***Source Code of Online Quiz Application***

**Creating Menu for Home Page**

< div id='cssmenu'>

< ul>

< li class=''>< a href='${pageContext.request.contextPath}'>< span>Home< /span>< /a>< /li>

< li>< a href='${pageContext.request.contextPath}/login'>< span>Login< /span>< /a>< /li>

< li>< a href='${pageContext.request.contextPath}/register'>< span>Register< /span>< /a>< /li>

< li class='#'>< a href='#'>< span>Submit a Question< /span>< /a>< /li>

< li class='#'>< a href='#'>< span>Feedback< /span>< /a>< /li>

< li>< a href='#'>< span>Contribute< /span>< /a>< /li>

< li>< a href='#'>< span>Contact us< /span>< /a>< /li>

< /ul>

< /div>

**Checking whether the user is logged in or not**

< c:if test='${not empty sessionScope.user}'>

< div style="position:absolute;top:70px;left:1100px">

Logged as < a href="#" class="button username">${sessionScope.user}< /a>

< /div>

< div style="position:absolute;top:70px;left:1300px">

< a href='${pageContext.request.contextPath}/logout'>Logout< /a>

< /div>

< /c:if>

**Showing the quiz images on home page**

< div style="position:absolute;left:120px;top:60px">

< table cellpadding="0" cellspacing="50">

< tr>

< td>< a href="takeExam?test=java">< img height="200" width="200" src="${pageContext.request.contextPath}/images/java.png"/>< /a>< /td>

< td>< a href="takeExam?test=javascript">< img height="200" width="200" src="${pageContext.request.contextPath}/images/javascript.png"/>< /a>< /td>

< td>< a href="takeExam?test=sql">< img height="200" width="200" src="${pageContext.request.contextPath}/images/sql-logo.png"/>< /a>< /td>

< td>< a href="takeExam?test=python">< img height="200" width="200" src="${pageContext.request.contextPath}/images/python.jpg"/>< /a>< /td>

< /tr>

< tr>

< td>< a href="takeExam?test=css">< img height="200" width="200" src="${pageContext.request.contextPath}/images/css.jpg"/>< /a>< /td>

< td>< a href="takeExam?test=php">< img height="200" width="200" src="${pageContext.request.contextPath}/images/php-logo.jpg"/>< /a>< /td>

< td>< a href="takeExam?test=linux">< img height="200" width="200" src="${pageContext.request.contextPath}/images/logo-linux.png"/>< /a>< /td>

< td>< a href="takeExam?test=mongodb">< img height="200" width="200" src="${pageContext.request.contextPath}/images/mongodb\_logo.png"/>< /a>< /td>

< /tr>

< /table>

< /div>

**Registration Code**

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String username=request.getParameter("username");

String email=request.getParameter("email");

String password=request.getParameter("password");

Connection con=DatabaseConnectionFactory.createConnection();

try

{

Statement st=con.createStatement();

String sql = "INSERT INTO users values ('"+username+"','"+password+"','"+email+"')";

System.out.println(sql);

st.executeUpdate(sql);

}catch(SQLException sqe){System.out.println("Error : While Inserting record in database");}

try

{

con.close();

}catch(SQLException se){System.out.println("Error : While Closing Connection");}

request.setAttribute("newUser",username);

RequestDispatcher dispatcher=request.getRequestDispatcher("/WEB-INF/jsps/regSuccess.jsp");

dispatcher.forward(request, response);

}

Getting Database Connection

**Users’ table**

create table users(username varchar(50),email varchar(50),password varchar(50))

If you are working with some other database like Oracle you have to change the properties of the DatabaseConnectionFactory class accordingly.

**DatabaseConnectionFactory.java**

public class DatabaseConnectionFactory {

private static String dbURL="jdbc:mysql://localhost/quiz";

private static String dbUser="root";

private static String dbPassword="";

public static Connection createConnection()

{

Connection con=null;

try{

try {

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

}

catch(ClassNotFoundException ex) {

System.out.println("Error: unable to load driver class!");

System.exit(1);

}

con = DriverManager.getConnection(dbURL,dbUser,dbPassword);

}

catch(SQLException sqe){ System.out.println("Error: While Creating connection to database");sqe.printStackTrace();}

return con;

}

}

## Creating the Login Page

Login page is very much similar to registration page where we are providing two input fields asking user to provide a username and password. Once we get the username and password entered by the user we pass it to LoginController to authenticate user.

**Login Validation Code**

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String username=request.getParameter("username");

String password=request.getParameter("password");

Connection con=DatabaseConnectionFactory.createConnection();

ResultSet set=null;

int i=0;

try

{

Statement st=con.createStatement();

String sql = "Select \* from users where username='"+username+"' and password='"+password+"' ";

System.out.println(sql);

set=st.executeQuery(sql);

while(set.next())

{

i=1;

}

if(i!=0)

{ HttpSession session=request.getSession();

session.setAttribute("user",username);

RequestDispatcher rd=request.getRequestDispatcher("/WEB-INF/jsps/home.jsp");

rd.forward(request, response);

}

else

{ request.setAttribute("errorMessage","Invalid username or password");

RequestDispatcher rd=request.getRequestDispatcher("/WEB-INF/jsps/login.jsp");

rd.forward(request, response);

}

}catch(SQLException sqe){System.out.println("Error : While Fetching records from database");}

try

{

con.close();

}catch(SQLException se){System.out.println("Error : While Closing Connection");}

}

## MainController for the Application

It is the MainController where we have written the code to redirect the user to appropriate page according to the incoming request url.

@WebServlet(urlPatterns = { "/login", "/register", "/takeExam", "/logout" })

public class MainController extends HttpServlet {

private static final long serialVersionUID = 1L;

protected void doGet(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

String applicationContextPath = request.getContextPath();

if (request.getRequestURI().equals(applicationContextPath + "/")) {

RequestDispatcher dispatcher = request

.getRequestDispatcher("/WEB-INF/jsps/home.jsp");

dispatcher.forward(request, response);

} else if (request.getRequestURI().equals(

applicationContextPath + "/login")) {

RequestDispatcher dispatcher = request

.getRequestDispatcher("/WEB-INF/jsps/login.jsp");

dispatcher.forward(request, response);

} else if (request.getRequestURI().equals(

applicationContextPath + "/register")) {

RequestDispatcher dispatcher = request

.getRequestDispatcher("/WEB-INF/jsps/register.jsp");

dispatcher.forward(request, response);

} else if (request.getRequestURI().equals(

applicationContextPath + "/takeExam")) {

request.getSession().setAttribute("currentExam", null);

String exam = request.getParameter("test");

request.getSession().setAttribute("exam", exam);

System.out.println(request.getSession().getAttribute("user"));

if (request.getSession().getAttribute("user") == null) {

request.getRequestDispatcher("/login").forward(request,

response);

} else {

RequestDispatcher dispatcher = request

.getRequestDispatcher("/WEB-INF/jsps/quizDetails.jsp");

dispatcher.forward(request, response);

}

} else if (request.getRequestURI().equals(

applicationContextPath + "/logout")) {

request.getSession().invalidate();

RequestDispatcher dispatcher = request

.getRequestDispatcher("/WEB-INF/jsps/home.jsp");

dispatcher.forward(request, response);

}

}

}

## Implementing the Logout Functionality

Once the user clicks on logout, link session is invalidated and all the objects bind in the session are removed.

request.getSession().invalidate();

## Storing the Quiz questions

Note that we have stored the questions in separate XML files, not in the database.

< quiz>

< title>MongoDB Quiz (01/09/2015)< /title>

< questions>

< question>

< quizquestion>MongoDB is a < /quizquestion>

< answer>Relational Database< /answer>

< answer>Object Relational Database< /answer>

< answer>Graph Database< /answer>

< answer>Document Database< /answer>

< correct>3< /correct>

< /question>

< question>

< quizquestion>What is the name of MongoDB server ?< /quizquestion>

< answer>mongoserver< /answer>

< answer>mongod< /answer>

< answer>mongodb< /answer>

< answer>mongo< /answer>

< correct>1< /correct>

< /question>

< question>

< quizquestion>What is the name of MongoDB client ?< /quizquestion>

< answer>mongo< /answer>

< answer>mongod< /answer>

< answer>mongodb< /answer>

< answer>mongo-client< /answer>

< correct>0< /correct>

< /question>

< /questions>

< /quiz>

**QuizQuestion.java**

public class QuizQuestion {

int questionNumber;

String question;

String questionOptions[];

int correctOptionIndex;

public String getQuestion()

{

return question;

}

public int getQuestionNumber()

{

return questionNumber;

}

public void setQuestionNumber(int i)

{

questionNumber=i;

}

public int getCorrectOptionIndex()

{

return correctOptionIndex;

}

public String[] getQuestionOptions()

{

return questionOptions;

}

public void setQuestion(String s)

{

question=s;

}

public void setCorrectOptionIndex(int i)

{

correctOptionIndex=i;

}

public void setQuestionOptions(String[]s)

{

questionOptions=s;

}

}

Let’s see what is there in the exam class

public class Exam {

Document dom;

public int currentQuestion=0;

public Map selections=new LinkedHashMap();

public ArrayList questionList = new ArrayList(10);

public Exam(String test) throws SAXException,ParserConfigurationException,IOException, URISyntaxException{

dom=CreateDOM.getDOM(test);

}

// code

}

## Submitting the Exam and Evaluating Exam Result

public int calculateResult(Exam exam){

int totalCorrect=0;

Map<Integer,Integer> userSelectionsMap=exam.selections;

List userSelectionsList=new ArrayList(10);

for (Map.Entry<Integer, Integer> entry :userSelectionsMap.entrySet()) {

userSelectionsList.add(entry.getValue());

}

List questionList=exam.questionList;

List correctAnswersList=new ArrayList(10);

for(QuizQuestion question: questionList){

correctAnswersList.add(question.getCorrectOptionIndex());

}

for(int i=0;i<selections.size();i++){

System.out.println(userSelectionsList.get(i)+" --- "+correctAnswersList.get(i));

if((userSelectionsList.get(i)-1)==correctAnswersList.get(i)){

totalCorrect++;

}

}

System.out.println("You Got "+totalCorrect+" Correct");

return totalCorrect;

}