

Assignment Number 5

Index.html →

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>EBookStore</title>
<style>
button{
  background-color: #008CBA;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 24px;
  border-radius : 8px;
  width : 25%;
}
</style>
</head>
<body>
  <h1>
    <center>E-BOOK STORE</center>
  </h1>
  <form action="/EBookStore" method = "get">
    <center><button>Show Books</button></center>
  </form>
</body>
</html>
```

EBookStore.java →

```
import java.io.*;
import java.sql.*;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;

@WebServlet("/EBookStore")

public class EBookStore extends HttpServlet {

  private static final long serialVersionUID = 1L;

  public void init() throws ServletException {

  }

}
```

```

public static Connection initializeDatabase() throws SQLException, ClassNotFoundException{
    Class.forName("com.mysql.jdbc.Driver");
    String url = "jdbc:mysql://localhost:3306/wtl?characterEncoding=utf8";
    String un = "root";
    String pass = "sameer";
    Connection con = DriverManager.getConnection(url,un,pass);
    return con;
}

```

```

public void doGet(HttpServletRequest request, HttpServletResponse response)

```

```

    throws ServletException, IOException {

```

```

    try {

```

```

        response.setContentType("text/html");

```

```

        PrintWriter out = response.getWriter();

```

```

        Connection con = initializeDatabase();

```

```

        PreparedStatement st = null;

```

```

        ResultSet rs = null;

```

```

        st = con.prepareStatement("SELECT * FROM ebookstore;");

```

```

        rs = st.executeQuery();

```

```

        int id;

```

```

        String name;

```

```

        String author;

```

```

        int price;

```

```

        int quantity;

```

```

        out.println("<html>");

```

```

        out.println("<h1> <center> EBOOK STORE </center> </h1>");

```

```

        out.println("<style>");

```

```

        out.println("table{

```

```

            + "    font-size : 25px;"

```

```

            + "text-align: center;"

```

```

            + "margin-left:auto;"

```

```

            + "margin-right:auto;"

```

```

            + "width:90%;"

```

```

        + "border-spacing: 10px 10px;"
        + "}"
        + "td{"
        + "padding : 10px;"
        + "}");
out.println("</style>");
out.println("<table>");
out.println("<tr>"

        + "<th>ID</th>"
        + "<th>Name</th>"
        + "<th>Author</th>"
        + "<th>Price</th>"
        + "<th>Quantity</th>"
        + "</tr>");

while(rs.next())
{
    id = rs.getInt(1);
    name = rs.getString(2);
    author = rs.getString(3);
    price = rs.getInt(4);
    quantity = rs.getInt(5);
    out.println("<tr>"

        + "<td>" + id + "</td>"
        + "<td>" + name + "</td>"
        + "<td>" + author + "</td>"
        + "<td>" + price + "</td>"
        + "<td>" + quantity + "</td>"
        + "</tr>");

}

out.println("</table>");

out.println("</html>");

```

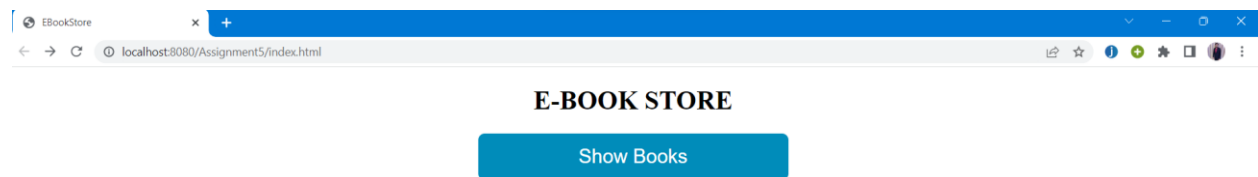
```

        con.close();
    }
    catch(Exception e) {
        System.out.println("Error : " + e);
    }
}

public void destroy() {
}
}

```

Output→



| EBOOK STORE | | | | |
|-------------|--|-------------|-------|----------|
| ID | Name | Author | Price | Quantity |
| 1 | Harry Potter and The Sorcerers Stone | J.K.Rowling | 350 | 10 |
| 2 | Harry Potter and The Chambers Of Secret | J.K.Rowling | 400 | 15 |
| 3 | Harry Potter and The Prisoner Of Askaban | J.K.Rowling | 440 | 20 |
| 4 | Harry Potter and The Goblet Of Fire | J.K.Rowling | 480 | 25 |
| 5 | Harry Potter and The Order Of Phoenix | J.K.Rowling | 500 | 30 |
| 6 | Harry Potter and The Half Blood Prince | J.K.Rowling | 550 | 35 |
| 7 | Harry Potter and The Deathly Hallows-I | J.K.Rowling | 575 | 40 |
| 8 | Harry Potter and The Deathly Hallows-II | J.K.Rowling | 600 | 45 |

Assignment Number 6

Index.html →

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>EBookStore</title>
<style>
button{
    background-color: #008CBA;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 24px;
    border-radius : 8px;
    width : 25%;
}
</style>
</head>
<body>
    <h1>
        <center>E-BOOK STORE</center>
    </h1>
    <form action="EBookStore.jsp" method = "get">
        <center><button>Show Books</button></center>
    </form>
</body>
</html>
```

EBookStore.jsp →

```
<%@ page language="java" contentType="text/html; charset=ISO-
8859-1" pageEncoding="ISO-8859-1"%>
<%@ page import = "java.io.*,java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>EBookStore</title>
<style>
table,th,td{
```

```

border : 2px solid;
}
table{
    font-size : 20px;
    text-align:center;
    margin-left:auto;
    margin-right:auto;
    width:90%;
    /* border-spacing: 5px; */
}
td{
    padding : 13px;
}
h1{
    font-size : 40px;
}
button{
    background-color: #008CBA;
    border: none;
    color: white;
    padding: 15px 32px;
    text-align: center;
    text-decoration: none;
    display: inline-block;
    font-size: 24px;
    border-radius : 8px;
    width : 25%;
}
</style>
</head>
<body>
    <h1>
        <center>E-BOOK STORE</center>
    </h1>
    <%
        try{
            Class.forName("com.mysql.jdbc.Driver");
            String url =
"jdbc:mysql://localhost:3306/wtl?characterEncoding=utf8";
            String un = "root";
            String pass = "sameer";
            Connection con =
DriverManager.getConnection(url,un,pass);

            PreparedStatement st = null;
            ResultSet rs = null;

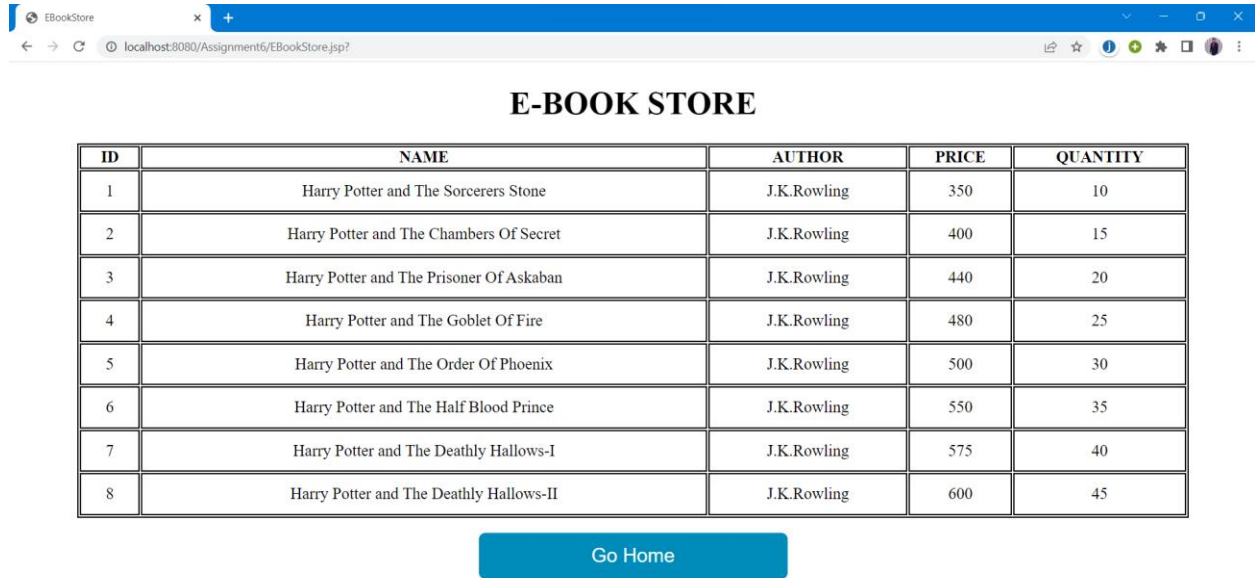
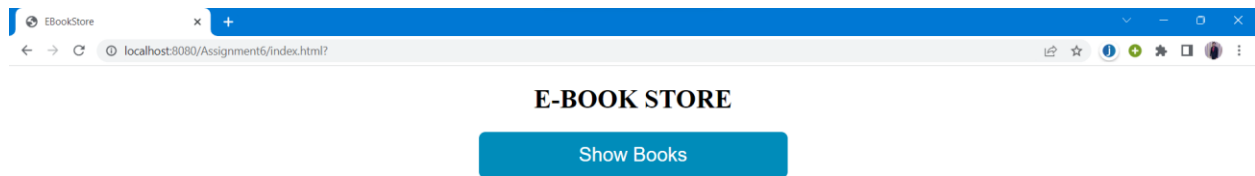
```

```

        st = con.prepareStatement("SELECT * FROM
ebookstore;");
        rs = st.executeQuery();
        int id;
        String name;
        String author;
        int price;
        int quantity;
    %>
    <table>
        <tr>
            <th>ID</th>
            <th>NAME</th>
            <th>AUTHOR</th>
            <th>PRICE</th>
            <th>QUANTITY</th>
        </tr>
        <%
        while(rs.next())
        {
        %>
        <tr>
            <td> <%= rs.getInt(1) %></td>
            <td> <%= rs.getString(2) %></td>
            <td> <%= rs.getString(3) %></td>
            <td> <%= rs.getInt(4) %></td>
            <td> <%= rs.getInt(5) %></td>
        </tr>
        <%} %>
    </table>
    <%
    }catch(Exception e){ %>
        <p><% e.printStackTrace(); %> </p>
    <%}%>
    <br>
    <form action="index.html">
        <center><button>Go Home</button></center>
    </form>
</body>
</html>

```

Output →



Assignment Number 7

Addform.html ->

```
<!DOCTYPE html>
<html>
  <head>
    <title>Books</title>
    <link rel="stylesheet" href="style.css">
    <meta charset="utf-8">
  </head>
  <body>
    <div style="display: flex;">
      <div>
        <form action="view.php" method="post">
          <h2>View All Books</h2>
          <input type="submit" value="Views Books"
/>
        </form>
      </div>
      <div>
        <form action="delete.php" method="post">
          <h2>Delete Data</h2>
          Book Id : <input type="text"
name="bid" />
          <input type="submit" value="DELETE"
/>
        </form>
      </div>
    </div>
    <div style="display: flex;">
      <div>
        <form action="index.php" method="post">
          <h2>Add Data</h2>
          Book Name : <input type="text"
name="bname" />
```

```

        Book Author : <input type="text"
name="bauthor" />
        Book Price : <input type="text"
name="price" />
        Book Quantity : <input type="text"
name="quantity" />
        <input type="submit" value="ADD BOOK"
/>
    </form>
</div>
<div>
    <form action="update.php" method="post">
        <h2>Update Data</h2>
        Book Id : <input type="text"
name="bid" />
        Book Quantity : <input type="text"
name="quantity" />
        <input type="submit" value="UPDATE"
/>
    </form>
</div>

</div>
</body>
</html>

```

Style.css->

```

form {
    width: 500px;
    margin: 0 auto;
    background-color: #fcfcfc;
    padding: 20px 50px 40px;
    box-shadow: 1px 4px 10px 1px #aaa;
    font-family: sans-serif;
    box-sizing: border-box;
}

```

```

}
form h2{
    text-align: center;
    margin-bottom: 30px;
}
form input {
    margin-bottom: 15px;
}

form input[type=text] {
    display: block;
    height: 32px;
    padding: 6px 16px;
    width: 100%;
    border: none;
    background-color: #f3f3f3;
}
form button[type=submit] {
    display: block;
    margin: 20px auto 0;
    width: 150px;
    height: 40px;
    border-radius: 25px;
    border: none;
    color: #eee;
    font-weight: 700;
    box-shadow: 1px 4px 10px 1px #aaa;
}

```

Index.php ->

```

<?php
$servername = "localhost";
$username = "root";
$password = "sameer";

```

```

$dbname = "wt1";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}
//echo "Connected successfully";

$bname = $_POST['bname'];
$author = $_POST['author'];
$price = $_POST['price'];
$quantity = $_POST['quantity'];

$query = $conn->prepare("insert into ebookstore
(book_name,book_author,book_price,quantity)
values (?, ?, ?, ?);");
$query->bind_param("ssii",
$bname, $author, $price, $quantity);

$query->execute();
echo "New records created successfully";
?><form action="addform.html">
    <center><button>Go Home</button></center>
</form>

View.php ->
<!DOCTYPE html>
<html>
<head>
<style>

```

```

table,th,td{
border : 2px solid;
}
table{
    font-size : 20px;
    text-align:center;
    margin-left:auto;
    margin-right:auto;
    width:90%;
/*      border-spacing: 5px; */
}
td{
    padding : 13px;
}
</style>
</head>
<body>

<?php
$servername = "localhost";
$username = "root";
$password = "sameer";
$dbname = "wt1";

// Create connection
$conn = new mysqli($servername, $username,
$password,$dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}

$sql = "SELECT * FROM ebookstore";

```

```

$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo
    "<table><tr><th>ID</th><th>Name</th><th>Author</th><th>Price</th><th>Quantity</th></tr>";
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo
        "<tr><td>".$row["book_id"]."</td><td>".$row["book_name"]."</td><td>".$row["book_author"]."</td><td>".$row["book_price"]."</td><td>".$row["quantity"]."</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
} ?>
<form action="addform.html">
    <button>Go Home</button>
</form>
</body>
</html>

```

Update.php ->

```

<?php
$servername = "localhost";
$username = "root";
$password = "sameer";
$dbname = "wt1";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);

```

```

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn-
>connect_error);
}

$bid = $_POST['bid'];
$quantity = $_POST['quantity'];
$sql = "UPDATE ebookstore SET quantity=" .
$quantity . " WHERE book_id=" . $bid . ";";

if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . $conn-
>error;
}

```

Delete.php ->

```

<?php
$servername = "localhost";
$username = "root";
$password = "sameer";
$dbname = "wt1";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);

// Check connection
if ($conn->connect_error) {

```

```
        die("Connection failed: " . $conn-
>connect_error);
    }

    $bid = $_POST['bid'];
    $sql = "DELETE from ebookstore WHERE book_id=" .
    $bid . ";";

    if ($conn->query($sql) === TRUE) {
        echo "Record deleted successfully";
    } else {
        echo "Error deleting record: " . $conn-
>error;
    }
}
```

OUTPUT ->

Books

localhost/PHPDemo/addform.html?

View All Books

Views Books

Delete Data

Book Id :

DELETE

Add Data

Book Name :

Book Author :

Book Price :

Book Quantity :

ADD BOOK

Update Data

Book Id :

Book Quantity :

UPDATE

localhost/PHPDemo/view.php

localhost/PHPDemo/view.php

| ID | Name | Author | Price | Quantity |
|----|--|-------------|-------|----------|
| 1 | Harry Potter and The Sorcerers Stone | J.K.Rowling | 350 | 10 |
| 2 | Harry Potter and The Chambers Of Secret | J.K.Rowling | 400 | 15 |
| 3 | Harry Potter and The Prisoner Of Askaban | J.K.Rowling | 440 | 20 |
| 4 | Harry Potter and The Goblet Of Fire | J.K.Rowling | 480 | 25 |
| 5 | Harry Potter and The Order Of Phoenix | J.K.Rowling | 500 | 30 |
| 6 | Harry Potter and The Half Blood Prince | J.K.Rowling | 550 | 35 |
| 7 | Harry Potter and The Deathly Hallows-I | J.K.Rowling | 575 | 40 |
| 8 | Harry Potter and The Deathly Hallows-II | J.K.Rowling | 600 | 45 |
| 10 | School Times | Ruskin Bond | 160 | 10 |

Go Home

Index.jsp

```
<%@ page language = "java" contentType = "text/html; charset = ISO-8859-1"
    pageEncoding = "ISO-8859-1"%>
<%@ taglib prefix = "s" uri = "/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
        <title>Employee Form</title>
    </head>

    <body>
        <s:form action = "empinfo" method = "post">
            <s:textfield name = "name" label = "Name" size = "20" />
            <s:textfield name = "age" label = "Age" size = "20" />
            <s:submit name = "submit" label = "Submit" align="center" />
        </s:form>
    </body>
</html>
```

Success.jsp

```
<%@ page language = "java" contentType = "text/html; charset = ISO-8859-1"
    pageEncoding = "ISO-8859-1"%>
<%@ taglib prefix = "s" uri = "/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
    <head>
```

```
<title>Success</title>
</head>

<body>
    Employee Information is captured successfully.
</body>
</html>
```

Validate.java

```
import com.opensymphony.xwork2.ActionSupport;

public class Employee extends ActionSupport {
    private String name;
    private int age;

    public String execute() {
        return SUCCESS;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }
}
```

```
}
```

```
public void setAge(int age) {
```

```
    this.age = age;
```

```
}
```

```
public void validate() {
```

```
    if (name == null || name.trim().equals("")) {
```

```
        addFieldError("name", "The name is required");
```

```
    }
```

```
    if (age < 28 || age > 65) {
```

```
        addFieldError("age", "Age must be in between 28 and 65");
```

```
    }
```

```
}
```

```
}
```



A web form with two input fields labeled "Name:" and "Age:", and a "Submit" button. The form is enclosed in a blue border. The "Name:" label is to the left of a text input field. The "Age:" label is to the left of another text input field. Below the "Age:" input field is a blue "Submit" button.

The name is required

Name:

Age must be in between 28 and 65

Age:

Employee Information is captured successfully.

Assignment Number 9

Signup.html ->

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <title>Registration Form Using Angular JS</title>
```

```
  <meta name="viewport" content="width=device-width" charset="UTF-8">
```

```
  <link rel="stylesheet" href="style.css">
```

```
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
```

```
  <script src="./app.js"></script>
```

```
</head>
```

```
<body>
```

```
  <div ng-app="myApp">
```

```
    <form ng-controller="frmController" novalidate>
```

```
      <h2>Register</h2>
```

```
      <label for="fname">First Name: </label>
```

```
      <input type="text" ng-model="newuser.fname" id="fname" name="fname" required />
```

```
      <br>
```

```
      <label for="lname">Last Name: </label>
```

```
      <input type="text" ng-model="newuser.lname" name="lname" id="lname" required />
```

```
      <br>
```

```
      <label for="uname">User Name: </label>
```

```
      <input type="text" ng-model="newuser.uname" id="uname" name="uname" required />
```

```

    <br>

    <label for="password">Password: </label>

    <input type="password" ng-model="newuser.password" name="password" id="password"
required />

    <br>

    <button type="submit" ng-click="saveUser()">Register</button>

    <p><a href="login.html">Login</a></p>

</form>

</div>

</body>

</html>

```

Login.html →

```

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Registration Form Using Angular JS</title>

    <meta name="viewport" content="width=device-width" charset="UTF-8">

    <link rel="stylesheet" href="style.css">

    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

    <script src="./app.js"></script>

</head>

<body>

    <div ng-app="myApp">

        <form ng-controller="frmController" novalidate>

```

```

<h2>Log In</h2>

<label for="uname">User Name: </label>

<input type="text" ng-model="newuser.uname" id="uname" name="uname" required />

<br>

<label for="password">Password: </label>

<input type="password" ng-model="newuser.password" name="password" id="password"
required />

<br>

<button type="submit" ng-click="verifyUser()">Register</button>

<p><a href="signup.html">Register</a></p>

</form>

</div>

</body>

</html>

```

Style.css →

```

@import
url('https://fonts.googleapis.com/css2?family=Quicksand:wght@300;400;500;600;700&display=swap');

```

h1,

h2,

h3,

h4,

ul,


```
li,  
a,  
p,  
input,  
label,  
button,  
div {  
    margin: auto;  
    padding: 0;  
    font-family: 'Quicksand', sans-serif;  
    color: #444;  
}
```

```
div {  
    background-color: #FEE996;  
    padding: 100px;  
}
```

```
form h2 {  
    font-size: 2.4em;  
    font-weight: 900;  
    margin-bottom: 40px;  
}
```

```
form {  
    width: 400px;  
    margin: 0 auto;  
    padding: 30px;  
    box-shadow: 1px 2px 3px rgba(0, 0, 0, 0.1);
```

```
border-radius: 10px;  
background: white;  
}
```

```
input {  
  padding: 10px 12px;  
  border-radius: 4px;  
  border: 1px solid #ddd;  
  font-size: 1em;  
  width: 90%;  
}
```

```
label {  
  display: block;  
  font-size: 20px;  
  margin: 20px 0 10px;  
  font-weight: 500;  
}
```

```
button {  
  margin-top: 30px;  
  border-radius: 36px;  
  background: #FEE996;  
  border: 0;  
  text-transform: uppercase;  
  font-weight: 700;  
  font-size: 0.8em;  
  display: block;  
  padding: 10px 16px;
```

```
letter-spacing: 2px;  
}
```

App.js →

```
var app = angular.module("myApp", []);
```

```
app.service("UserService", function () {
```

```
    var users = [{  
        'fname': 'Sameer',  
        'lname': 'Bramhecha',  
        'uname': 'samya',  
        'password': 'sameer123'  
    }];
```

```
    this.save = function (user) {  
        users.push(user);  
    };
```

```
    this.get = function (uname) {  
        for (var i in users) {  
            if (users[i].uname == uname) {  
                return users[i];  
            }  
        }  
    }  
};
```

```
    this.list = function () {
```

```

        return users;
    };
});

app.controller("frmController", function ($scope, UserService) {

    console.clear();

    $scope.ifSearchuser = false;
    $scope.users = UserService.list();

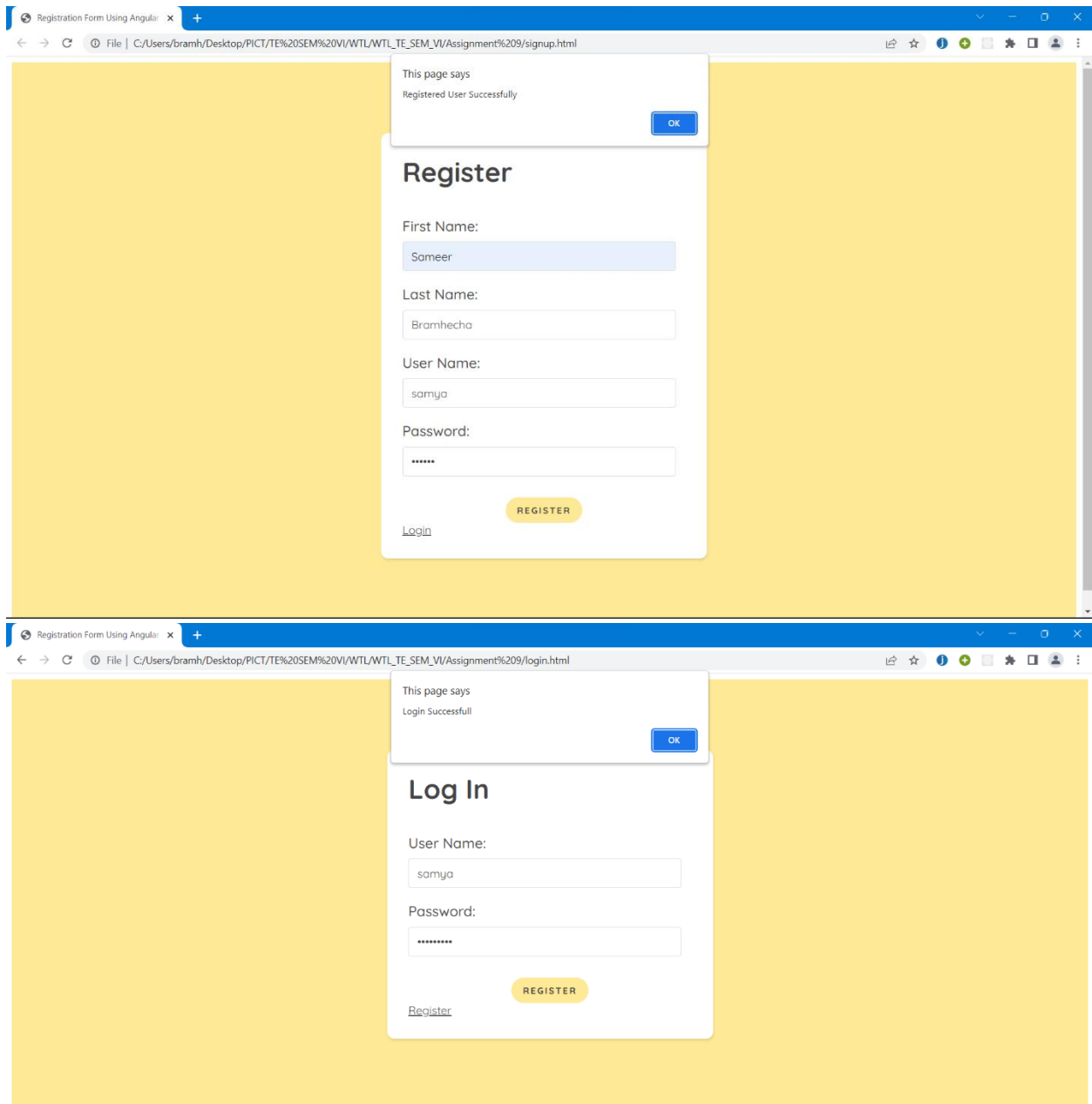
    $scope.saveUser = function () {
        console.log($scope.newuser);
        if ($scope.newuser == null || $scope.newuser == angular.undefined) {
            return;
        }
        UserService.save($scope.newUser);
        $scope.newuser = {};
        alert("Registered User Successfully");
    }

    $scope.verifyUser = function () {
        if ($scope.newuser == null || $scope.newuser == angular.undefined) {
            return;
        }
        $scope.user = UserService.get($scope.newuser.username);
        if ($scope.user.password == $scope.newuser.password) {
            alert("Login Successfull");
        }
        else {
            alert("Login Failure");
        }
    }
});

```

```
}  
}  
});
```

Output →



Assignment No 10

Code :

EJB Bank Application

@Entity public class

Transaction {

 @Id

 @GeneratedValue(strategy = GenerationType.IDENTITY)

private long transactionId; private long accountId;

private String transactionType; private double amount;

 // constructors, getters, and setters

}

@Stateless public class TransactionManagerBean implements

TransactionManagerRemote {

 @PersistenceContext(unitName = "bank")

private EntityManager entityManager;

 public void deposit(long accountId, double amount) {

 Account account = entityManager.find(Account.class, accountId);

double newBalance = account.getBalance() + amount;

account.setBalance(newBalance); entityManager.persist(account);

 Transaction transaction = new Transaction(accountId, "deposit", amount);

entityManager.persist(transaction);

 }

 public void withdraw(long accountId, double amount) throws InsufficientFundsException {

 Account account = entityManager.find(Account.class, accountId);

double newBalance = account.getBalance() - amount;

```

        if (newBalance < 0) {            throw new
InsufficientFundsException();
    }

```

```

        account.setBalance(newBalance);
entityManager.persist(account);

```

```

        Transaction transaction = new Transaction(accountId, "withdraw", amount);
entityManager.persist(transaction);
    }

```

```

    public List<Transaction> getTransactionHistory(long accountId) {
        TypedQuery<Transaction> query = entityManager.createQuery(
            "SELECT t FROM Transaction t WHERE t.accountId = :accountId", Transaction.class);
        query.setParameter("accountId", accountId);    return query.getResultList();
    }
}

```

```

public class TransactionServlet extends HttpServlet {

    @EJB    private TransactionManagerRemote
transactionManager;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException,    IOException    {        long    accountId    =
Long.parseLong(request.getParameter("accountId"));

        List<Transaction> transactions = transactionManager.getTransactionHistory(accountId);
        request.setAttribute("transactions", transactions);
        request.getRequestDispatcher("transaction_history.jsp").forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {    long accountId =
Long.parseLong(request.getParameter("accountId"));    double amount =
Double.parseDouble(request.getParameter("amount"));

```

```

String action = request.getParameter("action");
String message = "";

if (action.equals("deposit")) {
transactionManager.deposit(accountId, amount);    message = "Deposit
successful"; } else if (action.equals("withdraw")) {    boolean success
= transactionManager.withdraw(accountId, amount);
    if (success) {        message =
"Withdrawal successful";
    } else {
        message = "Insufficient funds";
    }
}

request.setAttribute("message", message);
doGet(request, response);
}

```

Frontend :

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Bank Transactions</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
    <div class="container mt-4">

```


<h1>Bank Transactions</h1>

<h2>Account Balance: \$</h2>

<form id="transaction-form">

<div class="form-group">

<label for="account-id">Account ID:</label>

<input type="number" class="form-control" id="account-id" name="accountId" required>

</div>

<div class="form-group">

<label for="amount">Amount:</label>

<input type="number" class="form-control" id="amount" name="amount" step="0.01" required>

</div>

<div class="form-group">

<label for="action">Action:</label>

<select class="form-control" id="action" name="action" required>

<option value="deposit">Deposit</option>

<option value="withdraw">Withdraw</option>

</select>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

<div id="message"></div>

<h3>Transaction History</h3>

<table class="table">

<thead>

<tr>

<th>Date</th>

<th>Type</th>

```

        <th>Amount</th>
    </tr>
</thead>
<tbody id="transaction-history">
</tbody>
</table>
</div>

```

```
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"
```

```

    integrity="sha384-
97VsVL8/MVJmQZvEi8cNzWyxVJvoHkDR7H7Pml1hBnK9zWwv4E0EmqKb/RHSnlTr"

```

```
    crossorigin="anonymous"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
```

```

    integrity="sha384-
vUKBRrN99AVC4lvG+9BEEVP2i+iAalqPZQyB+5NIhm3xRp1EY2V5fBcTUnTcbC8L"

```

```
    crossorigin="anonymous"></script>
```

```
<script>
```

```
    $(function() {
```

```
        // Fetch account balance
```

```
        $.get("/bank/balance", function(data) {
```

```
            $("#balance").text(data.balance.toFixed(2));
```

```
        });
```

```
        // Handle form submission
```

```
        $("#transaction-form").submit(function(event) {
```

```
            event.preventDefault();
```

```
            var formData = $(this).serialize();
```

```
            $.post("/bank/transaction", formData, function(data) {
```

```
                $("#message").text(data.message);
```

```

        $("#transaction-history").empty();

        $.each(data.transactions, function(i, transaction) {
            var row
            = "<tr><td>" + transaction.date + "</td><td>" + transaction.type +
                "</td><td>" + transaction.amount.toFixed(2) + "</td></tr>";
            $("#transaction-history").append(row);
        });

        // Update balance
        $.get("/bank/balance", function(data) {
            $("#balance").text(data.balance.toFixed(2));
        });

    }).fail(function(error) {
        $("#message").text(error.responseJSON.message);
    });

});

});

</script>

```

Bank Transactions

Account Balance: 400 \$

Amount:

33423

Action:

Deposit

Submit

Transaction History

| Date | Type | Amount |
|------------|------------|--------|
| 2022-05-01 | Withdrawal | 100.00 |
| 2022-05-02 | Deposit | 500.00 |
| 2022-05-03 | Withdrawal | 50.00 |