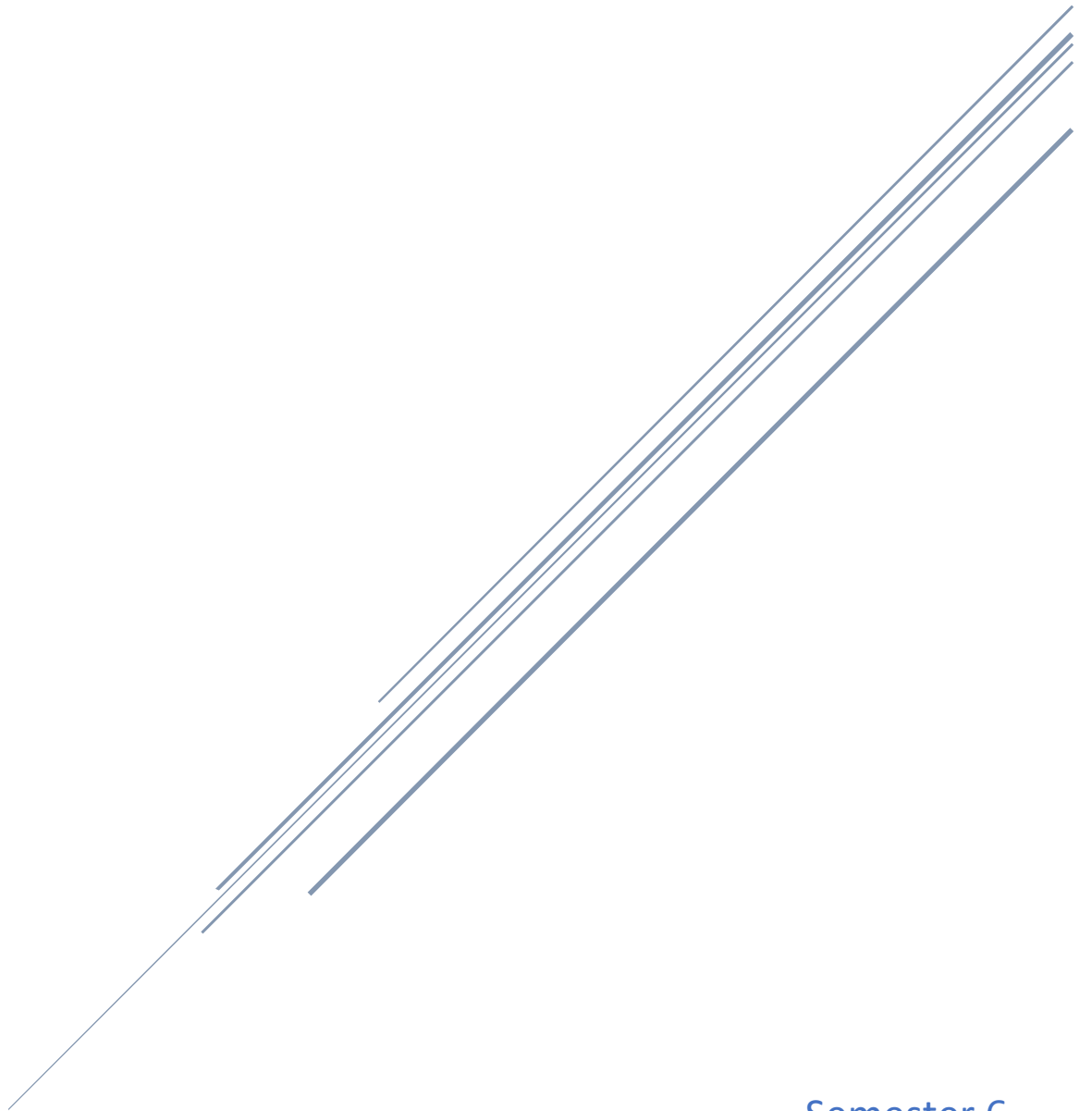


MATRIMONY WEB APP

eProject Report



Semester 6
eProject

This is to certify that:

Mr. SIHLELO DALUBUHLE DLAMINI (Student1354063)

Ms. MASELINA THANDEKA MATSE (Student1354076)

Mr. BANDISA LEORNARD DLAMINI (Student1354067)

Mr. LINDELANI NOTARRY NDLANGAMANDLA (Student1354065)

Successfully designed and developed:

Matrimony Web App

Start date:

01 October 2024

End date:

29 October 2024

Problem Statement

Problem definition:

The wedding event is no more a home manageable affair where one requires the different service providers to be contacted and negotiated. Furthermore, the precious time and resources consumed during the whole process of finding and searching the ideal match are inefficient and lacks a centralized platform to connect potential partners, leading to frustrating and limited options for individuals seeking suitable life partners.

Customer Requirement Specification (CRS)

Client:

Business/Project Objectives:

- Facilitate Matchmaking - to create a platform that efficiently connects eligible singles based on their preferences and compatibility
- Generate Revenue - to generate sustainable revenue through subscription fees.
- Enhance User Experience - to provide a seamless and enjoyable user experience, ensuring positive interaction with platform.
- Adapt to changing trends - to stay updated with evolving trends in matchmaking industry and incorporate relevant features.
- Simplify the matchmaking process - providing a user-friendly online platform

Inputs Provided by Client:

- Input to system
- Output from system
- Processes in system
- Expected delivery date
- List of deliverables

System Requirements:

| No. | Items | Discription |
|-----|-----------------------|--|
| 1 | Operating System (OS) | Windows |
| 2 | Software | Glassfish, Apache NetBeans IDE, JDK 17 |
| 3 | Database | MYSQL 8.X |

Scope of Work:

1. Requirement gathering and analysis
2. Designing of user interface (UI)
3. Backend development (EJB and Servlets)
4. Frontend development (HTML, CSS, JS and JSP)
5. Testing and quality assurance
6. Performance Optimization

7. Submission

Architecture and design of project

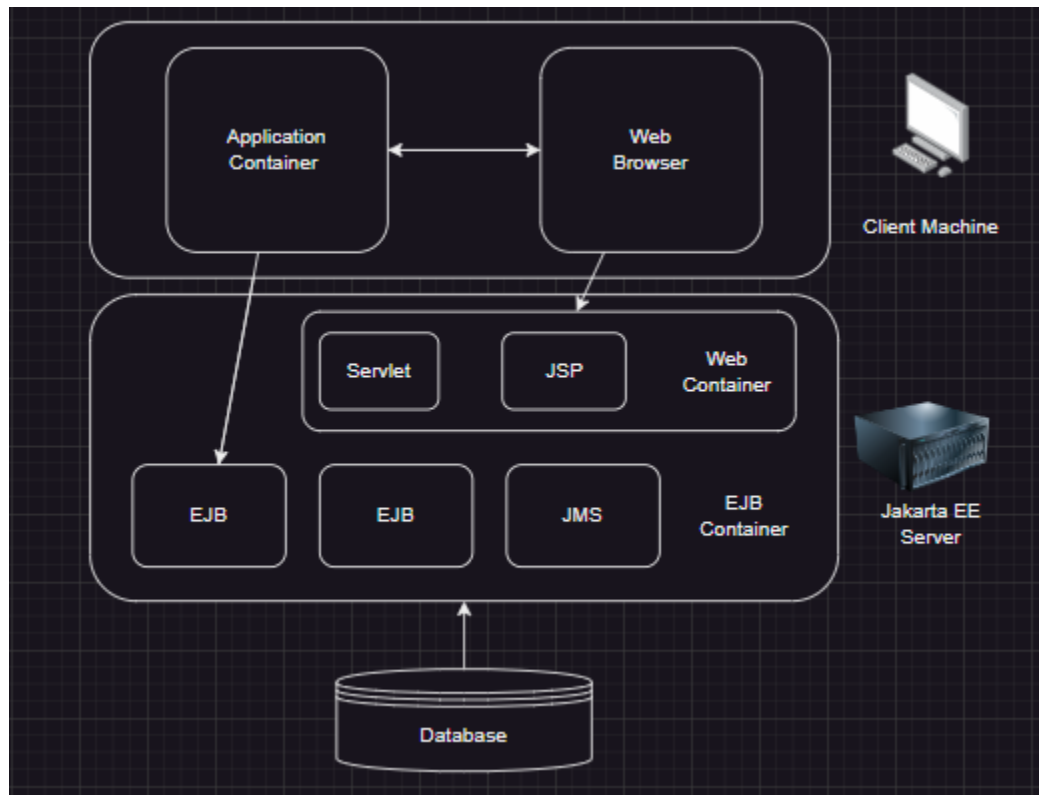


Fig 1.1: Application Architecture

Data flow diagram

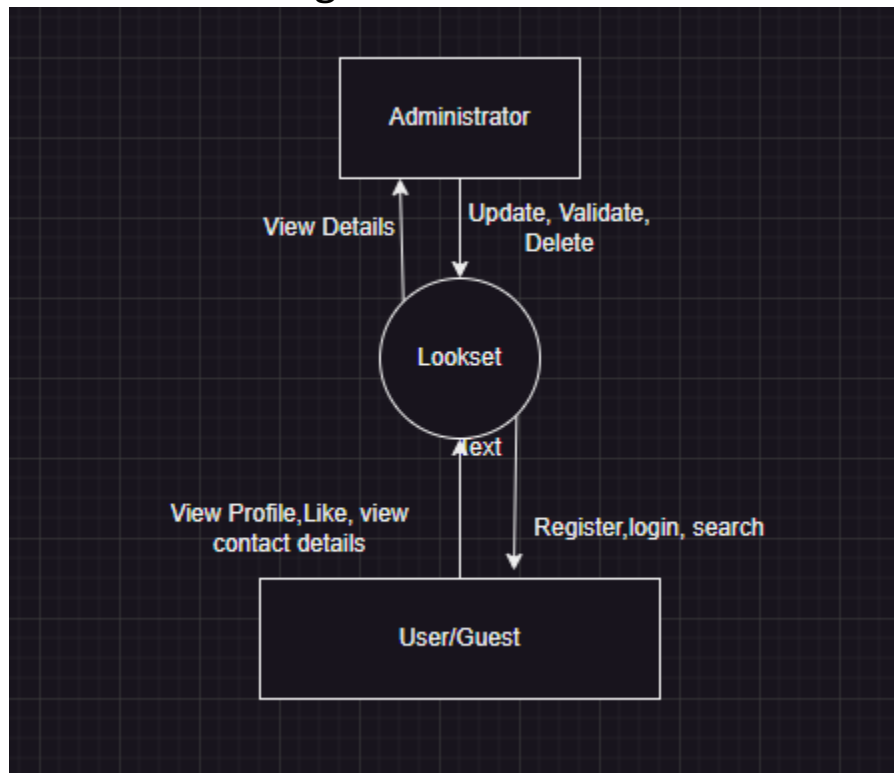


Fig 1.2: Context Level Diagram

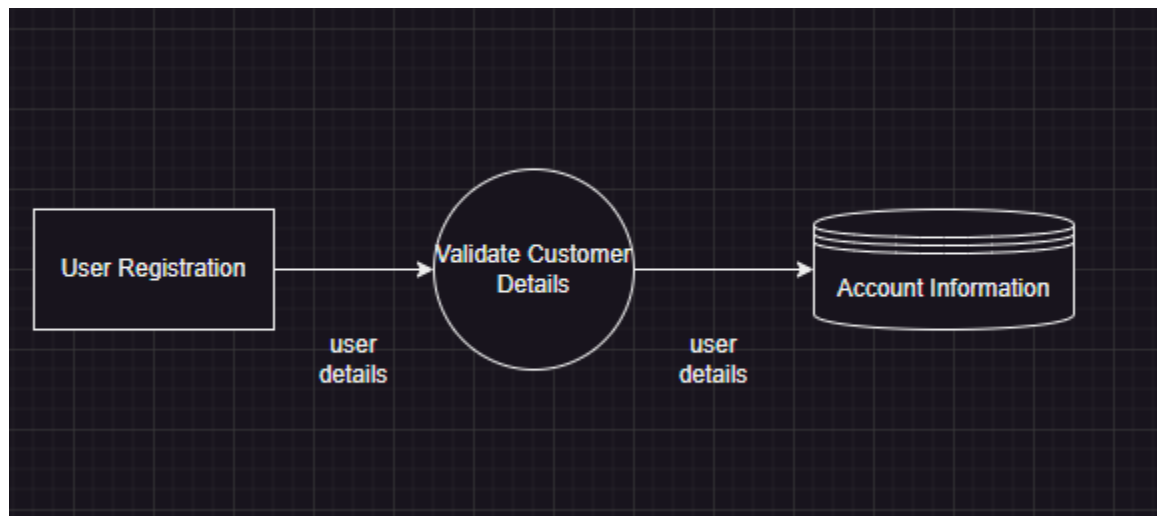


Fig 1.3: Registration

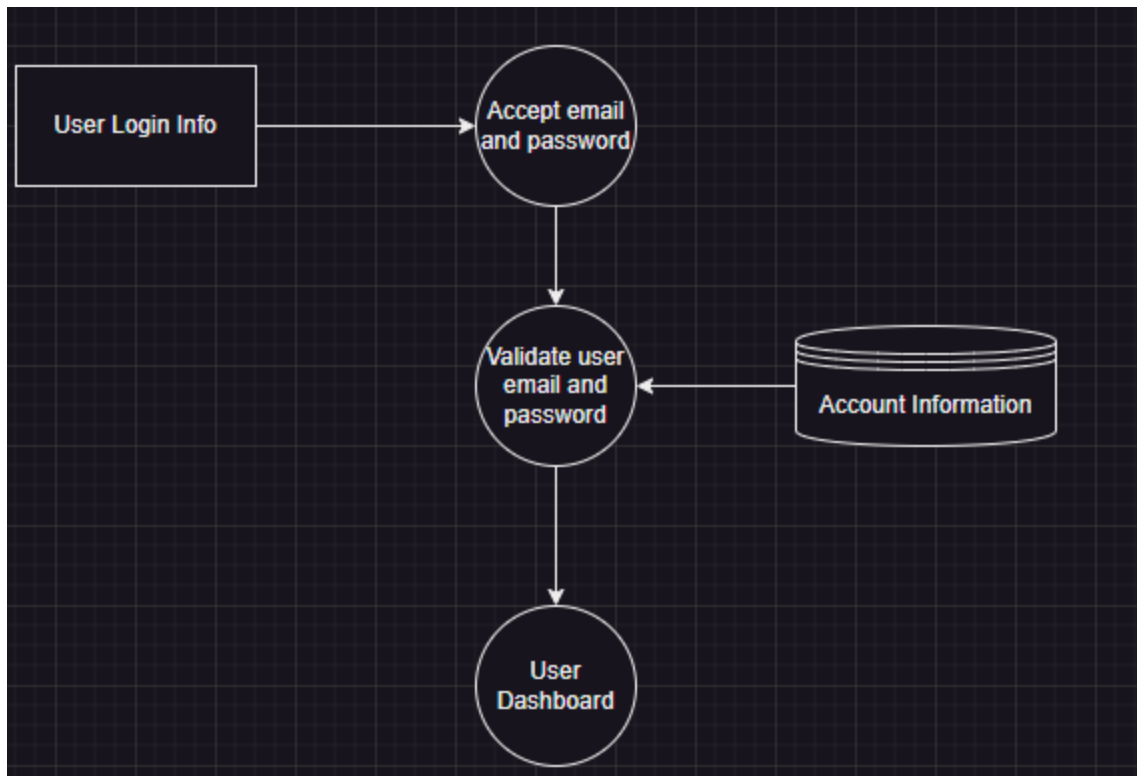


Fig 1.4: Login

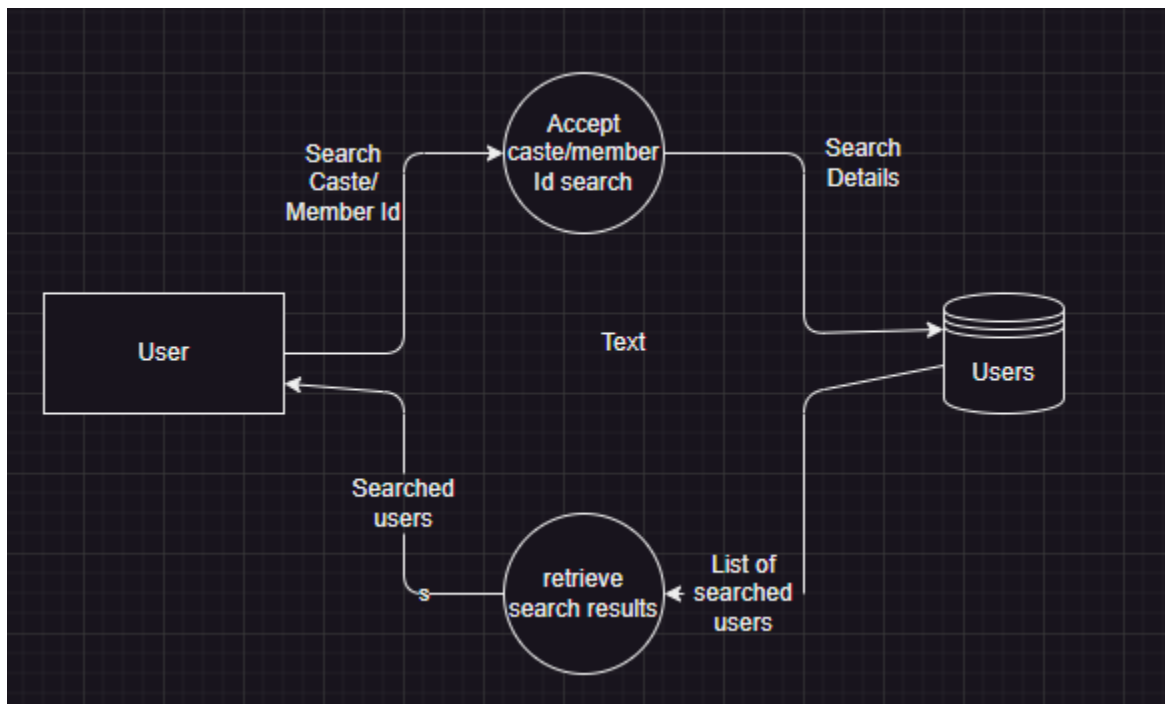


Fig 1.5: Search

Flow Chart

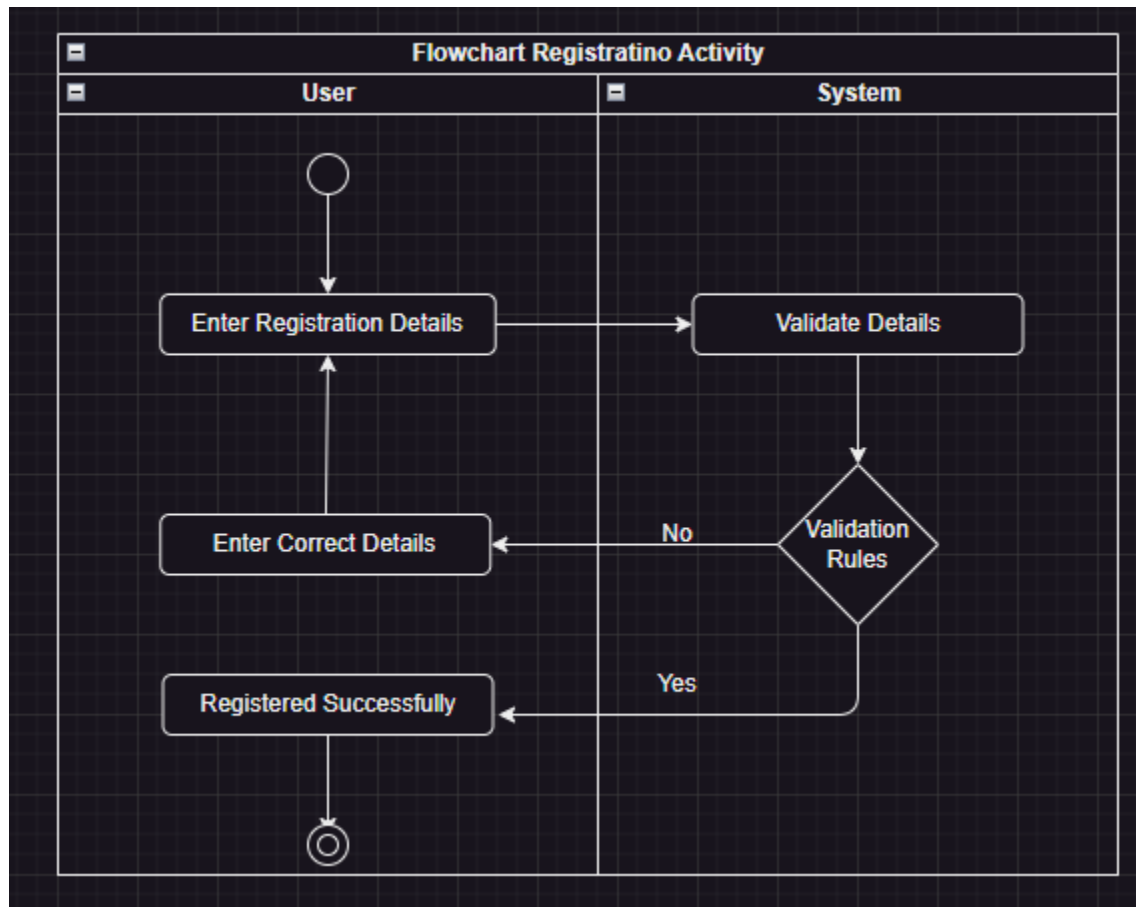


Fig 1.6: Register Flow Chart

Activity Diagram

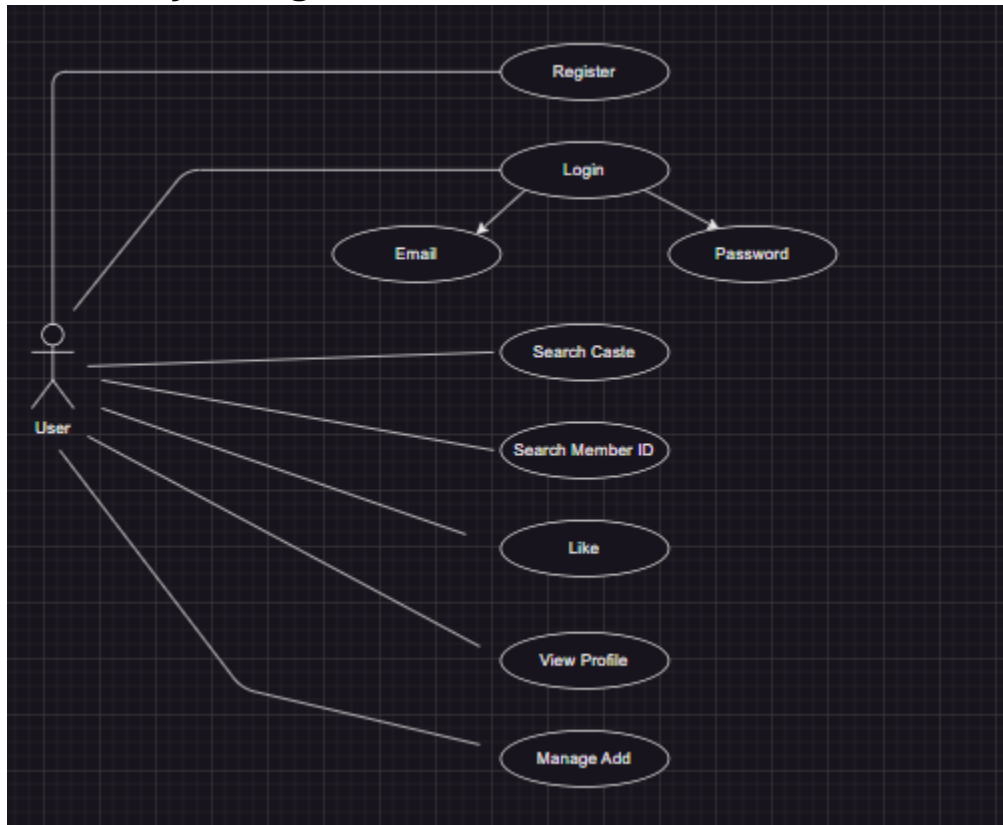


Fig 1.7: User Activities

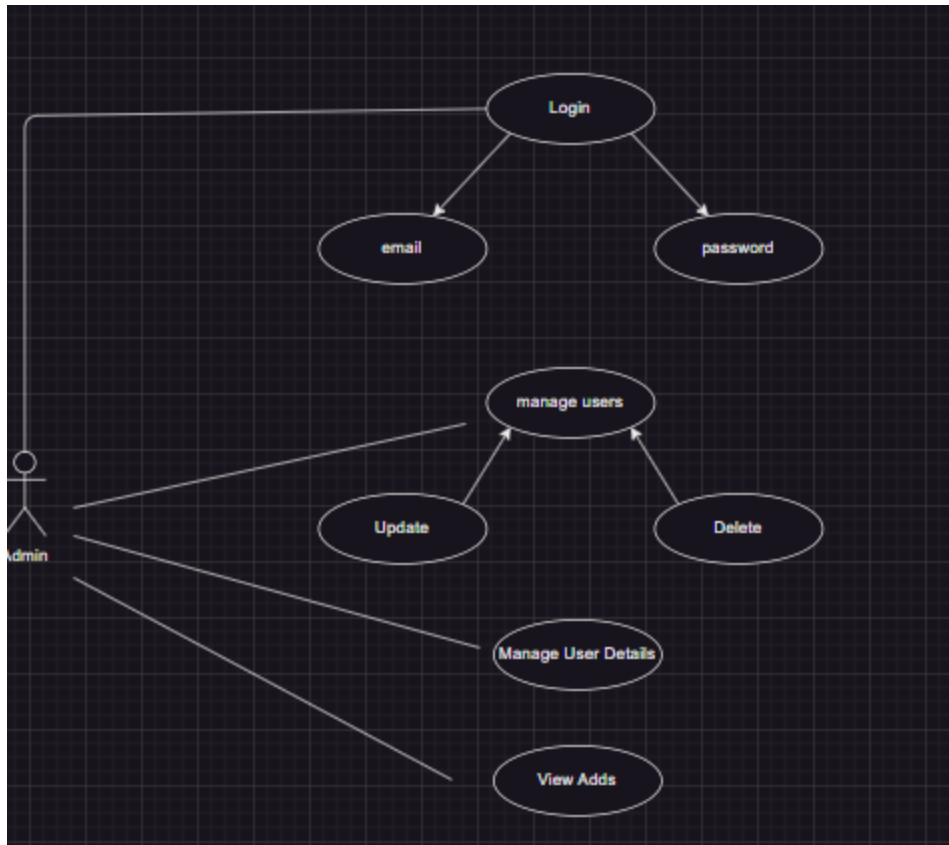


Fig 1.8: Admin Activities

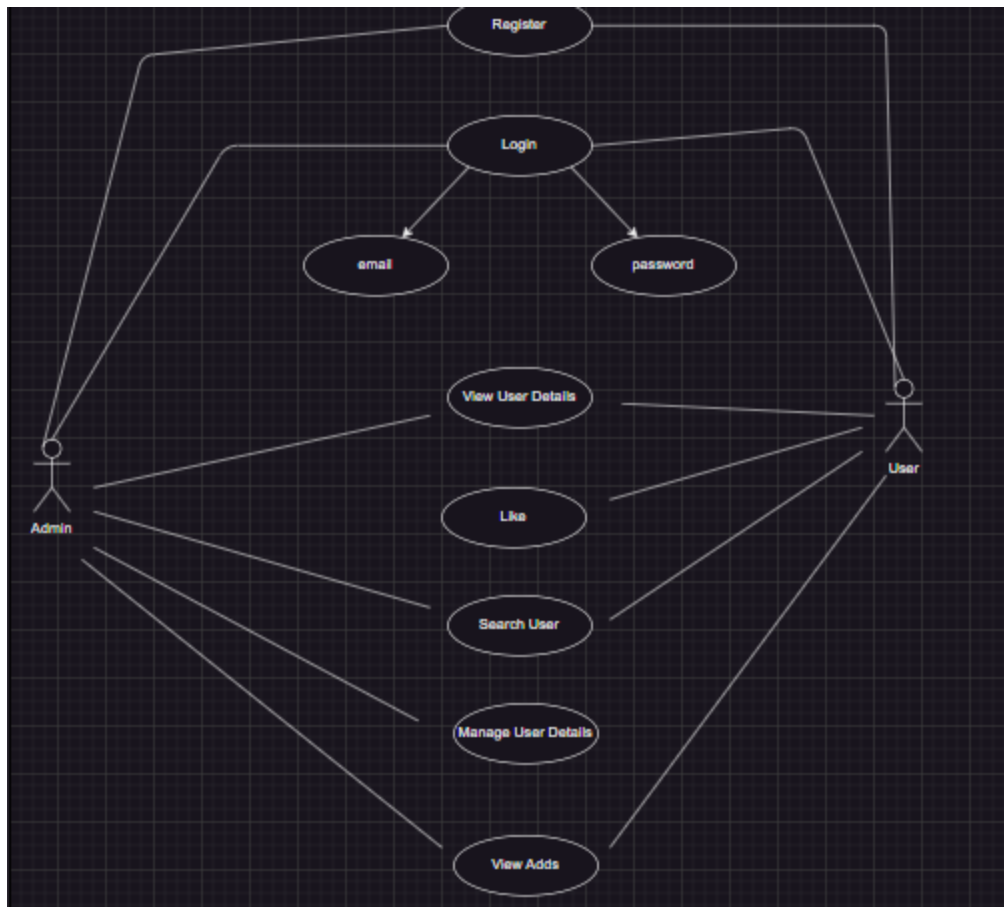


Fig 1.9: Activities in application

Entity Relationship (ER) Diagram

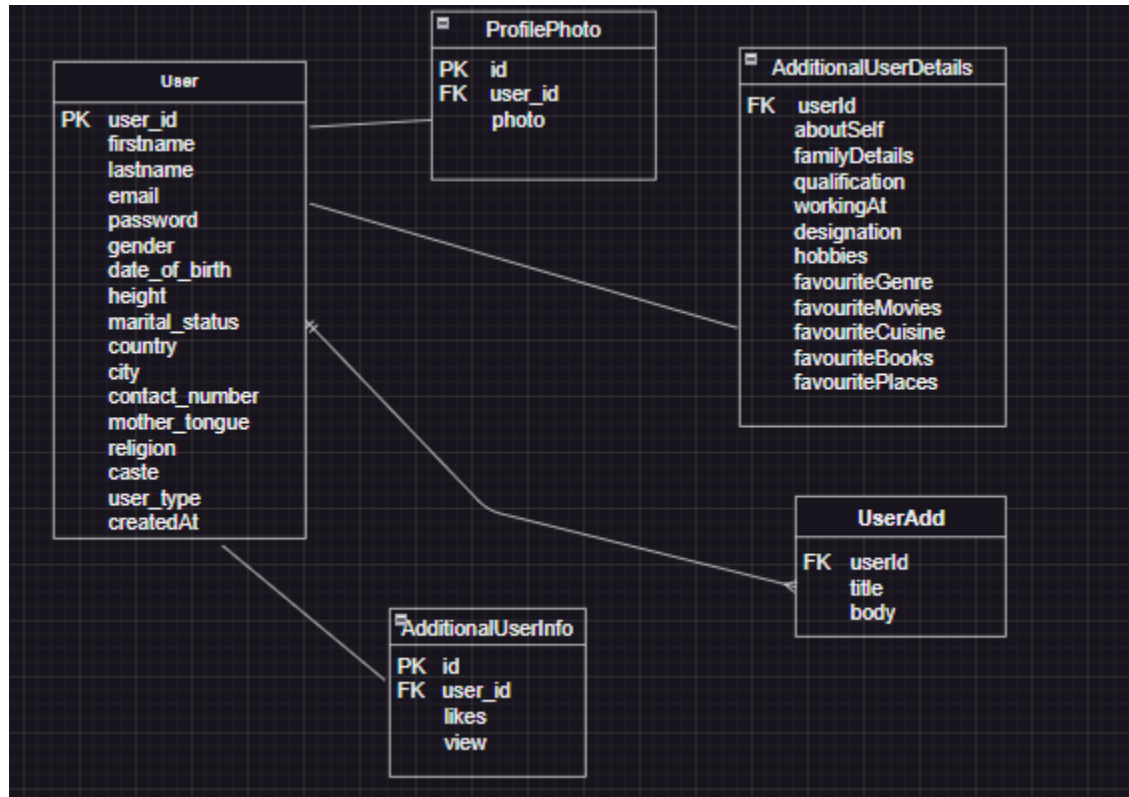


Fig 1.10: ER Diagram for Application

Database Design/ Structure

| Table: users | | | |
|----------------|---|-------------|--|
| Field name | Datatype | Key | Description |
| user_id | Int(11) | Primary Key | It stores the unique ID for user |
| firstname | Varchar(50) | Not Null | It stores the first name for user |
| lastname | Varchar(50) | Not Null | It stores the last name for user |
| email | Varchar(100) | Unique | It stores the email for user |
| password | Varchar(255) | Not Null | It stores the password for user |
| gender | enum("Male", "Femal", "Other") | Not Null | It stores the gender for user |
| date_of_birth | Date | Not Null | It stores the date of birth for user |
| height | Int(11) | Null | It stores the height for user |
| marital_status | enum("Married", "Never Married", "Divorced", "Widowed") | Not Null | It stores the marital status of user |
| country | Varchar(50) | Not Null | It stores the country name for user |
| city | Varchar(50) | Not Null | It stores the city name for user |
| contact_number | Varchar(15) | Not Null | It stores the contact number for user |
| mother_tongue | Varchar(50) | Not Null | It stores the mother tongue for user |
| religion | Varchar(50) | Not Null | It stores the religion for user |
| caste | Varchar(50) | Not Null | It stores the caste for user |
| user_type | enum("Admin", "User") | Not Null | It stores the user type of user |
| created_at | timestamp | Not Null | It stores the time when user created account |

| Table: additionaluserdetails | | | |
|------------------------------|--------------|-------------|---|
| Field name | Datatype | Key | Description |
| user_id | Int(11) | Foreign Key | It stores the unique ID for user |
| about_self | Text | Null | It stores the paragraph about user for user |
| family_details | Text | Null | It stores family details for user |
| qualifications | Varchar(255) | Null | It stores the qualifications for user |
| designation | Varchar(255) | Null | It stores the designation for user |
| hobbies | Text | Null | It stores the hobbies for user |
| favourite_music | Text | Null | It stores the favourite music of user |
| favourite_movies | Text | Null | It stores the favourite movies for user |
| favourite_cuisines | Text | Null | It stores the favourite cuisines for user |
| favourite_books | Text | Null | It stores the favourite books for user |
| favourite_places | Text | Not Null | It stores the mother tongue for user |

| Table: additionaluserinfo | | | |
|---------------------------|----------|-------------|--|
| Field name | Datatype | Key | Description |
| id | Int(11) | Primary Key | It stores the unique ID for info |
| user_id | Int(11) | Foreign Key | It stores the Unique ID user for user |
| likes | Int(11) | Null | It stores number of likes for user |
| views | Int(11) | Null | It stores the number of views for user |
| liked_user | Text | Null | It stores the users that likes as an array |

| Table: photos | | | |
|---------------|----------|-------------|---------------------------------------|
| Field name | Datatype | Key | Description |
| id | Int(11) | Primary Key | It stores the unique ID for info |
| user_id | Int(11) | Foreign Key | It stores the Unique ID user for user |
| photo | LongBlob | Null | It stores byte data for user photo |

| Table: photos | | | |
|---------------|-------------|-------------|--|
| Field name | Datatype | Key | Description |
| user_id | Int(11) | Foreign Key | It stores the Unique ID user for user |
| payment_type | Varchar(50) | Not Null | It stores the payment type user |
| amount | Varchar(50) | Not Null | It stores the amount paid by user |
| date_paid | timestamp | Not Null | It stores the last date when user paid |

User Guide

1. System Requirements:

| No. | Items | Discription |
|-----|-----------------------|--|
| 1 | Operating System (OS) | Windows |
| 2 | Software | Glassfish, Apache NetBeans IDE, JDK 17 |
| 3 | Database | MYSQL 8.X |

2. Deploy and run application

- a. Install required Apache NetBeans 22 and XAMMP.
- b. In XAMMP import database found in documentation file.
- c. Download glassfish server.
- d. Deploy app in NetBeans using glassfish server created.
- e. Enjoy and test the application.

Source Code

Models EJB

User

```
package com.matrimony.models;

import java.sql.Date;
import java.time.LocalDateTime;

public class User {
    private int user_id;
    private String firstname;
    private String lastname;
    private String email;
    private String password;
    private String gender;
    private Date date_of_birth;
    private int height;
    private String marital_status;
    private String country;
    private String city;
    private String contact_number;
    private String mother_tongue;
    private String religion;
    private String caste;
    private String user_type;
    private LocalDateTime createdAt = LocalDateTime.now();

    // New fields for additional user info
```

```

private int likes;
private int views;
private byte[] photo;

//new fields for additional user details
private String aboutSelf;
private String familyDetails;
private String qualification;
private String workingAt;
private String designation;
private String hobbies;
private String favouriteGenre;
private String favouriteMovies;
private String favouriteCuisine;
private String favouriteBooks;
private String favouritePlaces;

//    main class constructor

public User() {
}

//    constructor with all fields

public User(int user_id, String firstname, String lastname, String email,
String password, String gender, Date date_of_birth, int height, String
marital_status, String country, String city, String contact_number, String
mother_tongue, String religion, String caste, String user_type) {
    this.user_id = user_id;
    this.firstname = firstname;
    this.lastname = lastname;

```

```
this.email = email;
this.password = password;
this.gender = gender;
this.date_of_birth = date_of_birth;
this.height = height;
this.marital_status = marital_status;
this.country = country;
this.city = city;
this.contact_number = contact_number;
this.mother_tongue = mother_tongue;
this.religion = religion;
this.caste = caste;
this.user_type = user_type;
}
```

// getters and setter

```
public int getLikes() {
    return likes;
}
```

```
public void setLikes(int likes) {
    this.likes = likes;
}
```

```
public int getViews() {
    return views;
}
```

```
public void setViews(int views) {
    this.views = views;
}

public byte[] getPhoto() {
    return photo;
}

public void setPhoto(byte[] photo) {
    this.photo = photo;
}

// getters and setters for additional details
public String getAboutSelf() {
    return aboutSelf;
}

public void setAboutSelf(String aboutSelf) {
    this.aboutSelf = aboutSelf;
}

public String getFamilyDetails() {
    return familyDetails;
}

public void setFamilyDetails(String familyDetails) {
    this.familyDetails = familyDetails;
}

public String getQualification() {
```

```
        return qualification;
    }

    public void setQualification(String qualification) {
        this.qualification = qualification;
    }

    public String getWorkingAt() {
        return workingAt;
    }

    public void setWorkingAt(String workingAt) {
        this.workingAt = workingAt;
    }

    public String getDesignation() {
        return designation;
    }

    public void setDesignation(String designation) {
        this.designation = designation;
    }

    public String getHobbies() {
        return hobbies;
    }

    public void setHobbies(String hobbies) {
        this.hobbies = hobbies;
    }
}
```

```
public String getFavouriteGenre() {  
    return favouriteGenre;  
}  
  
public void setFavouriteGenre(String favouriteGenre) {  
    this.favouriteGenre = favouriteGenre;  
}  
  
public String getFavouriteMovies() {  
    return favouriteMovies;  
}  
  
public void setFavouriteMovies(String favouriteMovies) {  
    this.favouriteMovies = favouriteMovies;  
}  
  
public String getFavouriteCuisine() {  
    return favouriteCuisine;  
}  
  
public void setFavouriteCuisine(String favouriteCuisine) {  
    this.favouriteCuisine = favouriteCuisine;  
}  
  
public String getFavouriteBooks() {  
    return favouriteBooks;  
}  
  
public void setFavouriteBooks(String favouriteBooks) {
```

```
        this.favouriteBooks = favouriteBooks;
    }

    public String getFavouritePlaces() {
        return favouritePlaces;
    }

    public void setFavouritePlaces(String favouritePlaces) {
        this.favouritePlaces = favouritePlaces;
    }
}
```

```
public int getUser_id() {
    return user_id;
}
```

```
public void setUser_id(int user_id) {
    this.user_id = user_id;
}
```

```
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 getGender() {
    return gender;
}
```

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

public Date getDate_of_birth() {
    return date_of_birth;
}

public void setDate_of_birth(Date date_of_birth) {
    this.date_of_birth = date_of_birth;
}

public int getHeight() {
    return height;
}

public void setHeight(int height) {
    this.height = height;
}

public String getMarital_status() {
    return marital_status;
}

public void setMarital_status(String marital_status) {
    this.marital_status = marital_status;
}

public String getCountry() {
    return country;
}
```

```
}

public void setCountry(String country) {
    this.country = country;
}

public String getCity() {
    return city;
}

public void setCity(String city) {
    this.city = city;
}

public String getContact_number() {
    return contact_number;
}

public void setContact_number(String contact_number) {
    this.contact_number = contact_number;
}

public String getMother_tongue() {
    return mother_tongue;
}

public void setMother_tongue(String mother_tongue) {
    this.mother_tongue = mother_tongue;
}
```

```
public String getReligion() {  
    return religion;  
}  
  
public void setReligion(String religion) {  
    this.religion = religion;  
}  
  
public String getCaste() {  
    return caste;  
}  
  
public void setCaste(String caste) {  
    this.caste = caste;  
}  
  
public LocalDateTime getCreatedAt() {  
    return createdAt;  
}  
  
public void setCreatedAt(LocalDateTime createdAt) {  
    this.createdAt = createdAt;  
}  
  
public String getUser_type() {  
    return user_type;  
}  
  
public void setUser_type(String user_type) {  
    this.user_type = user_type;  
}
```

```
}
```

```
}
```

AdditionalUserDetails class

```
package com.matrimony.models;
```

```
/**
```

```
*
```

```
* @author dsihle875@gmail.com
```

```
*/
```

```
public class AdditionalUserDetails {
```

```
    private int userId;
```

```
    private String aboutSelf;
```

```
    private String familyDetails;
```

```
    private String qualification;
```

```
    private String workingAt;
```

```
    private String designation;
```

```
    private String hobbies;
```

```
    private String favouriteGenre;
```

```
    private String favouriteMovies;
```

```
    private String favouriteCuisine;
```

```
    private String favouriteBooks;
```

```
    private String favouritePlaces;
```

```
    public AdditionalUserDetails(int userId, String aboutSelf, String  
familyDetails, String qualification, String workingAt, String designation,  
String hobbies, String favouriteGenre, String favouriteMovies, String  
favouriteCuisine, String favouriteBooks, String favouritePlaces) {
```

```
        this.userId = userId;
```

```
    this.aboutSelf = aboutSelf;
    this.familyDetails = familyDetails;
    this.qualification = qualification;
    this.workingAt = workingAt;
    this.designation = designation;
    this.hobbies = hobbies;
    this.favouriteGenre = favouriteGenre;
    this.favouriteMovies = favouriteMovies;
    this.favouriteCuisine = favouriteCuisine;
    this.favouriteBooks = favouriteBooks;
    this.favouritePlaces = favouritePlaces;
}
```

```
public AdditionalUserDetails() {
}
```

```
public int getUserId() {
    return userId;
}
```

```
public void setUserId(int userId) {
    this.userId = userId;
}
```

```
public String getAboutSelf() {
    return aboutSelf;
}
```

```
public void setAboutSelf(String aboutSelf) {
    this.aboutSelf = aboutSelf;
}

public String getFamilyDetails() {
    return familyDetails;
}

public void setFamilyDetails(String familyDetails) {
    this.familyDetails = familyDetails;
}

public String getQualification() {
    return qualification;
}

public void setQualification(String qualification) {
    this.qualification = qualification;
}

public String getWorkingAt() {
    return workingAt;
}

public void setWorkingAt(String workingAt) {
    this.workingAt = workingAt;
}

public String getDesignation() {
    return designation;
}
```

```
}

public void setDesignation(String designation) {
    this.designation = designation;
}

public String getHobbies() {
    return hobbies;
}

public void setHobbies(String hobbies) {
    this.hobbies = hobbies;
}

public String getFavouriteGenre() {
    return favouriteGenre;
}

public void setFavouriteGenre(String favouriteGenre) {
    this.favouriteGenre = favouriteGenre;
}

public String getFavouriteMovies() {
    return favouriteMovies;
}

public void setFavouriteMovies(String favouriteMovies) {
    this.favouriteMovies = favouriteMovies;
}
```



```

    public String getFavouriteCuisine() {
        return favouriteCuisine;
    }

    public void setFavouriteCuisine(String favouriteCuisine) {
        this.favouriteCuisine = favouriteCuisine;
    }

    public String getFavouriteBooks() {
        return favouriteBooks;
    }

    public void setFavouriteBooks(String favouriteBooks) {
        this.favouriteBooks = favouriteBooks;
    }

    public String getFavouritePlaces() {
        return favouritePlaces;
    }

    public void setFavouritePlaces(String favouritePlaces) {
        this.favouritePlaces = favouritePlaces;
    }

}

```

AdditionalUserInfo Class

```
package com.matrimony.models;
```

```
/**
```

```

*
* @author dsihle875@gmail.com
*/
public class AdditionalUserInfo {
    private int id;
    private int user_id;
    private int likes;
    private int view;

    // empty constructor
    public AdditionalUserInfo() {
    }

    public AdditionalUserInfo(int id, int user_id, int likes, int view) {
        this.id = id;
        this.user_id = user_id;
        this.likes = likes;
        this.view = view;
    }

    public int getId() {
        return id;
    }

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

    public int getUser_id() {
        return user_id;
    }

```

```

    }

    public void setUser_id(int user_id) {
        this.user_id = user_id;
    }

    public int getLikes() {
        return likes;
    }

    public void setLikes(int likes) {
        this.likes = likes;
    }

    public int getView() {
        return view;
    }

    public void setView(int view) {
        this.view = view;
    }

}

package com.matrimony.models;

/**
 *
 * @author dsihle875@gmail.com

```

```

*/
public class ProfilePhoto {
    private int id;
    private int user_id;
    private byte[] photo;

    //    empty constructor
    public ProfilePhoto() {
    }

    public ProfilePhoto(int id, int user_id, byte[] photo) {
        this.id = id;
        this.user_id = user_id;
        this.photo = photo;
    }

    //    getters and setters
    public int getId() {
        return id;
    }

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

    public int getUser_id() {
        return user_id;
    }

    public void setUser_id(int user_id) {

```

```

        this.user_id = user_id;
    }

    public byte[] getPhoto() {
        return photo;
    }

    public void setPhoto(byte[] photo) {
        this.photo = photo;
    }
}

```

UserAdd Class

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
 this template
 */
package com.matrimony.models;

/**
 *
 * @author dsihl
 */
public class UserAdd {

    private int userId;

```

```
private String title;
private String body;

//    profile image
private byte[] photo;

//    get name
private String name;

public String getName() {
    return name;
}

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

public byte[] getPhoto() {
    return photo;
}

public void setPhoto(byte[] photo) {
    this.photo = photo;
}

public int getUserId() {
    return userId;
}

public void setUserId(int userId) {
```

```

        this.userId = userId;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getBody() {
        return body;
    }

    public void setBody(String body) {
        this.body = body;
    }
}

```

EJB Controller

```

package com.matrimony.stateful;

import com.matrimony.models.AdditionalUserDetails;
import com.matrimony.models.AdditionalUserInfo;
import com.matrimony.models.ProfilePhoto;
import com.matrimony.models.User;

```

```

import com.matrimony.models.UserAdd;
import com.matrimony.util.DBConnection;
import jakarta.ejb.Stateful;
import jakarta.ejb.TransactionAttribute;
import jakarta.ejb.TransactionAttributeType;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Timestamp;
import java.time.Duration;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.List;
import org.mindrot.jbcrypt.BCrypt;

/**
 *
 * @author dsihle875@gmail.com
 */
//stateful class for implementations of local session bean
@Stateful
public class MatrimonySessionBean implements MatrimonySessionBeanLocal {

    @Override
    public int insertBasicUserInformation(String firstname, String lastname,
String email, String password, String gender, String dob, int height, String

```



```

maritalStatus, String country, String city, String number, String language,
String religion, String caste, String userType, String paymentMethod, String
amount) throws SQLException {
    if (getUserByEmail(email) != null) {
        throw new SQLException("User Already Exsits In Database");
    }
    String hashed_password = hashPassword(password);
    LocalDateTime createdAt = LocalDateTime.now();
    LocalDate dateOfBirth = LocalDate.parse(dob,
DateTimeFormatter.ofPattern("yyyy-MM-dd"));
    String query = "INSERT INTO users (firstname, lastname, email,
password, gender, date_of_birth, height,marital_status, country, city,
contact_number, mother_tongue, religion, caste, user_type, createdAt) VALUES
(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
    try (Connection conn = DBConnection.getConnection();
PreparedStatement stmt = conn.prepareStatement(query,
Statement.RETURN_GENERATED_KEYS)) {
        stmt.setString(1, firstname);
        stmt.setString(2, lastname);
        stmt.setString(3, email);
        stmt.setString(4, hashed_password);
        stmt.setString(5, gender);
        stmt.setDate(6, java.sql.Date.valueOf(dateOfBirth));
        stmt.setInt(7, height);
        stmt.setString(8, maritalStatus);
        stmt.setString(9, country);
        stmt.setString(10, city);
        stmt.setString(11, number);
        stmt.setString(12, language);
        stmt.setString(13, religion);
        stmt.setString(14, caste);
        stmt.setString(15, userType);

```

```

        stmt.setTimestamp(16, java.sql.Timestamp.valueOf(createdAt));
        int affectedRow = stmt.executeUpdate();
        if (affectedRow == 0) {
            throw new SQLException("Failed to insert record");
        }
//        retrieving generated keys
        try (ResultSet generatedKeys = stmt.getGeneratedKeys()) {
            if (generatedKeys.next()) {
                int userId = generatedKeys.getInt(1);
                insertAdditionalUserInfo(userId);
                insertPhotoInfo(userId);
                insertUserId(userId);
                pay(userId, paymentMethod, amount);
                return userId; // Return the generated user ID
            } else {
                throw new SQLException("Inserting user failed, no ID
obtained.");
            }
        }
    }
}

// Method to insert additional user info
private void insertAdditionalUserInfo(int userId) throws SQLException {
    String query = "INSERT INTO additionaluserinfo (user_id, likes, view)
VALUES (?, ?, ?)";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(query)) {
        stmt.setInt(1, userId);
        stmt.setInt(2, 0);
        stmt.setInt(3, 0);
    }
}

```

```

        stmt.executeUpdate();
    }
}

private void pay(int userId, String paymentMethod, String amount) throws
SQLException {
    String query = "INSERT INTO payment (user_id, payment_type, amount)
VALUES (?, ?, ?)";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(query);) {
        stmt.setInt(1, userId);
        stmt.setString(2, paymentMethod);
        stmt.setString(3, amount);
        stmt.executeUpdate();
    }
}

// Method to insert additional user info
private void insertPhotoInfo(int userId) throws SQLException {
    String query = "INSERT INTO photos (user_id, photo) VALUES (?, ?)";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(query)) {
        stmt.setInt(1, userId);
        stmt.setString(2, "");
        stmt.executeUpdate();
    }
}

public void insertUserId(int userId) throws SQLException {
    String sql = "INSERT INTO additionalUserDetails (user_id) VALUES
(?)";
}

```

```

        try (Connection conn = DBConnection.getConnection();
PreparedStatement statement = conn.prepareStatement(sql)) {
            statement.setInt(1, userId);
            // Execute insert for each user ID
            statement.executeUpdate();
        }
    }

    @TransactionalAttribute(TransactionalAttributeType.REQUIRED)
    @Override
    public String loginUser(String username, String password) throws
SQLException {
        String sql = "SELECT password, user_type FROM users WHERE email = ?";
        try (Connection conn = DBConnection.getConnection();
PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setString(1, username);
            try (ResultSet rs = stmt.executeQuery()) {
                if (rs.next()) {
                    String storedHashedPassword = rs.getString("password");
                    String usertype = rs.getString("user_type");
                    if (BCrypt.checkpw(password, storedHashedPassword)) {
                        //logger.log(Level.INFO, "User {0} logged in
successfully", username);
                        return usertype;
                    } else {
                        throw new SQLException("Invalid username or
password");
                    }
                } else {
                    throw new SQLException("Invalid Username or Password");
                }
            }
        }
    }

```

```

    }
}

@Override

public void updateUser(int id, String firstname, String lastname, String
email, String gender, String dob,

    String height, String maritalStatus, String country, String city,
String number, String language,

    String religion, String caste, String userType) throws
SQLException {

    String query = "UPDATE users SET firstname = ?, lastname = ?, email =
?, gender = ?, date_of_birth = ?, height = ?, marital_status = ?, country =
?, city = ?, contact_number = ?, mother_tongue = ?, religion = ?, caste = ?,
user_type = ? WHERE user_id = ?";

    try (Connection conn = DBConnection.getConnection();
PreparedStatement stmt = conn.prepareStatement(query)) {

        stmt.setString(1, firstname);
        stmt.setString(2, lastname);
        stmt.setString(3, email);
        stmt.setString(4, gender);
        stmt.setDate(5, java.sql.Date.valueOf(dob));
        stmt.setString(6, height);
        stmt.setString(7, maritalStatus);
        stmt.setString(8, country);
        stmt.setString(9, city);
        stmt.setString(10, number);
        stmt.setString(11, language);
        stmt.setString(12, religion);
        stmt.setString(13, caste);
        stmt.setString(14, userType);
        stmt.setInt(15, id);
    }
}

```

```

        int rowsUpdated = stmt.executeUpdate();
        if (rowsUpdated > 0) {
            System.out.println("User Updated Successfully");
        } else {
            throw new SQLException("User update failed. User not
found.");
        }
    }
}

@Override
public void updatePassword(int id, String oldPassword, String
newPassword) throws SQLException {
    String fetchPassword = "SELECT password FROM users WHERE user_id =
?";
    String updatePassword = "UPDATE users SET password = ? WHERE user_id
= ?";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement fetchStmt = conn.prepareStatement(fetchPassword)) {
        fetchStmt.setInt(1, id);
        ResultSet rs = fetchStmt.executeQuery();
        if (rs.next()) {
            String storedHash = rs.getString("password");
            if (!BCrypt.checkpw(oldPassword, storedHash)) {
                throw new SQLException("The old password is incorrect.");
            }
            String hashedNewPassword = hashPassword(newPassword);
            try (PreparedStatement updateStmt =
conn.prepareStatement(updatePassword)) {
                updateStmt.setString(1, hashedNewPassword);
                updateStmt.setInt(2, id);
                int rowsUpdated = updateStmt.executeUpdate();

```

```

        if (rowsUpdated > 0) {
            System.out.println("Password for user: " + id + " has
been updated successfully");
        } else {
            throw new SQLException("Password update failed. User
not found.");
        }
    }
} else {
    throw new SQLException("User not found.");
}
}
}
}

```

```

@Override
public List<User> searchByCaste(String caste) throws SQLException {
    String sql = "SELECT user_id, firstname, lastname, email, gender,
date_of_birth, "
        + "height, marital_status, country, city, contact_number,
mother_tongue, "
        + "religion, caste, user_type, createdAt from users WHERE
caste = ?";
    // Initialize casteList to avoid null pointer exception
    List<User> casteList = new ArrayList();
    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(sql)) {
        stmt.setString(1, caste);
        try (ResultSet rs = stmt.executeQuery()) {
            while (rs.next()) {
                User user = new User();
                // Extract user data from the result set
                user.setUser_id(rs.getInt("user_id"));
            }
        }
    }
}

```

```

        user.setFirstname(rs.getString("firstname"));
        user.setLastname(rs.getString("lastname"));
        user.setEmail(rs.getString("email"));
        user.setGender(rs.getString("gender"));
        user.setDate_of_birth(rs.getDate("date_of_birth"));
        user.setHeight(rs.getInt("height"));
        user.setMarital_status(rs.getString("marital_status"));
        user.setCountry(rs.getString("country"));
        user.setCity(rs.getString("city"));
        user.setContact_number(rs.getString("contact_number"));
        user.setMother_tongue(rs.getString("mother_tongue"));
        user.setReligion(rs.getString("religion"));
        user.setCaste(rs.getString("caste"));
        user.setUser_type(rs.getString("user_type"));

        user.setCreatedAt(rs.getTimestamp("createdAt").toLocalDateTime());

        user.setLikes(getDetails(rs.getInt("user_id")).getLikes());

        user.setViews(getDetails(rs.getInt("user_id")).getView());
        user.setPhoto(getPhoto(rs.getInt("user_id")).getPhoto());
        casteList.add(user);
    }
}

return casteList;
}

@Override
public User searchByMemberId(int member_id) throws SQLException {

```



```

        String sql = "SELECT user_id, firstname, lastname, email, gender,
date_of_birth, "
                + "height, marital_status, country, city, contact_number,
mother_tongue, "
                + "religion, caste, user_type, createdAt from users WHERE
user_id = ?";

        User user = null;

        // Prepare the statement and set the email parameter
        try (Connection conn = DBConnection.getConnection();
PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setInt(1, member_id);
            try (ResultSet rs = stmt.executeQuery()) {
                if (rs.next()) {
                    user = extractUser(rs);
                }
            }
        }

        return user;
    }

```

```

@Override

    public User getUserByEmail(String email) throws SQLException {
        String sql = "SELECT user_id, firstname, lastname, email, gender,
date_of_birth, "
                + "height, marital_status, country, city, contact_number,
mother_tongue, "
                + "religion, caste, user_type, createdAt from users WHERE
email = ?";

        User user = null;

        // Prepare the statement and set the email parameter
        try (Connection conn = DBConnection.getConnection();
PreparedStatement stmt = conn.prepareStatement(sql)) {

```

```

        stmt.setString(1, email);
        try (ResultSet rs = stmt.executeQuery()) {
            if (rs.next()) {
                user = extractUser(rs);
            }
        }
    }
    return user;
}

```

@Override

```

    public void updateProfilePhoto(int userId, byte[] profilePhoto) throws
    SQLException {

```

```

        String sql = "UPDATE photos SET photo = ? WHERE user_id = ?";
        try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setInt(2, userId);
            stmt.setBytes(1, profilePhoto);
            int rowsUpdated = stmt.executeUpdate();
            if (rowsUpdated > 0) {
                System.out.println("User Updated Successfully");
            } else {
                throw new SQLException("User update failed. User not
found.");
            }
        }
    }
}

```

@Override

```

    public void updateUserDetails(int userId, String aboutSelf, String
familyDetails, String qualification,

```

```

        String workingAt, String designation, String hobbies, String
        favoriteMusic,

        String favoriteMovies, String favoriteCuisines, String
        favoriteBooks,

        String favoritePlaces) throws SQLException {

    String sql = "UPDATE additionaluserdetails SET about_self = ?,
family_details = ?, qualification = ?, working_at = ?, designation = ?,
hobbies = ?, favorite_music = ?, favorite_movies = ?, favorite_cuisines = ?,
favorite_books = ?, favorite_places = ? WHERE user_id = ?";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(sql)) {

        stmt.setString(1, aboutSelf);
        stmt.setString(2, familyDetails);
        stmt.setString(3, qualification);
        stmt.setString(4, workingAt);
        stmt.setString(5, designation);
        stmt.setString(6, hobbies);
        stmt.setString(7, favoriteMusic);
        stmt.setString(8, favoriteMovies);
        stmt.setString(9, favoriteCuisines);
        stmt.setString(10, favoriteBooks);
        stmt.setString(11, favoritePlaces);
        stmt.setInt(12, userId);

        int rowsUpdated = stmt.executeUpdate();

        if (rowsUpdated > 0) {
            System.out.println("User Updated Successfully");
        } else {
            throw new SQLException("User update failed. User not
found.");
        }
    }
}

```

```
//      extract user details from database

public User extractUser(ResultSet rs) throws SQLException {
    User user = new User();
    try {
        user.setUser_id(rs.getInt("user_id"));
        user.setFirstname(rs.getString("firstname"));
        user.setLastname(rs.getString("lastname"));
        user.setEmail(rs.getString("email"));
        user.setGender(rs.getString("gender"));
        user.setDate_of_birth(rs.getDate("date_of_birth"));
        user.setHeight(rs.getInt("height"));
        user.setMarital_status(rs.getString("marital_status"));
        user.setCountry(rs.getString("country"));
        user.setCity(rs.getString("city"));
        user.setContact_number(rs.getString("contact_number"));
        user.setMother_tongue(rs.getString("mother_tongue"));
        user.setReligion(rs.getString("religion"));
        user.setCaste(rs.getString("caste"));
        user.setUser_type(rs.getString("user_type"));
//          details come from external methods
        user.setLikes(getDetails(rs.getInt("user_id")).getLikes());
        user.setViews(getDetails(rs.getInt("user_id")).getView());
        user.setPhoto(getPhoto(rs.getInt("user_id")).getPhoto());
//          getting user informations

        user.setAboutSelf(getUserDetails(rs.getInt("user_id")).getAboutSelf());

        user.setFamilyDetails(getUserDetails(rs.getInt("user_id")).getFamilyDetails()
    );
}
```

```

user.setQualification(getUserDetails(rs.getInt("user_id")).getQualification()
);

user.setWorkingAt(getUserDetails(rs.getInt("user_id")).getWorkingAt());

user.setDesignation(getUserDetails(rs.getInt("user_id")).getDesignation());

user.setHobbies(getUserDetails(rs.getInt("user_id")).getHobbies());

user.setFavouriteGenre(getUserDetails(rs.getInt("user_id")).getFavouriteGenre
());

user.setFavouriteMovies(getUserDetails(rs.getInt("user_id")).getFavouriteMovi
es());

user.setFavouriteCuisine(getUserDetails(rs.getInt("user_id")).getFavouriteCui
sine());

user.setFavouriteBooks(getUserDetails(rs.getInt("user_id")).getFavouriteBooks
());

user.setFavouritePlaces(getUserDetails(rs.getInt("user_id")).getFavouritePlac
es());

    } catch (SQLException e) {
        throw new SQLException("An error occurred: " + e);
    }
    return user;
}

private String hashPassword(String password) {
    String salt = BCrypt.gensalt();
    String hashedPassword = BCrypt.hashpw(password, salt);
    return hashedPassword;
}

```

```

@Override

public List<User> getAllUsers() throws SQLException {

    String query = "SELECT * FROM users";

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

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(query); ResultSet rs =
        stmt.executeQuery()) {

        while (rs.next()) {

            // Extract user data from the result set
            User user = new User();

            user.setUser_id(rs.getInt("user_id"));
            user.setFirstname(rs.getString("firstname"));
            user.setLastname(rs.getString("lastname"));
            user.setEmail(rs.getString("email"));
            user.setGender(rs.getString("gender"));
            user.setDate_of_birth(rs.getDate("date_of_birth"));
            user.setHeight(rs.getInt("height"));
            user.setMarital_status(rs.getString("marital_status"));
            user.setCountry(rs.getString("country"));
            user.setCity(rs.getString("city"));
            user.setContact_number(rs.getString("contact_number"));
            user.setMother_tongue(rs.getString("mother_tongue"));
            user.setReligion(rs.getString("religion"));
            user.setCaste(rs.getString("caste"));
            user.setUser_type(rs.getString("user_type"));

            user.setCreatedAt(rs.getTimestamp("createdAt").toLocalDateTime());

            // getting likes and photos
            user.setLikes(getDetails(rs.getInt("user_id")).getLikes());
            user.setViews(getDetails(rs.getInt("user_id")).getView());

```

```

        user.setPhoto(getPhoto(rs.getInt("user_id")).getPhoto());
//        getting user informations

user.setAboutSelf(getUserDetails(rs.getInt("user_id")).getAboutSelf());

user.setFamilyDetails(getUserDetails(rs.getInt("user_id")).getFamilyDetails()
);

user.setQualification(getUserDetails(rs.getInt("user_id")).getQualification()
);

user.setWorkingAt(getUserDetails(rs.getInt("user_id")).getWorkingAt());

user.setDesignation(getUserDetails(rs.getInt("user_id")).getDesignation());

user.setHobbies(getUserDetails(rs.getInt("user_id")).getHobbies());

user.setFavouriteGenre(getUserDetails(rs.getInt("user_id")).getFavouriteGenre
());

user.setFavouriteMovies(getUserDetails(rs.getInt("user_id")).getFavouriteMovi
es());

user.setFavouriteCuisine(getUserDetails(rs.getInt("user_id")).getFavouriteCui
sine());

user.setFavouriteBooks(getUserDetails(rs.getInt("user_id")).getFavouriteBooks
());

user.setFavouritePlaces(getUserDetails(rs.getInt("user_id")).getFavouritePlac
es());

        userList.add(user);
    }
}
return userList;
}

```

```

@Override

public ProfilePhoto getPhoto(int userId) throws SQLException {
    ProfilePhoto profilePhoto = new ProfilePhoto();

    String query = "SELECT photo FROM photos WHERE user_id = ?";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(query)) {
        stmt.setInt(1, userId);

        try (ResultSet rs = stmt.executeQuery()) {
            if (rs.next()) {
                profilePhoto.setPhoto(rs.getBytes("photo"));
            }
        }
    }

    return profilePhoto;
}

```

```

@Override

public AdditionalUserInfo getDetails(int userId) throws SQLException {
    AdditionalUserInfo info = new AdditionalUserInfo();

    String query = "SELECT likes, view FROM additionaluserinfo WHERE
user_id = ?";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(query)) {
        stmt.setInt(1, userId);

        try (ResultSet rs = stmt.executeQuery()) {
            if (rs.next()) {
                info.setLikes(rs.getInt("likes"));
                info.setView(rs.getInt("view"));
            }
        }
    }
}

```



```

    }
    return info;
}

@Override
public void like(int profileUserId, int likingUserId) throws SQLException
{
    String selectSql = "SELECT likes, liked_users FROM additionaluserinfo
WHERE user_id = ?";

    String updateSql = "UPDATE additionaluserinfo SET likes = likes + 1,
liked_users = ? WHERE user_id = ?";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement selectStmt = conn.prepareStatement(selectSql);
        PreparedStatement updateStmt = conn.prepareStatement(updateSql)) {
        selectStmt.setInt(1, profileUserId);
        ResultSet rs = selectStmt.executeQuery();
        if (rs.next()) {
            String likedUsers = rs.getString("liked_users");
            int likes = rs.getInt("likes");
            // Check if the user has already liked this profile
            if (likedUsers == null ||
!likedUsers.contains(String.valueOf(likingUserId))) {
                // Add likingUserId to likedUsers list
                likedUsers = likedUsers == null ?
String.valueOf(likingUserId)
                : likedUsers + "," + likingUserId;
                // Update the database with the new likes count and
liked_users list
                updateStmt.setString(1, likedUsers);
                updateStmt.setInt(2, profileUserId);
                updateStmt.executeUpdate();
                System.out.println("User liked profile successfully.");
            }
        }
    }
}

```

```

        } else {
            System.out.println("User has already liked this
profile.");
        }
    } else {
        throw new SQLException("User profile not found.");
    }
}

// checking whether current user has liked other user

@Override
public boolean checkIfLiked(int likedUserId, int likingUserId) throws
SQLException {
    String query = "SELECT liked_users FROM additionaluserinfo WHERE
user_id = ?";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(query)) {
        stmt.setInt(1, likedUserId);
        ResultSet rs = stmt.executeQuery();
        if (rs.next()) {
            String likedUsers = rs.getString("liked_users");
            if (likedUsers != null && !likedUsers.isEmpty()) {
                String[] likedUserIds = likedUsers.split(",");
                for (String id : likedUserIds) {
                    if (Integer.parseInt(id.trim()) == likingUserId) {
                        return true;
                    }
                }
            }
        }
    }
}

```

```

    }
    return false;
}

@Override
public void addView(int userId) throws SQLException {
    String updateQuery = "UPDATE additionaluserinfo SET view = view + 1
WHERE user_id = ?";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(updateQuery)) {
        stmt.setInt(1, userId);
        stmt.executeUpdate();
    } catch (SQLException e) {
        throw new SQLException("Error updating view count for user ID: "
+ userId, e);
    }
}

@Override
public AdditionalUserDetails getUserDetails(int userId) throws
SQLException {
    String sql = "SELECT * FROM additionaluserdetails WHERE user_id = ?";
    AdditionalUserDetails userDetails = new AdditionalUserDetails();

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement statement = conn.prepareStatement(sql)) {
        statement.setInt(1, userId);
        try (ResultSet rs = statement.executeQuery()) {
            if (rs.next()) {
                userDetails.setUserId(rs.getInt("user_id"));
                userDetails.setAboutSelf(rs.getString("about_self"));

                userDetails.setFamilyDetails(rs.getString("family_details"));
            }
        }
    }
}

```

```

userDetails.setQualification(rs.getString("qualification"));
        userDetails.setWorkingAt(rs.getString("working_at"));
        userDetails.setDesignation(rs.getString("designation"));
        userDetails.setHobbies(rs.getString("hobbies"));

userDetails.setFavouriteGenre(rs.getString("favorite_music"));

userDetails.setFavouriteMovies(rs.getString("favorite_movies"));

userDetails.setFavouriteCuisine(rs.getString("favorite_cuisines"));

userDetails.setFavouriteBooks(rs.getString("favorite_books"));

userDetails.setFavouritePlaces(rs.getString("favorite_places"));
    }
}
}
return userDetails;
}

```

```

@Override

public List<AdditionalUserDetails> getAllDetails() throws SQLException {
    String sql = "SELECT * FROM additionaluserdetails";

    List<AdditionalUserDetails> userList = new ArrayList<>();

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement stmt = conn.prepareStatement(sql); ResultSet rs =
        stmt.executeQuery()) {
        while (rs.next()) {
            // Extract user data from the result set
            AdditionalUserDetails userDetails = new
AdditionalUserDetails();
            userDetails.setUserId(rs.getInt("user_id"));

```

```

        userDetails.setAboutSelf(rs.getString("about_self"));
        userDetails.setFamilyDetails(rs.getString("family_details"));
        userDetails.setQualification(rs.getString("qualification"));
        userDetails.setWorkingAt(rs.getString("working_at"));
        userDetails.setDesignation(rs.getString("designation"));
        userDetails.setHobbies(rs.getString("hobbies"));

        userDetails.setFavouriteGenre(rs.getString("favorite_music"));

        userDetails.setFavouriteMovies(rs.getString("favorite_movies"));

        userDetails.setFavouriteCuisine(rs.getString("favorite_cuisines"));

        userDetails.setFavouriteBooks(rs.getString("favorite_books"));

        userDetails.setFavouritePlaces(rs.getString("favorite_places"));
        userList.add(userDetails);
    }
}
return userList;
}

```

```

@Override
public void deleteUser(int userId) throws SQLException {
    String deleteUserQuery = "DELETE FROM users WHERE user_id = ?";
    String deletePhotosQuery = "DELETE FROM photo WHERE user_id = ?";
    String deleteAdditionalUserDetailsQuery = "DELETE FROM
additionaluserdetails WHERE user_id = ?";
    String deleteAdditionalUserInfoQuery = "DELETE FROM
additionaluserinfo WHERE user_id = ?";

    try (Connection conn = DBConnection.getConnection());
    PreparedStatement deletePhotosStmt =
conn.prepareStatement(deletePhotosQuery); PreparedStatement

```

```

deleteAdditionalUserDetailsStmt =
conn.prepareStatement(deleteAdditionalUserDetailsQuery); PreparedStatement
deleteAdditionalUserInfoStmt =
conn.prepareStatement(deleteAdditionalUserInfoQuery); PreparedStatement
deleteUserStmt = conn.prepareStatement(deleteUserQuery)) {

    // Begin transaction
    conn.setAutoCommit(false);

    // Delete from dependent tables first
    deletePhotosStmt.setInt(1, userId);
    deletePhotosStmt.executeUpdate();
    deleteAdditionalUserDetailsStmt.setInt(1, userId);
    deleteAdditionalUserDetailsStmt.executeUpdate();
    deleteAdditionalUserInfoStmt.setInt(1, userId);
    deleteAdditionalUserInfoStmt.executeUpdate();

    // Finally, delete from the main user table
    deleteUserStmt.setInt(1, userId);
    deleteUserStmt.executeUpdate();

    // Commit transaction
    conn.commit();
} catch (SQLException e) {
    throw e;
}
}

```

```

@Override
public int getTotalLikes() throws SQLException {

    String sql = "SELECT SUM(likes) FROM additionaluserinfo";

    try (Connection connection = DBConnection.getConnection();
        PreparedStatement stmt = connection.prepareStatement(sql); ResultSet rs =
        stmt.executeQuery()) {

        if (rs.next()) {

            return rs.getInt(1); // Get the total likes from the result

```

```

        }
    }

    return 0; // Return 0 if no data is found
}

// Method to get total views
@Override

public int getTotalViews() throws SQLException {
    String sql = "SELECT SUM(view) FROM additionaluserinfo";

    try (Connection connection = DBConnection.getConnection();
        PreparedStatement stmt = connection.prepareStatement(sql); ResultSet rs =
        stmt.executeQuery()) {
        if (rs.next()) {
            return rs.getInt(1); // Get the total views from the result
        }
    }

    return 0; // Return 0 if no data is found
}

@Override

public String getSubscriptionLeft(int userId) throws SQLException {
    String sql = "SELECT amount, date_paid FROM payment WHERE user_id =
?";

    try (Connection connection = DBConnection.getConnection();
        PreparedStatement ps = connection.prepareStatement(sql)) {
        ps.setInt(1, userId);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            int amount = rs.getInt("amount");
            Timestamp datePaid = rs.getTimestamp("date_paid");
            LocalDateTime paidDate = datePaid.toLocalDateTime();

```

```

    LocalDateTime currentDate = LocalDateTime.now();
    // Calculate subscription period based on amount
    LocalDateTime expiryDate;
    switch (amount) {
        case 50:
            expiryDate = paidDate.plusWeeks(1); // 1 week for 50
            break;
        case 500:
            expiryDate = paidDate.plusMonths(1); // 1 month for
500
            break;
        default:
            return "Invalid amount!";
    }
    // Calculate time remaining
    Duration duration = Duration.between(currentDate,
expiryDate);
    if (duration.isNegative()) {
        return "Subscription expired!";
    } else {
        long daysLeft = duration.toDays();
        long hoursLeft = duration.toHoursPart();
        return daysLeft + " days, " + hoursLeft + " hours left";
    }
} else {
    return "No subscription found for user!";
}
}
}

```


@Override

```
public void updatePaymentDetails(int userId, String newAmount, String
newPaymentType) throws SQLException {
```

```
    String sql = "UPDATE payment SET amount = ?, payment_type = ?,
date_paid = NOW() WHERE user_id = ?";
```

```
    try (Connection connection = DBConnection.getConnection();
PreparedStatement ps = connection.prepareStatement(sql)) {
```

```
        ps.setString(1, newAmount);
```

```
        ps.setString(2, newPaymentType);
```

```
        ps.setInt(3, userId);
```

```
        int rowsUpdated = ps.executeUpdate();
```

```
        if (rowsUpdated > 0) {
```

```
            System.out.println("Payment details updated successfully.");
```

```
        } else {
```

```
            System.out.println("No payment record found for user.");
```

```
        }
```

```
    }
```

```
}
```

@Override

```
public List<UserAdd> getAllAdds() throws SQLException {
```

```
    List<UserAdd> userAdds = new ArrayList<>();
```

```
    String query = "SELECT ua.user_id, ua.title, ua.body, u.firstname,
p.photo "
```

```
        + "FROM user_adds ua "
```

```
        + "JOIN users u ON ua.user_id = u.user_id "
```

```
        + "JOIN photos p ON ua.user_id = p.user_id";
```

```
    try (Connection connection = DBConnection.getConnection();
PreparedStatement preparedStatement = connection.prepareStatement(query);
ResultSet resultSet = preparedStatement.executeQuery()) {
```

```
        while (resultSet.next()) {
```

```
            UserAdd userAdd = new UserAdd();
```

```

        userAdd.setUserId(resultSet.getInt("user_id"));
        userAdd.setTitle(resultSet.getString("title"));
        userAdd.setBody(resultSet.getString("body"));
        // Fetch first name from 'users' table
        userAdd.setName(resultSet.getString("firstname"));
        // Fetch profile image from 'photos' table (as byte array)
        userAdd.setPhoto(resultSet.getBytes("photo"));
        userAdds.add(userAdd);
    }
}
return userAdds;
}

@Override
public void updateUserAdd(int userId, String title, String body) throws
SQLException {
    String sql = "UPDATE user_adds SET title = ?, body = ? WHERE user_id
= ?";

    try (Connection conn = DBConnection.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(sql)) {
        // Set the parameters for the prepared statement
        pstmt.setString(1, title);
        pstmt.setString(2, body);
        pstmt.setInt(3, userId);
        // Execute the update
        int rowsUpdated = pstmt.executeUpdate();
        if (rowsUpdated > 0) {
            System.out.println("An existing user_add was updated
successfully!");
        } else {

```

```

        System.out.println("No user_add found for the given
user_id.");
    }
    } catch (SQLException e) {
        throw new SQLException("Error updating user_add: " +
e.getMessage(), e);
    }
}

// Insert new user add
@Override
public void insertUserAdd(int userId, String title, String body) throws
SQLException {
    String sql = "INSERT INTO user_adds (user_id, title, body) VALUES (?,
?, ?)";

    try (Connection conn = DBConnection.getConnection();
PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setInt(1, userId);
        pstmt.setString(2, title);
        pstmt.setString(3, body);

        int rowsInserted = pstmt.executeUpdate();
        if (rowsInserted > 0) {
            System.out.println("A new user_add was inserted
successfully!");
        }
    } catch (SQLException e) {
        throw new SQLException("Error inserting user_add: " +
e.getMessage(), e);
    }
}

```

```
}  
}
```

EJB Interface

```
package com.matrimony.stateful;  
  
import com.matrimony.models.AdditionalUserDetails;  
import com.matrimony.models.AdditionalUserInfo;  
import com.matrimony.models.ProfilePhoto;  
import com.matrimony.models.User;  
import com.matrimony.models.UserAdd;  
import com.mysql.cj.jdbc.Blob;  
import jakarta.ejb.Local;  
import java.sql.SQLException;  
import java.util.List;  
  
/**  
 *  
 * @author dsihle875@gmail.com  
 */  
@Local  
public interface MatrimonySessionBeanLocal {  
  
    /**  
     *  
     * @param firstname  
     * @param lastname  
     * @param email  
     * @param password  
     * @param gender
```

```
* @param dob
* @param height
* @param maritalStatus
* @param country
* @param city
* @param number
* @param language
* @param religion
* @param caste
* @param userType
* @return
* @throws SQLException
*/
public int insertBasicUserInformation(String firstname,
    String lastname,
    String email,
    String password,
    String gender,
    String dob,
    int height,
    String maritalStatus,
    String country,
    String city,
    String number,
    String language,
    String religion,
    String caste,
    String userType,
    String paymentMethod,
    String amount
```

) throws SQLException;

/**

*

* @param caste

* @return

* @throws SQLException

*/

public List<User> searchByCaste(String caste) throws SQLException;

/**

*

* @param member_id

* @return

* @throws SQLException

*/

public User searchById(int member_id) throws SQLException;

/**

*

* @param email

* @return

* @throws SQLException

*/

public User getUserByEmail(String email) throws SQLException;

/**

*

```
* @param username
* @param password
* @return
* @throws SQLException
*/
public String loginUser(String username, String password) throws
SQLException;
```

```
/**
 *
 * @param id
 * @param firstname
 * @param lastname
 * @param email
 * @param gender
 * @param dob
 * @param height
 * @param maritalStatus
 * @param country
 * @param city
 * @param number
 * @param language
 * @param religion
 * @param caste
 * @param userType
 * @throws SQLException
 */
public void updateUser(
    int id,
```

```
        String firstname,
        String lastname,
        String email,
        String gender,
        String dob,
        String height,
        String maritalStatus,
        String country,
        String city,
        String number,
        String language,
        String religion,
        String caste,
        String userType
    ) throws SQLException;
```

```
/**
 *
 * @param id
 * @param oldPassword
 * @param newPassword
 * @throws SQLException
 */
public void updatePassword(int id, String oldPassword, String
newPassword) throws SQLException;
```

```
/**
 *
 * @param userId
```



```

    * @param profilePhoto
    * @throws SQLException
    */
    public void updateProfilePhoto(int userId, byte[] profilePhoto) throws
    SQLException;

    /**
     *
     * @param userId
     * @param aboutSelf
     * @param familyDetails
     * @param qualification
     * @param workingAt
     * @param designation
     * @param hobbies
     * @param favoriteMusic
     * @param favoriteMovies
     * @param favoriteCuisines
     * @param favoriteBooks
     * @param favoritePlaces
     * @throws SQLException
     */
    public void updateUserDetails(int userId, String aboutSelf, String
    familyDetails, String qualification,
        String workingAt, String designation, String hobbies, String
    favoriteMusic,
        String favoriteMovies, String favoriteCuisines, String
    favoriteBooks,
        String favoritePlaces) throws SQLException;

    /**

```

```

*

* @return returns a list of users
* @throws SQLException
*/
public List<User> getAllUsers() throws SQLException;

/**
*
* @return returns all user details
* @throws SQLException
*/
public List<AdditionalUserDetails> getAllDetails() throws SQLException;

/**
*
* @param userId
* @return
* @throws SQLException
*/
public ProfilePhoto getPhoto(int userId) throws SQLException;

/**
*
* @param userId
* @return
* @throws SQLException
*/
public AdditionalUserInfo getDetails(int userId) throws SQLException;

/**

```

```

*
* @param userId
* @param likingUserId
* @throws SQLException
*/
public void like(int userId, int likingUserId) throws SQLException;

/**
*
* @param likedUserId
* @param likingUserId
* @return
* @throws SQLException
*/
public boolean checkIfLiked(int likedUserId, int likingUserId) throws
SQLException;

/**
*
* @param userId
* @throws SQLException
*/
public void addView(int userId) throws SQLException;

/**
*
* @param userId
* @return
* @throws SQLException

```

```
*/  
  
    public AdditionalUserDetails getUserDetails(int userId) throws  
SQLException;
```

```
/**  
 *  
 * @param userId  
 * @throws SQLException  
 */
```

```
public void deleteUser(int userId) throws SQLException;
```

```
/**  
 *  
 * @return  
 * @throws SQLException  
 */
```

```
public int getTotalViews() throws SQLException;
```

```
/**  
 *  
 * @return  
 * @throws SQLException  
 */
```

```
public int getTotalLikes() throws SQLException;
```

```
/**  
 *  
 * @param userId  
 * @return  
 * @throws SQLException
```

```

    */

    public String getSubscriptionLeft(int userId) throws SQLException;

    /**
     *
     * @param userId
     * @param newAmount
     * @param newPaymentType
     * @throws SQLException
     */
    public void updatePaymentDetails(int userId, String newAmount, String
newPaymentType) throws SQLException;

    /**
     *
     * @return
     * @throws SQLException
     */
    public List<UserAdd> getAllAdds() throws SQLException;

    /**
     *
     * @param userId
     * @param title
     * @param body
     * @throws SQLException
     */
    public void updateUserAdd(int userId, String title, String body) throws
SQLException;

```

```

/**
 *
 * @param userId
 * @param title
 * @param body
 * @throws SQLException
 */
    public void insertUserAdd(int userId, String title, String body) throws
SQLException;
}

```

EJB Util Class

```

package com.matrimony.util;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/*
 *
 * @author dsihle875@gmail.com
 */
public class DBConnection{
    // connection variables and constants
    private static String dbName = "matrinomydb";
    private static String username = "root";
    private static String password = "";
    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/" +
dbName;

```

```

private static final String JDBC_USERNAME = username;
private static final String JDBC_PASSWORD = password;

static{
    try{
        Class.forName("com.mysql.cj.jdbc.Driver");
    }catch(ClassNotFoundException e){
        throw new ExceptionInInitializerError(e);
    }
}

public static Connection getConnection() throws SQLException {
    return DriverManager.getConnection(JDBC_URL, JDBC_USERNAME,
JDBC_PASSWORD);
}

public static void closeConnection(Connection c){
    if(c != null){
        try {
            c.close();
        } catch (SQLException e) {
            throw new ExceptionInInitializerError(e);
        }
    }
}
}

```

Web Controllers

Login

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
 default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
 edit this template
 */
package com.matrinomy.controllers;

import com.matrimony.models.User;
import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.sql.SQLException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 */
```



```

* @author dsihl
*/

@WebServlet(name = "Login", urlPatterns = {"/login-user"})
public class Login extends HttpServlet {
    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();

            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");
        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override

```

```

protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    //      session object
    HttpSession session = request.getSession();
    String email = request.getParameter("email");
    String password = request.getParameter("password");

    User user = null;
    try {
        String usertype = matrimonySessionBean.loginUser(email,
password);
        List<User> users = matrimonySessionBean.getAllUsers();
        if(null == usertype){
            request.setAttribute("errorMessage", "login failed email or
password incorrect");
            request.getRequestDispatcher("login.jsp").forward(request,
response);
        }else switch (usertype) {
            case "User":
                user = matrimonySessionBean.getUserByEmail(email);
                session.setAttribute("users", users);
                session.setAttribute("user", user);
                response.sendRedirect("user");
                break;
            case "Admin":
                user = matrimonySessionBean.getUserByEmail(email);
                session.setAttribute("users", users);
                session.setAttribute("user", user);
                response.sendRedirect("admin");
                break;
        }
    }
}

```

```

        default:
            request.setAttribute("errorMessage", "login failed email
or password incorrect");

request.getRequestDispatcher("login.jsp").forward(request, response);
            break;
        }

    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null,
ex);

        request.setAttribute("errorMessage", "login failed email or
password incorrect " + ex.getMessage());

        request.getRequestDispatcher("login.jsp").forward(request,
response);
    }

}

}

```

[Register](#)

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */

package com.matrinomy.controllers;

import com.matrimony.stateful.MatrimonySessionBeanLocal;

```

```

import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "Register", urlPatterns = {"/register-user"})
public class Register extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();
            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");
        } catch (NamingException e) {

```

```

        throw new ServletException("Failed to lookup Admin Service EJB",
e);
    }
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {

    try {
        //      parameters from register form
        String firstname = request.getParameter("firstname");
        String lastname = request.getParameter("lastname");
        String email = request.getParameter("email");
        String password = request.getParameter("password");
        String gender = request.getParameter("gender");
        String date_of_birth = request.getParameter("date_of_birth");
        String heightString = request.getParameter("height");
        String marital_status = request.getParameter("status");
        String country = request.getParameter("country");

```

```
String city = request.getParameter("city");
String contact_number = request.getParameter("number");
String mother_tongue = request.getParameter("language");
String religion = request.getParameter("religion");
String caste = request.getParameter("caste");
String usertype = request.getParameter("usertype");
String paymentMethod = request.getParameter("payment_method");
String amount = request.getParameter("payment_amount");
int height = Integer.parseInt(heightString);

matrimonySessionBean.insertBasicUserInformation(
    firstname,
    lastname,
    email,
    password,
    gender,
    date_of_birth,
    height,
    marital_status,
    country,
    city,
    contact_number,
    mother_tongue,
    religion,
    caste,
    usertype,
    paymentMethod,
    amount
);
```

```

        request.getRequestDispatcher("login").forward(request, response);
    } catch (SQLException ex) {
        Logger.getLogger(Register.class.getName()).log(Level.SEVERE,
null, ex);
        request.setAttribute("errorMessage", "Registration failed: " +
ex.getMessage());
        request.getRequestDispatcher("register.jsp").forward(request,
response);
    }

}

}

```

[Browse](#)

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */

package com.matrinomy.controllers;

import com.matrimony.models.User;
import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;

```

```

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.sql.SQLException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihle875@gmail.com
 */
@WebServlet(name = "Browse", urlPatterns = {"/browse"})
public class Browse extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();

            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");

        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);

```



```

    }
}

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    HttpSession session = request.getSession();
    try {
        List<User> users = matrimonySessionBean.getAllUsers();
        session.setAttribute("matrimonySessionBean",
matrimonySessionBean);

        request.setAttribute("users", users);
        request.getRequestDispatcher("browse.jsp").forward(request,
response);
    } catch (SQLException ex) {
        Logger.getLogger(Browse.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}
}

```

Edit

```
/*  
    * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-  
    default.txt to change this license  
    * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to  
    edit this template  
*/  
  
package com.matrinomy.controllers;  
  
import com.matrimony.models.User;  
import com.matrimony.stateful.MatrimonySessionBeanLocal;  
import java.io.IOException;  
import java.io.PrintWriter;  
import jakarta.servlet.ServletException;  
import jakarta.servlet.annotation.WebServlet;  
import jakarta.servlet.http.HttpServlet;  
import jakarta.servlet.http.HttpServletRequest;  
import jakarta.servlet.http.HttpServletResponse;  
import jakarta.servlet.http.HttpSession;  
import java.sql.SQLException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.naming.Context;  
import javax.naming.InitialContext;  
import javax.naming.NamingException;  
  
/**  
    *  
    * @author dsihl
```

```

*/
@WebServlet(name = "Edit", urlPatterns = {"/edit"})
public class Edit extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();

            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");
        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {

```

```

String firstname = request.getParameter("firstname");
String lastname = request.getParameter("lastname");
String email = request.getParameter("email");
String gender = request.getParameter("gender");
String dob = request.getParameter("date_of_birth");
String height = request.getParameter("height");
String maritalStatus = request.getParameter("marital_status");
String country = request.getParameter("country");
String city = request.getParameter("city");
String number = request.getParameter("contact_number");
String language = request.getParameter("mother_tongue");
String religion = request.getParameter("religion");
String caste = request.getParameter("caste");
String userType = request.getParameter("userType");
String userId = request.getParameter("userId");

int id = Integer.parseInt(userId);
User user = null;
HttpSession session = request.getSession();
try {
    matrimonySessionBean.updateUser(id, firstname, lastname, email,
gender, dob, height, maritalStatus, country, city, number, language,
religion, caste, userType);
    user = matrimonySessionBean.getUserByEmail(email);
    session.setAttribute("user", user);
    response.sendRedirect("user");
} catch (SQLException ex) {
    Logger.getLogger(EditDetails.class.getName()).log(Level.SEVERE,
null, ex);
    response.sendRedirect("editprofile?id=" + userId);
}

```

```
}
```

```
}
```

[Edit](#) [Add](#)

```
/*
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-  
default.txt to change this license
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to  
edit this template
```

```
*/
```

```
package com.matrinomy.controllers;
```

```
import com.matrimony.stateful.MatrimonySessionBeanLocal;
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import jakarta.servlet.ServletException;
```

```
import jakarta.servlet.annotation.WebServlet;
```

```
import jakarta.servlet.http.HttpServlet;
```

```
import jakarta.servlet.http.HttpServletRequest;
```

```
import jakarta.servlet.http.HttpServletResponse;
```

```
import java.sql.SQLException;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
import javax.naming.Context;
```

```
import javax.naming.InitialContext;
```

```
import javax.naming.NamingException;
```

```
/**
```

```
 *
```

```

* @author dsihl
*/
@WebServlet(name = "EditAdd", urlPatterns = {"/edit-add"})
public class EditAdd extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();
            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");
        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)

```

```

        throws ServletException, IOException {

    try {
        String title = request.getParameter("title");
        String body = request.getParameter("body");
        String id = request.getParameter("id");

        int userId = Integer.parseInt(id);

        matrimonySessionBean.updateUserAdd(userId, title, body);

        response.sendRedirect("services");
    } catch (SQLException ex) {
        Logger.getLogger(EditAdd.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}

```

Edit Details

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */

package com.matrinomy.controllers;

import com.matrimony.models.User;
import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;

```

```

import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "EditDetails", urlPatterns = {"/editdetails"})
public class EditDetails extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();
            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");

```



```

        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }
}

```

```

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {

```

```

    String aboutSelf = request.getParameter("aboutSelf");
    String familyDetails= request.getParameter("familyDetails");
    String qualification= request.getParameter("qualification");
    String workingAt= request.getParameter("workingAt");
    String designation= request.getParameter("designation");
    String hobbies = request.getParameter("hobbies");
    String favoriteMusic = request.getParameter("favoriteMusic");
    String favoriteMovies = request.getParameter("favoriteMovies");
    String favoriteCuisine = request.getParameter("favoriteCuisines");
    String favoriteBooks = request.getParameter("favoriteBooks");
    String favoritePlaces = request.getParameter("favoritePlace");

```

```

String userId = request.getParameter("user_id");

int id = Integer.parseInt(userId);
User user = null;
HttpSession session = request.getSession();

try {
    matrimonySessionBean.updateUserDetails(id, aboutSelf,
familyDetails, qualification, workingAt, designation, hobbies, favoriteMusic,
favoriteMovies, favoriteCuisine, favoriteBooks, favoritePlaces);

    user = matrimonySessionBean.searchByMemberId(id);
    session.setAttribute("user", user);
    response.sendRedirect("user");
} catch (SQLException ex) {
    Logger.getLogger(EditDetails.class.getName()).log(Level.SEVERE,
null, ex);
    response.sendRedirect("editdetailsroute?id="+userId);
}
}
}

```

Edit Profile

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */
package com.matrinomy.controllers;

import com.matrimony.models.User;

```

```

import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "EditProfile", urlPatterns = {"/editprofile"})
public class EditProfile extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();
            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");

```

```

        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    String id = request.getParameter("id");
    int user_id = Integer.parseInt(id);

    try {
        User user = (User)
matrimonySessionBean.searchByMemberId(user_id);
        request.setAttribute("user", user);

        request.getRequestDispatcher("edit-profile.jsp").forward(request,
response);
    }
}

```

```

        } catch (SQLException ex) {
            Logger.getLogger(EditProfile.class.getName()).log(Level.SEVERE,
null, ex);
        }

    }

}

```

Insert Add

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */
package com.matrimony.controllers;

import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;

```

```

import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "InsertAdd", urlPatterns = {"/insert-add"})
public class InsertAdd extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();

            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");
        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs

```

```

        * @throws IOException if an I/O error occurs
        */
        @Override
        protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
            try {
                String title = request.getParameter("title");
                String body = request.getParameter("body");
                String id = request.getParameter("id");

                int userId = Integer.parseInt(id);

                matrimonySessionBean.insertUserAdd(userId, title, body);

                response.sendRedirect("services");
            } catch (SQLException ex) {
                Logger.getLogger(InsertAdd.class.getName()).log(Level.SEVERE,
null, ex);
            }
        }
    }
}

```

Like

/*

* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license

* Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template

```
*/  
  
package com.matrimony.controllers;  
  
  
import com.matrimony.stateful.MatrimonySessionBeanLocal;  
import java.io.IOException;  
import java.io.PrintWriter;  
import jakarta.servlet.ServletException;  
import jakarta.servlet.annotation.WebServlet;  
import jakarta.servlet.http.HttpServlet;  
import jakarta.servlet.http.HttpServletRequest;  
import jakarta.servlet.http.HttpServletResponse;  
import java.sql.SQLException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.naming.Context;  
import javax.naming.InitialContext;  
import javax.naming.NamingException;  
  
/**  
 *  
 * @author dsihl  
 */  
@WebServlet(name = "Like", urlPatterns = {"/like"})  
public class Like extends HttpServlet {  
  
    private MatrimonySessionBeanLocal matrimonySessionBean;  
  
    @Override
```



```

public void init() throws ServletException {
    try {
        Context ctx = new InitialContext();
        matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");
    } catch (NamingException e) {
        throw new ServletException("Failed to lookup Admin Service EJB",
e);
    }
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {

    response.setContentType("application/json");
    PrintWriter out = response.getWriter();
    try {
//        user that will be liked

        int profileUserId =
Integer.parseInt(request.getParameter("profileUserId"));

```

```

        //            user that will be liking
        int likingUserId =
Integer.parseInt(request.getParameter("likingUserId"));
        matrimonySessionBean.like(profileUserId, likingUserId);
        response.setStatus(200);
        out.print("{\"status\": \"success\"}");
    } catch (SQLException ex) {
        Logger.getLogger(Like.class.getName()).log(Level.SEVERE, null,
ex);
        response.setStatus(500);
        out.print("{\"status\": \"error\"}");
    } finally {
        out.close();
    }
}

}

```

Search Result

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */
package com.matrinomy.controllers;

import com.matrimony.models.User;
import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;

```

```

import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "SearchResult", urlPatterns = {"/searchresult"})
public class SearchResult extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();

```

```

        matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");

        } catch (NamingException e) {

            throw new ServletException("Failed to lookup Admin Service EJB",
e);

        }

    }

}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)

    throws ServletException, IOException {

    List<User> casteList;

    HttpSession session = request.getSession();

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

    if (!caste.isEmpty()) {

        try {

            casteList = matrimonySessionBean.searchByCaste(caste);

            request.setAttribute("casteList", casteList);

            session.setAttribute("matrimonySessionBean",
matrimonySessionBean);

```

```

        request.getRequestDispatcher("search-
result.jsp").forward(request, response);
    } catch (SQLException ex) {

Logger.getLogger(SearchResult.class.getName()).log(Level.SEVERE, null, ex);
    }
    } else {
        //request.setAttribute("casteList", casteList);
        // Forward the request to the JSP page to display the results
        request.getRequestDispatcher("search-
result.jsp").forward(request, response);
    }
}
}
}

```

Search Result ID

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */
package com.matrinomy.controllers;

import com.matrimony.models.User;
import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;

```

```

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.sql.SQLException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "SearchResultID", urlPatterns = {"/searchresultid"})
public class SearchResultID extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();

            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");

        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);

```

```

    }
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {

    User user;

    String userId = request.getParameter("member_id");
    int id = Integer.parseInt(userId);
    HttpSession session = request.getSession();
    if (userId != null) {
        try {
            user = matrimonySessionBean.searchByMemberId(id);
            request.setAttribute("userId", user);
            session.setAttribute("matrimonySessionBean",
matrimonySessionBean);

            // Forward the request to the JSP page to display the results
            request.getRequestDispatcher("search-
resultID.jsp").forward(request, response);
        } catch (SQLException ex) {

```

```

Logger.getLogger(SearchResult.class.getName()).log(Level.SEVERE, null, ex);
    }
    } else {
        //request.setAttribute("casteList", casteList);
        // Forward the request to the JSP page to display the results
        request.getRequestDispatcher("search-
resultID.jsp").forward(request, response);
    }
}
}
}

```

Update Password

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */
package com.matrinomy.controllers;

import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;

```



```

import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "UpdatePassword", urlPatterns = {"/update-password"})
public class UpdatePassword extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();
            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");
        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }

    /**
     * Handles the HTTP <code>POST</code> method.

```

```

*
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {

    String oldPassword = request.getParameter("old-password");
    String newPassword = request.getParameter("new-password");
    String id = request.getParameter("id");

    int userId = Integer.parseInt(id);

    try {
        matrimonySessionBean.updatePassword(userId, oldPassword,
newPassword);

        HttpSession session = request.getSession(false);
        if (session != null) {
            session.invalidate();
        }

        response.sendRedirect("login.jsp");
    } catch (SQLException ex) {
        request.setAttribute("errorMessage", "update failed password
incorrect" + ex.getMessage());
    }
}

```

```

        request.getRequestDispatcher("update-
password.jsp").forward(request, response);

Logger.getLogger(UpdatePassword.class.getName()).log(Level.SEVERE, null, ex);
    }

}

}

```

Update Payment

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to
edit this template
 */
package com.matrimony.controllers;

import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;

```

```

import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "UpdatePayment", urlPatterns = {"/update-payment"})
public class UpdatePayment extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();

            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");
        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs

```

```

        * @throws IOException if an I/O error occurs
        */
        @Override
        protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

            try {
                String paymentMethod = request.getParameter("payment_method");
                String paymentAmount = request.getParameter("payment_amount");
                String id = request.getParameter("id");

                int userId = Integer.parseInt(id);

                matrimonySessionBean.updatePaymentDetails(userId, paymentAmount,
paymentMethod);

                response.sendRedirect("user");
            } catch (SQLException ex) {
                Logger.getLogger(UpdatePayment.class.getName()).log(Level.SEVERE,
null, ex);
            }
        }
    }
}

```

Update Photo

```

package com.matrinomy.controllers;

import com.matrimony.models.User;
import com.matrimony.stateful.MatrimonySessionBeanLocal;
import java.io.IOException;

```

```

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.MultipartConfig;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import jakarta.servlet.http.Part;
import java.io.InputStream;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihle875@gmail.com
 */
@WebServlet(name = "UpdatePhoto", urlPatterns = {"/updatephoto"})
@MultipartConfig(maxFileSize = 11124776)
public class UpdatePhoto extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();

```

```

        matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");

        } catch (NamingException e) {

            throw new ServletException("Failed to lookup Admin Service EJB",
e);

        }

    }

}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse
response)

    throws ServletException, IOException {

    try {

        HttpSession session = request.getSession(true);
        User user = (User) session.getAttribute("user");
        int user_id = user.getUser_id();

        // Get the uploaded file as a part (since it's a file upload)
        Part filePart = request.getPart("photo");

        if (filePart != null && filePart.getSize() > 0) {

```



```

import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

/**
 *
 * @author dsihl
 */
@WebServlet(name = "ViewProfile", urlPatterns = {"/viewprofile"})
public class ViewProfile extends HttpServlet {

    private MatrimonySessionBeanLocal matrimonySessionBean;

    @Override
    public void init() throws ServletException {
        try {
            Context ctx = new InitialContext();
            matrimonySessionBean = (MatrimonySessionBeanLocal)
ctx.lookup("java:global/Matrimony/MatrimonyEJB/MatrimonySessionBean!com.matri
mony.stateful.MatrimonySessionBeanLocal");

```

```

        } catch (NamingException e) {
            throw new ServletException("Failed to lookup Admin Service EJB",
e);
        }
    }

/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {

    String userId = request.getParameter("id");

    try {
//        user filtered by id
        User user = (User)
matrimonySessionBean.searchByMemberId(Integer.parseInt(userId));
        matrimonySessionBean.addView(Integer.parseInt(userId));
        request.setAttribute("user", user);
        request.getRequestDispatcher("view-user.jsp").forward(request,
response);
    } catch (SQLException ex) {

```

```

        Logger.getLogger(ViewProfile.class.getName()).log(Level.SEVERE,
null, ex);
    }

}

}

```

JSP Pages

[Index](#)

```
<%--
```

```
    Document    : index
```

```
    Created on  : Oct 3, 2024, 6:20:23 PM
```

```
    Author      : dsihle875@gmail.com
```

```
--%>
```

```
<%@page import="com.matrimony.models.User"%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

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

```
    <%
```

```
        // Get the current session user (if available)
```

```
        User user = (User) session.getAttribute("user");
```

```
    %>
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```
    <meta name="description"
```

```

        content="Find your perfect match on  Matrimony - the leading
matrimony site for genuine connections." />

    <meta name="author" content=" Projects" />

    <meta name="keywords"

        content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />

    <meta name="theme-color" content="#ff4081" />


    <!-- Open Graph (for Facebook, LinkedIn) -->

    <meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />

    <meta property="og:type" content="website" />

    <meta property="og:url" content="https://www.matrimony.com" />

    <meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

    <meta property="og:description"

        content=" Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />

    <meta property="og:locale" content="en_US" />


    <!-- Twitter Cards -->

    <meta name="twitter:card" content="summary_large_image" />

    <meta name="twitter:title" content=" Matrimony - Your Trusted
Matchmaking Platform" />

    <meta name="twitter:description"

        content="Find your life partner on  Matrimony - where real
connections are made." />

    <meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

    <meta name="twitter:site" content="@Matrimony" />


    <!-- css -->

```

```

    <link rel="stylesheet" href="./assets/css/style.css">
    <title>Matrinomy Site</title>
</head>

<body>
    <div class="wrapper">
        <header class="site-header">
            <nav>
                <a href="javascript:void(0)" class="active-link"><strong>MATRIMONY</strong></a> |
                <a href="search">Search</a>
                <a href="about">About Us</a>
                <a href="browse">Looking For</a>
                <a href="services">Services</a>
                <a href="contact">Contact Us</a>
            </nav>

            <div class="nav-btn">
                <% if (user == null) { %>
                <!-- If user is NOT logged in -->
                <p><a href="login.jsp" class="login-btn">Have an
account?</a></p>
                <p><a href="register.jsp" class="signup-btn">Sign
up</a></p>
                <% } else { %>
                <!-- If user is logged in -->
                <p><a href="user.jsp" class="login-btn">View
Profile?</a></p>
                <p><a href="logout" class="signup-btn">Logout</a></p>
                <% }%>
            </div>

```

```

        <div class="mobile-menu-icon"
onclick="toggleMobileMenu()">≡</div>

    </header>

    <main>

        <!-- home section -->

        <section class="home-page">

            <div class="block">

                <% if (user == null) { %>

                    <!-- If user is NOT logged in -->

                    <h1>Welcome to Our Matrimony Site.</h1>

                    <p class="intro">We are dedicated to helping
individuals find their perfect life partner. Our platform offers a user-
friendly experience that prioritizes inclusivity and accessibility for all
users.</p>

                    <p>Join us to explore profiles, connect with
potential matches, and embark on your journey towards love and
companionship.</p>

                    <a href="register.jsp" class="button">Get Started
Now</a>

                    <% } else {%>

                    <!-- If user is logged in -->

                    <h1>Welcome back, <%= user.getFirstname()%> <%=
user.getLastname()%></h1>

                    <p class="intro">Start searching for your match or
edit your profile.</p>

                    <a href="search.jsp" class="button">Search for other
users</a>

                    <% }%>

                </div>

                <div class="block">

```

```
</div>
</section>
```

```
</main>
```

```
<footer>
```

```
<small>All content &copy; Matrimony Site. All rights
reserved.</small>
```

```
</footer>
```

```
</div>
```

```
</body>
```

```
</html>
```

[Register](#)

```
<%--
```

```
Document    : register
```

```
Created on  : Oct 2, 2024, 7:28:05 PM
```

```
Author      : dsihle875@gmail.com
```

```
--%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8">
```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <meta name="description"

        content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />

    <meta name="author" content="Projects" />

    <meta name="keywords"

        content="Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />

    <meta name="theme-color" content="#ff4081" />


    <!-- Open Graph (for Facebook, LinkedIn) -->

    <meta property="og:title" content="Matrimony - Find Your Perfect
Match" />

    <meta property="og:type" content="website" />

    <meta property="og:url" content="https://www.matrimony.com" />

    <meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

    <meta property="og:description"

        content="Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />

    <meta property="og:locale" content="en_US" />


    <!-- Twitter Cards -->

    <meta name="twitter:card" content="summary_large_image" />

    <meta name="twitter:title" content="Matrimony - Your Trusted
Matchmaking Platform" />

    <meta name="twitter:description" content="Find your life partner on
Matrimony - where real connections are made." />

    <meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

    <meta name="twitter:site" content="@Matrimony" />

```



```

    <!-- Font Awesome -->

    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" />

    <!-- CSS -->

    <link rel="stylesheet" href="./assets/css/style.css">
    <link rel="stylesheet" href="./assets/css/form.css">

    <title>Matrimony</title>
</head>

<body>
    <div class="login-card">
        <div class="login-card-content">
            <div class="header">
                <h2>Matrimony</h2>
                <h3>Find your perfect match</h3>
                <p>Already have an account? <a href="login">Login</a></p>
            </div>
            <form class="form" method="post" action="register-user"
id="multiStepForm">
                <div class="progress-bar">
                    <div class="progress" id="progress"></div>
                </div>
                <%
                    String errorMessage = (String)
request.getAttribute("errorMessage");
                    if (errorMessage != null) {
                        %>
                        <div class="error-message">

```

```

        <p><%= errorMessage%></p>
    </div>
    <% }%>

    <input type="hidden" name="usertype" id="usertype"
value="User"/>

    <!-- Step 1: User Information -->
    <div class="form-step" id="step1">
        <div class="form-field firstname">
            <div class="icon">
                <i class="fas fa-user"></i>
            </div>
            <input type="text" id="firstname"
name="firstname" placeholder="First Name" required>
        </div>
        <div class="form-field lastname">
            <div class="icon">
                <i class="fas fa-user"></i>
            </div>
            <input type="text" id="lastname" name="lastname"
placeholder="Last Name" required>
        </div>
        <div class="form-field email">
            <div class="icon">
                <i class="fas fa-envelope"></i>
            </div>
            <input type="email" id="email" name="email"
placeholder="Email" required>
        </div>
        <div class="form-field password">
            <div class="icon">

```

```

        <i class="fas fa-lock"></i>
    </div>
    <input type="password" id="password"
name="password" placeholder="Password">
    </div>
    <button type="button" class="next-btn">Next</button>
</div>

<!-- Step 2: Contact Information -->
<div class="form-step" id="step2" style="display: none;">
    <div class="form-field gender">
        <div class="icon">
            <i class="fas fa-venus-mars"></i>
        </div>
        <select id="gender" name="gender" required>
            <option value="" disabled selected>Select
Gender</option>
            <option value="Male">Male</option>
            <option value="Female">Female</option>
            <option value="Other">Other</option>
        </select>
    </div>
    <div class="form-field contact_number">
        <div class="icon">
            <i class="fas fa-phone"></i>
        </div>
        <input type="tel" id="number" name="number"
placeholder="Contact Number"
            required>
    </div>
    <div class="form-field caste">

```

```

        <div class="icon">
            <i class="fas fa-users"></i>
        </div>
        <select id="caste" name="caste" required>
            <option value="" disabled selected>Select
your caste/ethnic group</option>
            <option value="Zulu">Zulu</option>
            <option value="Xhosa">Xhosa</option>
            <option value="Sotho">Sotho</option>
            <option value="Tswana">Tswana</option>
            <option value="Ndebele">Ndebele</option>
            <option value="Venda">Venda</option>
            <option value="Shona">Shona</option>
            <option value="Swazi">Swazi</option>
            <option value="Other">Other</option>
        </select>
    </div>
    <button type="button" class="prev-
btn">Previous</button>
    <button type="button" class="next-btn">Next</button>
</div>

```

```

<!-- Step 3: Personal Details -->
<div class="form-step" id="step3" style="display: none;">
    <div class="form-field date_of_birth">
        <p id="error-message" style="color: red;"></p>
        <div class="icon">
            <i class="fas fa-calendar-alt"></i>
        </div>
        <input type="date" id="date_of_birth"
name="date_of_birth" required onblur="validateAgeOnLeaveFocus()">
    </div>

```

```

</div>
<div class="form-field height">
  <div class="icon">
    <i class="fas fa-arrows-alt-v"></i>
  </div>
  <input type="number" id="height" name="height"
placeholder="Height (in cm)" required>
</div>
<div class="form-field marital_status">
  <div class="icon">
    <i class="fas fa-heart"></i>
  </div>
  <select id="status" name="status" required>
    <option value="" disabled selected>Select
Marital Status</option>
    <option value="Never Married">Never
Married</option>
    <option value="Divorced">Divorced</option>
    <option value="Widowed">Widowed</option>
  </select>
</div>
<button type="button" class="prev-
btn">Previous</button>
<button type="button" class="next-btn">Next</button>
</div>

<!-- Step 4: Location and Background -->
<div class="form-step" id="step4" style="display: none;">
  <div class="form-field country">
    <div class="icon">
      <i class="fas fa-globe"></i>

```

```

        </div>
        <input type="text" id="country" name="country"
placeholder="Country" required>
    </div>
    <div class="form-field city">
        <div class="icon">
            <i class="fas fa-map-marker-alt"></i>
        </div>
        <input type="text" id="city" name="city"
placeholder="City" required>
    </div>
    <div class="form-field mother_tongue">
        <div class="icon">
            <i class="fas fa-comments"></i>
        </div>
        <select id="language" name="language" required>
            <option value="" disabled selected>Select
your mother tongue</option>
            <option value="siSwati">siSwati</option>
            <option value="English">English</option>
            <option value="Zulu">Zulu</option>
            <option value="Afrikaans">Afrikaans</option>
            <option value="Tswana">Tswana</option>
            <option value="Ndebele">Ndebele</option>
            <option value="Other">Other</option>
        </select>
    </div>
    <div class="form-field religion">
        <div class="icon">
            <i class="fas fa-pray"></i>
        </div>
    </div>

```

```

        <select id="religion" name="religion" required>
            <option value="" disabled selected>Select
your religion</option>
            <option
value="Christianity">Christianity</option>
            <option value="Zionism">Zionism</option>
            <option
value="Catholicism">Catholicism</option>
            <option
value="Anglicanism">Anglicanism</option>
            <option value="Methodism">Methodism</option>
            <option value="Islam">Islam</option>
            <option value="Traditional African
Religions">Traditional African Religions</option>
            <option value="Hinduism">Hinduism</option>
            <option value="Other">Other</option>
        </select>
    </div>
    <!-- Payment Section -->
    <div class="form-field payment">
        <div class="icon">
            <i class="fas fa-credit-card"></i>
        </div>
        <select id="payment_method"
name="payment_method">
            <option value="" disabled selected>Select
Payment Method</option>
            <option value="Credit Card">Credit
Card</option>
            <option value="Debit Card">Debit
Card</option>
            <option value="PayPal">PayPal</option>

```

```

        <option value="Bank Transfer">Bank
Transfer</option>

        </select>
    </div>

    <div class="form-field payment_amount">
        <div class="icon">
            <i class="fas fa-dollar-sign"></i>
        </div>
        <select id="payment_amount"
name="payment_amount">
            <option value="" disabled selected>Select
Payment Amount</option>
            <option value="50">Monthly - $50</option>
            <option value="500">Yearly - $500</option>
        </select>
    </div>
    <button type="submit">Submit</button>
    <div class="multi-btn">
        <button type="button" class="prev-
btn">Previous</button>
    </div>
</div>
</form>
</div>
</div>

<!-- JavaScript -->
<script src="./assets/js/multi-step-form.js"></script>
<script>
    function validateAgeOnLeaveFocus() {

```



```

        var dob = document.getElementById('date_of_birth').value;
        var today = new Date();
        var birthDate = new Date(dob);
        var age = today.getFullYear() - birthDate.getFullYear();
        var monthDiff = today.getMonth() - birthDate.getMonth();

        if (monthDiff < 0 || (monthDiff === 0 && today.getDate() <
        birthDate.getDate())) {
            age--;
        }

        if (age < 18) {
            document.getElementById('error-message').textContent =
            "You must be at least 18 years old to sign up.";
        } else {
            document.getElementById('error-message').textContent =
            ""; // Clear error if age is valid
        }
    }
}

```

</script>

</body>

</html>

[Login](#)

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html lang="en">

<head>

```

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-
scale=1.0">

<meta name="description"
    content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />
<meta name="author" content=" Projects" />
<meta name="keywords"
    content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />
<meta name="theme-color" content="#ff4081" />

<!-- Open Graph (for Facebook, LinkedIn) -->
<meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />
<meta property="og:type" content="website" />
<meta property="og:url" content="https://www.matrimony.com" />
<meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
<meta property="og:description"
    content=" Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />
<meta property="og:locale" content="en_US" />

<!-- Twitter Cards -->
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:title" content=" Matrimony - Your Trusted
Matchmaking Platform" />
<meta name="twitter:description"
    content="Find your life partner on Matrimony - where real
connections are made." />

```

```

        <meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
        <meta name="twitter:site" content="@Matrimony" />

        <!-- fontawesome -->

        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" />

        <!-- css -->

        <link rel="stylesheet" href="./assets/css/style.css">
        <link rel="stylesheet" href="./assets/css/form.css">

        <title>Matrimony</title>
</head>

<body>
    <div class="login-card">
        <div class="login-card-content">
            <div class="header">
                <div class="logo">
                    <div></div>
                </div>
                <h2>Matri<span class="highlight">mony</span></h2>
                <h3>Find you your perfect match</h3>
            </div>
            <form class="form" method="POST" action="login-user">

                <%

```

```

        String errorMessage = (String)
request.getAttribute("errorMessage");
        if (errorMessage != null) {
%>
<div class="error-message">
    <p><%= errorMessage%></p>
</div>
<% }%>
<div class="form-field email">
    <div class="icon">
        <i class="fas fa-envelope"></i>
    </div>
    <input type="email" id="email" name="email"
placeholder="Email" required>
</div>

<div class="form-field password">
    <div class="icon">
        <i class="fas fa-lock"></i>
    </div>
    <input type="password" id="password" name="password"
placeholder="Password">
</div>

<button type="submit">
    Login
</button>
<div>
    Don't have an account? <a href="register">Sign Up
Now</a>
</div>

```

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

</html>
```

Search

```
<%--
```

```
    Document    : search
```

```
    Created on  : Oct 4, 2024, 1:07:55 PM
```

```
    Author      : dsihl
```

```
--%>
```

```
<%@page import="com.matrimony.models.User"%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%
```

```
    // Get the current session user (if available)
```

```
    User user = (User) session.getAttribute("user");
```

```
%>
```

```
<!DOCTYPE html>
```

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

```
    <head>
```

```
        <meta charset="UTF-8">
```

```
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```
        <meta name="description"
```

```
            content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />
```

```

<meta name="author" content=" Projects" />

<meta name="keywords"
    content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />

<meta name="theme-color" content="#ff4081" />

<!-- Open Graph (for Facebook, LinkedIn) -->
<meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />
<meta property="og:type" content="website" />
<meta property="og:url" content="https://www.matrimony.com" />
<meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
<meta property="og:description"
    content=" Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />
<meta property="og:locale" content="en_US" />

<!-- Twitter Cards -->
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:title" content=" Matrimony - Your Trusted
Matchmaking Platform" />
<meta name="twitter:description"
    content="Find your life partner on Matrimony - where real
connections are made." />
<meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
<meta name="twitter:site" content="@Matrimony" />

<!-- Font Awesome -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" />

```

```

<!-- css -->
<link rel="stylesheet" href="./assets/css/style.css">
<link rel="stylesheet" href="./assets/css/search.css">
<title>Matrinomy Site</title>
</head>
<body>
  <header class="site-header">
    <nav>
      <a href="user"><strong>MATRIMONY</strong></a> |
      <a href="javascript:void(0)" class="active-link">Search</a>
      <a href="about">About Us</a>
      <a href="browse">Looking For</a>
      <a href="services">Services</a>
      <a href="contact">Contact Us</a>
    </nav>

    <div class="nav-btn">
      <% if (user == null) { %>
        <!-- If user is NOT logged in -->
        <p><a href="login.jsp" class="login-btn">Have an
account?</a></p>
        <p><a href="register.jsp" class="signup-btn">Sign up</a></p>
      <% } else { %>
        <!-- If user is logged in -->
        <p><a href="user.jsp" class="login-btn">View Profile?</a></p>
        <p><a href="logout" class="signup-btn">Logout</a></p>
      <% }%>
    </div>
  </header>

```

```

<main>
  <div class="search-box">
    <h2 style="text-align:center;">Search by Caste</h2>
    <form action="searchresult" method="post" class="search-
caste">

      <div class="form-field caste">
        <div class="icon">
          <i class="fas fa-users"></i>
        </div>
        <select id="caste" name="caste" required>
          <option value="" disabled selected>Select your
caste/ethnic group</option>
          <option value="Zulu">Zulu</option>
          <option value="Xhosa">Xhosa</option>
          <option value="Sotho">Sotho</option>
          <option value="Tswana">Tswana</option>
          <option value="Ndebele">Ndebele</option>
          <option value="Venda">Venda</option>
          <option value="Shona">Shona</option>
          <option value="Swazi">Swazi</option>
          <option value="Other">Other</option>
        </select>
      </div>
      <button type="submit">
        Search Caste <i class="fas fa-search"></i>
      </button>
    </form>
    <h2 style="text-align:center;">Search by Member ID</h2>
    <form action="searchresultid" method="post" class="search-
caste">

      <div class="form-field caste">

```



```

        <div class="icon">
            <i class="fas fa-id-card"></i>
        </div>
        <input type="number" name="member_id" id="member_id"
placeholder="Member ID"/>
    </div>
    <button type="submit">
        Search ID <i class="fas fa-search"></i>
    </button>
</form>
</div>
</main>
</body>
</html>

```

Search result

```
<%--
```

```
    Document    : search-result
```

```
    Created on  : Oct 4, 2024, 1:27:34 PM
```

```
    Author      : dsihle875@gmail.com
```

```
--%>
```

```
<%@page import="com.matrimony.stateful.MatrimonySessionBeanLocal"%>
```

```
<%@page import="java.util.Base64"%>
```

```
<%@page import="java.util.List"%>
```

```
<%@page import="com.matrimony.models.User"%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%
```

```
    List<User> casteList = (List<User>) request.getAttribute("casteList");
```

```
    User loggedInUser = (User) session.getAttribute("user");
```

```

    MatrimonySessionBeanLocal bean = (MatrimonySessionBeanLocal)
session.getAttribute("matrimonySessionBean");
%>

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta name="viewport" content="width=device-width, initial-
scale=1.0">

        <meta name="description"

            content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />

        <meta name="author" content=" Projects" />

        <meta name="keywords"

            content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />

        <meta name="theme-color" content="#ff4081" />

        <!-- Open Graph (for Facebook, LinkedIn) -->

        <meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />

        <meta property="og:type" content="website" />

        <meta property="og:url" content="https://www.matrimony.com" />

        <meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

        <meta property="og:description"

            content=" Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />

        <meta property="og:locale" content="en_US" />

```

```

<!-- Twitter Cards -->

<meta name="twitter:card" content="summary_large_image" />

<meta name="twitter:title" content=" Matrimony - Your Trusted
Matchmaking Platform" />

<meta name="twitter:description"
        content="Find your life partner on Matrimony - where real
connections are made." />

<meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

<meta name="twitter:site" content="@Matrimony" />


<!-- Font Awesome -->

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" />


<!-- css -->

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

<link rel="stylesheet" href="./assets/css/search-result.css">

<title>Matrimony Site</title>

</head>

<body>

    <header class="site-header">

        <nav>

            <a href="user"><strong>MATRIMONY</strong></a> |

            <a href="search">Search</a>

            <a href="about">About Us</a>

            <a href="browse">Browse</a>

            <a href="services">Services</a>

            <a href="contact">Contact Us</a>

```

```

</nav>

<div class="nav-btn">
    <% if (loggedInUser == null) { %>
        <!-- If user is NOT logged in -->
        <p><a href="login.jsp" class="login-btn">Have an
account?</a></p>
        <p><a href="register.jsp" class="signup-btn">Sign up</a></p>
        <% } else { %>

        <!-- If user is logged in -->
        <p><a href="user.jsp" class="login-btn">View Profile?</a></p>
        <p><a href="logout" class="signup-btn">Logout</a></p>
        <% }%>
    </div>
</header>
<main>
    <section>
        <% if (casteList != null && !casteList.isEmpty()) {%>
            <p>Caste Search Succesful Find The Perfect One For
Yourself</p>
            <% for (User user : casteList) {%>
                <% if (loggedInUser != null) {%>

                <%
                    boolean foundProfiles = false;
                    String oppositeGender = (user != null &&
"male".equalsIgnoreCase(loggedInUser.getGender())) ? "female" : "male";

                    if (oppositeGender.equalsIgnoreCase(user.getGender())) {
                        foundProfiles = true;

```

```

%>
<div class="search-result">
    <div class="profile-photo">
        <%
user.getPhoto();
                                byte[] photoBytes =
                                String base64Image =
Base64.getEncoder().encodeToString(photoBytes);
                                %>

                                <% if (!base64Image.equals("")) {%>
                                
                                <% } else { %>
                                
                                <% }%>
        </div>
    <div class="user-details">
        <div class="member-id">
            <%= user.getGender()%> #<%= user.getUser_id()%>
        </div>
        <div class="member-details">
            <p>Name <span><%= user.getFirstname() + " " +
user.getLastname()%></span></p>
            <p>Age, Height <span><%
                                // Get the date of birth from the User object
as java.sql.Date
                                java.sql.Date dob = user.getDate_of_birth();

                                if (dob != null) {

```

```

        java.time.LocalDate dobLocalDate =
dob.toLocalDate();

        // Get today's date
        java.time.LocalDate today =
java.time.LocalDate.now();

        // Calculate the age
        int age =
java.time.Period.between(dobLocalDate, today).getYears();

        // Display the calculated age
        out.println(age);
    }

    %> Yrs, <%=
user.getHeight()%>cm</span></p>
        <p>Religion, Caste <span><%= user.getReligion()%>
: <%= user.getCaste()%></span></p>
        <p>Mother Tongue <span><%=
user.getMother_tongue()%></span></p>
        <p>Phone Number: <a href="tel: +<%=
user.getContact_number()%>"><%= user.getContact_number()%></a></p>
        <p>Email: <a href="mailto:<%=
user.getEmail()%>"><%= user.getEmail()%></a></p>
    </div>
</div>
    <div class="user-buttons">
        <form action="like" method="POST" id="likeForm">

            <input type="hidden" name="profileUserId"
value="<%= user.getUser_id()%>">

            <input type="hidden" name="likingUserId"
value="<%= loggedInUser.getUser_id()%>">

```

```

        <%
            if (!bean.checkIfLiked(user.getUser_id(),
loggedInUser.getUser_id())) {
                %>
                <button class="likeButton" id="likebtn" data-
profile-id="<%= user.getUser_id()%>" data-liking-id="<%=
loggedInUser.getUser_id()%>">
                    <i class="fas fa-heart"></i> Like
                </button>
            <%
            } else {
                %>

                <button class="likeButton" type="disabled">
                    <i class="fas fa-heart"></i> Liked
                </button>
            <%}%>

        </form>

        <button onclick="window.location.href =
'viewprofile?id=<%= user.getUser_id()%>'"><i class="fas fa-user"></i> View
Profile</button>

    </div>
</div>

<%}%>

<%
} else { %>

```

```

<div class="search-result">
    <div class="profile-photo">
        <%
            byte[] photoBytes = user.getPhoto();
            String base64Image =
Base64.getEncoder().encodeToString(photoBytes);
        %>

        <% if (!base64Image.equals("")) {%>
            
        %> } else { %>
            
        %> }%>
    </div>
    <div class="user-details">
        <div class="member-id">
            <%= user.getGender()%> #<%= user.getUser_id()%>
        </div>
        <div class="member-details">
            <p>Name <span><%= user.getFirstname() + " " +
user.getLastname()%></span></p>
            <p>Age, Height <span><%
                // Get the date of birth from the User object
                as java.sql.Date
                java.sql.Date dob = user.getDate_of_birth();

                if (dob != null) {
                    java.time.LocalDate dobLocalDate =
                    dob.toLocalDate();

```



```

        // Get today's date
        java.time.LocalDate today =
java.time.LocalDate.now();

        // Calculate the age
        int age =
java.time.Period.between(dobLocalDate, today).getYears();

        // Display the calculated age
        out.println(age);
    }

    %> Yrs, <%=
user.getHeight()%>cm</span></p>
        <p>Religion, Caste <span><%= user.getReligion()%>
: <%= user.getCaste()%></span></p>
        <p>Mother Tongue <span><%=
user.getMother_tongue()%></span></p>
        <p>Phone Number: <i class='fas fa-lock'></i> Sign
up to view</p>
        <p>Email: <i class='fas fa-lock'></i> Sign up to
view</p>
    </div>

</div>
<div class="user-buttons">
    <button><i class='fas fa-lock'></i></button>
    <button><i class='fas fa-lock'></i></button>
</div>
</div>
<% }
    } %>
<% } else { %>

```

```

        <p>No users found for the provided search criteria.</p>
        <% }%>
    </section>
</main>
</body>

<script>
    document.querySelectorAll(".likeButton").forEach(button => {
        button.addEventListener("click", function (e) {
            e.preventDefault();

            const profileUserId = this.getAttribute("data-profile-id");
            const likingUserId = this.getAttribute("data-liking-id");

            const xhr = new XMLHttpRequest();
            xhr.open("POST", "like", true);
            xhr.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");

            xhr.onload = function () {
                if (xhr.status === 200) {
                    const response = JSON.parse(xhr.responseText);
                    if (response.status === "success") {
                        alert("You liked the profile!");
                        button.innerHTML = '<i class="fas fa-heart"></i>
Liked';
                        button.setAttribute("disabled", true);
                    } else {
                        alert("An error occurred while liking the
profile.");
                    }
                }
            }
        });
    });

```

```

        }
    };
    xhr.send("profileUserId=" + profileUserId + "&likingUserId="
+ likingUserId);
    });
});
</script>

```

```
</html>
```

User

```
<%--
```

```
Document    : user
```

```
Created on  : Oct 3, 2024, 10:25:23 AM
```

```
Author      : dsihle875@gmail.com
```

```
--%>
```

```
<%@page import="java.util.Base64"%>
```

```
<%@page import="java.time.LocalDate"%>
```

```
<%@page import="java.util.List"%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%@ page import="com.matrimony.models.User" %>
```

```
<%@ page import="jakarta.servlet.http.HttpSession" %>
```

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8">
```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <meta name="description"
        content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />
    <meta name="author" content=" Projects" />
    <meta name="keywords"
        content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />
    <meta name="theme-color" content="#ff4081" />

    <!-- Open Graph (for Facebook, LinkedIn) -->
    <meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />
    <meta property="og:type" content="website" />
    <meta property="og:url" content="https://www.matrimony.com" />
    <meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
    <meta property="og:description"
        content=" Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />
    <meta property="og:locale" content="en_US" />

    <!-- Twitter Cards -->
    <meta name="twitter:card" content="summary_large_image" />
    <meta name="twitter:title" content=" Matrimony - Your Trusted
Matchmaking Platform" />
    <meta name="twitter:description"
        content="Find your life partner on Matrimony - where real
connections are made." />
    <meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

```

```

<meta name="twitter:site" content="@Matrimony" />

<!-- Font Awesome -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" />

<!-- css -->
<link rel="stylesheet" href="./assets/css/style.css">
<link rel="stylesheet" href="./assets/css/user.css">

<style>
    select
    {
        border: 1px solid #000;
        text-align: center;
        width: 100%;
        border-radius: 16px;
        height: 36px;
        background: rgba(255, 255, 255, 0.1);
        color: black;
        outline: none;
        transition: all 0.2s;
        font-family: "Nunito", system-ui;
    }

    .modal {
        display: none; /* Hidden by default */
        position: fixed; /* Stay in place */
        z-index: 1; /* Sit on top */

```

```

    left: 0;
    top: 0;
    width: 100%; /* Full width */
    height: 100%; /* Full height */
    background-color: rgb(0,0,0); /* Fallback color */
    background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
    overflow: auto; /* Enable scroll if needed */
}

/* Modal Content */
.modal-content {
    background-color: #ffe4ec ;
    margin: 15% auto; /* 15% from the top and centered */
    padding: 20px;
    border: 1px solid #888;
    width: 50%; /* Could be more or less, depending on screen
size */

    box-shadow: 0 5px 15px rgba(0,0,0,0.3);
    position: relative;
}

/* The Close Button */
.close {
    color: #aaa;
    float: right;
    font-size: 28px;
    font-weight: bold;
    cursor: pointer;
    position: absolute;
    top: 10px;

```

```

        right: 20px;
    }

    .close:hover, .close:focus {
        color: black;
        text-decoration: none;
        cursor: pointer;
    }
</style>

<!-- test -->
<script src="assets/js/calculateAge.js"></script>
<title>Matrinomy Site</title>
</head>

<body>
    <div class="wrapper">
        <header class="site-header">
            <nav>
                <a href="user" class="active-link"><strong>MATRIMONY</strong></a> |
                <a href="search">Search</a>
                <a href="about">About Us</a>
                <a href="browse">Looking For</a>
                <a href="services">Services</a>
                <a href="contact">Contact Us</a>
            </nav>

            <div class="nav-btn">
                <p><a href="logout" class="signup-btn">Logout</a></p>

```

```

        </div>
    </header>

    <main>

        <%
            User user = (User) session.getAttribute("user");
            List<User> users = (List<User>)
session.getAttribute("users");
            String oppositeGender = (user != null &&
"male".equalsIgnoreCase(user.getGender())) ? "female" : "male";
            if (user == null || session.getAttribute("user") == null)
{
                response.sendRedirect("login.jsp");
            }
        %>

        <section class="home-page">
            <div class="block">
                <!-- profile card start -->
                <div class="layout">
                    <div class="profile">
                        <div class="profile__picture">
                            <a href="updatephotoroute">

                                <%
                                    byte[] photoBytes =
user.getPhoto();

                                    String base64Image =
Base64.getEncoder().encodeToString(photoBytes);
                                %>

```



```

                                <% if (!base64Image.equals("")) {%>
                                
                                <% } else { %>
                                
                                <% }%>

                                </a>
                                </div>
                                <div class="profile__header">
                                    <div class="profile__account">
                                        <h4 class="profile__username"><%=
user.getFirstname()%></h4>
                                        </div>

                                        <div class="profile__edit">
                                            <a class="profile__button"
href="editprofile?id=<%= user.getUser_id()%>">Edit Profile</a>
                                            <br>
                                            <a class="profile__button"
href="editdetailsroute?id=<%= user.getUser_id()%>">Edit Details</a>
                                        </div>

                                </div>
                                <div class="profile__stats">

                                    <div class="profile__stat">
                                        <div class="profile__icon
profile__icon--gold">

```

```

        <i class="fas fa-eye"></i>
    </div>

    <div class="profile__value"><%=
user.getViews()%></div>

    <div class="profile__key">Views</div>
</div>

<div class="profile__stat">
    <div class="profile__icon
profile__icon--blue">

        <i class="fas fa-signal"></i>
    </div>

    <div class="profile__value">1</div>
    <div
class="profile__key">Ranking</div>
    </div>

<div class="profile__stat">
    <div class="profile__icon
profile__icon--pink">

        <i class="fas fa-heart"></i>
    </div>

    <div class="profile__value"><%=
user.getLikes()%></div>

    <div class="profile__key">Likes</div>
</div>
</div>
</div>

```

```

        </div>
        <!-- profile card end -->
    </div>
    <div class="block">
        <div class="profile-details">
            <h5 class="profile-details__heading">Profile
Details</h5>
            <p class="profile-details__bio">
                Your name is <%= user.getFirstname()%> your
lastname is <%= user.getLastname()%>, you speak <%= user.getMother_tongue()%>
                your religion is <%= user.getReligion()%>
that is <%= user.getCaste()%> born <%= user.getDate_of_birth().toString()%>
            </p>
            <div class="profile-details__info">
                <div class="profile-details__item">
                    <i class="fas fa-map-marker-alt"></i>
                    <strong>Location: </strong> <%=
user.getCountry()%>
                </div>
                <div class="profile-details__item">
                    <i class="fas fa-envelope"></i>
                    <strong>Email: </strong> <%=
user.getEmail()%>
                </div>
                <div class="profile-details__item">
                    <i class="fas fa-calendar-alt"></i>
                    <strong>Joined: </strong> <%=
user.getCreatedAt().toString()%>
                </div>
                <div class="profile-details__item"
style="cursor: pointer;" onclick="openModal()">
                    <i class="fas fa-dollar-sign"></i>

```

```

                                <strong>Payment: </strong> <%=
session.getAttribute("timeLeft")%>
                                </div>
                                <div class="profile-details__item">
                                    <i class="fas fa-dollar-sign"></i>
                                    <strong>Update Password: </strong><p><a
href="password" class="signup-btn">Update Password</a></p>
                                </div>

                                <div id="paymentModal" class="modal">
                                    <!-- Modal content -->
                                    <div class="modal-content">
                                        <span class="close"
onclick="closeModal()">&times;</span>
                                        <form action="update-payment"
method="POST">
                                            <input type="hidden" name="id"
value="<%= user.getUser_id() %>">
                                            <div class="form-field payment">
                                                <div class="icon">
                                                    <i class="fas fa-credit-
card"></i>
                                                </div>
                                                <select id="payment_method"
name="payment_method">
                                                    <option value="" disabled
selected>Select Payment Method</option>
                                                    <option value="Credit
Card">Credit Card</option>
                                                    <option value="Debit
Card">Debit Card</option>
                                                    <option
value="PayPal">PayPal</option>

```

```

        <option value="Bank
Transfer">Bank Transfer</option>

    </select>
</div>


<div class="form-field
payment_amount">

    <div class="icon">
        <i class="fas fa-dollar-
sign"></i>

    </div>

    <select id="payment_amount"
name="payment_amount">

        <option value="" disabled
selected>Select Payment Amount</option>

        <option
value="50">Monthly - $50</option>

        <option
value="500">Yearly - $500</option>

    </select>
</div>

<button type="submit">Update
Payment</button>

</form>

</div>

</div>

</div>

</div>

</section>

<h2 style="text-align: center;">Your Info</h2>

```

D. R.


```

                                <strong>Designation: &nbsp; </strong> <%=
user.getDesignation()%>
                                </div>
                                <%> else {<%>
                                <div class="profile-details__item">
                                    <i class="fas fa-user-tie"></i>
                                    <strong>Designation: &nbsp; </strong> No
designations
                                </div>
                                <%}%>

                                <% if (user.getHobbies() != null) {<%>
                                <div class="profile-details__item">
                                    <i class="fas fa-palette"></i>
                                    <strong>Hobbies: &nbsp; </strong> <%=
user.getHobbies()%>
                                </div>
                                <%> else {<%>
                                <div class="profile-details__item">
                                    <i class="fas fa-palette"></i>
                                    <strong>Hobbies: &nbsp; </strong> No hobbies
                                </div>
                                <%}%>

                                <% if (user.getFavouriteGenre() != null) {<%>
                                <div class="profile-details__item">
                                    <i class="fas fa-music"></i>
                                    <strong>Favourite Genre: &nbsp; </strong> <%=
user.getFavouriteGenre()%>
                                </div>
                                <%> else {<%>

```


Favourite Cuisines

Favourite Cuisine: No

</div>

<%}%>

<% if (user.getFavouriteBooks() != null) {%>

<div class="profile-details__item">

<i class="fas fa-book"></i>

Favourite Books: <%=
user.getFavouriteBooks()%>

</div>

<%} else {%>

<div class="profile-details__item">

<i class="fas fa-book"></i>

Favourite Books

Favourite Books: No

</div>

<%}%>

<% if (user.getFavouritePlaces() != null) {%>

<div class="profile-details__item">

<i class="fas fa-globe"></i>

Favourite Place: <%=
user.getFavouritePlaces()%>

</div>

<%} else {%>

<div class="profile-details__item">

<i class="fas fa-globe"></i>

Favourite Place

Favourite Place: No

</div>

```

        <%}%>

    </div>
</section>

<h2 style="text-align: center;">Similar users</h2>
<!-- section 2 -->
<section class="section2">

    <% if (users != null && !users.isEmpty()) { %>
    <%
        boolean foundProfiles = false;
        for (User userList : users) {
            byte[] photoByteUsers = userList.getPhoto();
            String base64ImageUsers =
Base64.getEncoder().encodeToString(photoByteUsers);

            // Show profiles of the opposite gender
            if
(oppositeGender.equalsIgnoreCase(userList.getGender())) {
                foundProfiles = true;
            }
        }
    <%>
        <div class="user-card-suggestion">
            <div class="profile-photo">
                <!-- Dynamically display user photo, fallback to
default if no photo is available -->
                <% if (!base64ImageUsers.isEmpty()) {%>
                    
                <% } else { %>

```

```

        <% } %>
    </div>
    <div class="profile-info">
        <!-- Dynamically display user details -->
        <p class="age" id="age">
            <%
                // Get the date of birth from the User
                object as java.sql.Date
                java.sql.Date dob =
                userList.getDate_of_birth();

                if (dob != null) {
                    java.time.LocalDate dobLocalDate =
                dob.toLocalDate();

                    // Get today's date
                    java.time.LocalDate today =
                java.time.LocalDate.now();

                    // Calculate the age
                    int age =
                java.time.Period.between(dobLocalDate, today).getYears();

                    // Display the calculated age
                    out.println("<p>Age: " + age +
                "</p>");
                }
            %>
        </p>
    
```

```

                <p class="details"><%= userList.getFirstname()%>
<%= userList.getLastname()%></p>

                <p style="margin-top: 10px;"><a
href="viewprofile?id=<%= userList.getUser_id()%>" class="view-profile">View
profile</a></p>

            </div>
        </div>
    <%
    }
    }
    // If no profiles of opposite gender are found, show
a message

        if (!foundProfiles) {
    %>
    <p>No profiles available for your preferences at the
moment.</p>

    <% } %>
    <% } else { %>
    <p>No user suggestions available at the moment.</p>
    <% }%>
    </section>
</main>

    <footer>
        <small>All content &copy; Matrimony Site. All rights
reserved.</small>
    </footer>
</div>

<script>
    // Get the modal
    var modal = document.getElementById("paymentModal");

```

```

// Function to open the modal
function openModal() {
    modal.style.display = "block";
}

// Function to close the modal
function closeModal() {
    modal.style.display = "none";
}

// Close the modal when the user clicks anywhere outside of it
window.onclick = function (event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>

</body>

</html>

```

Services

```
<%--
```

```
Document    : services
```

```
Created on  : Oct 23, 2024, 10:58:49 AM
```

```
Author      : dsihle875@gmail.com
```

```
--%>
```

```

<%@page import="java.util.Base64"%>
<%@page import="com.matrimony.models.UserAdd"%>
<%@page import="java.util.List"%>
<%@page import="com.matrimony.models.User"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    // Get the current session user (if available)
    User user = (User) session.getAttribute("user");
    List<UserAdd> adds = (List<UserAdd>) request.getAttribute("adds");
%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">

        <meta name="description"
            content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />
        <meta name="author" content=" Projects" />
        <meta name="keywords"
            content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />
        <meta name="theme-color" content="#ff4081" />

        <!-- Open Graph (for Facebook, LinkedIn) -->
        <meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />
        <meta property="og:type" content="website" />

```

```

    <meta property="og:url" content="https://www.matrimony.com" />

    <meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

    <meta property="og:description"
        content=" Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />

    <meta property="og:locale" content="en_US" />

<!-- Twitter Cards -->

    <meta name="twitter:card" content="summary_large_image" />

    <meta name="twitter:title" content=" Matrimony - Your Trusted
Matchmaking Platform" />

    <meta name="twitter:description"
        content="Find your life partner on Matrimony - where real
connections are made." />

    <meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

    <meta name="twitter:site" content="@Matrimony" />

<!-- css -->

<link rel="stylesheet" href="./assets/css/style.css">
<link rel="stylesheet" href="./assets/css/services.css">

<style>

    .blog-author h3::before, .blog-author--no-cover h3::before {

        background-size: cover;
        border-radius: 50%;
        content: " ";
        display: inline-block;
        height: 32px;

```



```

        margin-right: 0.5rem;
        position: relative;
        top: 8px;
        width: 32px;
    }

    input
    {
        border: 1px solid #000;
        text-align: center;
        width: 100%;
        border-radius: 16px;
        height: 36px;
        background: rgba(255, 255, 255, 0.1);
        color: black;
        outline: none;
        transition: all 0.2s;
        font-family: "Nunito", system-ui;
    }

    textarea{
        border: 1px solid #000;
        width: 100%;
        border-radius: 16px;
        background: rgba(255, 255, 255, 0.1);
        color: black;
        outline: none;
        transition: all 0.2s;
        font-family: "Nunito", system-ui;
    }

```

```

.modal {
    display: none; /* Hidden by default */
    position: fixed; /* Stay in place */
    z-index: 1; /* Sit on top */
    left: 0;
    top: 0;
    width: 100%; /* Full width */
    height: 100%; /* Full height */
    background-color: rgb(0,0,0); /* Fallback color */
    background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
    overflow: auto; /* Enable scroll if needed */
}

/* Modal Content */
.modal-content {
    background-color: #ffe4ec ;
    margin: 15% auto; /* 15% from the top and centered */
    padding: 20px;
    border: 1px solid #888;
    width: 50%; /* Could be more or less, depending on screen
size */

    box-shadow: 0 5px 15px rgba(0,0,0,0.3);
    position: relative;
}

/* The Close Button */
.close {
    color: #aaa;
    float: right;

```

```

        font-size: 28px;
        font-weight: bold;
        cursor: pointer;
        position: absolute;
        top: 10px;
        right: 20px;
    }

    .close:hover, .close:focus {
        color: black;
        text-decoration: none;
        cursor: pointer;
    }
</style>
<title>Matrinomy Site</title>
</head>
<body>
    <div class="wrapper">
        <header class="site-header">
            <nav>
                <a href="user"><strong>MATRIMONY</strong></a> |
                <a href="search">Search</a>
                <a href="about">About Us</a>
                <a href="browse">Looking For</a>
                <a href="javascript:void(0)" class="active-
link">Services</a>
                <a href="contact">Contact Us</a>
                <% if (user != null) { %>
                    <p class="signup-btn cursor:pointer;"
onclick="openModal1()">Add Blog</p>

```

```

        <div id="addModal" class="modal">
            <!-- Modal content -->
            <div class="modal-content">
                <span class="close"
onclick="closeModal1()">&times;</span>
                <form action="insert-add" method="POST">
                    <input type="hidden" value="<%=
user.getUser_id()%>" name="id">
                    <input type="text" name="title"
placeholder="Title">
                    <textarea name="body">

                    </textarea>
                    <button type="submit">Submit</button>
                    <button type="reset">Reset</button>
                </form>
            </div>
        </div>
    <% }%>
</nav>

<div class="nav-btn">
    <% if (user == null) { %>
    <!-- If user is NOT logged in -->
    <p><a href="login.jsp" class="login-btn">Have an
account?</a></p>
    <p><a href="register.jsp" class="signup-btn">Sign
up</a></p>
    <% } else { %>
    <!-- If user is logged in -->
    <p><a href="user.jsp" class="login-btn">View
Profile?</a></p>

```

```

        <p><a href="logout" class="signup-btn">Logout</a></p>
        <% }%>
    </div>

    <div class="mobile-menu-icon"
onclick="toggleMobileMenu()">≡</div>
</header>

<main>

    <% for (UserAdd add : adds) { %>
    <% if (!add.equals(null)) {
        byte[] photoByteUsers = add.getPhoto();
        String base64ImageUser =
Base64.getEncoder().encodeToString(photoByteUsers);
        String backgroundImage = !base64ImageUser.isEmpty()
            ? "data:image/png;base64," + base64ImageUser
            : "./assets/images/profile.png";
    %>
    <head>
        <script defer>
            const style = document.createElement('style');
            style.innerHTML = `
                .blog-author h3::before, .blog-author--no-cover
h3::before {
                    background: url("<%=backgroundImage%>");
                }
            `;
            document.head.appendChild(style);
        </script>
    </head>
    <div class="blog-container">

```

```

<div class="blog-header">
    <div class="blog-author--no-cover">
        <h3><%= add.getName()%></h3>
    </div>
</div>

<div class="blog-body">
    <div class="blog-title">
        <h1><%= add.getTitle()%></h1>
    </div>
    <div class="blog-summary">
        <p>
            <%= add.getBody()%>
        </p>
    </div>
</div>

<div class="blog-footer">
    <ul>
        <% if (user != null) {%>
        <% if (add.getName().equals(user.getFirstname()))
{%>
            <li class="published-date"><p style="color:white;
cursor:pointer;" onclick="openModal()">Edit</p></li>
            <div id="editModal" class="modal">
                <!-- Modal content -->
                <div class="modal-content">
                    <span class="close"
onclick="closeModal()">&times;</span>
                    <form action="edit-add" method="POST">

```

```

                                <input type="hidden" value="<%=
add.getUserId()%>" name="id">
                                <input type="text" value="<%=
add.getTitle()%>" name="title" placeholder="Title">
                                <textarea name="body">
                                    <%= add.getBody()%>
                                </textarea>
                                <button type="submit">Submit</button>
                                <button type="reset">Reset</button>
                                </form>
                            </div>
                        </div>
                    <% } else { %>

                    <% } %>
                    <% } %>
                </ul>
            </div>

        </div>
        <%= } else { %>
        <div class="blog-container">
            <h1>no adds to view</h1>
        </div>
        <% } %>
        <%=}%>
    </main>
    <footer>
        <small>All content &copy; Matrimony Site. All rights
reserved.</small>

```

```
</footer>
</div>
```

```
<script>
    // Get the modal
    var modal = document.getElementById("editModal");
    var modal1 = document.getElementById("addModal");

    // Function to open the modal
    function openModal() {
        modal.style.display = "block";
    }

    function openModal1() {
        modal1.style.display = "block";
    }

    // Function to close the modal
    function closeModal() {
        modal.style.display = "none";
    }

    function closeModal1() {
        modal1.style.display = "none";
    }

    // Close the modal when the user clicks anywhere outside of it
    window.onclick = function (event) {
        if (event.target == modal) {
```



```

        modal.style.display = "none";
    }

    if (event.target == modal1){
        modal1.style.display = "none";
    }
}
</script>
</body>
</html>

```

Edit Details

```
<!--
```

```
Document    : edit-details
```

```
Created on  : Oct 14, 2024, 11:21:59 AM
```

```
Author      : dsihl
```

```
--%>
```

```

<%@page import="com.matrimony.models.User"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.Base64"%>
<%
    // Get the current session user (if available)
    User user = (User) request.getAttribute("user");
%>
<!DOCTYPE html>
<html lang="en">

```

```
<head>
```

```
<meta charset="UTF-8">
```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <meta name="description"
        content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />
    <meta name="author" content=" Projects" />
    <meta name="keywords"
        content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />
    <meta name="theme-color" content="#ff4081" />

    <!-- Open Graph (for Facebook, LinkedIn) -->
    <meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />
    <meta property="og:type" content="website" />
    <meta property="og:url" content="https://www.matrimony.com" />
    <meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
    <meta property="og:description"
        content=" Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />
    <meta property="og:locale" content="en_US" />

    <!-- Twitter Cards -->
    <meta name="twitter:card" content="summary_large_image" />
    <meta name="twitter:title" content=" Matrimony - Your Trusted
Matchmaking Platform" />
    <meta name="twitter:description"
        content="Find your life partner on Matrimony - where real
connections are made." />
    <meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />

```

```

<meta name="twitter:site" content="@Matrimony" />

<!-- fontawesome -->
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" />

<!-- css -->
<link rel="stylesheet" href="./assets/css/style.css">
<link rel="stylesheet" href="./assets/css/form.css">

<style>
    .form{
        width: 600px;

    }

</style>

<title>Matrimony</title>
</head>
<body>
    <section>
        <div class="login-card">
            <div class="login-card-content">
                <div class="header">
                    <h2>Matri<span class="highlight">mony</span></h2>
                    <h3>Tell Us More About Yourself</h3>
                </div>
                <form action="editdetails" method="POST" class="form">

```

```

<!-- Personal Section -->
<fieldset id="personalSection">
    <legend>Personal Information</legend>
    <input type="hidden" value="<%= user.getUser_id()
%>" name="user_id" id="user_id" >
    <div class="form-field">
        <div class="icon">
            <i class="fas fa-info-circle"></i>
        </div>
        <textarea id="aboutSelf" name="aboutSelf"
placeholder="About Self" rows="4" required>
            <%= user.getAboutSelf() %>
        </textarea>
    </div>
    <div class="form-field">
        <div class="icon">
            <i class="fas fa-users"></i>
        </div>
        <textarea id="familyDetails"
name="familyDetails" placeholder="Family Details" rows="4" required>
            <%= user.getFamilyDetails() %>
        </textarea>
    </div>
    <button type="button"
onclick="showNextSection('professionalSection')">Next</button>
</fieldset>

<!-- Professional Section -->
<fieldset id="professionalSection" style="display:
none;">

```

```

<legend>Professional Information</legend>
<div class="form-field">
    <div class="icon">
        <i class="fas fa-graduation-cap"></i>
    </div>
    <input type="text" id="qualification"
name="qualification" value="<%= user.getQualification() %>"
placeholder="Qualification" required>
    </div>
<div class="form-field">
    <div class="icon">
        <i class="fas fa-building"></i>
    </div>
    <input type="text" id="workingAt"
name="workingAt" value="<%= user.getWorkingAt() %>" placeholder="Working At"
required>
    </div>
<div class="form-field">
    <div class="icon">
        <i class="fas fa-user-tie"></i>
    </div>
    <input type="text" id="designation"
name="designation" value="<%= user.getDesignation() %>"
placeholder="Designation" required>
    </div>
    <button type="button"
onclick="showNextSection('personalSection')">Previous</button>
    <button type="button"
onclick="showNextSection('likingsSection')">Next</button>
</fieldset>

<!-- Likings Section -->

```

```

<fieldset id="likingsSection" style="display: none;">
    <legend>Likings</legend>
    <div class="form-field">
        <div class="icon">
            <i class="fas fa-palette"></i>
        </div>
        <input type="text" id="hobbies"
name="hobbies" value="<%= user.getHobbies() %>" placeholder="Hobbies"
required>
    </div>
    <div class="form-field">
        <div class="icon">
            <i class="fas fa-music"></i>
        </div>
        <input type="text" id="favoriteMusic"
name="favoriteMusic" value="<%= user.getFavouriteGenre() %>"
placeholder="Favorite Genre" required>
    </div>
    <div class="form-field">
        <div class="icon">
            <i class="fas fa-film"></i>
        </div>
        <input type="text" id="favoriteMovies"
name="favoriteMovies" value="<%= user.getFavouriteMovies() %>"
placeholder="Favorite Movies" required>
    </div>
    <div class="form-field">
        <div class="icon">
            <i class="fas fa-utensils"></i>
        </div>

```

```

        <input type="text" id="favoriteCuisines"
name="favoriteCuisines" value="<%= user.getFavouriteCuisine() %>"
placeholder="Favorite Cuisines" required>

        </div>

        <div class="form-field">

            <div class="icon">

                <i class="fas fa-book"></i>

            </div>

            <input type="text" id="favoriteBooks"
name="favoriteBooks" value="<%= user.getFavouriteBooks() %>"
placeholder="Favorite Books" required>

            </div>

            <div class="form-field">

                <div class="icon">

                    <i class="fas fa-globe"></i>

                </div>

                <input type="text" id="favoritePlace"
name="favoritePlace" value="<%= user.getFavouritePlaces() %>"
placeholder="Favorite Place" required>

                </div>

                <button type="button"
onclick="showNextSection('professionalSection')">Previous</button>

                <button type="submit">Submit</button>

            </fieldset>

        </form>

    </div>

</div>

</section>

<script>

    function showNextSection(sectionId) {

```

```

        const sections = ['personalSection', 'professionalSection',
        'likingsSection'];

        sections.forEach(id =>
document.getElementById(id).style.display = 'none');

        document.getElementById(sectionId).style.display = 'block';

    }

</script>

</body>

</html>

```

Update Profile

```
<!--
```

```
    Document    : edit-profile
```

```
    Created on  : Oct 9, 2024, 7:08:16 PM
```

```
    Author      : dsihle875@gmail.com
```

```
--%>
```

```
<%@page import="java.util.Base64"%>
```

```
<%@page import="com.matrimony.models.User"%>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%
```

```
    // Get the current session user (if available)
```

```
    User user = (User) request.getAttribute("user");
```

```
%>
```

```
<!DOCTYPE html>
```

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

```
    <head>
```

```
        <meta charset="UTF-8">
```

```
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```



```

<meta name="description"
      content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />
<meta name="author" content=" Projects" />
<meta name="keywords"
      content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />
<meta name="theme-color" content="#ff4081" />

<!-- Open Graph (for Facebook, LinkedIn) -->
<meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />
<meta property="og:type" content="website" />
<meta property="og:url" content="https://www.matrimony.com" />
<meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
<meta property="og:description"
      content=" Matrimony is your trusted platform to connect with
your ideal life partner. Join today to find your match." />
<meta property="og:locale" content="en_US" />

<!-- Twitter Cards -->
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:title" content=" Matrimony - Your Trusted
Matchmaking Platform" />
<meta name="twitter:description"
      content="Find your life partner on Matrimony - where real
connections are made." />
<meta name="twitter:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
<meta name="twitter:site" content="@Matrimony" />

```

```

        <!-- fontawesome -->

        <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css" />

        <!-- css -->

        <link rel="stylesheet" href="./assets/css/style.css">
        <link rel="stylesheet" href="./assets/css/form.css">

        <title>Matrimony</title>
    </head>

    <body>
        <div class="login-card">
            <div class="login-card-content">
                <div class="header">
                    <div class="logo" >
                        <%
                            byte[] photoBytes = user.getPhoto();
                            String base64Image =
Base64.getEncoder().encodeToString(photoBytes);
                        %>

                        <% if (!base64Image.equals("")) {%>
                            
                        <% } else { %>
                            

```

```

        <% }%>
    </div>

    <h2>Matri<span class="highlight">mony</span></h2>
    <h3>Find you your perfect match</h3>
</div>

<form class="form" action="edit" method="POST">

    <input type="hidden" id="userId" name="userId" value="<%=
user.getUser_id()%>">

    <input type="hidden" id="userType" name="userType"
value="<%= user.getUser_type()%>">

    <div class="form-field firstname">
        <div class="icon">
            <i class="far fa-user"></i>
        </div>
        <input type="text" id="firstname" name="firstname"
placeholder="First Name" value="<%= user.getFirstname()%>" required>
    </div>

    <div class="form-field lastname">
        <div class="icon">
            <i class="fas fa-user"></i>
        </div>
        <input type="text" id="lastname" name="lastname"
placeholder="Last Name" value="<%= user.getLastname()%>" required>
    </div>

    <div class="form-field email">
        <div class="icon">
            <i class="fas fa-envelope"></i>

```

```

        </div>
        <input type="email" id="email" name="email"
placeholder="Email" value="<%= user.getEmail()%>" required>
    </div>

```

```

<div class="form-field gender">
    <div class="icon">
        <i class="fas fa-venus-mars"></i>
    </div>
    <select id="gender" name="gender" required>
        <option value="<%= user.getGender()%>"
selected><%= user.getGender()%></option>
        <option value="Male">Male</option>
        <option value="Female">Female</option>
        <option value="Other">Other</option>
    </select>
</div>

```

```

<div class="form-field date_of_birth">
    <div class="icon">
        <i class="fas fa-calendar-alt"></i>
    </div>
    <input type="date" id="date_of_birth"
name="date_of_birth" value="<%= user.getDate_of_birth()%>"required>
</div>

```

```

<div class="form-field height">
    <div class="icon">
        <i class="fas fa-arrows-alt-v"></i>
    </div>

```

```

        <input type="number" id="height" name="height"
placeholder="Height (in cm)" value="<%= user.getHeight()%>" required>
    </div>

    <div class="form-field marital_status">
        <div class="icon">
            <i class="fas fa-heart"></i>
        </div>
        <select id="marital_status" name="marital_status"
required>
            <option value="<%=
user.getMarital_status()%>"selected><%= user.getMarital_status()%></option>
            <option value="Never Married">Never
Married</option>
            <option value="Divorced">Divorced</option>
            <option value="Widowed">Widowed</option>
        </select>
    </div>

    <div class="form-field country">
        <div class="icon">
            <i class="fas fa-globe"></i>
        </div>
        <input type="text" id="country" name="country"
placeholder="Country" value="<%= user.getCountry()%>" required>
    </div>

    <div class="form-field city">
        <div class="icon">
            <i class="fas fa-map-marker-alt"></i>
        </div>

```

```

        <input type="text" id="city" name="city"
placeholder="City" value="<%= user.getCity()%>" required>
    </div>

    <div class="form-field contact_number">
        <div class="icon">
            <i class="fas fa-phone"></i>
        </div>
        <input type="tel" id="contact_number"
name="contact_number" placeholder="Contact Number" value="<%=
user.getContact_number()%>" required>
    </div>

    <div class="form-field mother_tongue">
        <div class="icon">
            <i class="fas fa-comments"></i>
        </div>
        <select id="mother_tongue" name="mother_tongue"
required>
            <option value="<%= user.getMother_tongue()%>"
selected><%= user.getMother_tongue()%></option>
            <option value="siSwati">siSwati</option>
            <option value="English">English</option>
            <option value="Zulu">Zulu</option>
            <option value="Afrikaans">Afrikaans</option>
            <option value="Tswana">Tswana</option>
            <option value="Ndebele">Ndebele</option>
            <option value="Other">Other</option>
        </select>
    </div>

```

```

<div class="form-field religion">
    <div class="icon">
        <i class="fas fa-pray"></i>
    </div>
    <select id="religion" name="religion" required>
        <option value="<%= user.getReligion()%>"
selected><%= user.getReligion()%></option>
        <option
value="Christianity">Christianity</option>
        <option value="Zionism">Zionism</option>
        <option value="Catholicism">Catholicism</option>
        <option value="Anglicanism">Anglicanism</option>
        <option value="Methodism">Methodism</option>
        <option value="Islam">Islam</option>
        <option value="Traditional African
Religions">Traditional African Religions</option>
        <option value="Hinduism">Hinduism</option>
        <option value="Other">Other</option>
    </select>
</div>

```

```

<div class="form-field caste">
    <div class="icon">
        <i class="fas fa-users"></i>
    </div>
    <select id="caste" name="caste" required>
        <option value="<%= user.getCaste()%>"
selected><%= user.getCaste()%></option>
        <option value="Zulu">Zulu</option>
        <option value="Xhosa">Xhosa</option>
        <option value="Sotho">Sotho</option>
    </select>
</div>

```

```

        <option value="Tswana">Tswana</option>
        <option value="Ndebele">Ndebele</option>
        <option value="Venda">Venda</option>
        <option value="Shona">Shona</option>
        <option value="Swazi">Swazi</option>
        <option value="Other">Other</option>
    </select>
</div>

<input type="hidden" name="user-type">

<button type="submit">
    Update Profile
</button>
</form>
</div>
</div>
</body>

</html>

```

Update Photo

```
<%--
```

```
    Document    : update-photo
```

```
    Created on  : Oct 5, 2024, 10:43:46 AM
```

```
    Author      : dsihle875@gmail.com
```

```
--%>
```

```
<%@page import="com.matrimony.models.User"%>
```



```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    // Get the current session user (if available)
    User user = (User) session.getAttribute("user");

%>
<!DOCTYPE html>
<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">

        <meta name="description"
            content="Find your perfect match on Matrimony - the leading
matrimony site for genuine connections." />
        <meta name="author" content=" Projects" />
        <meta name="keywords"
            content=" Matrimony, marriage, life partner, matrimony site,
find a match, wedding, matchmaking" />
        <meta name="theme-color" content="#ff4081" />

        <!-- Open Graph (for Facebook, LinkedIn) -->
        <meta property="og:title" content=" Matrimony - Find Your Perfect
Match" />
        <meta property="og:type" content="website" />
        <meta property="og:url" content="https://www.matrimony.com" />
        <meta property="og:image"
content="https://www.matrimony.com/images/logo/Logo.png" />
        <meta property="og:description"

```

```
        content=" Matrimony is your trusted platform to connect with  
your ideal life partner. Join today to find your match." />
```

```
    <meta property="og:locale" content="en_US" />
```

```
    <!-- Twitter Cards -->
```

```
    <meta name="twitter:card" content="summary_large_image" />
```

```
    <meta name="twitter:title" content=" Matrimony - Your Trusted  
Matchmaking Platform" />
```

```
    <meta name="twitter:description"
```

```
        content="Find your life partner on Matrimony - where real  
connections are made." />
```

```
    <meta name="twitter:image"  
content="https://www.matrimony.com/images/logo/Logo.png" />
```

```
    <meta name="twitter:site" content="@Matrimony" />
```

```
    <!-- fontawesome -->
```

```
    <link rel="stylesheet"  
href="https://cdnjs.cloudflare.com/ajax/libs/font-  
awesome/5.15.4/css/all.min.css" />
```

```
    <!-- css -->
```

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

```
    <link rel="stylesheet" href="./assets/css/form.css">
```

```
    <title>Matrimony</title>
```

```
</head>
```

```
<body>
```

```
    <div class="login-card">
```

```
        <div class="login-card-content">
```

```
            <div class="header">
```

```
                <div class="logo">
```

```

        <div></div>
    </div>

    <h2>Matri<span class="highlight">mony</span></h2>
    <h3>Add Profile Photo</h3>
</div>

    <form class="form" action="updatephoto" method="POST"
enctype="multipart/form-data">
    <div class="form-field photo">
        <div class="icon">
            <i class="far fa-user"></i>
        </div>
        <input type="file" id="photo" name="photo">
    </div>

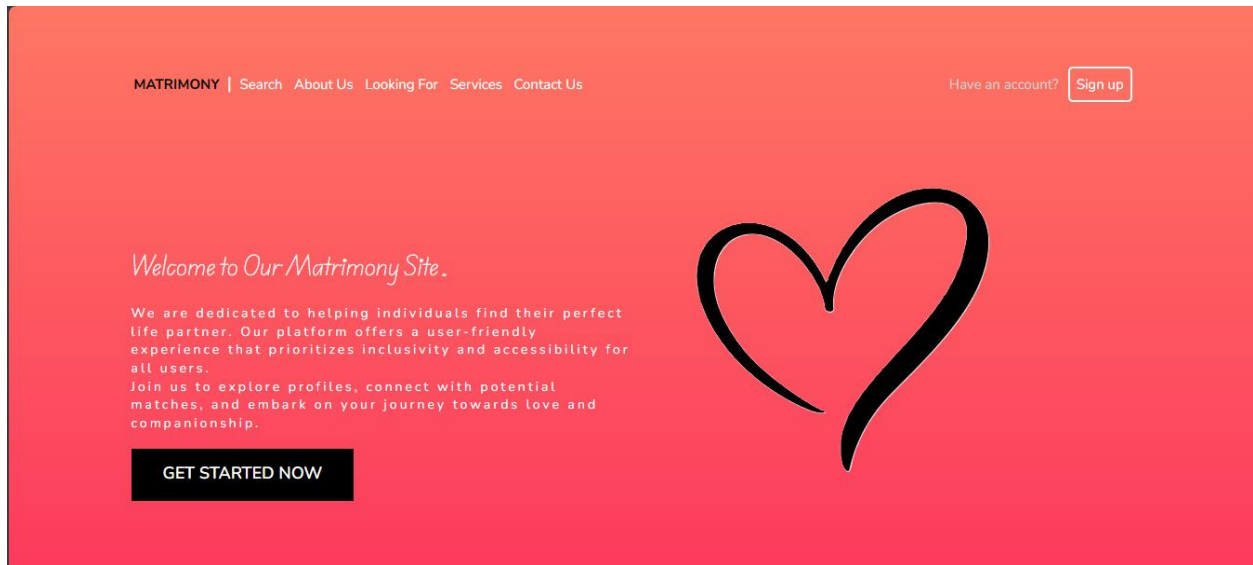
    <button type="submit">
        Update Profile Photo
    </button>
    </form>
</div>
</div>
</body>

</html>

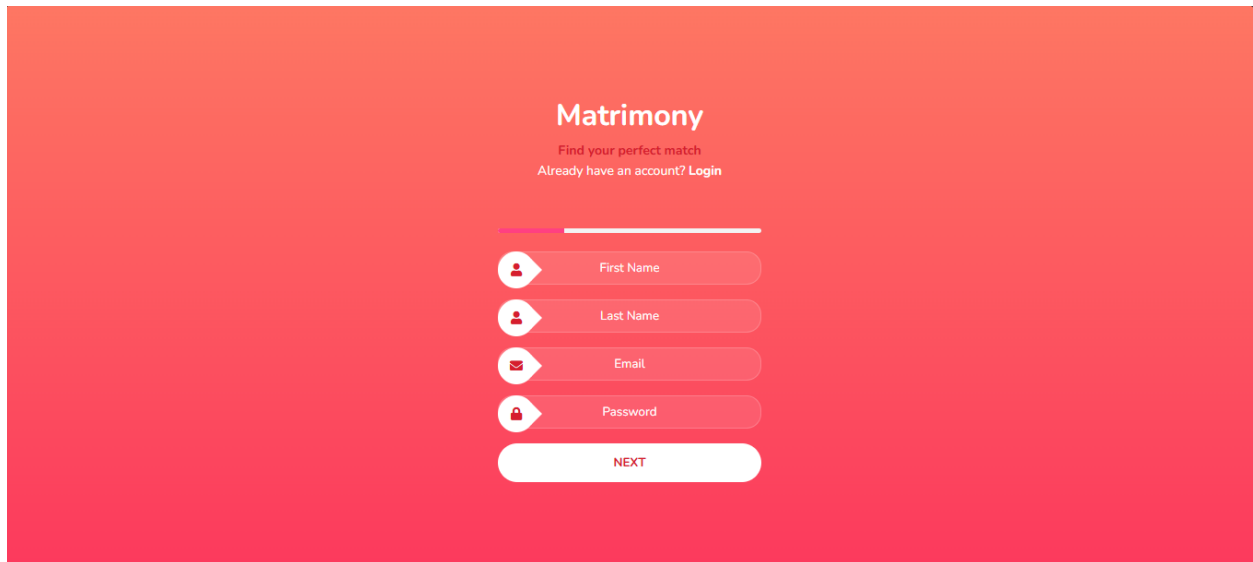
```

Screenshots

Landing page



Sign up page



Matrimony

Find your perfect match
Already have an account? [Login](#)

♂

Male

▼

📞

78234532

👥

Zulu

▼

PREVIOUS

NEXT

When age is less than 18 error shows

Matrimony

Find your perfect match
Already have an account? [Login](#)

📅

Must be at least 18 years old to sign up.

12/23/2010

📅

👤

176

❤️

Never Married

▼

PREVIOUS

NEXT

Matrimony

Find your perfect match

Already have an account? [Login](#)

12/23/2004

176

Never Married

PREVIOUS

NEXT

Matrimony

Find your perfect match

Already have an account? [Login](#)

Eswatini

Manzini

English

Anglicanism

Debit Card

Monthly - \$50

SUBMIT

User dashboard



Jackson

[Edit Profile](#)

[Edit Details](#)

0 Views

1 Ranking

0 Likes

Profile Details

Your name is Jackson your lastname is Jim, you speak English your religion is Anglicanism that is Zulu born 2004-12-23

Location:Eswatini

Email:jackson@gmail.com

Joined:2024-10-24T15:28:07.059203500

Payment:6 days, 23 hours left

Update Password: [Update Password](#)

Your Info

Details: No Details

Your Info

Details: No Details

Family Details: No Details

Qualification: No Details

Job: Not Working yet

Designation: No designations

Hobbies: No hobbies

Favourite Genre: No Favourite Genre

Favourite Movie: No Favourite Movies

Favourite Cuisine: No Favourite Cuisines

Favourite Books: No Favourite Books

Favourite Place: No Favourite Place

Similar users



Age: 34

Gugulethu bunandzi

[View profile](#)



Age: 28

Temaswati Dlamini

[View profile](#)



Age: 24

Nomfundo Mhoni

[View profile](#)



Age: 34

Jane Smith

[View profile](#)



Age: 36

Emily Brown

[View profile](#)



Age: 37

Sarah Miller

[View profile](#)



Age: 33

Laura Moore

[View profile](#)



Age: 38

Sophia Anderson

[View profile](#)



Age: 35

Olivia Jackson

[View profile](#)



Age: 39







Emma Harris

[View profile](#)

User dashboard when details updated

Tell Us More About Yourself

Likings

| | |
|--|--------------|
|  | Soccer |
|  | Jazz |
|  | Titanic |
|  | Pie, Chicken |
|  | Goodfellas |
|  | Spain, Japan |

PREVIOUS

SUBMIT

Matrimony

Tell Us More About Yourself

Professional Information



Havard



Central Bank of Africa



Software Engineer

PREVIOUS

NEXT

Matrimony

Tell Us More About Yourself

Personal Information



null



null

NEXT



Matrimony

Add Profile Photo

Choose File No file chosen

UPDATE PROFILE PHOTO

MATRIMONY | Search About Us Looking For Services Contact Us

Logout



Jackson

Edit Profile

Edit Details

0 Views

1 Ranking

0 Likes

Profile Details

Your name is Jackson your lastname is Jim, you speak English your religion is Anglicanism that is Zulu born 2001-12-23

Location:Eswatini

Email:jackson@gmail.com

Joined:2024-10-24T15:31:14.313656500

Payment:6 days, 23 hours left

Update Password: Update Password

Your Info

Details: No Details



Matrimony

Find you your perfect match

Jackson

Jim

jackson@gmail.com

Male

12/23/2004

176

Never Married

[illegible]

Search Procedures

MATRIMONY | Search About Us Looking For Services Contact Us

View Profile? Logout

Search by Caste

Select your caste/ethnic group

SEARCH CASTE Q

Search by Member ID

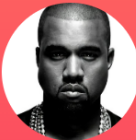
Member ID

SEARCH ID Q

MATRIMONY | Search About Us Looking for Services Contact Us

View Profile? Logout

Showing result for Member ID search retrieved 1 results



Male #3

Name Dalubuhle Dlamini

Age, Height 22 Yrs, 187cm

Religion, Caste Christianity : Swazi

Mother Tongue siSwati

76635980

dsihle875@gmail.com


LIKE

VIEW PROFILE

MATRIMONY | Search About Us Browse Services Contact Us

View Profile? Logout

Caste Search Succesful Find The Perfect One For Yourself



Female #11

Name Jane Smith

Age, Height 34 Yrs, 160cm

Religion, Caste Christianity : Other


Mother Tongue English

Phone Number: 1234567891

Email: janesmith1@example.com

LIKE

VIEW PROFILE



Female #13

Name Emily Brown

Age, Height 26 Yrs, 165cm


LIKE

Looking for page

[MATRIMONY](#) | [Search](#) [About Us](#) [Looking For](#) [Services](#) [Contact Us](#)


[View Profile?](#) [Logout](#)

Showing result 31 users



Female #6
Name [Gugulethu bunandzi](#)
Age, Height [34 Yrs, 167cm](#)
Religion, Caste [Zionism : Zulu](#)
Mother Tongue [siSwati](#)
Phone Number: [768680357](#)
Email: gugu@gmail.com

[❤️ LIKE](#)
[👤 VIEW PROFILE](#)



Female #7

Services page

Add Blog

View Profile?

Logout

 Dalubuhle

This is my first Add

i managed to find the woman of my dreams in this web app and i really appreciate what it has done for me

 Amelia

I am Married

this app really helped me i managed to get married and found the woman of my dreams here

Add Blog

View Profile?

Logout

Matrimony has helped me find the love of life

Matrimony has helped me find the love of my life and i really appreciate what they have done for me thank you

Submit Reset

appreciate what it has done for me

 Amelia

I am Married

this app really helped me i managed to get married and found the woman of my dreams here

dreams here

 James

THis is my blog

this is my blog body

 Jackson

Matrimony has helped me find the love of life

Matrimony has helped me find the love of my life and i really appreciate what they have done for me thank you

EDIT

Admin

D. R.

Tables

Additional Tables

Tables

Home / Tables

Total Users | All Time

 31

Views | Total Views

 37

Likes | Total Likes

 24

Users | Table

Tables

Additional Tables


Users | Table

| ID | Photo | Firstname | Lastname | Email | Gender | dob |
|----|---|-----------|----------|---------------------|--------|------------|
| #3 |  | Dalubuhle | Dlamini | dsihle875@gmail.com | Male | 02/14/2002 |
| #4 |  | James | Stewart | james@gmail.com | Male | 03/22/1999 |
| #5 |  | Phumlani | Nxumalo | pomy@gmail.com | Male | 12/19/1990 |
| #6 |  | Gugulethu | bunandzi | gugu@gmail.com | Female | 02/02/1990 |

User Details | Table

Tables

Additional Tables

| | | | | | | |
|-----|---|-------|--------|------------------------|------|------------|
| #18 |  | James | Taylor | jamestaylor1@example.c | Male | 03/05/1996 |
|-----|---|-------|--------|------------------------|------|------------|

User Details | Table

| User ID | About Self | Family Details | Qualification | Working At | Designation |
|---------|---------------------------|----------------------------|--------------------------|------------------------|-----------------|
| #3 | I am a software engineer | My family is fully swazi a | null | Central Bank of Africa | Engineeer |
| #4 | I am a software engineer | My family is fully Swazi a | Software Engineer | FNB | Engineer |
| #5 | A dedicated teacher with | Close-knit family, one sib | Bachelor of Arts in Educ | Swazi High School | Teacher |
| #6 | Entrepreneur focused or | Married with two childre | MBA | Innovatech Ltd. | CEO |
| #7 | Adventurer and nature lk | Parents and three sibling | Environmental Science | Green World Org | Conservationist |
| #8 | Food enthusiast and culi | Only child, supportive pa | Culinary Arts | Foodies Paradise | Chef |
| #9 | Fitness trainer focused o | Family of fitness enthusi | Sports Science | Health First Gym | Trainer |

Additional Tables Section

MatrimonyAdmin

Sihlelo





Tables

Additional Tables

Tables

Home / Additional Tables

User Adds | Table

| User ID | User image | Username | Title | Body | Action |
|---------|---|-----------|---------------------------|-------------------------------|---|
| # 3 |  | Dalubuhle | This is my first Add | i managed to find the woma | <button>Update</button> <button>Delete</button> |
| # 27 |  | Amelia | I am Married | this app really helped me i r | <button>Update</button> <button>Delete</button> |
| # 4 |  | James | THis is my blog | | <button>Update</button> <button>Delete</button> |
| # 37 |  | Jackson | Matrimony has helped me f | | <button>Update</button> <button>Delete</button> |