



ASSIGNMENT 4

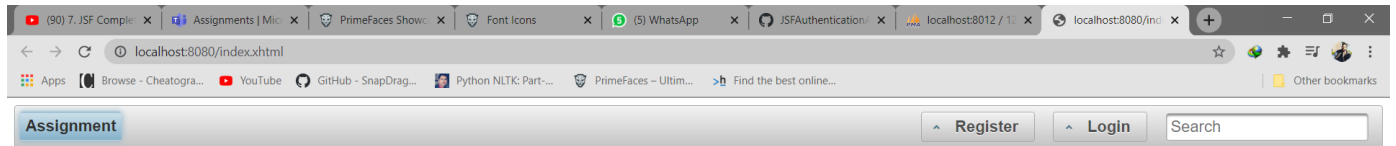
Enterprise App Development



ASIM ZUBAIR
BSE173121

SCREENSHOTS

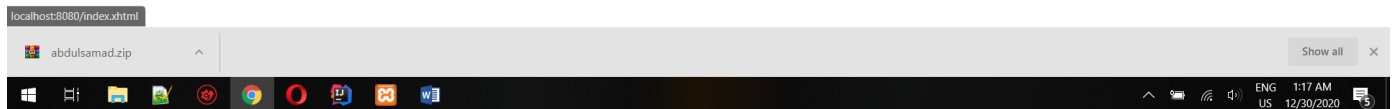
Index.xhtml



ASSALAM-O-ALIKUM

Description of project

1. Create a Registration Page with following fields in JSF with Primefaces and store the data in database.
 - a) Username
 - b) First Name
 - c) Last Name
 - d) Email
 - e) Password
 - f) Confirm Password
 - g) Date of Birth
 - h) Gender
2. Create a Login Page and login with username and password from the saved data.
3. User should login into a page showing first name, last name, email, date of birth and gender of all users in tabular form.



Register.xhtml

validations

The screenshot shows a web browser window displaying the 'Register' form. The form has two columns of input fields: Username, Lastname, Password, DateOfBirth, and Firstname, Email, ConfirmPassword, Gender. An 'Ajax Submit' button is at the bottom left. On the right side, a vertical stack of six error messages is displayed, each with a red 'X' icon and the text 'Validation Error: Value is required.' The error messages are for the following fields: j_idt4:j_idt14 (Username), j_idt4:j_idt17 (Firstname), j_idt4:j_idt20 (Email), j_idt4:j_idt23 (ConfirmPassword), j_idt4:inlineFeedback (Gender), and j_idt4:j_idt28 (DateOfBirth). The browser's address bar shows 'localhost:8080/register.xhtml?_afid=8jfi=register.xhtml'. The taskbar at the bottom shows various application icons and the system clock indicating 1:18 AM on 12/30/2020.

The screenshot shows the same 'Register' form, but now with valid data entered in the input fields. The Username field contains 'khan', the Firstname field contains 'khan', the Lastname field contains 'khan', the Email field contains 'khan@gmail.com', the Password field contains '*****', and the ConfirmPassword field contains 'ConfirmPassword'. The DateOfBirth field contains 'Date-Of-Birth' and the Gender field contains 'Gender'. The 'Ajax Submit' button is still present at the bottom left. The browser's address bar shows 'localhost:8080/register.xhtml?_afid=8jfi=register.xhtml'. The taskbar at the bottom shows various application icons and the system clock indicating 1:20 AM on 12/30/2020.

On successful submission

Assignment

Register

Username: khan

Firstname: khan

Lastname: khan

Email: khan@gmail.com

Password: ****

ConfirmPassword: ****

DateOfBirth: 121472

Gender: male

Ajax Submit

abdulesamad.zip

ENG US 12:21 AM 12/30/2020

Login.xhtml

Assignment

Login

Email: khan@gmail.com

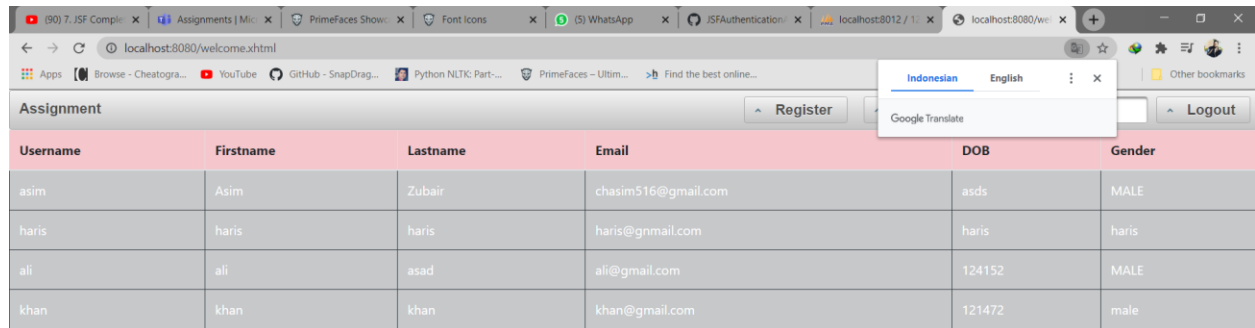
Password: ****

Login

abdulesamad.zip

ENG US 12:22 AM 12/30/2020

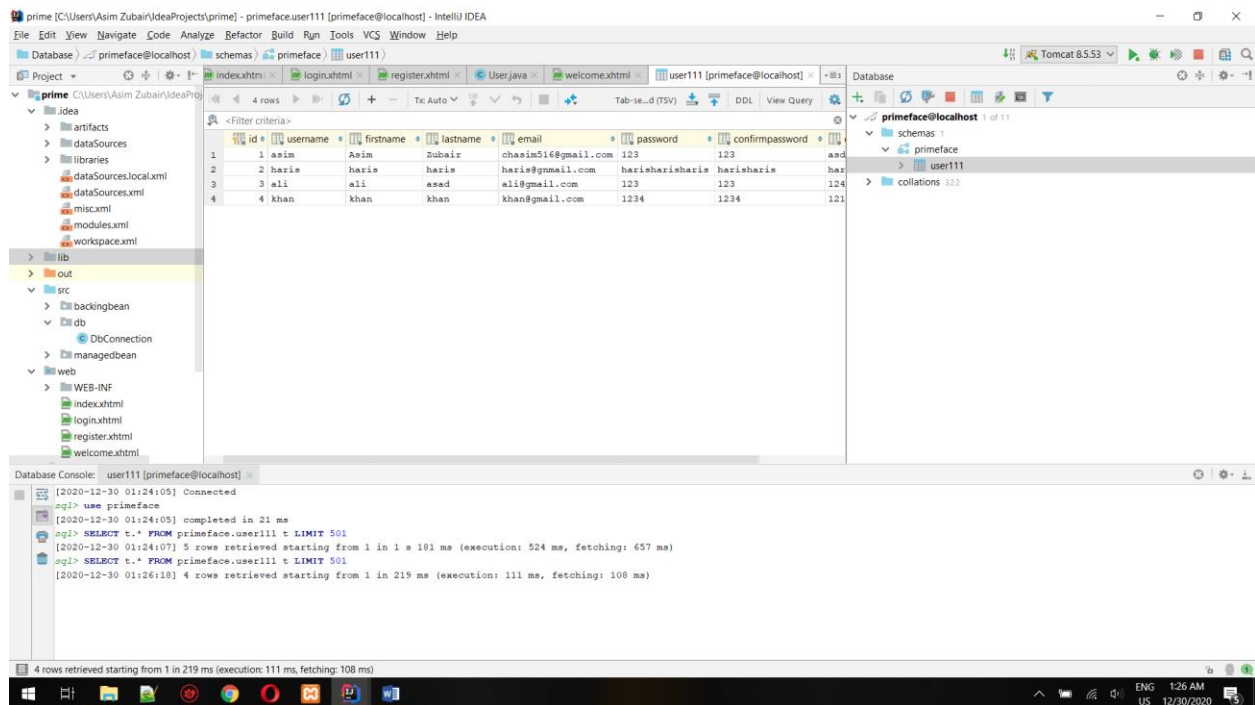
Unsuccessful login welcome.xhtml will load



The screenshot shows a web browser window with the URL `localhost:8080/welcome.xhtml`. The page displays a table with user information. Above the table, there are buttons for 'Register' and 'Logout', and a Google Translate widget. The table has columns for Username, Firstname, Lastname, Email, DOB, and Gender.

Username	Firstname	Lastname	Email	DOB	Gender
asim	Asim	Zubair	chasim516@gmail.com	asds	MALE
haris	haris	haris	haris@gmail.com	haris	haris
ali	ali	asad	ali@gmail.com	124152	MALE
khan	khan	khan	khan@gmail.com	121472	male

Database screenshots



The screenshot shows the IntelliJ IDEA interface. The main editor displays a table with 4 rows and 8 columns. The columns are: id, username, firstname, lastname, email, password, confirmpassword, and an unnamed column. The rows contain data for users: asim, haris, ali, and khan.

id	username	firstname	lastname	email	password	confirmpassword	
1	asim	Asim	Zubair	chasim516@gmail.com	123	123	
2	haris	haris	haris	haris@gmail.com	harisharisharis	harisharis	
3	ali	ali	asad	ali@gmail.com	123	123	
4	khan	khan	khan	khan@gmail.com	1234	1234	

The Database Console at the bottom shows the following SQL queries and results:

```
[2020-12-30 01:24:05] Connected
sql> use primeface
[2020-12-30 01:24:05] completed in 21 ms
sql> SELECT t.* FROM primeface.user111 t LIMIT 501
[2020-12-30 01:24:07] 5 rows retrieved starting from 1 in 1 s 101 ms (execution: 524 ms, fetching: 657 ms)
sql> SELECT t.* FROM primeface.user111 t LIMIT 501
[2020-12-30 01:24:10] 4 rows retrieved starting from 1 in 219 ms (execution: 111 ms, fetching: 108 ms)
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://xmlns.jcp.org/jsf/html"
xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
xmlns:f="http://xmlns.jcp.org/jsf/core"
xmlns:p="http://primefaces.org/ui">
<h:head>

</h:head>
<f:view contentType="text/html">

    <h:form>
        <p:menubar>
            <p:menuitem style="font-weight: bolder" url="index.xhtml" value="Assignment"></p:menuitem>
            <f:facet name="options">
                <p:button outcome="register.xhtml" value="Register" icon="pi pi-star"
                    style="margin-right:20px;"></p:button>
                <p:button outcome="login.xhtml" value="Login" icon="pi pi-star" style="margin-
right:20px;"></p:button>
            </f:facet>

            <f:facet name="options">
                <p:inputText style="margin:1px 10px 1px 1px;" placeholder="Search"/>
            </f:facet>
        </p:menubar>

        <p:outputPanel deferred="true">
            <div class="all-classes-container">
                <h2>ASSALAM-O-ALIKUM</h2>
                <h3>Description of project</h3>
                <p>1. Create a Registration Page with following fields in JSF with Primefaces and store the
data in database.</p>
                <ul>
                    <li>a) Username</li>
                    <li>b) First Name</li>
                </ul>
            </div>
        </p:outputPanel>
    </h:form>
</f:view>
</html>
```

```

        <li>c) Last Name</li>

        <li>d) Email</li>

        <li>e) Password</li>

        <li>f) Confirm Password</li>

        <li>g) Date of Birth</li>

        <li>h) Gender</li>
    </ul>

    <p>2. Create a Login Page and login with username and password from the saved data.</p>

    <p>3. User should login into a page showing first name, last name, email, date of birth and
gender of all users in tabular form.</p>
</div>
</p:outputPanel>

</h:form>
</f:view>
</html>

```

Register.xhtml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns:p="http://primefaces.org/ui">
<h:head>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"></link>
</h:head>
<f:view contentType="text/html">

    <h:form>
        <p:menubar style="color: crimson">
            <p:menuitem style="font-weight: bolder" value="Assignment" url="index.xhtml"></p:menuitem>

            <f:facet name="options">
                <p:commandButton outcome="register.xhtml" type="button" value="Register" icon="pi pi-home"/>
                <p:commandButton outcome="login.xhtml" type="button" value="Login" icon="pi pi-home"/>
            </f:facet>

            <f:facet name="options">
                <p:inputText style="margin: 1px 10px 1px 1px;" placeholder="Search"/>
            </f:facet>
        </p:menubar>
        <h3 style="padding: 25px">Register</h3>

        <p:separator/>
        <div class="ui-g ui-fluid">
            <div class="ui-g-12 ui-md-4">
                <div class="ui-inputgroup">
                    <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="Username"/>
                    <p:inputText placeholder="Username" value="#{userBean.user.username}" required="true"/>
                </div>
            </div>

            <div class="ui-g-12 ui-md-4">
                <div class="ui-inputgroup">
                    <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="Firstname"/>
                    <p:inputText placeholder="Firstname" value="#{userBean.user.firstname}" required="true"/>
                </div>
            </div>

            <div class="ui-g ui-fluid">
                <div class="ui-g-12 ui-md-4">
                    <div class="ui-inputgroup">
                        <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="Lastname"/>

```

```

        <p:inputText placeholder="Lastname" value="#{userBean.user.lastname}" required="true"/>
    </div>
</div>

<div class="ui-g-12 ui-md-4">
    <div class="ui-inputgroup">
        <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="Email"/>
        <p:inputText placeholder="Email" value="#{userBean.user.email}" required="true"/>
    </div>
</div>

</div>
<div class="ui-g ui-fluid">
    <div class="ui-g-12 ui-md-4">
        <div class="ui-inputgroup">
            <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="Password"/>
            <p:password id="inlineFeedback" placeholder="Password" value="#{userBean.user.password}"
                feedback="true" inline="true" required="true"/>
        </div>
    </div>

    <div class="ui-g-12 ui-md-4">
        <div class="ui-inputgroup">
            <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="ConfirmPassword"
                required="true"/>
            <p:password placeholder="ConfirmPassword" value="#{userBean.user.confirmpassword}"
                required="true"/>
        </div>
    </div>

</div>
<div class="ui-g ui-fluid">
    <div class="ui-g-12 ui-md-4">
        <div class="ui-inputgroup">
            <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="DateOfBirth"/>
            <p:inputText placeholder="Date-Of-Birth" value="#{userBean.user.dob}" required="true"/>
        </div>
    </div>

    <div class="ui-g-12 ui-md-4">
        <div class="ui-inputgroup">
            <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="Gender"/>
            <p:inputText placeholder="Gender" value="#{userBean.user.gender}" required="true"/>
        </div>
    </div>

</div>
<p:spacer/>
<h:panelGrid columns="1" cellpadding="5">
    <p:growl id="growl" life="2000" />
    <p:commandButton value="Ajax Submit" id="ajax" update="growl" action="#{userBean.registerUser}"
        style="margin-right:20px;" styleClass="ui-priority-primary" />
</h:panelGrid>
<p:spacer/>

</h:form>
</f:view>
</html>

```

Login.xhtml

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns:p="http://primefaces.org/ui">
<h:head>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"></link>
</h:head>
<f:view contentType="text/html">

    <h:form>
        <p:menubar>
            <p:menuItem style="font-weight: bolder" value="Assignment" url="index.xhtml"></p:menuItem>

            <f:facet name="options">
                <p:commandButton outcome="register.xhtml" type="button" value="Register" icon="pi pi-home"/>
            </f:facet>
        </p:menubar>
    </h:form>
</f:view>

```



```

        <p:commandButton outcome="login.xhtml" type="button" value="Login" icon="pi pi-home"/>
    </f:facet>

    <f:facet name="options">
        <p:inputText style="margin:1px 10px 1px 1px;" placeholder="Search"/>
    </f:facet>
</p:menubar>
<h3 style="padding: 25px">Login</h3>

<p:separator />

<div class="ui-g ui-fluid">
    <div class="ui-g-12 ui-md-4">
        <div class="ui-inputgroup">
            <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="Email"/>
            <p:inputText placeholder="Email" value="#{userBean.user.email}" required="true"/>
        </div>
    </div>
    <div class="ui-g-12 ui-md-4">
        <div class="ui-inputgroup">
            <p:outputLabel styleClass="p-col-fixed" style="width:100px" value="Password"/>
            <p:password id="inlineFeedback" placeholder="Password" value="#{userBean.user.password}"
required="true"/>
        </div>
    </div>

</div>

<p:spacer/>
<h:panelGrid columns="1" cellpadding="5">
    <p:growl id="growl" life="2000" />
    <p:commandButton value="Login" id="ajax" update="growl" action="#{userBean.loginUser}"
style="margin-right:20px;" styleClass="ui-priority-primary" />
</h:panelGrid>

<p:spacer/>

</h:form>
</f:view>
</html>

```

Welcome.xhtml

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns:p="http://primefaces.org/ui">
<h:head>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"></link>
</h:head>
<f:view contentType="text/html">

    <h:form>
        <p:menubar>
            <p:menuitem style="font-weight: bolder" url="index.xhtml" value="Assignment"></p:menuitem>
            <f:facet name="options">
                <p:button outcome="register.xhtml" value="Register" icon="pi pi-star"
                    style="margin-right:20px;"></p:button>
                <p:button outcome="login.xhtml" value="Login" icon="pi pi-star" style="margin-
right:20px;"></p:button>
            </f:facet>

            <f:facet name="options">
                <p:inputText style="margin:1px 10px 1px 1px;" placeholder="Search"/>
                <p:commandButton type="button" value="Logout" action="#{userBean.logoutUser}" icon="pi pi-
home"/>
            </f:facet>
        </p:menubar>

        <table class="table table-bordered table-danger">
            <thead>
                <tr>
                    <th scope="col">Username</th>
                    <th scope="col">Firstname</th>
                    <th scope="col">Lastname</th>

```

```

        <th scope="col">Email</th>
        <th scope="col">DOB</th>
        <th scope="col">Gender</th>

    </tr>
</thead>
<tbody>

    <ui:repeat value="#{userBean.users}" var="user">

        <tr class="table table-bordered table-dark">

            <td>#{user.username}</td>
            <td>#{user.firstname}</td>
            <td>#{user.lastname}</td>
            <td>#{user.email}</td>

            <td>#{user.dob}</td>
            <td>#{user.gender}</td>

        </tr>

    </ui:repeat>

</tbody>
</table>
</h:form>
</f:view>
</html>

```

User.java

```

package backingbean;

public class User {
    private int id;
    private String username;
    private String firstname;
    private String lastname;
    private String email;
    private String password;
    private String confirmpassword;
    private String dob;
    private String gender;

    public User() {
    }

    public int getId() {
        return id;
    }

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

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public String getFirstname() {
        return firstname;
    }

    public void setFirstname(String firstname) {
        this.firstname = firstname;
    }

    public String getLastname() {
        return lastname;
    }
}

```

```

    }

    public void setLastname(String lastname) {
        this.lastname = lastname;
    }

    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 String getConfirmpassword() {
        return confirmpassword;
    }

    public void setConfirmpassword(String confirmpassword) {
        this.confirmpassword = confirmpassword;
    }

    public String getDob() {
        return dob;
    }

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

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }
}

```

UserBean.java

```

package managedbean;

import backingbean.User;
import db.DbConnection;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.context.FacesContext;
import javax.servlet.http.HttpSession;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

@ManagedBean
public class UserBean {
    private User user;
    private DbConnection dbConnection;

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public UserBean() {

```

```

        user = new User();
        dbConnection = new DbConnection();
    }

    public void registerUser() {

        addMessage("Data Submitted To Database!!");

        System.out.println(user.getEmail());
        String username = user.getUsername();
        String firstname = user.getFirstname();
        String lastname = user.getLastname();
        String email = user.getEmail();
        String password = user.getPassword();
        String confirmpassword = user.getConfirmpassword();
        String dob = user.getDob();
        String gender = user.getGender();
        dbConnection.insertRecord(username, firstname, lastname, email, password, confirmpassword, dob,
gender);
    }

    public String loginUser() {
        dbConnection.loginUser(user.getEmail(), user.getPassword());
        HttpSession session = (HttpSession)
FacesContext.getCurrentInstance().getExternalContext().getSession(false);
        session.setAttribute("email", user.getEmail());
        session.setMaxInactiveInterval(15 * 60);
        addMessage("Login Sucessfull!!");
        return "welcome.xhtml?faces-redirect=true";
    }
    public String logoutUser(){
        HttpSession session = (HttpSession)
FacesContext.getCurrentInstance().getExternalContext().getSession(false);
        session.invalidate();
        return "login.xhtml?faces-redirect=true";
    }

    public void addMessage(String summary) {
        FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO, summary, null);
        FacesContext.getCurrentInstance().addMessage(null, message);
    }

    public List<User> getUsers() {

        List<User> abc = new ArrayList<User>();

        ResultSet rs = dbConnection.getRecords();

        try {

            while (rs.next()) {

                User user = new User();
                user.setUsername(rs.getString("username"));
                user.setFirstname(rs.getString("firstname"));
                user.setLastname(rs.getString("lastname"));
                user.setEmail(rs.getString("email"));
                user.setDob(rs.getString("dob"));
                user.setGender(rs.getString("gender"));

                abc.add(user);

            }

        } catch (Exception e) {
        }

        return abc;
    }

}

```

DbConnection.java

```

package db;

import java.sql.*;

/**
 * Created by fawad.tariq on 9/26/2019.
 */
public class DbConnection {
    private String dbURL = "jdbc:mysql://localhost:3306/primeface";
    private String username = "root";
    private String password = "";
    private Connection connection;

    public DbConnection() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection(dbURL, username, password);
            if (connection != null) {
                System.out.println("Success");
            }
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void insertRecord(String username, String firstname, String lastname, String email, String password,
String confirmpassword, String dob, String gender) {
        try {
            String sqlQuery = "INSERT INTO
user111(username,firstname,lastname,email,password,confirmpassword,dob,gender) VALUES(?,?,?,?,?,?,?,?)";
            PreparedStatement preparedStatement = connection.prepareStatement(sqlQuery);
            preparedStatement.setString(1, username);
            preparedStatement.setString(2, firstname);
            preparedStatement.setString(3, lastname);
            preparedStatement.setString(4, email);
            preparedStatement.setString(5, password);
            preparedStatement.setString(6, confirmpassword);
            preparedStatement.setString(7, dob);
            preparedStatement.setString(8, gender);

            int noOfRowsInserted = preparedStatement.executeUpdate();
            if (noOfRowsInserted > 0) {
                System.out.println(noOfRowsInserted + " rows inserted!");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public ResultSet loginUser(String email, String password) {
        try {
            String sqlQuery = "SELECT * from user111 where email=? and password=?";
            PreparedStatement preparedStatement = connection.prepareStatement(sqlQuery);
            preparedStatement.setString(1, email);
            preparedStatement.setString(2, password);

            ResultSet result= preparedStatement.executeQuery();

            if (result != null) {
                return result;
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null;
    }

    public ResultSet getRecords(){
        try {
            String sqlQuery = "SELECT * FROM user111";
            Statement statement = connection.createStatement();
            ResultSet result = statement.executeQuery(sqlQuery);
            return result;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null;
    }
}

```

