:02.659 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Launching a Selenium Grid node on port 5566
2020-12-02 21:34:04.846:INFO::main: Logging initialized @2742ms to org.seleniumhq.jetty9.util.log.StdErrLog
21:34:05.225 INFO [WebDriverServlet.<init>] - Initialising WebDriverServlet
21:34:05.345 INFO [SeleniumServer.boot] - Selenium Server is up and running on port 5566
21:34:05.345 INFO [GridLauncherV3.lambda$buildLaunchers$7] - Selenium Grid node is up and ready to register to the hub
21:34:06.063 INFO [SelfRegisteringRemote$1.run] - Starting auto registration thread. Will try to register every 5000 ms.
21:34:07.317 INFO [SelfRegisteringRemote.registerToHub] - Registering the node to the hub: http://192.168.44.1:4444/grid
/register
21:34:08.225 INFO [SelfRegisteringRemote.registerToHub] - The node is registered to the hub and ready to use
21:36:48.646 INFO [ActiveSessionFactory.apply] - Capabilities are: {
"acceptInsecureCerts": true,
"browserName": "firefox",
"marionette": true,
"version": ""
21:36:48.649 INFO [ActiveSessionFactory.lambda$apply$11] - Matched factory org.openqa.selenium.grid.session.remote.Servi
cedSession$Factory (provider: org.openqa.selenium.firefox.GeckoDriverService)
1606925208807 geckodriver INFO Listening on 127.0.0.1:38887
1606925209260 mozrunner::runner INFO Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "--ma
rionette" "-foreground" "-no-remote" "-profile" "C:\\Users\\HP\\AppData\\Local\\Temp\\rust_mozprofilekhKWJg"
Can't find symbol 'eglSwapBuffersWithDamageEXT'.
Can't find symbol 'eglSetDamageRegionKHR'.
JavaScript error: resource://gre/modules/XULStore.jsm, line 66: Error: Can't find profile directory.
Select C:\Windows\System32\cmd.exe - java -jar selenium-server-standalone-3.141.59.jar -role hub
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.
H:\WT Assignments>java -jar selenium-server-standalone-3.141.59.jar -role hub
21:33:08.799 INFO [GridLauncherV3.parse] - Selenium server version: 3.141.59, revision: e82be7d358
21:33:08.991 INFO [GridLauncherV3.lambda$buildLaunchers$5] - Launching Selenium Grid hub on port 4444
2020-12-02 21:33:09.440:INFO::main: Logging initialized @1042ms to org.seleniumhq.jetty9.util.log.StdErr
log
21:33:11.283 INFO [Hub.start] - Selenium Grid hub is up and running
21:33:11.285 INFO [Hub.start] - Nodes should register to http://192.168.44.1:4444/grid/register/
21:33:11.285 INFO [Hub.start] - Clients should connect to http://192.168.44.1:4444/wd/hub
node http://192.168.44.1:5566
to create a new session: Capabilities {accept
ate a new session on test slot {server:CONFIG_
l=WebDriver, marionette=true, browserName=fire
to create a new session: Capabilities {accept
ate a new session on test slot {server:CONFIG_
l=WebDriver, marionette=true, browserName=fire
to create a new session: Capabilities {accept
ate a new session on test slot {server:CONFIG_
l=WebDriver, marionette=true, browserName=fire
to create a new session: Capabilities {accept
ate a new session on test slot {server:CONFIG_
l=WebDriver, marionette=true, browserName=fire
to create a new session: Capabilities {accept
ate a new session on test slot {server:CONFIG_
l=WebDriver, marionette=true, browserName=fire
to create a new session: Capabilities {accept
to create a new session: Capabilities {browse
rName: chrome, platform: WIN10, version: }
21:47:56.542 INFO [RequestHandler.process] - Got a request to create a new session: Capabilities {browse
rName: chrome, platform: WIN10, version: }
21:53:31.172 INFO [DefaultGridRegistry.add] - Registered a node http://192.168.44.1:26254
21:53:31.176 INFO [TestSlot.getNewSession] - Trying to create a new session on test slot {server:CONFIG_
UUID=36cb79c9-1f96-4e08-a776-a15d7947c50b, seleniumProtocol=WebDriver, browserName=chrome, maxInstances=
Search
ENG
22:26
02-12-2020

Selenium WebDriver

webdriver.java

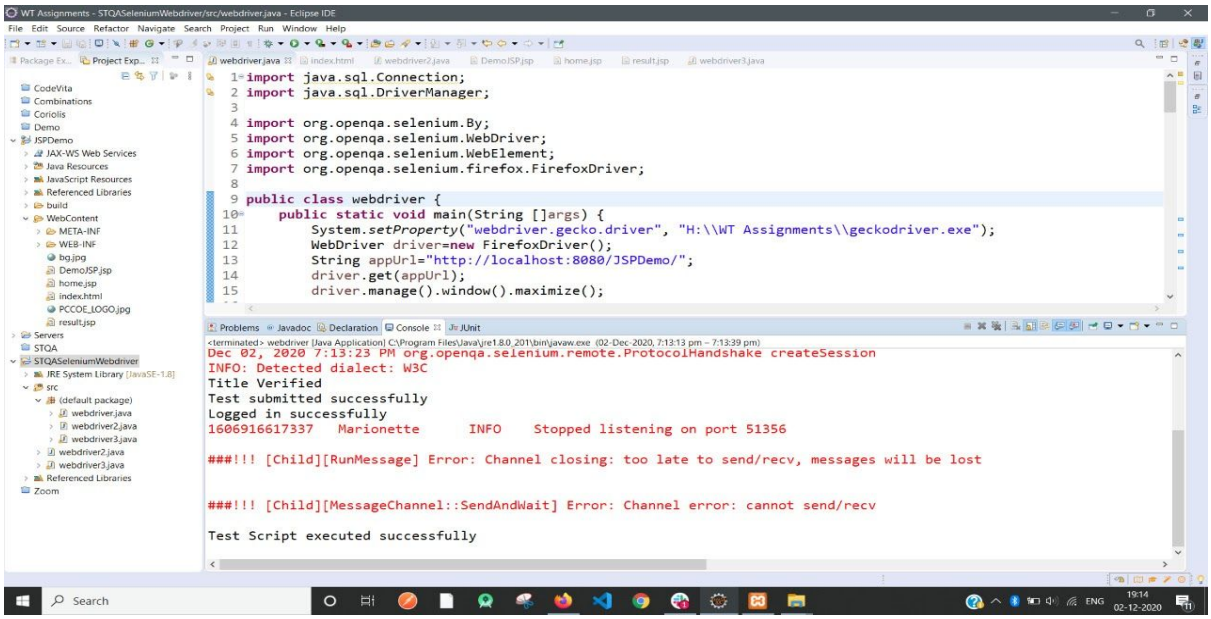
```
//This Class tests the login,home page and exit feature of web app on firefox using selenium webdriver.

import java.sql.Connection;

import java.sql.DriverManager;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class webdriver {

public static void main(String []args) {

        System.setProperty("webdriver.gecko.driver", "H:\\WT Assignments\\geckodriver.exe");

        WebDriver driver=new FirefoxDriver();

        String appUrl="http://localhost:8080/JSPDemo/";

        driver.get(appUrl);

        driver.manage().window().maximize();

        //Title check

        String actual_title =driver.getTitle();

        String title="HOME";

        if(actual_title.contentEquals(title))
```

```
{
        System.out.println("Title Verified");
}
else
        System.out.println("Title Mismatched");
//              Checking database
Connection conn = null;
String dbUrl = "jdbc:mysql://localhost:3306/demo";
String user = "root";
String pass = "";
try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(dbUrl, user, pass);
        if (conn != null) {
                System.out.print("Connected successfully to database");
        }
}catch(Exception e) {
        System.out.print("Connection Failed");
}
//Locating HTMLElements
WebElement username=driver.findElement(By.name("uname"));
```

```
username.click();
username.sendKeys("x@gmail.com");
WebElement password=driver.findElement(By.name("pass"));
password.click();
password.sendKeys("123456");
System.out.println("Test submitted successfully");
WebElement login=driver.findElement(By.id("login"));
login.click();
System.out.println("Logged in successfully");
WebElement option2=driver.findElement(By.id("o2"));
option2.click();
try {
        Thread.sleep(1000);
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
WebElement next=driver.findElement(By.id("next"));
next.click();
try {
        Thread.sleep(3000);
```

```
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
driver.navigate().to("http://localhost:8080/JSPDemo/result.jsp");
try {
        Thread.sleep(2000);
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
WebElement exit=driver.findElement(By.linkText("Exit"));
exit.click();
try {
        Thread.sleep(2000);
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
driver.close();
System.out.println("Test Script executed successfully");
```

```
        System.exit(0);
}}
```



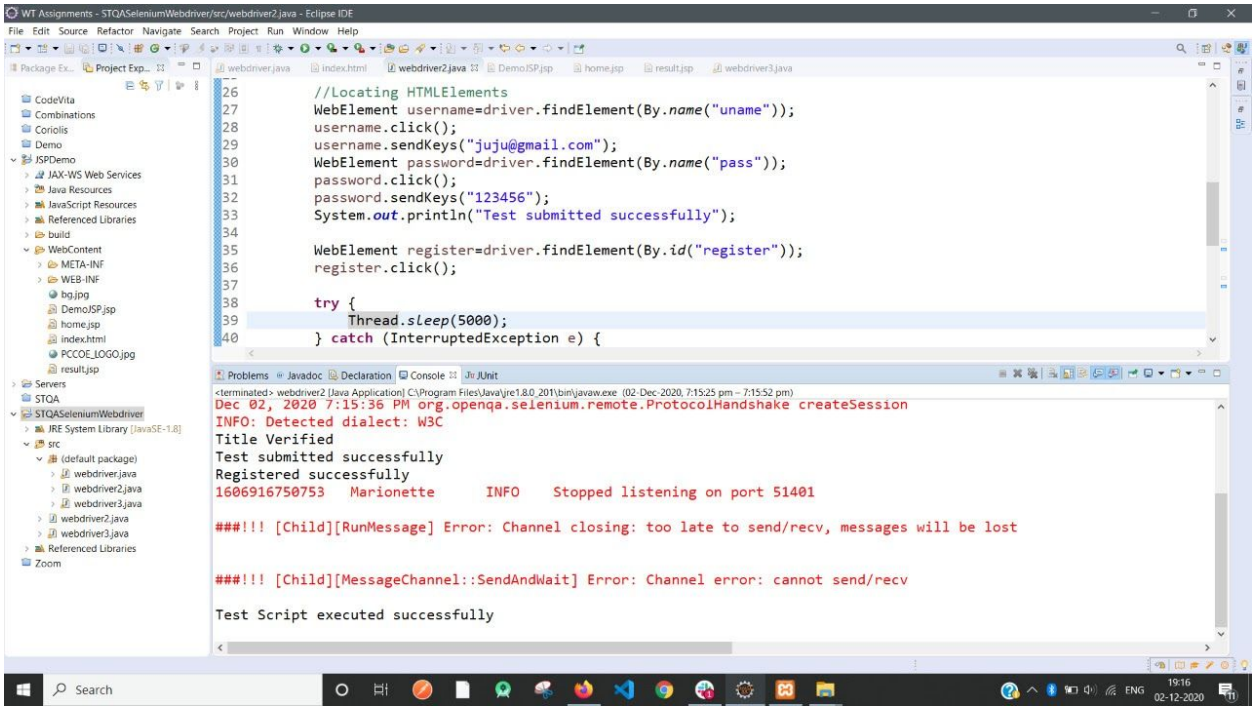
webdriver2.java

```
//This Class tests the registration feature of web app on firefox using selenium webdriver.
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
public class webdriver2 {
public static void main(String []args) {
        System.setProperty("webdriver.gecko.driver", "H:\\WT Assignments\\geckodriver.exe");
        WebDriver driver=new FirefoxDriver();
```

```
String appUrl="http://localhost:8080/JSPDemo/";
driver.get(appUrl);
driver.manage().window().maximize();
//Title check
String actual_title =driver.getTitle();
String title="HOME";
if(actual_title.contentEquals(title))
{
        System.out.println("Title Verified");
}
else
        System.out.println("Title Mismatched");
//Locating HTMLElements
WebElement username=driver.findElement(By.name("uname"));
username.click();
username.sendKeys("juju@gmail.com");
WebElement password=driver.findElement(By.name("pass"));
password.click();
password.sendKeys("123456");
System.out.println("Test submitted successfully");
WebElement register=driver.findElement(By.id("register"));
```

```
		register.click();
		try {
			Thread.sleep(5000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		driver.navigate().to("http://localhost:8080/JSPDemo/");
		System.out.println("Registered successfully");
		try {
			Thread.sleep(5000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		driver.close();
		System.out.println("Test Script executed successfully");
		System.exit(0);
}
```

}



webdriver3.java

//This Class tests the login,home page and exit feature of web app on chrome browser using selenium webdriver.

import java.sql.Connection;

import java.sql.DriverManager;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

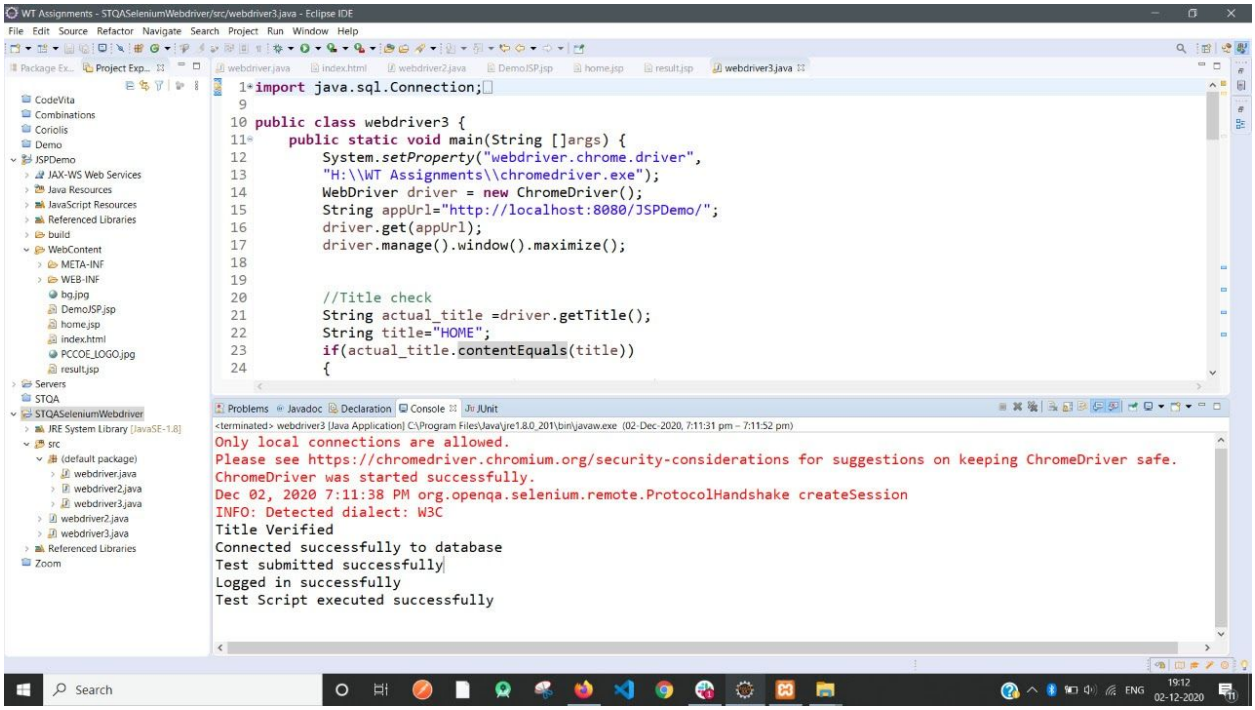
import org.openqa.selenium.firefox.FirefoxDriver;

```
public class webdriver3 {
public static void main(String []args) {
        System.setProperty("webdriver.chrome.driver",
                        "H:\\WT Assignments\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        String appUrl="http://localhost:8080/JSPDemo/";
        driver.get(appUrl);
        driver.manage().window().maximize();
        //Title check
        String actual_title =driver.getTitle();
        String title="HOME";
        if(actual_title.contentEquals(title))
        {
                System.out.println("Title Verified");
        }
        else
                System.out.println("Title Mismatched");
        //Checking database
        Connection conn = null;
        String dbUrl = "jdbc:mysql://localhost:3306/demo";
        String user = "root";
```

```
String pass = "";
try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(dbUrl, user, pass);
        if (conn != null) {
                System.out.println("Connected successfully to database");
        }
}catch(Exception e) {
        System.out.println("Connection Failed");
}
//Locating HTMLElements
WebElement username=driver.findElement(By.name("uname"));
username.click();
username.sendKeys("x@gmail.com");
WebElement password=driver.findElement(By.name("pass"));
password.click();
password.sendKeys("123456");
System.out.println("Test submitted successfully");
WebElement login=driver.findElement(By.id("login"));
login.click();
System.out.println("Logged in successfully");
```

```
WebElement option2=driver.findElement(By.id("o2"));
option2.click();
try {
        Thread.sleep(1000);
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
WebElement next=driver.findElement(By.id("next"));
next.click();
try {
        Thread.sleep(3000);
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
}
driver.navigate().to("http://localhost:8080/JSPDemo/result.jsp");
try {
        Thread.sleep(2000);
} catch (InterruptedException e) {
        // TODO Auto-generated catch block
```

```
            e.printStackTrace();
        }
        WebElement exit=driver.findElement(By.linkText("Exit"));
        exit.click();
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        driver.close();
        System.out.println("Test Script executed successfully");
        System.exit(0);
}
}
```


WT Assignments - STQASeleniumWebdriver/src/webdriver3.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
webdriver.java index.html webdriver2.java DemoJSP.jsp home.jsp result.jsp webdriver3.java
1 import java.sql.Connection;
9
10 public class webdriver3 {
11 public static void main(String []args) {
12 System.setProperty("webdriver.chrome.driver",
13 "H:\\WT Assignments\\chromedriver.exe");
14 WebDriver driver = new ChromeDriver();
15 String appUrl="http://localhost:8080/JSPDemo/";
16 driver.get(appUrl);
17 driver.manage().window().maximize();
18
19
20 //Title check
21 String actual_title =driver.getTitle();
22 String title="HOME";
23 if(actual_title.contentEquals(title))
24 {
Problems Javadoc Declaration Console JUnit
<terminated> webdriver3 [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (02-Dec-2020, 7:11:31 pm – 7:11:52 pm)
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Dec 02, 2020 7:11:38 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Title Verified
Connected successfully to database
Test submitted successfully
Logged in successfully
Test Script executed successfully
Search
ENG
19:12
02-12-2020