index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="ISO-8859-1">
    <title>Online Quiz Portal</title>
    <style>
        body {

            text-align: center;
        }

        h1 {
            margin-top: 50px;
        }

        h3 {
            margin-top: 30px;
        }

        ul {
            list-style-type: none;
            padding: 0;
        }

        ul li {
            display: inline-block;
            margin: 10px;
        }

        ul li a {
            text-decoration: none;
        }
    </style>
</head>
<body>
    <h1>Online Quiz Portal</h1>
    <h3><a href="userregister.jsp">User Registration</a></h3>
    <h3><a href="userlogin.jsp">User Login</a></h3>
    <h3><a href="adminlogin.jsp">Admin Login</a></h3>
</body>
</html>
```

userregistration.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Registration Page</title>
<style>
    body {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
    }
    h1 {
            margin-top: 50px;
        }

    h2 {
        text-align: center;
    }

    form {
        text-align: center;
    }

    table {
        width: 80%;
    }

    input[type="text"],
    input[type="email"],
    input[type="password"],
    input[type="date"] {
        width: 100%;
        padding: 10px;
        margin-bottom: 10px;
    }
</style>
</head>
<body>
    <div align="center">
    <h1>Online Quiz Portal</h1>
        <h2>Registration Page</h2>
        <form action="registerServlet" method="POST">
            <table>
                <tr>
                    <td><input type="text" name="name"
placeholder="Name"></td>
                </tr>
                <tr>
                    <td><input type="email" name="email"
placeholder="Email"></td>
                </tr>
                <tr>
                    <td><input type="password" name="password"
placeholder="Password"></td>
```

```
                </tr>
                <tr>
                    <td><input type="date" name="dob" placeholder="Date of
Birth"></td>
                </tr>
            </table>
            <br>
            <input type="submit" value="Register"> <br><br> <h3><a
href="index.jsp">Main</a></h3>
        </form>
    </div>
</body>
</html>
```

userlogin.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<style>
    body {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
    }

    h2 {
        text-align: center;
    }
    form {
        text-align: center;
    }

    input[type="email"],
    input[type="password"] {
        width: 100%;
        padding: 10px;
        margin-bottom: 10px;
    }


</style>
</head>
<body>
    <div align="center">
    <h1>Online Quiz Portal</h1>
        <h2>User Login</h2>
        <form action="userloginServlet" method="post">
            Email: <input type="email" name="email"><br>
            Password: <input type="password" name="password"><br>
            <br>
            <input type="submit" value="Login">
        </form> <br>
       <h3><a href="index.jsp" class="button">Main</a></h3>
    </div>
</body>
</html>
```

adminlogin.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<style>
body {
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
}
h1 {
}
text-align: center;
form {
text-align: center;
}
input[type="email"],
input[type="password"] {
width: 100%;
padding: 10px;
margin-bottom: 10px;
}
</style>
</head>
<body>
<div align="center">
<h1>Admin  Login</h1>
<form action="adminLoginServlet" method="post">
```

```
Email: <input type="email" name="email"><br>
Password: <input type="password" name="password"><br>
<br>
<input type="submit" value="Login">
</form> <br>
<h3><a href="index.jsp" class="button">Main</a></h3>
</div>
</body>
</html>
```

admindashboardpage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%><%@ include file="header.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>

</head>
<body>
<h1>Welcome, Admin!</h1>

<h3><a href="addquestion.jsp">Add Questions to quiz</a></h3>



<h3><a href="viewuser.jsp">View User List </a></h3>
<br>
<br>



</body>
```

```
</html>
```

userdashboardpage.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ include file="header.jsp" %>
<!DOCTYPE html>
<html>
<head>
  <meta charset="ISO-8859-1">
  <title>Dashboard</title>
  <style>


    .container {
      max-width: 800px;
      margin: 0 auto;
      padding: 20px;
    }

    h2 {
      text-align: center;
    }
    h1{
    text-align: center;
    }

    .welcome-message {
      text-align: center;
      margin-bottom: 20px;
    }
```

```
        .action-link {
            display: block;
            margin: 0 auto 10px;
            text-align: center;
        }

        .logout-form {
            text-align: center;
        }
    </style>
</head>
<body>
    <div class="container">
    <h1>Online Quiz Portal</h1>
        <h2>Welcome to the User Dashboard</h2>
        <p class="welcome-message">Welcome, <%=
session.getAttribute("name") %>!</p>
        <button class="action-link"
onclick="window.location.href='sqlTestServlet'">Take SQL Test</button>
    </div>
</body>
</html>
```

addquestion.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<%@ include file="header.jsp" %>
<html>
<head>
<meta charset="ISO-8859-1">
```

```
<title>Insert title here</title>
<style>
  table {
    width: 100%;
    border-collapse: collapse;
  }

  th, td {
    border: 1px solid black;
    padding: 8px;
    text-align: left;
  }

  tr:nth-child(even) {
    background-color: #f2f2f2;
  }
</style>
</head>
<body>
  <h1>Add Questions to Quiz</h1>

  <h2>Subjects:</h2>
  <table>
    <thead>
      <tr>
        <th>Subject</th>
        <th>Add Question</th>
        <th>View Question</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>SQL</td>
        <td><a href="addquestionsql.jsp">Add SQL Question</a></td>
```

```
                <td><a href="viewsqllist.jsp">View SQL Questions</a></td>
            </tr>


        </tbody>
    </table>


</body>
</html>
```

addquestionsql.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@ include file="header.jsp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Add SQL Questions</h1>

    <form action="addsqlquestion" method="post">
        <label>Question:</label>
        <input type="text" name="question">
        <br>
        <br>
        <label>Option 1:</label>
        <input type="text" name="option1">
        <br>
        <br>
        <label>Option 2:</label>
        <input type="text" name="option2">
        <br>
        <br>
        <label>Option 3:</label>
        <input type="text" name="option3">
        <br>
        <br>
        <label>Option 4:</label>
        <input type="text" name="option4">
        <br>
        <br>
        <label>Correct Option:</label>
        <input type="text" name="correctOption">
        <br>
        <br>
        <input type="submit" value="Add Question">
    </form>
</body>
</html>
```

editsqlquestion.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%><%@ include file="header.jsp" %>
<%@ page import="com.questions.SQLQuestion" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Edit SQL Question</h1>
<form action="editsqlquestion" method="post">
<input type="hidden" name="id" value="<%= ((SQLQuestion)
request.getAttribute("question")).getId() %>">
<label>Question:</label>
```

```
<input type="text" name="question" value="<%= ((SQLQuestion)
request.getAttribute("question")).getQuestion() %>">
<br>
<label>Option 1:</label>
<input type="text" name="option1" value="<%= ((SQLQuestion)
request.getAttribute("question")).getOption1() %>">
<br>
<label>Option 2:</label>
<input type="text" name="option2" value="<%= ((SQLQuestion)
request.getAttribute("question")).getOption2() %>">
<br>
<label>Option 3:</label>
<input type="text" name="option3" value="<%= ((SQLQuestion)
request.getAttribute("question")).getOption3() %>">
<br>
<label>Option 4:</label>
<input type="text" name="option4" value="<%= ((SQLQuestion)
request.getAttribute("question")).getOption4() %>">
<br>
<label>Correct Option:</label>
<input type="number" name="correctOption" value="<%= ((SQLQuestion)
request.getAttribute("question")).getCorrectOption() %>">
<br>
<input type="submit" value="Update">
</form>
<a href="viewsqllist.jsp">Back to List</a>
</body>
</html>
```

edituser.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%><%@ include file="header.jsp" %><%@ include
file="header.jsp" %>
<!DOCTYPE html>
<html>
```

```
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Edit User</h1>
<form action="updateuser" method="post">
<input type="hidden" name="userId" value="${user.id}">
<label>Name: </label>
<input type="text" name="name" value="${user.name}">
<br>
<label>Email: </label>
<input type="email" name="email" value="${user.email}">
<br>
<input type="submit" value="Update">
</form>
</body>
</html>
```

error.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Something error happen!</h1>
</body>
</html>
```

header.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="logoutServlet">Logout</a>
<hr>
</body>
</html>
```

loggedout.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Logged Out</title>
</head>
<body>
<h1>Logged Out</h1>
<b>Your session has terminated.</b><br>
<a href="index.jsp">Login again</a>
</body>
</html>
```

sqlTest.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1" %>
<%@ page import="com.questions.SQLQuestion" %><%@ include file="header.jsp"
%>
<%@ page import="com.questions.SQLQuestionDAO" %>
<%@ page import="java.util.List" %>
<!DOCTYPE html>
```

```
<html>
<head>
<meta charset="ISO-8859-1">
<title>SQL Test</title>
</head>
<body>
<h1 >Online Quiz Portal</h1>
<h3>SQL Test</h3>
<form action="submitAnswerServlet" method="post">
<%
List<SQLQuestion> questionList =
SQLQuestionDAO.getAllSQLQuestions();
if (questionList != null) {
for (SQLQuestion question : questionList) { %>
<h3><%= question.getQuestion() %></h3>
<input type="hidden" name="questionId" value="<%=
question.getId() %>">
<input type="radio" name="answer<%= question.getId() %>"
value="1"> <%= question.getOption1() %><br>
<input type="radio" name="answer<%= question.getId() %>"
value="2"> <%= question.getOption2() %><br>
<input type="radio" name="answer<%= question.getId() %>"
value="3"> <%= question.getOption3() %><br>
<input type="radio" name="answer<%= question.getId() %>"
value="4"> <%= question.getOption4() %><br>
<hr>
<% } } %>
<input type="submit" value="Submit">
</form>
<br>
<br>
<form action="userdashboard.jsp" method="get">
<input type="submit" value="Back to Dashboard">
</form>
```

```
</body>
</html>
```

sqlTestResult.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1" %>
<!DOCTYPE html><%@ include file="header.jsp" %>
<html>
<head>
<meta charset="ISO-8859-1">
<title>SQL Test Result</title>
</head>
<body>
<h1>SQL Test Result</h1>
<p>Your score: <%= request.getAttribute("score") %></p>
<a href="userdashboard.jsp">Back to Dashboard</a>
</body>
</html>
```

userquizdetails.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%><%@ include file="header.jsp" %>
<%@ page import="com.user.User" %>
<%@ page import="com.user.UserDAO" %>
<%@ page import="java.util.List" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Admin Dashboard</h1>
<table>
<tr>
<th>User ID</th>
```

```
<th>Name</th>
<th>Email</th>
<th>Score</th>
</tr>
<%
UserDAO userDAO = new UserDAO();
List<User> userList = userDAO.getAllUsers();
if (userList != null) {
for (User user : userList) { %>
<tr>
<td><%= user.getId() %></td>
<td><%= user.getName() %></td>
<td><%= user.getEmail() %></td>
<td><%=  request.getAttribute("score")  %></td>
</tr>
<% } } %>
</table>
</body>
</html>
```

viewsqllist.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1" %>
<%@ page import="com.questions.SQLQuestion" %><%@ include file="header.jsp"
%>
<%@ page import="com.questions.SQLQuestionDAO" %>
<%@ page import="java.util.List" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>View SQL Questions</title>
<style>
table {
width: 100%;
```

```
border-collapse: collapse;
}
th, td {
border: 1px solid black;
padding: 8px;
text-align: left;
}
tr:nth-child(even) {
background-color: #f2f2f2;
}
</style>
</head>
<body>
<h1>View SQL Questions</h1>
<table>
<thead>
<tr>
<th>ID</th>
<th>Question</th>
<th>Options</th>
<th>Correct Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<%
SQLQuestionDAO sqlQuestionDAO = new SQLQuestionDAO();
List<SQLQuestion> questionList =
sqlQuestionDAO.getAllSQLQuestions();
for (SQLQuestion question : questionList) { %>
<tr>
            <td><%= question.getId() %></td>
            <td><%= question.getQuestion() %></td>
            <td><%= question.getOption1() %>, <%=
question.getOption2() %>, <%= question.getOption3() %>, <%=
question.getOption4() %></td>
                    <td><%= question.getCorrectOption() %></td>
                    <td>
                        <a href="editsqlquestion?id=<%= question.getId()
%>">Edit</a>
                        <a href="deletesqlquestion?id=<%= question.getId()
%>">Delete</a>
                    </td>
                </tr>
            <% } %>
        </tbody>
    </table>

    <br> <br>
    <a href="admindashboardpage.jsp">Go Back Admin Dashboard</a>
</body>
</html>
```

viewuser.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ page import="com.user.User"%>
<%@ page import="com.user.UserDAO"%>
<%@ page import="java.util.List"%><%@ include file="header.jsp" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>User List</title>
<style>
```

```
    table {
        width: 100%;
        border: 1px solid black;
    }

    th, td {
        border: 1px solid black;
        padding: 8px;
        text-align: left;
    }

    tr:nth-child(even) {
        background-color: #f2f2f2;
    }

    tr:hover {
        background-color: #e6e6e6;
    }
</style>
</head>
<body>
<h1>User List</h1>
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Email</th>
<th>Date of Birth</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<%
```

```
UserDAO userDAO = new UserDAO();
List<User> userList = userDAO.getAllUsers();
for (User user : userList) { %>
<tr>
<td><%= user.getId() %></td>
<td><%= user.getName() %></td>
<td><%= user.getEmail() %></td>
<td><%= user.getDob() %></td>
<td>
<a href="edituser?userId=<%= user.getId() %>">Edit</a>
<a href="deleteuser?userId=<%= user.getId()
%>">Delete</a>
</td>
</tr>
<% } %>
</tbody>
</table>
<br> <br>
<a href="admindashboardpage.jsp">Go Back to Admin Dashboard</a>
</body>
</html>
```

Java Code

Package Name : com.question

AddSQLQuestionServlet.java

```
package com.questions;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.hibernate.Session;
import org.hibernate.Transaction;
import com.user.HibernateUtil;
```

```
@WebServlet("/addsqlquestion")
public class AddSQLQuestionServlet extends HttpServlet {
private static final long serialVersionUID = 1L;
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
String question = request.getParameter("question");
String option1 = request.getParameter("option1");
String option2 = request.getParameter("option2");
String option3 = request.getParameter("option3");
String option4 = request.getParameter("option4");
int correctOption =
Integer.parseInt(request.getParameter("correctOption"));
// Create a new SQLQuestion object and set the values
SQLQuestion sqlQuestion = new SQLQuestion();
sqlQuestion.setQuestion(question);
sqlQuestion.setOption1(option1);
sqlQuestion.setOption2(option2);
sqlQuestion.setOption3(option3);
sqlQuestion.setOption4(option4);
sqlQuestion.setCorrectOption(correctOption);
// Save the SQLQuestion object to the database using Hibernate
Session session = HibernateUtil.getSessionFactory().openSession();
Transaction transaction = session.beginTransaction();
session.save(sqlQuestion);
try {
transaction.commit();
} catch (Exception e) {
if (transaction != null) {
transaction.rollback();
}
e.printStackTrace();
} finally {
session.close();
}
response.sendRedirect("addquestion.jsp");
}
}
```

DeleteSQLQuestionServlet.java

```
package com.questions;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/deletesqlquestion")
public class DeleteSQLQuestionServlet extends HttpServlet {
private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
// Retrieve the question ID from the request parameter
Long questionId = Long.parseLong(request.getParameter("id"));
// Delete the SQL question from the database
SQLQuestionDAO.deleteSQLQuestion(questionId);
// Redirect back to the view page
response.sendRedirect(request.getContextPath() + "/viewsqllist");
}
}
```

EditSQLQuestionServlet.java

```
package com.questions;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/editsqlquestion")
public class EditSQLQuestionServlet extends HttpServlet {
```

```
private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
// Retrieve the question ID from the request parameter
Long questionId = Long.parseLong(request.getParameter("id"));
// Retrieve the SQL question from the database
SQLQuestion sqlQuestion =
SQLQuestionDAO.getSQLQuestionById(questionId);
// Forward the question object to the JSP page for editing
request.setAttribute("question", sqlQuestion);
request.getRequestDispatcher("editsqlquestion.jsp").forward(request,
response);
}
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
// Retrieve the question ID from the request parameter
Long questionId = Long.parseLong(request.getParameter("id"));
// Retrieve the SQL question from the database
SQLQuestion sqlQuestion =
SQLQuestionDAO.getSQLQuestionById(questionId);
// Update the SQL question with the form data
sqlQuestion.setQuestion(request.getParameter("question"));
sqlQuestion.setOption1(request.getParameter("option1"));
sqlQuestion.setOption2(request.getParameter("option2"));
sqlQuestion.setOption3(request.getParameter("option3"));
sqlQuestion.setOption4(request.getParameter("option4"));
sqlQuestion.setCorrectOption(Integer.parseInt(request.getParameter("correct
 Option")));
// Update the SQL question in the database
SQLQuestionDAO.updateSQLQuestion(sqlQuestion);
// Redirect to the SQL question list page
response.sendRedirect("viewsqllist.jsp");
}
}
```

SQLQuestion.java

```
package com.questions;

import java.util.List;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

import javax.persistence.criteria.CriteriaBuilder;

import javax.persistence.criteria.CriteriaQuery;

import javax.persistence.criteria.Root;

import javax.transaction.HeuristicMixedException;

import javax.transaction.HeuristicRollbackException;

import javax.transaction.RollbackException;

import javax.transaction.SystemException;

import javax.transaction.Transaction;

import org.hibernate.Session;

import com.user.HibernateUtil;


@Entity

@Table(name = "sql_question")

public class SQLQuestion {

    @Id

    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;


    private String question;

    private String option1;

    private String option2;

    private String option3;

    private String option4;

    private int correctOption;

 public Long getId() {

  return id;
 }
 public void setId(Long id) {
  this.id = id;
 }
 public String getQuestion() {
  return question;
 }
 public void setQuestion(String question) {
  this.question = question;
 }
 public String getOption1() {
  return option1;
 }
 public void setOption1(String option1) {
  this.option1 = option1;
 }
 public String getOption2() {
  return option2;
 }
 public void setOption2(String option2) {
  this.option2 = option2;
 }
 public String getOption3() {
  return option3;
 }
 public void setOption3(String option3) {
  this.option3 = option3;
 }
 public String getOption4() {
  return option4;
 }
 public void setOption4(String option4) {
  this.option4 = option4;
 }
```

```
 public int getCorrectOption() {
  return correctOption;
 }
 public void setCorrectOption(int correctOption) {
  this.correctOption = correctOption;
 }
 public String formatOptions() {
        return option1 + ", " + option2 + ", " + option3 + ", " + option4;
    }
}
```

SQLQuestionDAO.java

```
package com.questions;

import java.util.List;

import javax.persistence.criteria.CriteriaBuilder;
import javax.persistence.criteria.CriteriaQuery;
import javax.persistence.criteria.Root;

import org.hibernate.Session;
import org.hibernate.Transaction;

import com.user.HibernateUtil;

public class SQLQuestionDAO {
 public static List<SQLQuestion> getAllSQLQuestions() {
  try (Session session =
HibernateUtil.getSessionFactory().openSession()) {
   CriteriaBuilder builder = session.getCriteriaBuilder();
   CriteriaQuery<SQLQuestion> query =
builder.createQuery(SQLQuestion.class);
   Root<SQLQuestion> root = query.from(SQLQuestion.class);
   query.select(root);

   return session.createQuery(query).getResultList();
  }
 }

 public static String formatOptions(SQLQuestion question) {
  String options = question.getOption1() + ", " +
question.getOption2() + ", " + question.getOption3() + ", "
    + question.getOption4();
  return options;
 }

 public static SQLQuestion getSQLQuestionById(Long questionId) {
  try (Session session =
HibernateUtil.getSessionFactory().openSession()) {
   return session.get(SQLQuestion.class, questionId);
  }
  // Implement other methods for CRUD operations
  // ...

 }

 public static void deleteSQLQuestion(Long questionId) {
  try (Session session =
HibernateUtil.getSessionFactory().openSession()) {
   Transaction transaction = session.beginTransaction();

   SQLQuestion question = session.get(SQLQuestion.class,
questionId);
   if (question != null) {
    session.delete(question);
 }

}
transaction.commit();
}
public static void updateSQLQuestion(SQLQuestion sqlQuestion) {
try (Session session =
HibernateUtil.getSessionFactory().openSession()) {
Transaction transaction = session.beginTransaction();
session.update(sqlQuestion);
transaction.commit();
}
}
}
```

SQLTestServlet.java

```
package com.questions;
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/sqlTestServlet")
public class SQLTestServlet extends HttpServlet {
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
List<SQLQuestion> sqlQuestions =
SQLQuestionDAO.getAllSQLQuestions();
request.setAttribute("sqlQuestions", sqlQuestions);
request.getRequestDispatcher("sqlTest.jsp").forward(request,
response);
}
}
```

SubmitAnswerServlet.java

```
package com.questions;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
@WebServlet("/submitAnswerServlet")
public class SubmitAnswerServlet extends HttpServlet {
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
Map<Long, Integer> userAnswers = new HashMap<>();
// Get the submitted answers from the request parameters
List<SQLQuestion> questionList =
SQLQuestionDAO.getAllSQLQuestions();
for (SQLQuestion question : questionList) {
Long questionId = question.getId();
int selectedOption =
Integer.parseInt(request.getParameter("answer" + questionId));
userAnswers.put(questionId, selectedOption);
}
// Calculate the score
int score = calculateScore(userAnswers, questionList);
// Set the score as a request attribute and forward to a result
page
request.setAttribute("score", score);
request.getRequestDispatcher("/sqlTestResult.jsp").forward(request,
response);
}
private int calculateScore(Map<Long, Integer> userAnswers,
List<SQLQuestion> questionList) {
int score = 0;
for (SQLQuestion question : questionList) {
Long questionId = question.getId();
int selectedOption = userAnswers.getOrDefault(questionId, -1);
int correctOption = question.getCorrectOption();
if (selectedOption == correctOption) {
score++; // Increment score for each correct answer
}
}
return score;
}
}
```

ViewSQLQuestionServlet.java

```
package com.questions;
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/viewsqllist")
public class ViewSQLQuestionServlet extends HttpServlet {
private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
// Retrieve the list of SQL questions from the database
        List<SQLQuestion> questionList =
SQLQuestionDAO.getAllSQLQuestions();

        // Forward the list to the JSP page for rendering
        request.setAttribute("questionList", questionList);
        request.getRequestDispatcher("viewsqllist.jsp").forward(request,
response);
    }
}
```

Package Name : com.user

Admin.java

```
package com.user;

import javax.persistence.*;

@Entity
@Table(name = "admin")
public class Admin {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "email")
    private String email;

    @Column(name = "password")
    private String password;

 public int getId() {
  return id;
 }

 public void setId(int id) {
  this.id = id;
 }

 public String getEmail() {
  return email;
 }

 public void setEmail(String email) {
  this.email = email;
 }

 public String getPassword() {
  return password;
 }

 public void setPassword(String password) {
  this.password = password;
 }

}
```

AdminDAO.java

```
package com.user;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.query.Query;
public class AdminDAO {
private SessionFactory sessionFactory;
public AdminDAO() {
sessionFactory = HibernateUtil.getSessionFactory();
}
public Admin validateAdmin(String email, String password) {
try (Session session = sessionFactory.openSession()) {
Query<Admin> query = session.createQuery("FROM Admin WHERE
email = :email AND password = :password", Admin.class);
query.setParameter("email", email);
query.setParameter("password", password);
return query.uniqueResult();
} catch (Exception e) {
e.printStackTrace();
}
return null;
}
}
```

AdminLoginServlet.java

```
package com.user;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/adminLoginServlet")
public class AdminLoginServlet extends HttpServlet {
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
String email = request.getParameter("email");
String password = request.getParameter("password");
AdminDAO adminDAO = new AdminDAO();
Admin admin = adminDAO.validateAdmin(email, password);
if (admin != null) {
request.getSession().setAttribute("adminAuthenticated", true);
response.sendRedirect("admindashboardpage.jsp");
} else {
// Invalid admin credentials, display an error message or
redirect to login page
response.sendRedirect("adminlogin.jsp?error=invalidAdminCredentials");
}
}
}
```

DeleteUserServlet.java

```
package com.user;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/deleteuser")
public class DeleteUserServlet extends HttpServlet {
/**
*
*/
private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
Long userId = Long.parseLong(request.getParameter("userId"));
UserDAO userDAO = new UserDAO();
userDAO.deleteUser(userId);
response.sendRedirect("viewuser.jsp");
}
}
```

EditUserServlet.java

```
package com.user;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/edituser")
public class EditUserServlet extends HttpServlet {
/**
*
*/
private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
Long userId = Long.parseLong(request.getParameter("userId"));
UserDAO userDAO = new UserDAO();
User user = userDAO.getUserById(userId); // to to userdao class
call the method
request.setAttribute("user", user);
request.getRequestDispatcher("edituser.jsp").forward(request,
response);
}
}
```

HibernateUtil.java

```
package com.user;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class HibernateUtil {
private static final SessionFactory sessionFactory =
buildSessionFactory();
private static SessionFactory buildSessionFactory() {
try {
// Create the SessionFactory from hibernate.cfg.xml
Configuration configuration = new Configuration();
configuration.configure("hibernate.cfg.xml");
return configuration.buildSessionFactory();
} catch (Throwable ex) {
// Handle any initial SessionFactory creation errors
System.err.println("Initial SessionFactory creation failed: " +
ex);
}
throw new ExceptionInInitializerError(ex);
}
public static SessionFactory getSessionFactory() {
return sessionFactory;
}
public static void shutdown() {
// Close caches and connection pools
getSessionFactory().close();
}
}
```

LogoutServlet.java

```
package com.user;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet("/logoutServlet")
public class LogoutServlet extends HttpServlet {
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
HttpSession session = request.getSession();
session.invalidate();
response.sendRedirect("loggedout.jsp");
}
}
```

RegisterServlet.java

```
package com.user;
import java.io.IOException;
import java.sql.Date;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/registerServlet")
public class RegisterServlet extends HttpServlet {
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
String name = request.getParameter("name");
String email = request.getParameter("email");
String password = request.getParameter("password");
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
Date dob = null;
try {
dob = new
Date(dateFormat.parse(request.getParameter("dob")).getTime());
} catch (ParseException e) {
e.printStackTrace();
}
User user = new User();
user.setName(name);
user.setEmail(email);
user.setPassword(password);
user.setDob(dob);
UserDAO userDAO = new UserDAO();
userDAO.registerUser(user);
response.sendRedirect("userdashboard.jsp");
}
}
```

UpdateUserServlet.java

```
package com.user;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


@WebServlet("/updateuser")
public class UpdateUserServlet extends HttpServlet {
   /**
  *
  */
 private static final long serialVersionUID = 1L;


 protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
   String userIdParam = request.getParameter("userId");
      if (userIdParam != null && !userIdParam.isEmpty()) {
          try {
              Long userId = Long.parseLong(userIdParam);
              String name = request.getParameter("name");
              String email = request.getParameter("email");


              UserDAO userDAO = new UserDAO();
              User user = userDAO.getUserById(userId);
              if (user != null) {
                  user.setName(name);
                  user.setEmail(email);
                  userDAO.updateUser(user);


                  response.sendRedirect("viewuser.jsp");
                  return;
              }
          } catch (NumberFormatException e) {
              // Handle parsing error
          }
      }
      // Handle invalid userId or other errors
      response.sendRedirect("error.jsp");
  }
}
```

User.java

```
package com.user;

import java.sql.Date;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String email;
    private String password;
    private Date dob;
    public Long getId() {
  return id;
 }

 public void setId(Long id) {
  this.id = id;
}

public String getName() {
 return name;
}

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

public String getEmail() {
 return email;
}

public void setEmail(String email) {
 this.email = email;
}

public String getPassword() {
 return password;
}

public void setPassword(String password) {
 this.password = password;
}

public Date getDob() {
 return dob;
}

public void setDob(Date dob) {
	this.dob = dob;
}
```

```
}
```

UserDAO.java

```
package com.user;

import java.util.List;

import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

public class UserDAO {
    public void registerUser(User user) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction transaction = session.beginTransaction();
        session.save(user);
        transaction.commit();
        session.close();
    }

    public User validateUser(String email, String password) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Query<User> query = session.createQuery("FROM User WHERE email =
:email AND password = :password", User.class);
        query.setParameter("email", email);
        query.setParameter("password", password);
        User user = query.uniqueResult();
        session.close();
        return user;

    }
    // ------------------------------- show the list of user -------------------
    public List<User> getAllUsers() {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Query<User> query = session.createQuery("FROM User", User.class);
        List<User> userList = query.list();
        session.close();
        return userList;
    }

    // -------------------------------- show the list of user -------------------
    public User getUserById(Long userId) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        User user = session.get(User.class, userId);
        session.close();
        return user;
    }
 // -------------------------------- Update the list of user -------------------
    public void updateUser(User user) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction transaction = session.beginTransaction();
        session.update(user);
        transaction.commit();
        session.close();
    }
 // -------------------------------- Delete the list of user -------------------
    public void deleteUser(Long userId) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction transaction = session.beginTransaction();

        User user = session.get(User.class, userId);
        if (user != null) {
            session.delete(user);
        }

        transaction.commit();
        session.close();
    }

}
```

UserLoginServlet.java

```
package com.user;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet("/userloginServlet")
public class UserLoginServlet extends HttpServlet {
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
String email = request.getParameter("email");
String password = request.getParameter("password");
UserDAO userDAO = new UserDAO();
User user = userDAO.validateUser(email, password);
if (user != null) {
HttpSession session = request.getSession();
session.setAttribute("name", user.getName());
response.sendRedirect("userdashboard.jsp");
} else {
request.setAttribute("error", "Invalid email or password");
request.getRequestDispatcher("userlogin.jsp").forward(request,
response);
}
}
}
```

hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
<property name="hibernate.connection.password">root</property>
<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/Register</prope
 rty>
<property name="hibernate.connection.username">root</property>
<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
<mapping class="com.user.User"/>
<mapping class="com.user.Admin"/>
<mapping class="com.questions.SQLQuestion" />
</session-factory>
</hibernate-configuration>
```

MYSQL Query

```
create database Register;
USE register;
CREATE TABLE users (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255),
email VARCHAR(255),
password VARCHAR(255),
dob DATE
);
select *from users;
CREATE TABLE admin (
id INT AUTO_INCREMENT PRIMARY KEY,
email VARCHAR(255) NOT NULL,
password VARCHAR(255) NOT NULL
);
INSERT INTO admin (email, password)
VALUES ('admin@gmail.com', 'admin');
CREATE TABLE sql_question (
id INT PRIMARY KEY AUTO_INCREMENT,
question VARCHAR(255),
option1 VARCHAR(255),
option2 VARCHAR(255),
option3 VARCHAR(255),
option4 VARCHAR(255),
correctOption VARCHAR(255)
);
```