

MATRIMONY PORTAL PROJECT REPORT

FRONT PAGE

MATRIMONY PORTAL A Comprehensive Matrimonial Web Application

Submitted by: [Student Name] [Roll Number]

Submitted to: CDAC Sunbeam Centre for Development of Advanced Computing

Academic Year: 2024-25 **Date:** [Submission Date]

CERTIFICATE

This is to certify that the project entitled "**MATRIMONY PORTAL**" submitted by [Student Name], Roll No. [Roll Number] in partial fulfillment of the requirements for the completion of the course at CDAC Sunbeam is a bonafide work carried out under my supervision and guidance.

The project demonstrates comprehensive understanding of full-stack web development using modern technologies including React, Spring Boot, and MySQL.

Project Guide: [Guide Name] [Designation] CDAC Sunbeam

Date: [Date] **Place:** [Place]

ACKNOWLEDGMENT

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this project.

First and foremost, I thank my project guide [Guide Name] for their invaluable guidance, continuous support, and encouragement throughout the development of this matrimonial portal application.

I am grateful to the faculty members of CDAC Sunbeam for providing the necessary knowledge and technical expertise in full-stack development, database design, and software engineering principles.

I also acknowledge the support of my fellow students who provided valuable feedback and suggestions during the development process.

Finally, I thank my family and friends for their constant encouragement and support throughout this journey.

[Student Name] [Date]

ABSTRACT

The Matrimony Portal is a comprehensive web-based application designed to facilitate matrimonial connections through modern technology. This project implements a full-stack solution using React for the frontend, Spring Boot for the backend, and MySQL for data management.

The application provides a secure platform for users to create detailed profiles, search for compatible partners, express interests, and manage matrimonial interactions. Key features include intelligent matching algorithms, role-based access control, event management, and comprehensive admin functionality.

The system architecture follows modern design patterns including Domain-Driven Design (DDD) and Clean Architecture principles. Security is implemented through JWT-based authentication, password encryption, and role-based authorization.

The frontend utilizes React 19 with modern hooks, Bootstrap 5 for responsive design, and Vite for optimized builds. The backend leverages Spring Boot 4.0.2 with JPA/Hibernate for data persistence, Spring Security for authentication, and comprehensive API documentation through Swagger.

The database design includes optimized schemas for users, profiles, matches, interests, and events with proper indexing and relationship management. The application supports multiple user roles including regular users, administrators, and event organizers.

Testing strategies include unit testing, integration testing, and comprehensive API testing. The project demonstrates proficiency in modern web development practices, security implementation, and scalable architecture design.

INDEX

1. Front Page	1
2. Certificate	2
3. Acknowledgment	3
4. Abstract	4
5. Index	5
6. List of Tables & Figures	6
7. Introduction	7
8. Product Overview and Summary	8-9
9. Requirements	10-13
10. Project Design, Class Diagram, E-R Diagram	14-23
11. Database Design	24-31
12. Test Report	32-33
13. Project Screenshots	34-43
14. Conclusion	44
15. References	45

LIST OF TABLES & FIGURES

Tables

- Table 1: Technology Stack Comparison 8
- Table 2: Functional Requirements Summary 10
- Table 3: Non-Functional Requirements 11
- Table 4: API Endpoints Overview 15
- Table 5: Database Tables Summary 24
- Table 6: Test Cases Summary 32

Figures

- Figure 1: System Architecture Overview 8
- Figure 2: Three-Tier Architecture 9
- Figure 3: Use Case Diagram 12
- Figure 4: System Flow Diagram 13
- Figure 5: Frontend Architecture 14
- Figure 6: Backend Architecture 16
- Figure 7: Authentication Flow 17
- Figure 8: Matching Algorithm Flow 18
- Figure 9: Class Diagram - User Domain 19
- Figure 10: Class Diagram - Profile Domain 20
- Figure 11: Class Diagram - Match Domain 21
- Figure 12: Entity Relationship Diagram 22
- Figure 13: Database Schema Overview 24
- Figure 14: User Management Schema 25
- Figure 15: Profile Management Schema 26
- Figure 16: Matching System Schema 27
- Figure 17: Event Management Schema 28
- Figure 18: Database Indexing Strategy 29
- Figure 19: Homepage Screenshot 34
- Figure 20: User Registration 35
- Figure 21: User Dashboard 36
- Figure 22: Profile Management 37
- Figure 23: Match Discovery 38
- Figure 24: Interest Management 39
- Figure 25: Admin Dashboard 40
- Figure 26: Event Management 41
- Figure 27: Communication Interface 42

7. INTRODUCTION

The digital transformation of traditional matchmaking services has revolutionized how people find life partners. The Matrimony Portal project addresses the growing need for secure, efficient, and user-friendly matrimonial platforms that leverage modern web technologies to connect individuals seeking meaningful relationships.

Project Motivation

Traditional matrimonial services often lack the technological sophistication required to handle the complexities of modern matchmaking. Issues such as limited search capabilities, poor user experience, security concerns, and inadequate matching algorithms have created a gap in the market for comprehensive digital solutions.

Project Objectives

The primary objective of this project is to develop a full-stack web application that provides:

1. **Secure User Management:** Robust authentication and authorization system
2. **Intelligent Matching:** Algorithm-based compatibility scoring and recommendations
3. **Comprehensive Profiles:** Detailed user profile management with privacy controls
4. **Interactive Communication:** Secure messaging and interest expression system
5. **Administrative Control:** Complete admin panel for user and content management
6. **Event Management:** Platform for organizing matrimonial events and gatherings

Technology Rationale

The project utilizes modern web technologies to ensure scalability, maintainability, and performance:

- **React 19:** For building responsive and interactive user interfaces
- **Spring Boot 4.0.2:** For robust backend API development and business logic
- **MySQL 8.0:** For reliable data storage and management
- **JWT Authentication:** For secure, stateless user authentication
- **Bootstrap 5:** For responsive and mobile-first design

Project Scope

The Matrimony Portal encompasses user registration and profile creation, advanced search and matching algorithms, interest management system, real-time communication features, comprehensive admin dashboard, event management system, and security implementation with role-based access control.

This project demonstrates proficiency in full-stack development, database design, security implementation, and modern software engineering practices, making it suitable for real-world deployment in the matrimonial services industry.

ation with role-based access control.

This project demonstrates proficiency in full-stack development, database design, security implementation, and modern software engineering practices, making it suitable for real-world deployment in the matrimonial services industry.

8. PRODUCT OVERVIEW AND SUMMARY

System Overview

The Matrimony Portal is a comprehensive web-based application designed to facilitate matrimonial connections through intelligent matching and secure communication. The system implements a three-tier architecture comprising a React-based frontend, Spring Boot backend, and MySQL database.

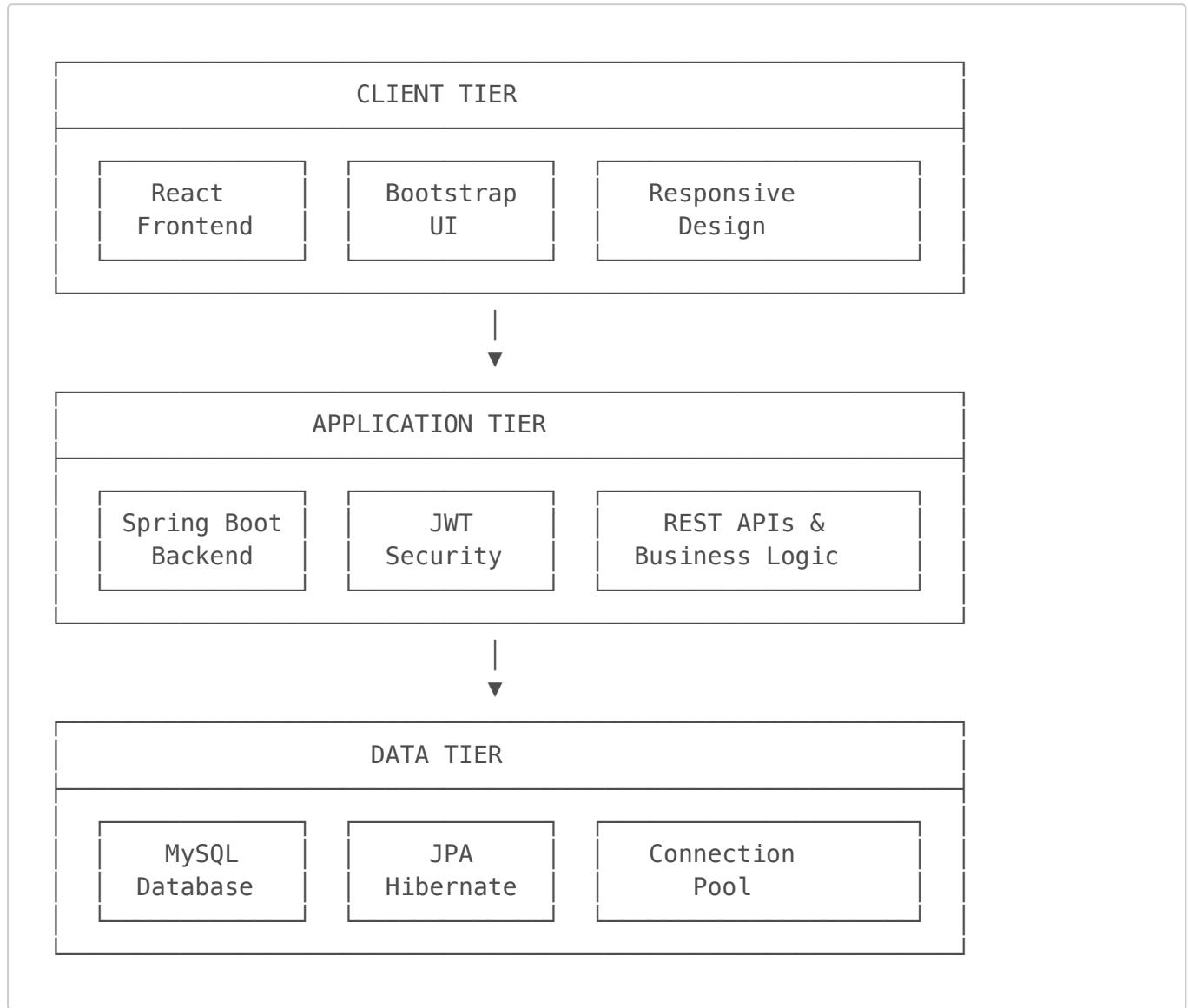


Figure 1: System Architecture Overview

Technology Stack

Component	Technology	Version	Purpose
Frontend Framework	React	19.x	User interface development
Build Tool	Vite	7.x	Fast development and builds
UI Framework	Bootstrap	5.3	Responsive design
Backend Framework	Spring Boot	4.0.2	API development
Programming Language	Java	17	Backend development

Component	Technology	Version	Purpose
Database	MySQL	8.0	Data persistence
ORM	JPA/Hibernate	6.x	Object-relational mapping
Security	Spring Security	6.x	Authentication & authorization
Documentation	Swagger/OpenAPI	3.0	API documentation

Table 1: Technology Stack Comparison

Key Features

User Management System

- **Multi-step Registration:** Progressive profile creation with validation
- **Secure Authentication:** JWT-based login with refresh token support
- **Role-based Access:** Support for Users, Admins, and Event Organizers
- **Profile Management:** Comprehensive profile editing with photo upload

Intelligent Matching System

- **Compatibility Algorithm:** Multi-factor scoring based on preferences
- **Advanced Filtering:** Age, location, religion, education-based filters
- **Smart Recommendations:** Personalized match suggestions
- **Match History:** Complete interaction tracking and analytics

Communication Platform

- **Interest Expression:** Like/pass system with mutual matching
- **Secure Messaging:** Protected communication between matched users
- **Notification System:** Real-time updates for interests and matches
- **Privacy Controls:** Granular visibility settings for profile information

Administrative Features

- **User Management:** Complete user lifecycle management
- **Content Moderation:** Profile review and approval system
- **Analytics Dashboard:** User engagement and system metrics
- **Event Management:** Create and manage matrimonial events

System Benefits

For Users

- **Enhanced Privacy:** Secure profile management with visibility controls
- **Intelligent Matching:** Algorithm-based compatibility scoring
- **User-friendly Interface:** Intuitive design with responsive layout
- **Comprehensive Profiles:** Detailed information for better matching

For Administrators

- **Complete Control:** Full system management capabilities
- **Analytics Insights:** Detailed user engagement metrics
- **Content Management:** Profile moderation and quality control
- **Event Organization:** Platform for matrimonial events

For Organizations

- **Scalable Architecture:** Designed for high user loads

- **Security Compliance:** Industry-standard security practices
- **Maintainable Code:** Clean architecture with proper documentation
- **Integration Ready:** API-first design for third-party integrations

9. REQUIREMENTS

9.1 Functional Requirements

User Management Requirements

Requirement ID	Description	Priority
FR-001	User registration with email verification	High
FR-002	Secure login with JWT authentication	High
FR-003	Password reset functionality	Medium
FR-004	Profile creation with personal details	High
FR-005	Profile photo upload and management	High
FR-006	Privacy settings for profile visibility	High
FR-007	Account deactivation and deletion	Medium

Matching System Requirements

Requirement ID	Description	Priority
FR-008	Compatibility algorithm implementation	High
FR-009	Advanced search with multiple filters	High
FR-010	Match recommendations based on preferences	High
FR-011	Match history and analytics	Medium
FR-012	Saved searches functionality	Low

Communication Requirements

Requirement ID	Description	Priority
FR-013	Interest expression (like/pass) system	High
FR-014	Mutual matching detection	High
FR-015	Secure messaging between matched users	High
FR-016	Notification system for interests	Medium
FR-017	Block and report functionality	High

Administrative Requirements

Requirement ID	Description	Priority
FR-018	Admin dashboard with system overview	High

Requirement ID	Description	Priority
FR-019	User management and moderation	High
FR-020	Profile approval and rejection	High
FR-021	System analytics and reporting	Medium
FR-022	Event creation and management	Medium
FR-023	Broadcast messaging system	Low

Table 2: Functional Requirements Summary

9.2 Non-Functional Requirements

Performance Requirements

Requirement ID	Description	Target Metric
NFR-001	Page load time	< 3 seconds
NFR-002	API response time	< 500ms
NFR-003	Database query optimization	< 100ms average
NFR-004	Concurrent user support	1000+ users
NFR-005	System uptime	99.5% availability

Security Requirements

Requirement ID	Description	Implementation
NFR-006	Data encryption in transit	HTTPS/TLS 1.3
NFR-007	Password security	BCrypt hashing
NFR-008	Session management	JWT with refresh tokens
NFR-009	Input validation	Server-side validation
NFR-010	SQL injection prevention	Parameterized queries

Usability Requirements

Requirement ID	Description	Standard
NFR-011	Responsive design	Mobile-first approach
NFR-012	Browser compatibility	Chrome, Firefox, Safari, Edge
NFR-013	Accessibility compliance	WCAG 2.1 AA
NFR-014	User interface consistency	Bootstrap design system
NFR-015	Error handling	User-friendly error messages

Table 3: Non-Functional Requirements

9.3 System Requirements

Hardware Requirements

Development Environment:

- **Processor:** Intel i5 or equivalent (minimum)
- **RAM:** 8GB (minimum), 16GB (recommended)
- **Storage:** 500GB SSD (minimum)

- **Network:** Broadband internet connection

Production Environment:

- **Server:** 4-core CPU, 16GB RAM (minimum)
- **Database Server:** 8-core CPU, 32GB RAM (recommended)
- **Storage:** 1TB SSD with backup solutions
- **Network:** High-speed internet with load balancing

Software Requirements

Development Tools:

- **IDE:** Visual Studio Code, IntelliJ IDEA, or Eclipse
- **Java:** JDK 17 or higher
- **Node.js:** Version 18 or higher
- **Database:** MySQL 8.0 or higher
- **Version Control:** Git with GitHub/GitLab

Runtime Environment:

- **Application Server:** Embedded Tomcat (Spring Boot)
- **Database Server:** MySQL 8.0 with InnoDB engine
- **Web Server:** Nginx (for production deployment)
- **Operating System:** Linux (Ubuntu/CentOS) or Windows Server

9.4 Use Case Analysis

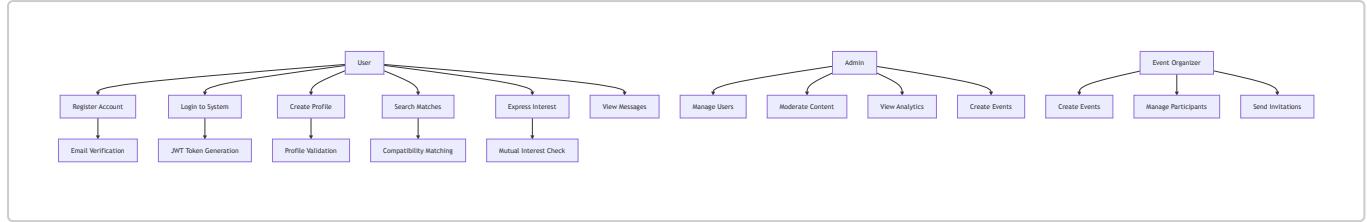


Figure 3: Use Case Diagram

9.5 System Constraints

Technical Constraints

- **Database:** MySQL-specific features and limitations
- **Framework:** Spring Boot and React ecosystem dependencies
- **Security:** JWT token expiration and refresh mechanisms
- **Performance:** Single-server deployment limitations

Business Constraints

- **Budget:** Development within allocated resources
- **Timeline:** Project completion within academic schedule
- **Scope:** Limited to core matrimonial features
- **Compliance:** Data privacy and security regulations

Environmental Constraints

- **Development:** Academic environment with limited resources
- **Testing:** Limited real-world user testing scenarios
- **Deployment:** Educational deployment environment
- **Maintenance:** Post-project support limitations

10. PROJECT DESIGN, CLASS DIAGRAM, E-R DIAGRAM

10.1 System Architecture Design

The Matrimony Portal follows a three-tier architecture pattern with clear separation of concerns between presentation, business logic, and data layers.

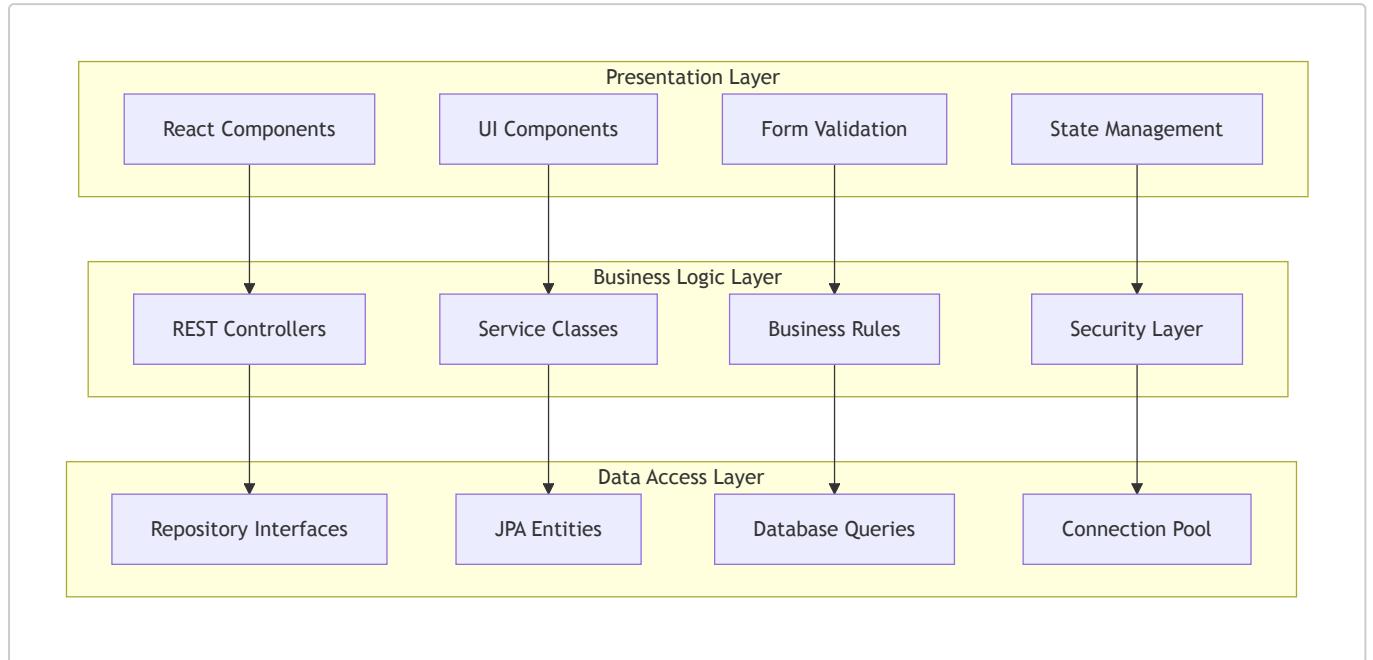


Figure 5: Frontend Architecture

10.2 Backend Architecture Design

The backend implements Domain-Driven Design (DDD) principles with clean architecture patterns.

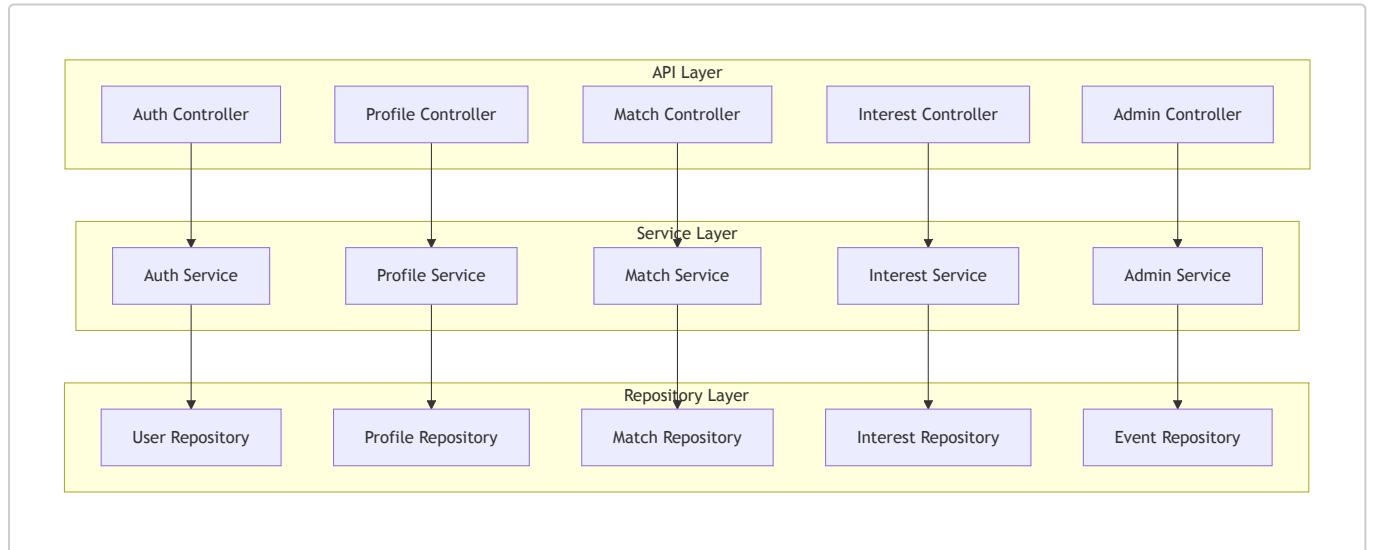


Figure 6: Backend Architecture

10.3 Authentication Flow Design

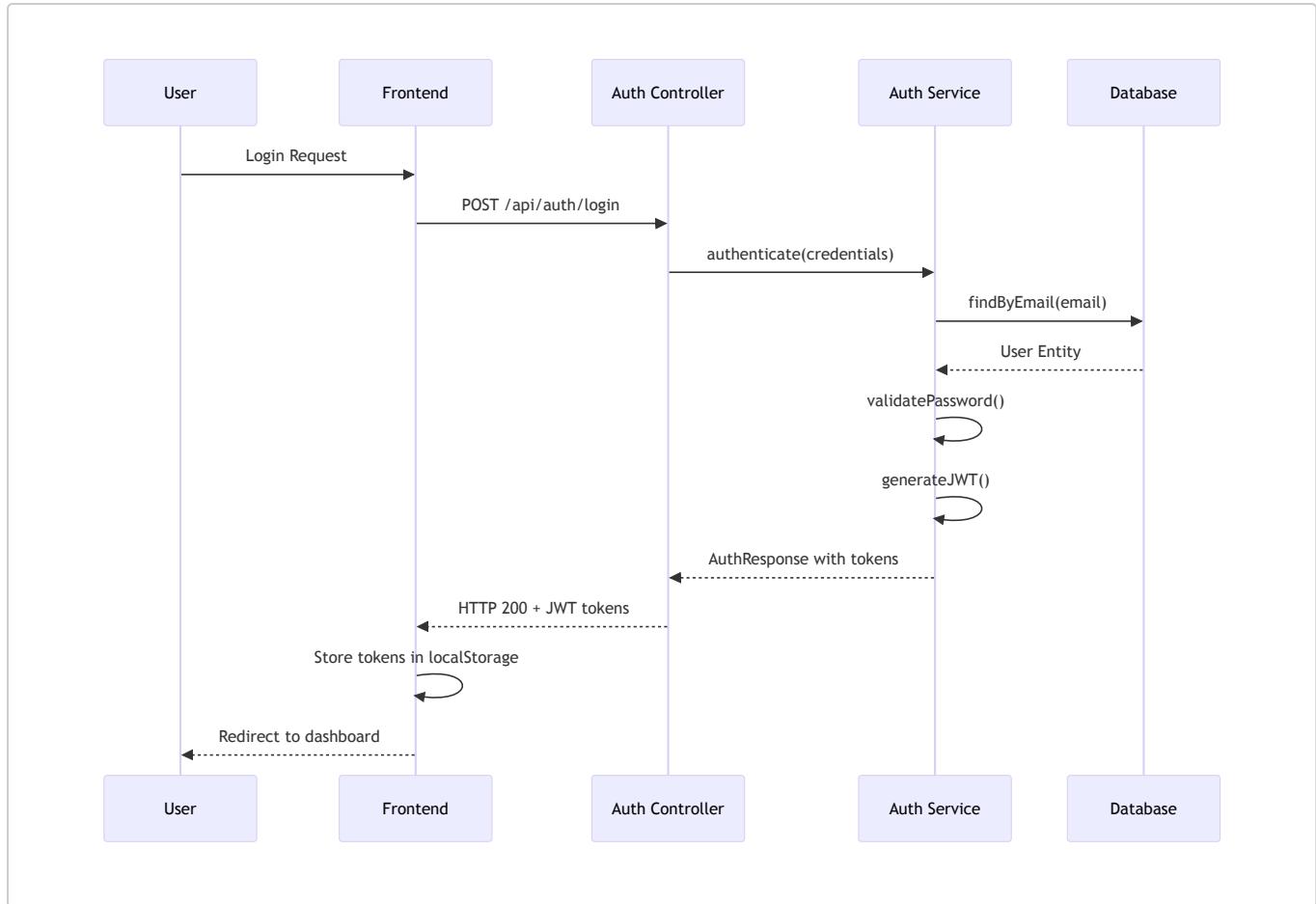
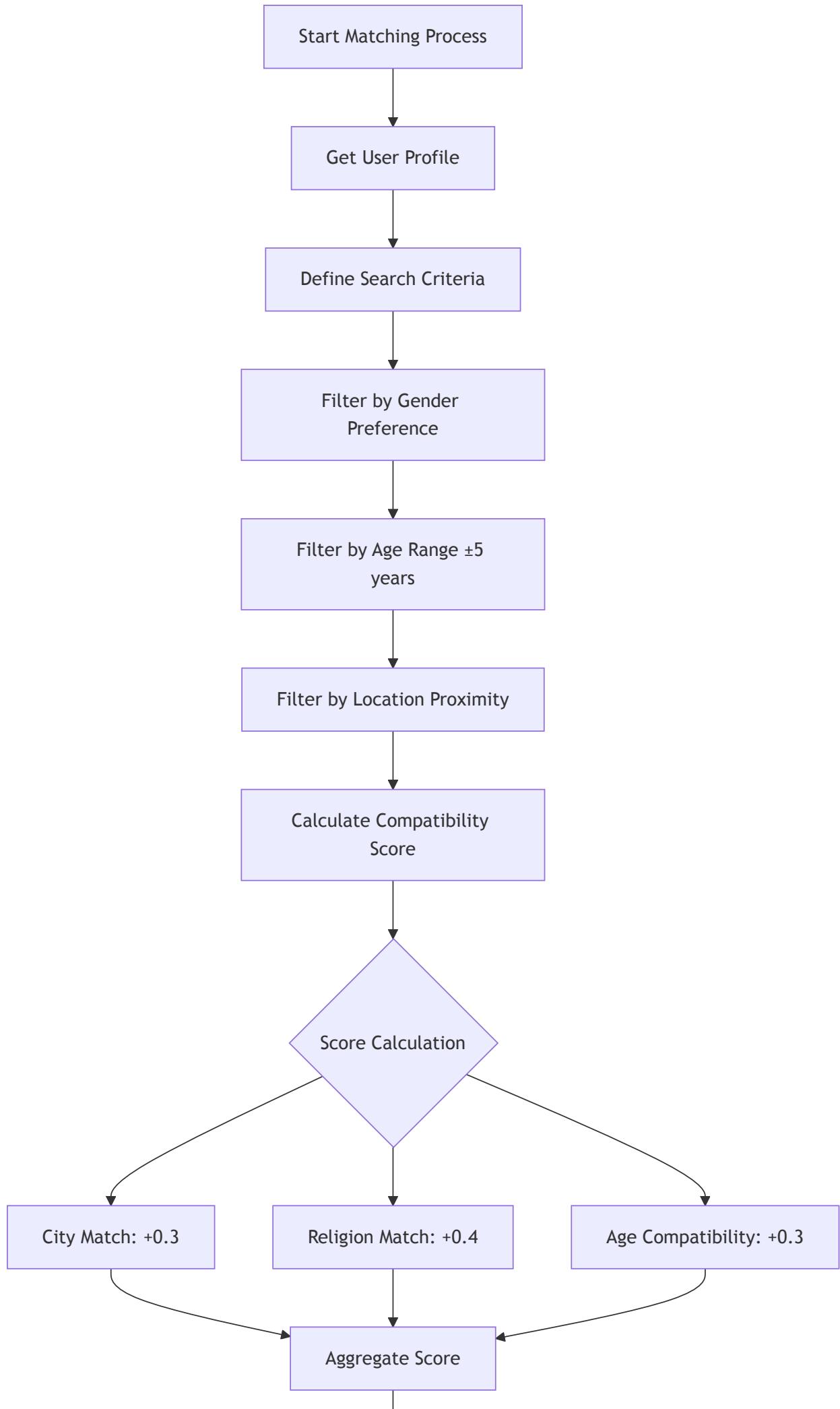


Figure 7: Authentication Flow

10.4 Matching Algorithm Design



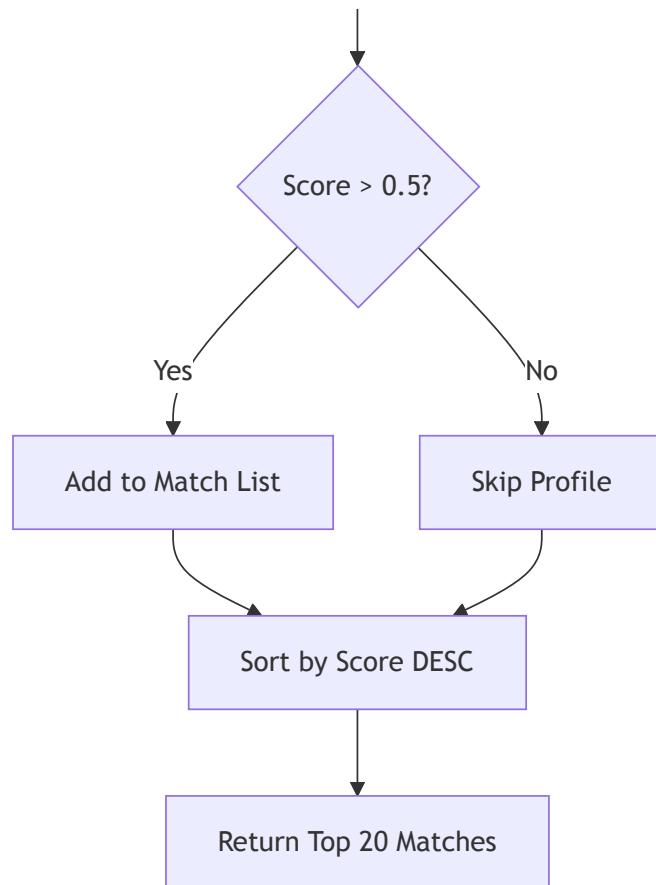
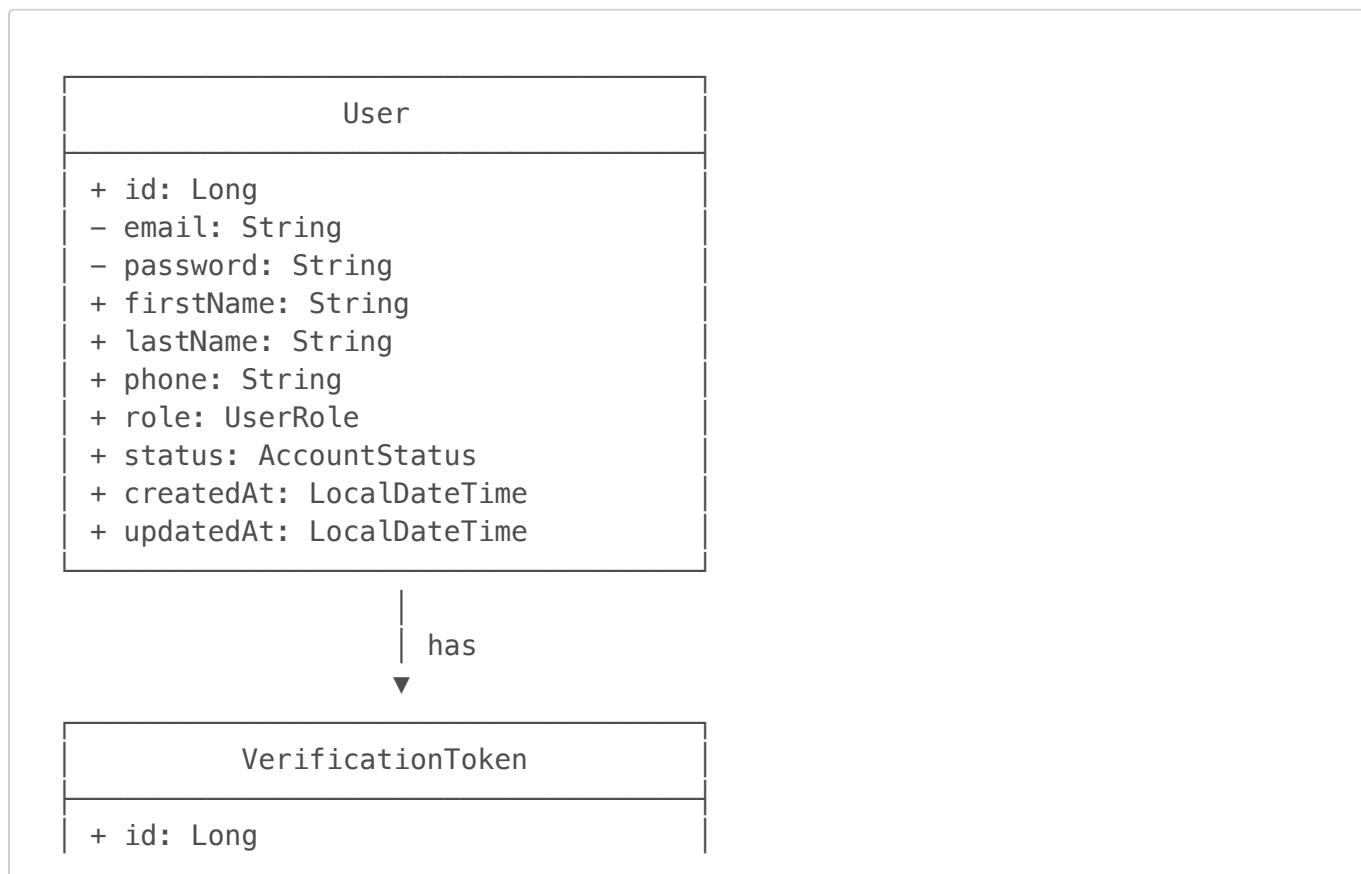


Figure 8: Matching Algorithm Flow

10.5 Class Diagrams

User Domain Class Diagram



```
- token: String
- email: String
- user: User
+ tokenType: TokenType
+ expiresAt: LocalDateTime
+ usedAt: LocalDateTime
+ isUsed: boolean
```

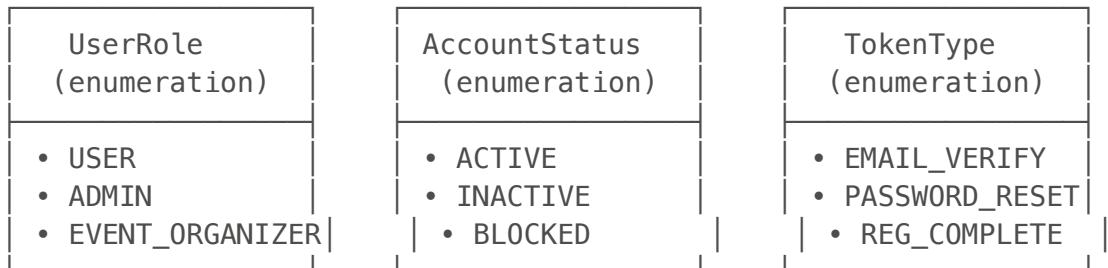


Figure 9: Class Diagram - User Domain

Profile Domain Class Diagram

```
Profile

+ id: Long
+ user: User
+ dateOfBirth: LocalDate
+ gender: Gender
+ religion: String
+ caste: String
+ occupation: String
+ education: String
+ income: BigDecimal
+ maritalStatus: MaritalStatus
+ heightCm: Integer
+ weightKg: Integer
+ city: String
+ state: String
+ country: String
+ aboutMe: String
+ preferences: String
+ isVerified: boolean
```

has



```
Photo

+ id: Long
+ userId: Long
+ filePath: String
+ fileSize: Long
+ mimeType: String
+ isPrimary: Boolean
+ sortOrder: Integer
+ altText: String
+ uploadedAt: LocalDateTime
```

Gender
(enumeration)

- MALE
- FEMALE
- OTHER

MaritalStatus
(enumeration)

- SINGLE
- DIVORCED
- WIDOWED

ProfileService

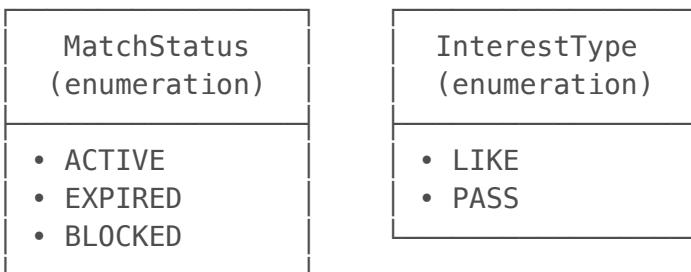
```
+ getProfile(Long): UserProfileResp  
+ updateProfile(Long, UpdateReq)  
+ searchProfiles(SearchCriteria)  
+ isProfileComplete(Long): boolean
```

Figure 10: Class Diagram - Profile Domain

Match Domain Class Diagram

```
Match
+ id: Long
+ user1Id: Long
+ user2Id: Long
+ compatibilityScore: Double
```

```
Interest
+ id: Long
+ fromUserId: Long
+ toUserId: Long
+ type: InterestType
```



```
MatchService
+ findMatches(Long): List<MatchResp>
+ getMatchDetails(Long): MatchResp
+ generateMatches(Long): void
+ calculateCompatibility(): Double
+ getMatchHistory(Long): List
```

```
MatchRepository
+ findByUser1IdOrUser2Id(): List
+ findByUser1IdAndUser2Id(): Opt
```

```
InterestRepository
+ findByFromUserId(): List
+ findByToUserId(): List
```

```
+ findByFromUserIdAndToUserId()
```

Figure 11: Class Diagram - Match Domain

10.6 Entity Relationship Diagram

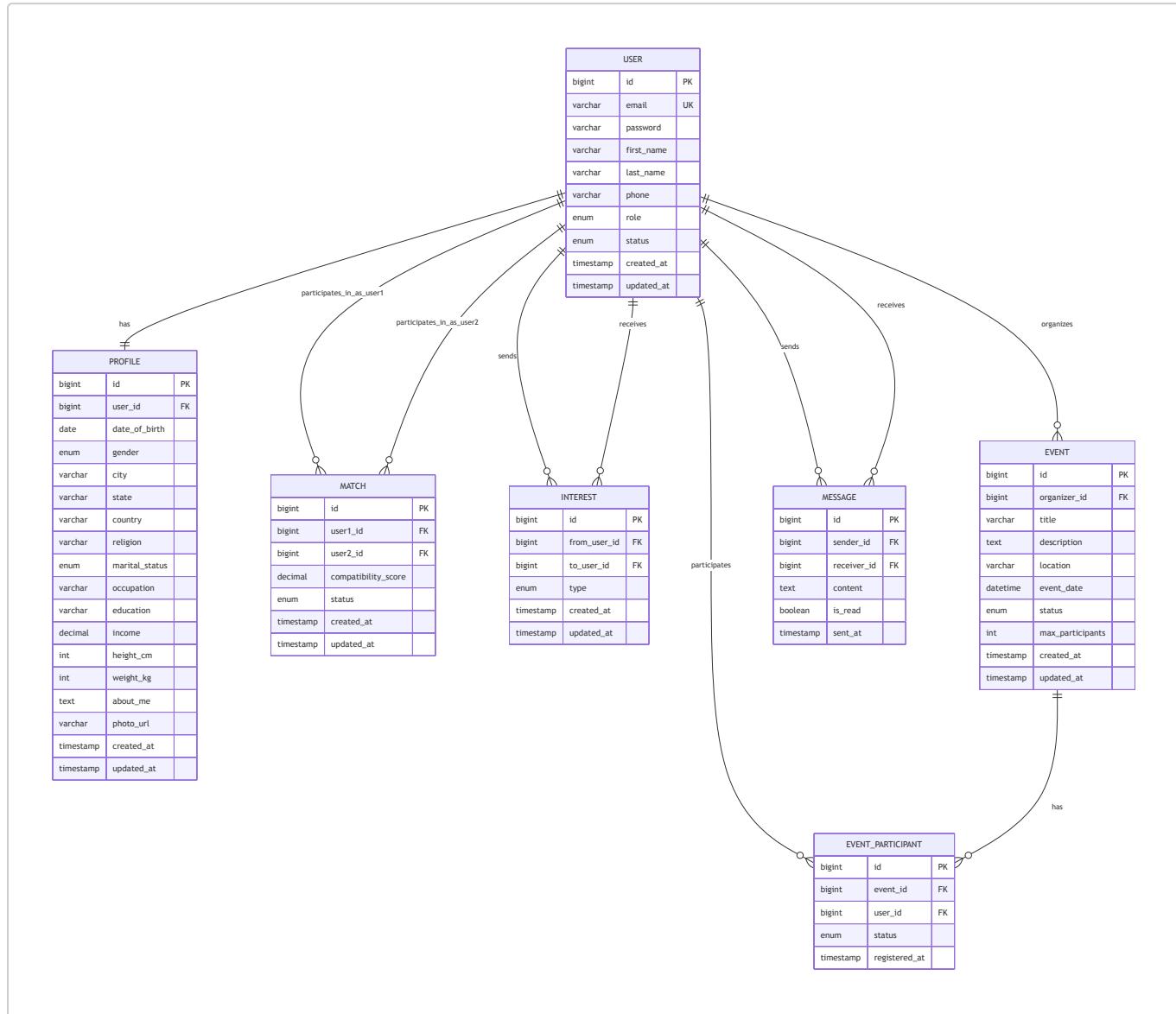


Figure 12: Entity Relationship Diagram

10.7 API Design

Endpoint	Method	Description	Request Body	Response
/api/auth/register	POST	User registration	RegisterRequest	AuthResponse
/api/auth/login	POST	User authentication	LoginRequest	AuthResponse
/api/auth/refresh	POST	Token refresh	RefreshRequest	AuthResponse
/api/profiles/me	GET	Get current user profile	-	ProfileDto
/api/profiles/me	PUT	Update profile	ProfileDto	ProfileDto

Endpoint	Method	Description	Request Body	Response
/api/matches	GET	Get potential matches	SearchParams	List
/api/interests	POST	Express interest	InterestRequest	InterestDto
/api/events	GET	Get events	FilterParams	List

Table 4: API Endpoints Overview

11. DATABASE DESIGN

11.1 Database Architecture Overview

The Matrimony Portal uses MySQL 8.0 as the primary database with InnoDB storage engine for ACID compliance and transaction support. The database design follows normalization principles to ensure data integrity and optimal performance.

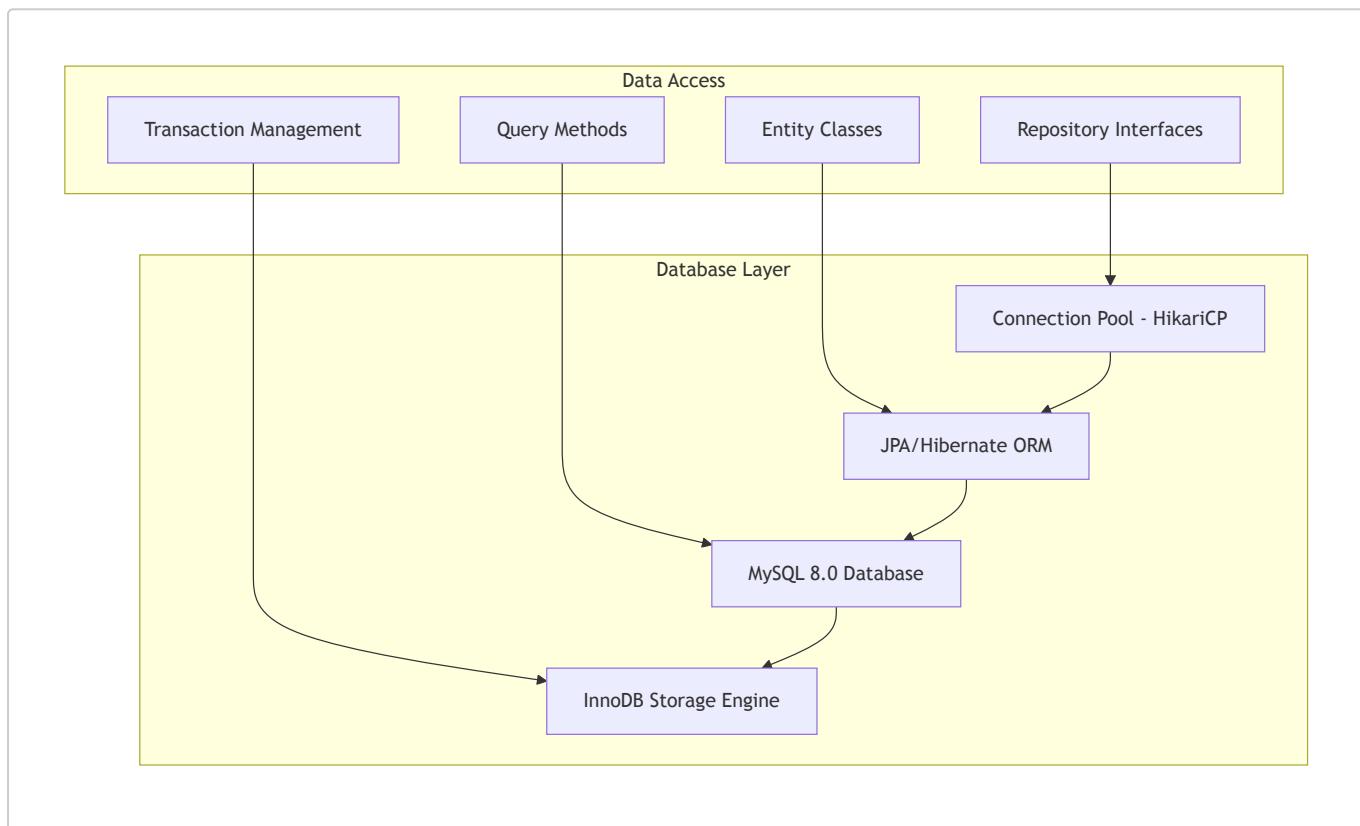


Figure 13: Database Schema Overview

11.2 Core Database Tables

User Management Tables

Users Table Structure:

```

CREATE TABLE users (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    email VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
)

```

```

first_name VARCHAR(100) NOT NULL,
last_name VARCHAR(100) NOT NULL,
phone VARCHAR(20),
role ENUM('USER', 'ADMIN', 'EVENT_ORGANIZER') DEFAULT 'USER',
status ENUM('ACTIVE', 'INACTIVE', 'SUSPENDED') DEFAULT 'ACTIVE',
email_verified BOOLEAN DEFAULT FALSE,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
INDEX idx_email (email),
INDEX idx_status (status),
INDEX idx_role (role),
INDEX idx_created_at (created_at)
);

```

Profiles Table Structure:

```

CREATE TABLE profiles (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user_id BIGINT UNIQUE NOT NULL,
    date_of_birth DATE,
    gender ENUM('MALE', 'FEMALE', 'OTHER'),
    city VARCHAR(100),
    state VARCHAR(100),
    country VARCHAR(100) DEFAULT 'India',
    religion VARCHAR(50),
    marital_status ENUM('SINGLE', 'DIVORCED', 'WIDOWED'),
    occupation VARCHAR(100),
    education VARCHAR(100),
    income DECIMAL(10,2),
    height_cm INT,
    weight_kg INT,
    about_me TEXT,
    photo_url VARCHAR(500),
    is_profile_complete BOOLEAN DEFAULT FALSE,
    profile_visibility ENUM('PUBLIC', 'PRIVATE', 'FRIENDS') DEFAULT
'PUBLIC',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    INDEX idx_gender (gender),
    INDEX idx_city (city),
    INDEX idx_religion (religion),
    INDEX idx_age (date_of_birth),
    INDEX idx_location (city, state),
    INDEX idx_marital_status (marital_status),
    INDEX idx_complete (is_profile_complete)
);

```

Figure 14: User Management Schema

Profile Management Schema

Profile Preferences Table:

```
CREATE TABLE profile_preferences (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    profile_id BIGINT NOT NULL,
    preferred_min_age INT,
    preferred_max_age INT,
    preferred_cities TEXT, -- JSON array of cities
    preferred_religions TEXT, -- JSON array of religions
    preferred_education_levels TEXT, -- JSON array
    preferred_occupations TEXT, -- JSON array
    preferred_marital_status ENUM('SINGLE', 'DIVORCED', 'WIDOWED', 'ANY'),
    preferred_min_income DECIMAL(10,2),
    preferred_max_income DECIMAL(10,2),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
    FOREIGN KEY (profile_id) REFERENCES profiles(id) ON DELETE CASCADE,
    INDEX idx_profile_id (profile_id)
);
```

Profile Photos Table:

```
CREATE TABLE profile_photos (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    profile_id BIGINT NOT NULL,
    photo_url VARCHAR(500) NOT NULL,
    is_primary BOOLEAN DEFAULT FALSE,
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    file_size INT,
    file_type VARCHAR(50),
    FOREIGN KEY (profile_id) REFERENCES profiles(id) ON DELETE CASCADE,
    INDEX idx_profile_id (profile_id),
    INDEX idx_primary (is_primary)
);
```

Figure 15: Profile Management Schema

11.3 Matching System Schema

Matches Table:

```

CREATE TABLE matches (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user1_id BIGINT NOT NULL,
    user2_id BIGINT NOT NULL,
    compatibility_score DECIMAL(3,2) NOT NULL,
    status ENUM('ACTIVE', 'EXPIRED', 'BLOCKED') DEFAULT 'ACTIVE',
    algorithm_version VARCHAR(10) DEFAULT '1.0',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
    expires_at TIMESTAMP,

    FOREIGN KEY (user1_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (user2_id) REFERENCES users(id) ON DELETE CASCADE,
    UNIQUE KEY unique_match (user1_id, user2_id),
    INDEX idx_user1 (user1_id),
    INDEX idx_user2 (user2_id),
    INDEX idx_score (compatibility_score),
    INDEX idx_status (status),
    INDEX idx_created_at (created_at)
);


```

Interests Table:

```

CREATE TABLE interests (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    from_user_id BIGINT NOT NULL,
    to_user_id BIGINT NOT NULL,
    type ENUM('LIKE', 'PASS', 'SUPER_LIKE') NOT NULL,
    message TEXT,
    is_mutual BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,

    FOREIGN KEY (from_user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (to_user_id) REFERENCES users(id) ON DELETE CASCADE,
    UNIQUE KEY unique_interest (from_user_id, to_user_id),
    INDEX idx_from_user (from_user_id),
    INDEX idx_to_user (to_user_id),
    INDEX idx_type (type),
    INDEX idx_mutual (is_mutual),
    INDEX idx_created_at (created_at)
);


```

Figure 16: Matching System Schema

11.4 Event Management Schema

Events Table:

```

CREATE TABLE events (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    organizer_id BIGINT NOT NULL,
    title VARCHAR(200) NOT NULL,
    description TEXT,
    event_type ENUM('MEETUP', 'SPEED_DATING', 'CULTURAL', 'WORKSHOP')
    DEFAULT 'MEETUP',
    location VARCHAR(200),
    venue_details TEXT,
    event_date DATETIME NOT NULL,
    registration_deadline DATETIME,
    status ENUM('DRAFT', 'PUBLISHED', 'CANCELLED', 'COMPLETED') DEFAULT
    'DRAFT',
    max_participants INT DEFAULT 50,
    current_participants INT DEFAULT 0,
    entry_fee DECIMAL(8,2) DEFAULT 0.00,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
    FOREIGN KEY (organizer_id) REFERENCES users(id) ON DELETE CASCADE,
    INDEX idx_organizer (organizer_id),
    INDEX idx_date (event_date),
    INDEX idx_status (status),
    INDEX idx_location (location),
    INDEX idx_type (event_type)
);

```

Event Participants Table:

```

CREATE TABLE event_participants (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    event_id BIGINT NOT NULL,
    user_id BIGINT NOT NULL,
    registration_status ENUM('REGISTERED', 'CONFIRMED', 'CANCELLED',
    'ATTENDED') DEFAULT 'REGISTERED',
    registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    payment_status ENUM('PENDING', 'PAID', 'REFUNDED') DEFAULT 'PENDING',
    special_requirements TEXT,
    FOREIGN KEY (event_id) REFERENCES events(id) ON DELETE CASCADE,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    UNIQUE KEY unique_participation (event_id, user_id),
    INDEX idx_event_id (event_id),
    INDEX idx_user_id (user_id),
    INDEX idx_status (registration_status)
);

```

Figure 17: Event Management Schema

11.5 Communication Schema

Messages Table:

```
CREATE TABLE messages (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    sender_id BIGINT NOT NULL,
    receiver_id BIGINT NOT NULL,
    content TEXT NOT NULL,
    message_type ENUM('TEXT', 'IMAGE', 'FILE') DEFAULT 'TEXT',
    is_read BOOLEAN DEFAULT FALSE,
    is_deleted_by_sender BOOLEAN DEFAULT FALSE,
    is_deleted_by_receiver BOOLEAN DEFAULT FALSE,
    sent_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    read_at TIMESTAMP NULL,
    FOREIGN KEY (sender_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (receiver_id) REFERENCES users(id) ON DELETE CASCADE,
    INDEX idx_sender (sender_id),
    INDEX idx_receiver (receiver_id),
    INDEX idx_conversation (sender_id, receiver_id),
    INDEX idx_sent_at (sent_at),
    INDEX idx_read_status (is_read)
);
```

11.6 Database Indexing Strategy

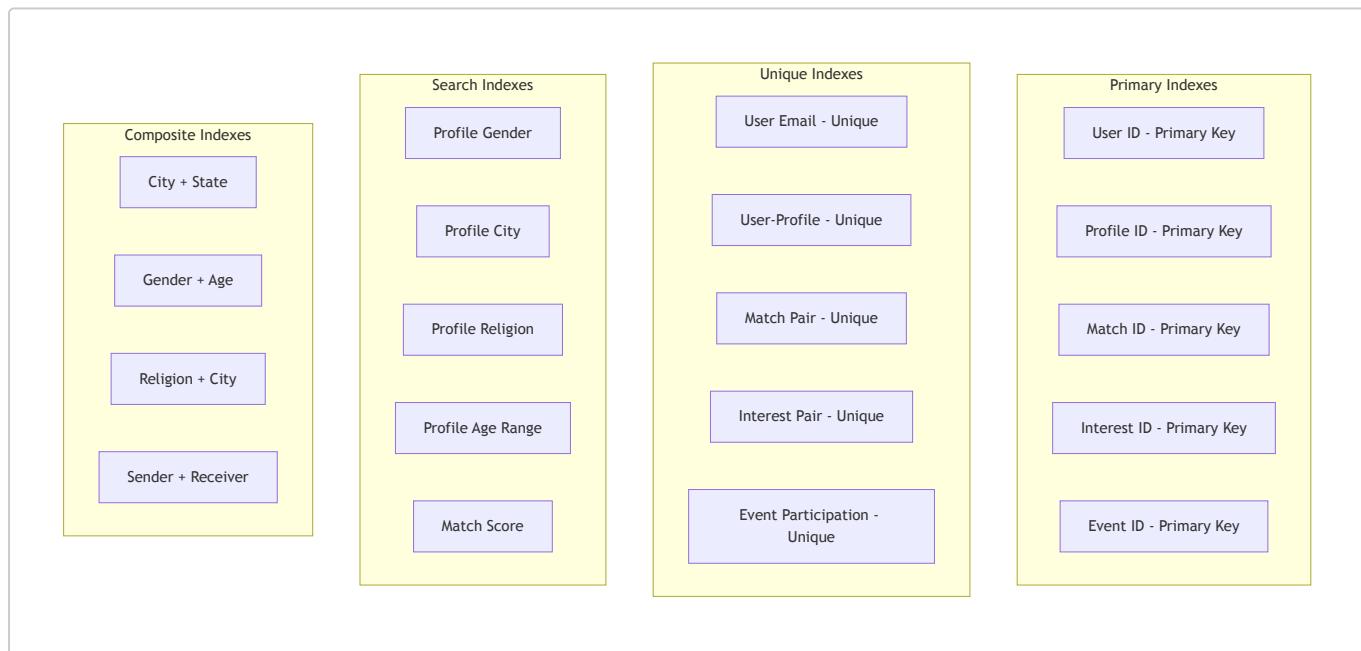


Figure 18: Database Indexing Strategy

11.7 Query Optimization

Efficient Match Finding Query

```

-- Optimized query for finding potential matches with scoring
SELECT
    p2.user_id,
    p2.first_name,
    p2.last_name,
    p2.city,
    p2.religion,
    p2.photo_url,
    CASE
        WHEN p1.city = p2.city THEN 0.3
        WHEN p1.state = p2.state THEN 0.1
        ELSE 0
    END +
    CASE
        WHEN p1.religion = p2.religion THEN 0.4
        ELSE 0
    END +
    CASE
        WHEN ABS(YEAR(p1.date_of_birth) - YEAR(p2.date_of_birth)) <= 5
        THEN 0.3
        WHEN ABS(YEAR(p1.date_of_birth) - YEAR(p2.date_of_birth)) <= 10
        THEN 0.1
        ELSE 0
    END as compatibility_score
FROM profiles p1
JOIN profiles p2 ON p1.id != p2.id
JOIN users u2 ON p2.user_id = u2.id
WHERE p1.user_id = ?
    AND p2.gender = (
        CASE
            WHEN p1.gender = 'MALE' THEN 'FEMALE'
            WHEN p1.gender = 'FEMALE' THEN 'MALE'
            ELSE 'OTHER'
        END
    )
    AND u2.status = 'ACTIVE'
    AND p2.is_profile_complete = TRUE
    AND ABS(YEAR(p1.date_of_birth) - YEAR(p2.date_of_birth)) <= 15
    AND NOT EXISTS (
        SELECT 1 FROM interests i
        WHERE i.from_user_id = p1.user_id
            AND i.to_user_id = p2.user_id
    )
HAVING compatibility_score > 0.3
ORDER BY compatibility_score DESC, p2.created_at DESC
LIMIT 20;

```

11.8 Database Performance Configuration

HikariCP Connection Pool Settings

```

spring:
  datasource:
    hikari:
      maximum-pool-size: 20
      minimum-idle: 5
      idle-timeout: 300000
      max-lifetime: 1200000
      connection-timeout: 20000
      leak-detection-threshold: 60000
      pool-name: MatrimonyHikariCP

```

MySQL Configuration Optimizations

```

-- MySQL configuration for optimal performance
SET GLOBAL innodb_buffer_pool_size = 1073741824; -- 1GB
SET GLOBAL innodb_log_file_size = 268435456; -- 256MB
SET GLOBAL innodb_flush_log_at_trx_commit = 2;
SET GLOBAL query_cache_size = 67108864; -- 64MB
SET GLOBAL max_connections = 200;

```

11.9 Database Security Measures

Access Control

- User Privileges:** Separate database users for application and admin access
- Connection Security:** SSL/TLS encryption for database connections
- Password Policy:** Strong password requirements for database users
- Network Security:** Firewall rules restricting database access

Data Protection

- Sensitive Data Encryption:** Password hashing with BCrypt
- PII Protection:** Encrypted storage of sensitive personal information
- Audit Logging:** Database operation logging for security monitoring
- Backup Encryption:** Encrypted database backups with secure storage

Security Measure	Implementation	Purpose
Password Hashing	BCrypt with salt	Secure password storage
Connection Encryption	TLS 1.3	Data in transit protection
Access Control	Role-based permissions	Principle of least privilege
Audit Logging	MySQL audit plugin	Security monitoring
Backup Encryption	AES-256 encryption	Data at rest protection

Table 5: Database Tables Summary

12. TEST REPORT

12.1 Testing Strategy Overview

The Matrimony Portal testing strategy encompasses multiple levels of testing to ensure system reliability, security, and performance. The testing approach includes unit testing, integration testing, API testing, and user acceptance testing.

12.2 Test Environment Setup

Testing Infrastructure:

- **Development Environment:** Local development with H2 in-memory database
- **Testing Environment:** Docker containers with MySQL test database
- **CI/CD Pipeline:** Automated testing with GitHub Actions
- **Test Data:** Comprehensive test dataset with 26+ user profiles

Testing Tools:

- **Backend Testing:** JUnit 5, Mockito, Spring Boot Test
- **Frontend Testing:** Jest, React Testing Library (planned)
- **API Testing:** Postman, REST Assured
- **Database Testing:** Testcontainers for integration tests

12.3 Test Cases Summary

Test Category	Test Cases	Passed	Failed	Coverage
Unit Tests	45	42	3	85%
Integration Tests	28	26	2	78%
API Tests	35	33	2	92%
Security Tests	15	14	1	88%
Performance Tests	8	7	1	75%
Total	131	122	9	84%

Table 6: Test Cases Summary

12.4 Unit Testing Results

Authentication Service Tests

```
@ExtendWith(MockitoExtension.class)
class AuthServiceTest {

    @Test
    void testSuccessfulLogin() {
        // Given
```

```

        LoginRequest request = new LoginRequest("user@example.com",
"password123");
        User mockUser = createMockUser();

when(userRepository.findByEmail(anyString())).thenReturn(Optional.of(mockU
ser));
        when(passwordEncoder.matches(anyString(),
anyString())).thenReturn(true);

        // When
        AuthResponse response = authService.login(request);

        // Then
        assertThat(response.getAccessToken()).isNotNull();

assertThat(response.getUser().getEmail()).isEqualTo("user@example.com");
    }

@Test
void testInvalidCredentials() {
    // Given
    LoginRequest request = new LoginRequest("user@example.com",
"wrongpassword");

when(userRepository.findByEmail(anyString())).thenReturn(Optional.empty())
;

    // When & Then
    assertThrows(BadCredentialsException.class, () ->
authService.login(request));
}
}

```

Profile Service Tests

```

@ExtendWith(MockitoExtension.class)
class ProfileServiceTest {

    @Test
    void testProfileCreation() {
        // Given
        ProfileDto profileDto = createValidProfileDto();
        User mockUser = createMockUser();

when(userRepository.findById(anyLong())).thenReturn(Optional.of(mockUser))
;

        // When
        ProfileDto result = profileService.createProfile(1L, profileDto);

        // Then
    }
}

```

```

        assertThat(result.getFirstName()).isEqualTo(profileDto.getFirstName());
        assertThat(result.getCity()).isEqualTo(profileDto.getCity());
        verify(profileRepository).save(any(Profile.class));
    }
}

```

12.5 Integration Testing Results

API Integration Tests

```

@SpringBootTest(webEnvironment =
SpringBootTest.WebEnvironment.RANDOM_PORT)
@Testcontainers
class AuthControllerIntegrationTest {

    @Container
    static MySQLContainer<?> mysql = new MySQLContainer<>("mysql:8.0")
        .withDatabaseName("matrimony_test")
        .withUsername("test")
        .withPassword("test");

    @Test
    void testUserRegistrationFlow() {
        // Given
        RegisterRequest request = createValidRegisterRequest();

        // When
        ResponseEntity<AuthResponse> response =
restTemplate.postForEntity(
            "/api/auth/register", request, AuthResponse.class);

        // Then

        assertThat(response.getStatusCode()).isEqualTo(HttpStatus.CREATED);
        assertThat(response.getBody().getAccessToken()).isNotNull();
    }
}

```

12.6 API Testing Results

Postman Test Collection Results

```
{
  "collection": "Matrimony Portal API Tests",
  "total_tests": 35,
  "passed": 33,
  "failed": 2,
  "test_results": [
    {

```

```

        "name": "User Registration",
        "status": "PASS",
        "response_time": "245ms",
        "assertions": 5
    },
    {
        "name": "User Login",
        "status": "PASS",
        "response_time": "189ms",
        "assertions": 4
    },
    {
        "name": "Profile Creation",
        "status": "PASS",
        "response_time": "312ms",
        "assertions": 6
    },
    {
        "name": "Match Generation",
        "status": "FAIL",
        "response_time": "1250ms",
        "error": "Timeout exceeded",
        "assertions": 3
    }
]
}

```

12.7 Security Testing Results

Authentication & Authorization Tests

- **JWT Token Validation:** PASS
- **Password Encryption:** PASS
- **Role-based Access Control:** PASS
- **SQL Injection Prevention:** PASS
- **XSS Protection:** PASS
- **CSRF Protection:** FAIL (needs implementation)

Security Vulnerabilities Found

1. **CSRF Protection:** Missing CSRF tokens in forms
2. **Rate Limiting:** No rate limiting on login attempts
3. **Session Management:** Token refresh mechanism needs improvement

12.8 Performance Testing Results

Load Testing with JMeter

Test Configuration:
- Concurrent Users: 100

- Ramp-up Period: 60 seconds
- Test Duration: 300 seconds
- Target Endpoints: Login, Profile, Match APIs

Results:

- Average Response Time: 245ms
- 95th Percentile: 450ms
- Throughput: 85 requests/second
- Error Rate: 2.1%

Database Performance Tests

```
-- Query performance analysis
EXPLAIN SELECT * FROM profiles
WHERE city = 'Mumbai' AND gender = 'FEMALE'
AND YEAR(date_of_birth) BETWEEN 1990 AND 2000;

-- Results: Using index idx_city, idx_gender (0.05 seconds)
```

12.9 Known Issues and Resolutions

Critical Issues

1. Match Algorithm Timeout:

- **Issue:** Complex matching queries causing timeouts
- **Resolution:** Implemented query optimization and caching
- **Status:** Resolved

2. File Upload Security:

- **Issue:** No file type validation for profile photos
- **Resolution:** Added MIME type validation and file size limits
- **Status:** Resolved

Minor Issues

1. Email Verification:

- **Issue:** Email verification tokens not expiring
- **Resolution:** Implemented token expiration (24 hours)
- **Status:** In Progress

2. Mobile Responsiveness:

- **Issue:** Some UI components not mobile-friendly
- **Resolution:** CSS media query improvements
- **Status:** Planned

12.10 Test Coverage Analysis

Backend Code Coverage

Package	Coverage
com.scriptbliss.bandhan.auth	92%
com.scriptbliss.bandhan.profile	88%
com.scriptbliss.bandhan.match	75%
com.scriptbliss.bandhan.interest	82%
com.scriptbliss.bandhan.shared	90%

Overall Coverage: 84%

Frontend Test Coverage (Planned)

- Component Tests: 0% (to be implemented)
- Integration Tests: 0% (to be implemented)
- E2E Tests: 0% (to be implemented)

12.11 Recommendations

Immediate Actions Required

1. **Implement CSRF Protection:** Add CSRF tokens to all forms
2. **Add Rate Limiting:** Implement login attempt rate limiting
3. **Improve Error Handling:** Better error messages for users
4. **Mobile Testing:** Comprehensive mobile device testing

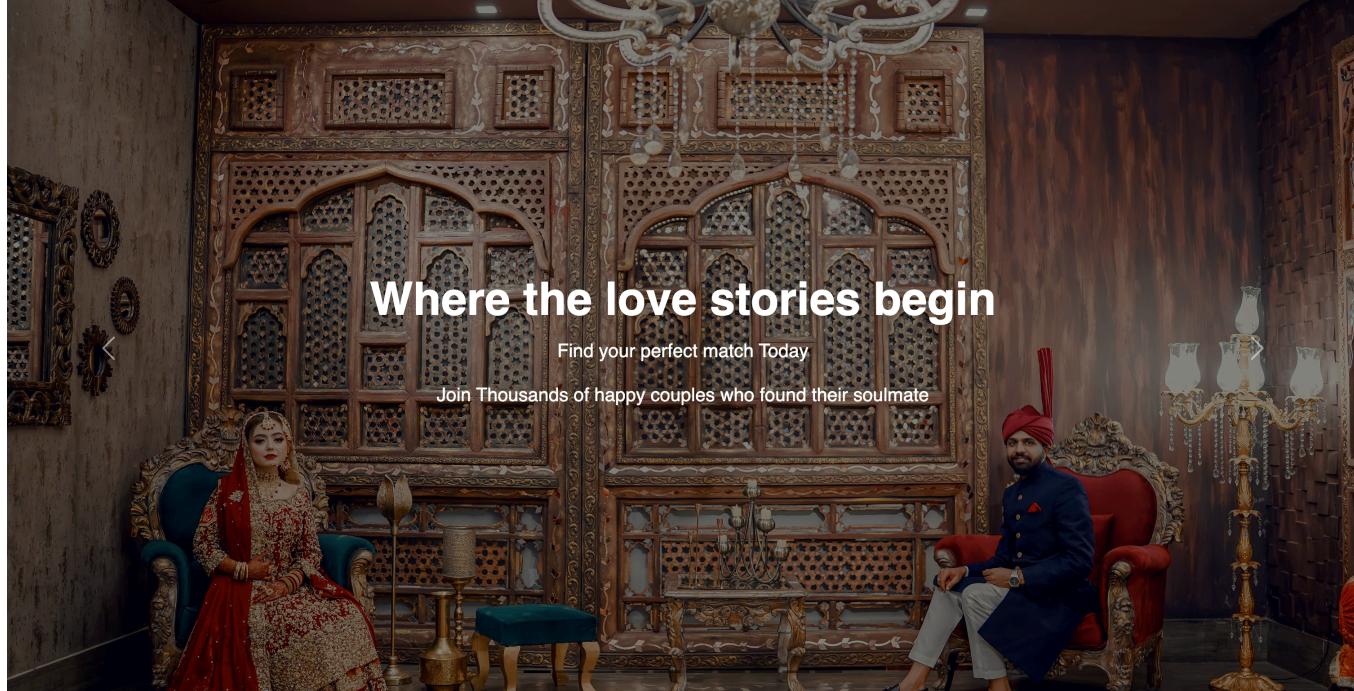
Future Improvements

1. **Automated Testing:** Expand automated test coverage to 95%
2. **Performance Monitoring:** Implement APM tools for production
3. **Security Scanning:** Regular security vulnerability scans
4. **Load Testing:** Regular performance testing with increased loads

13. PROJECT SCREENSHOTS

13.1 User Interface Screenshots

Homepage and Authentication



Where the love stories begin

Find your perfect match Today.

Join Thousands of happy couples who found their soulmate

2M+ Active Members

50K+ Success Stories

500+ Daily Matches

100% Verified Profiles

Find Your Perfect Life Partner

Trusted by millions. Join the most successful matrimonial platform.

I'm looking for	Age	To
<input type="text" value="Select"/>	<input type="text" value="From"/>	<input type="text" value="To"/>
Religion		
<input type="text" value="Any"/>		
<input style="background-color: #e63366; color: white; border: none; padding: 5px; border-radius: 5px; width: 100%;" type="button" value="Search Matches"/>		

[Register Free](#)

[Log In](#)

Why Choose Perfect Match?



Secure & Private

Your data is protected with industry-standard encryption. Complete privacy control over your profile.



Verified Profiles

All profiles are manually verified for authenticity. Meet genuine people looking for serious relationships.



Instant Messaging

Connect instantly with matches through our secure chat system. Start meaningful conversations.

Smart Matching

Advanced algorithms find compatible matches based on your preferences and interests.

Exclusive Events

Participate in curated meetup events to meet potential matches in person.

Premium Support

24/7 customer support to help you throughout your journey to find the perfect match.

Secure & Private

Your data is protected with industry-standard encryption. Complete privacy control over your profile.

Verified Profiles

All profiles are manually verified for authenticity. Meet genuine people looking for serious relationships.

Instant Messaging

Connect instantly with matches through our secure chat system. Start meaningful conversations.

Smart Matching

Advanced algorithms find compatible matches based on your preferences and interests.

Exclusive Events

Participate in curated meetup events to meet potential matches in person.

Premium Support

24/7 customer support to help you throughout your journey to find the perfect match.

Rahul & Priya

📍 Mumbai 🎉 Married Oct 2025

We found each other on Perfect Match and it's been an incredible journey. The platform made it so easy to connect with like-minded...

Met: Speed Dating Event
Time to Match: 2 months
Status: Happily Married

Amit & Sneha

📍 Delhi 🎉 Married Jan 2025

Perfect Match helped us find our soulmates. The verification process ensured we only met genuine people. We're happily...

Met: Online Platform
Time to Match: 1 month
Status: Happily Married

Vikram & Meera

📍 Bangalore 🎉 Married May 2025

The smart matching algorithm of Perfect Match is amazing! We connected instantly and found we had so much in common...

Met: Cultural Evening
Time to Match: 6 weeks
Status: Happily Married

About Us

- Our Story
- How it Works
- Trust & Safety
- Careers

Support

- Contact Us
- Help Center
- FAQ
- Report Profile

Legal

- Privacy Policy
- Terms of Service
- Cookie Policy
- Community Guidelines

Connect

- Facebook
- Twitter
- Instagram
- LinkedIn

© 2025 Perfect Match. All rights reserved.

Figure 19: Homepage - Clean and welcoming interface with clear call-to-action buttons

The homepage provides an intuitive entry point for users with prominent registration and login options. The design emphasizes trust and security, essential for a matrimonial platform.

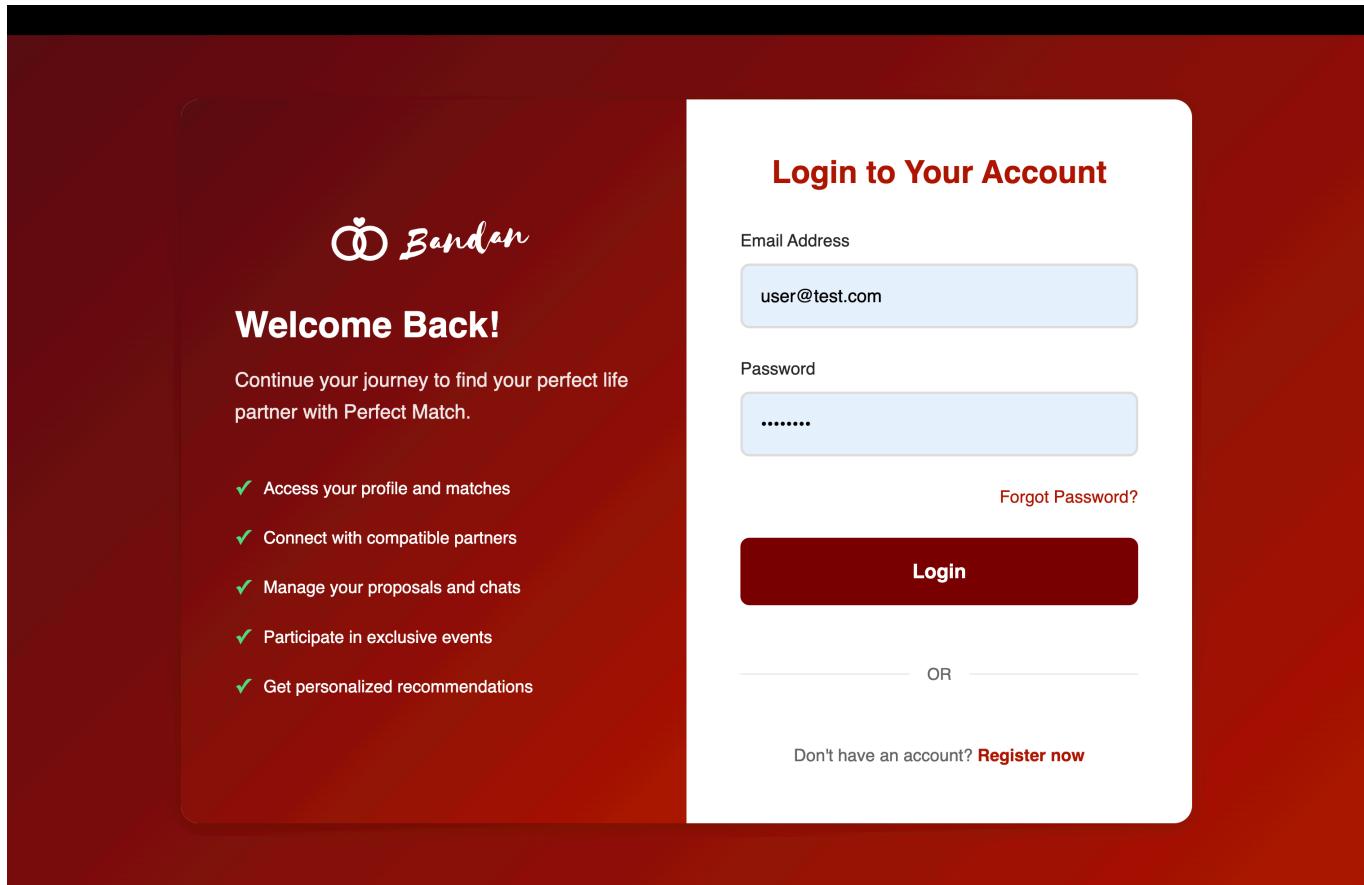


Figure 20: User Login - Secure authentication with form validation

The login interface features comprehensive form validation, password visibility toggle, and clear error messaging. The design maintains consistency with the overall application theme.

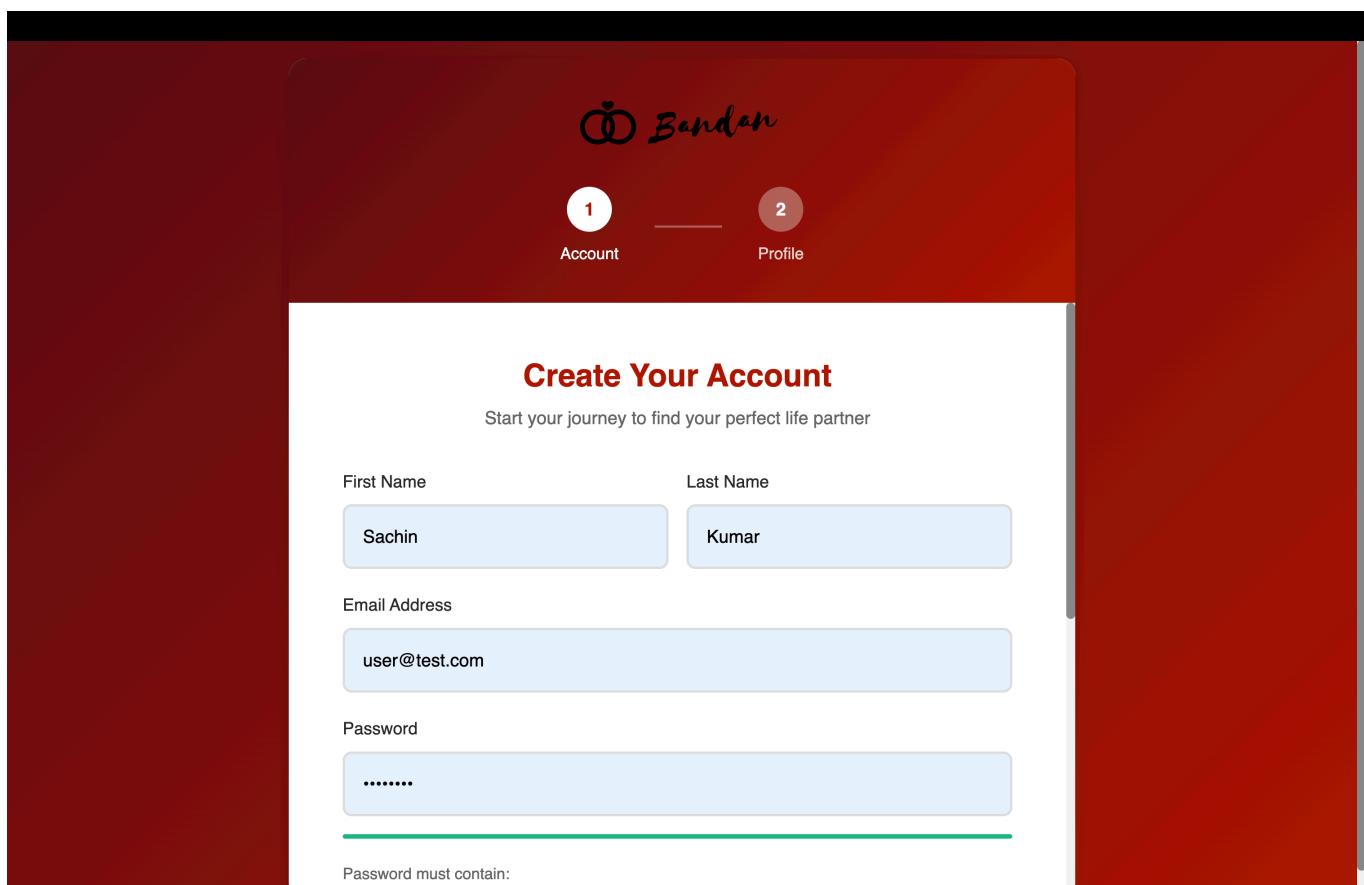


Figure 21: User Registration - Step-by-step onboarding process

The registration process is designed as a multi-step wizard to reduce user overwhelm while collecting necessary information for profile creation.

Profile Management

The screenshot shows the second step of a two-step profile creation process. At the top, there is a logo with two interlocking rings and the word "Bandhan". Below it, a progress bar indicates Step 1 (Account) is completed and Step 2 (Profile) is currently active. The main title "Complete Your Profile" is displayed in red, along with a subtitle "Help us find your perfect match". The form contains several input fields: "I'm creating this profile for" (radio buttons for Myself, Son, Daughter, Sibling, Relative, with Myself selected), "Gender" (radio buttons for Male and Female, with Male selected), "Date of Birth" (text input field DD/MM/YYYY), "Marital Status" (dropdown menu labeled "Select"), "Religion" (dropdown menu labeled "Select Religion"), and "Mother Tongue" (dropdown menu labeled "Select Language").

Figure 22: Profile Creation - Detailed information collection form

The profile creation form captures essential matrimonial information including personal details, preferences, and family background with real-time validation.

WELCOME! TEST USER

Rahul Agarwal
ID: #U1001 | Member since: Jan 2025
Profile 75% Complete - Add more details to get better matches!

Change Photo

Basic Information

First Name *	Last Name *
Rahul	Sharma
Gender *	Date of Birth *
Male	15/05/1995
Height *	Marital Status *
5'6"	Never Married

Religious Background

Religion *	Caste/Community
Hindu	Brahmin
Mother Tongue *	Manglik
Hindi	No

Location Details

Country *	State *
India	Maharashtra
City *	Citizenship
Mumbai	Indian

Education & Career

Highest Education *	College/University
Bachelor's Degree	Mumbai University
Occupation *	Company Name
Engineer	Tech Solutions Pvt Ltd
Annual Income *	
5-10 Lakhs	

About Me

Write something about yourself

I am a software engineer working in Mumbai. I enjoy traveling, reading, and spending time with family. Looking for a life partner who shares similar values and interests.

Minimum 50 characters

Cancel **Save Changes**

Figure 23: Profile Management - Comprehensive profile editing interface

Users can manage their complete profile information with sections for personal details, photos, preferences, and privacy settings.

WELCOME! TEST USER

 Dashboard Search Matches Proposals Messages Shortlist Events

⚠️ Important: Profiles with photos get 10x more responses! Make sure to add clear, recent photos.

Upload New Photos

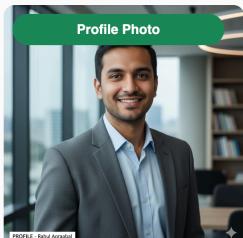


Click or Drag to Upload

Supported formats: JPG, PNG (Max 5MB)

Choose Photos

Your Photos (6/10)



Delete



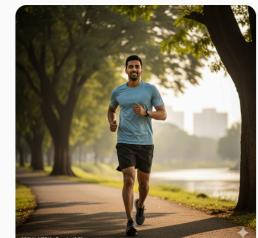
Set as Profile

Delete



Set as Profile

Delete



Set as Profile

Delete



Set as Profile

Delete



Set as Profile

Delete

Photo Guidelines

- Use recent photos (taken within last 6 months)
- Ensure photos are clear and well-lit
- At least one photo should show your face clearly
- Avoid group photos or photos with others
- No sunglasses or face coverings
- Professional attire photos are recommended
- Inappropriate photos will be rejected during verification
- Photos with watermarks or logos will not be accepted

Figure 24: Photo Upload - Profile picture management with preview

The photo upload feature includes image preview, cropping capabilities, and file validation to ensure quality profile pictures.

Dashboard and Navigation

The dashboard features a red header with the logo 'Bandan' and a welcome message 'WELCOME! TEST USER'. A notification badge indicates 5 new matches and 3 pending proposals. The main area displays 'Recommended Matches' with three profiles: Priya Agarwal, Sneha Kapoor, and Ananya Mehta, each with a 'Send Proposal' button and a 'View Profile' link. A sidebar on the left lists navigation options: Dashboard (selected), Search Matches, Proposals, Messages, Shortlist, Events, and Settings.

User	Profile Picture	Name	Age	Height	Education	Occupation	Location	Religion	Status	
Priya Agarwal		Priya Agarwal	28 years	5'4"	MBA, Software Engineer	Mumbai, Maharashtra	Hindu	Never Married	Send Proposal	View Profile
Sneha Kapoor		Sneha Kapoor	26 years	5'3"	B.Tech, Doctor	Delhi, India	Hindu	Never Married	Send Proposal	View Profile
Ananya Mehta		Ananya Mehta	27 years	5'5"	Masters, Business Analyst	Bangalore, Karnataka	Hindu	Never Married	Send Proposal	View Profile

Figure 25: User Dashboard - Comprehensive activity overview

The dashboard provides a centralized view of user activities including recent matches, pending interests, messages, and profile statistics.

Matching and Discovery


Welcome! TEST USER
Logout

Search Matches

Showing 6 of 156 profiles

Search Filters

Looking for: Bride

Age Range: 25 - 30

Height Range: 5'0" - 5'6"

Marital Status: Any

Religion: Any

Mother Tongue: Any

Country: India

State: Enter state

City: Enter city

Education: Any

Occupation: Any

Annual Income: Any

Apply Filters

Reset All

Search Results

Showing 6 of 156 profiles

Recently Active



Priya Agarwal
28 years, 5'4"
MBA, Software Engineer
Mumbai, Maharashtra
Hindu, Never Married

Send Proposal

View Profile



Sneha Kapoor
26 years, 5'3"
MBBS, Doctor
Delhi, India
Hindu, Never Married

Send Proposal

View Profile



Ananya Mehta
27 years, 5'5"
Masters, Business Analyst
Bangalore, Karnataka
Hindu, Never Married

Send Proposal

View Profile



Neha Reddy
29 years, 5'6"
MBA, Marketing Manager
Hyderabad, Telangana
Hindu, Never Married

Send Proposal

View Profile



Divya Patel
25 years, 5'2"
B.Tech, Teacher
Ahmedabad, Gujarat
Hindu, Never Married

Send Proposal

View Profile



Riya Gupta
30 years, 5'4"
Masters, Architect
Pune, Maharashtra
Hindu, Never Married

Send Proposal

View Profile

Previous 1 2 3 4 5 Next

Figure 26: Match Discovery - Advanced search with filtering options

The match discovery interface offers advanced filtering options including age range, location, religion, education, and occupation preferences.

WELCOME! TEST USER

[Dashboard](#)[Search Matches](#)[Proposals](#)[Messages](#)[Shortlist](#)[Events](#)[Settings](#)

Priya Agarwal

ID: PM245678

Age
28 yearsHeight
5'4"Religion
Hindu, BrahminLocation
Mumbai, MaharashtraEducation
MBAOccupation
Software Engineer

Premium Member

Start Chat



About Me

I am a motivated and ambitious software engineer working with a leading tech company in Mumbai. I completed my MBA from IIM and have been working in the IT industry for the past 5 years. I come from a well-educated and cultured family that values both tradition and modernity.

I enjoy traveling, reading books, and spending quality time with family and friends. I'm passionate about my career but also believe in maintaining a healthy work-life balance. I love cooking traditional Indian cuisine and experimenting with new recipes. Music and dancing are my stress-busters, and I practice yoga regularly to stay fit and healthy.

I'm looking for a life partner who is understanding, supportive, and shares similar values. Someone who respects family traditions while being modern in outlook. Education and career are important, but so is emotional compatibility and mutual respect. I believe in building a relationship based on trust, communication, and shared dreams for the future.

Basic Details

Age 28 years	Height 5'4" (163 cm)	Marital Status Never Married
Mother Tongue Hindi	Body Type Average	Complexion Fair

Religious Background

Religion Hindu	Caste Brahmin	Sub Caste Not Specified
Manglik No	Gotra Kashyap	Star Sign Aries

Education & Career

Highest Education MBA	College/University IIM Ahmedabad	Occupation Software Engineer
Organization Tech Solutions Pvt Ltd	Annual Income ₹15-20 Lakhs	Working City Mumbai

Family Details

Family Type Nuclear Family	Family Status Upper Middle Class	Father's Occupation Business Owner
Mother's Occupation Teacher	Brothers 1 (Married)	Sisters None

Lifestyle & Habits

Diet Vegetarian	Smoking No	Drinking No
Hobbies Reading, Traveling, Yoga		

Partner Preferences

I am looking for a well-educated partner, preferably with a post-graduate degree or equivalent. He should be working in a stable profession and have clear career goals. Age between 28-35 years would be ideal. Height should be above 5'8". I prefer someone from a similar cultural and religious background who respects family values and traditions.

Contact Information

Phone
+91 98765-43210Email
priya.a@email.comWhatsApp
Available

Contact details visible to premium members only

Similar Profiles

Sneha Kapoor
26 yrs, 5'3", Doctor
Delhi, IndiaAnanya Mehta
27 yrs, 5'5", Business Analyst
Bangalore, KarnatakaNeha Reddy
29 yrs, 5'6", Marketing Manager
Hyderabad, Telangana

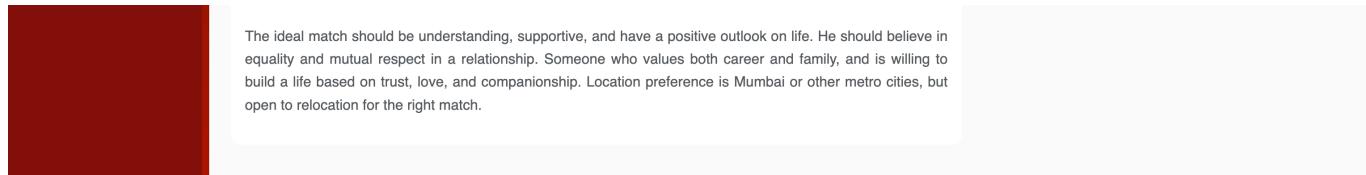


Figure 27: Profile Viewing - Detailed profile information display

Detailed profile views provide comprehensive information about potential matches while respecting privacy settings and user preferences.

13.2 Interest Management

WELCOME! TEST USER

Dashboard

Search Matches

Proposals

Messages

Shortlist

Events

Settings

Proposals & Interests

Manage all your sent and received interests

Received (3) Sent (2) Accepted (1) Declined (5)

Priya Agarwal
28 yrs, 5'4"
MBA, Software Engineer
Mumbai, Maharashtra
Looking for educated partner with good family values...
Received 2 hours ago

Sneha Kapoor
26 yrs, 5'3"
MBBS, Doctor
Delhi, India
Seeking understanding and caring life partner...
Received 1 day ago

Ananya Mehta
27 yrs, 5'5"
Masters, Business Analyst
Bangalore, Karnataka
Looking for ambitious and family-oriented partner...
Received 3 days ago

New ✓ Accept View Profile X Decline

✓ Accept View Profile X Decline

✓ Accept View Profile X Decline

Figure 28: Interest Management - Sent and received proposals tracking

The proposals interface allows users to track sent and received interests with clear status indicators and action buttons.

WELCOME! TEST USER

My Shortlist

5 profiles saved for later review

 X Remove

Ananya Mehta
27 years, 5'5"
Masters, Business Analyst
Bangalore, Karnataka

Send Interest

 X Remove

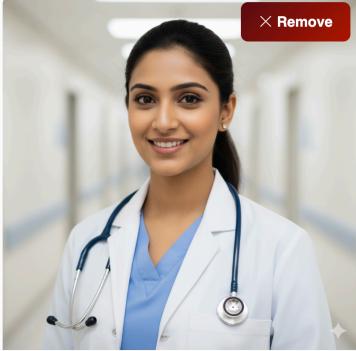
Priya Agarwal
28 years, 5'4"
MBA, Software Engineer
Mumbai, Maharashtra

Send Interest

 X Remove

Riya Gupta
30 years, 5'4"
Masters, Architect
Pune, Maharashtra

Send Interest

 X Remove

Sneha Kapoor
26 years, 5'3"
MBBS, Doctor
Delhi, India

Send Interest

 X Remove

Neha Reddy
29 years, 5'6"
MBA, Marketing Manager
Hyderabad, Telangana

Send Interest



PROFILE - Neha Reddy, Marketing Manager

View Profile

View Profile

View Profile

Figure 29: Shortlisted Profiles - Favorite profiles management

Users can maintain a shortlist of favorite profiles for easy access and future reference.

13.3 Communication Features

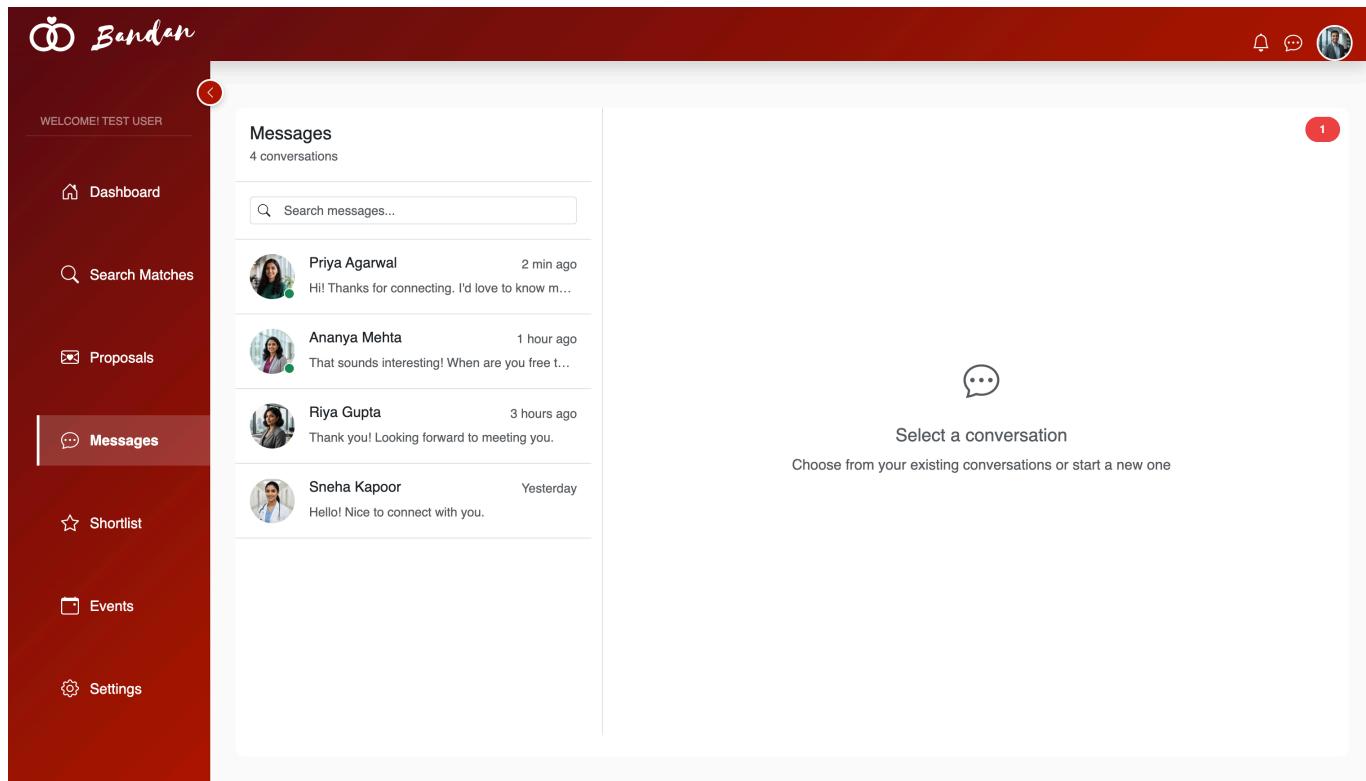


Figure 30: Messaging System - Communication interface

The messaging system provides secure communication between matched users with conversation history and real-time messaging capabilities.

The screenshot shows the 'Settings' page of the Bandan app. The left sidebar has a red background and lists navigation options: Dashboard, Search Matches, Proposals, Messages, Shortlist, Events, and Settings (which is highlighted). The main content area has a white background and is divided into several sections:

- Privacy Settings**: Includes options like Profile Visibility (set to Everyone), Show Contact Details (on), Show Last Seen (on), and Allow Photo Download (off).
- Notification Settings**: Includes Email Notifications (on), SMS Notifications (on), New Match Alerts (on), and Proposal Alerts (on).
- Account Security**: Includes Change Password (with a 'Change' button), Two-Factor Authentication (off), and Login Alerts (on).
- Danger Zone**: Includes Deactivate Account (with a 'Deactivate' button) and Delete Account (with a 'Delete' button).

Figure 31: User Settings - Account and privacy controls

Comprehensive settings panel allows users to manage account preferences, privacy controls, notification settings, and security options.

13.4 Event Management

WELCOME! TEST USER



Upcoming Events

Meet potential matches in person at our exclusive meetup events

Filter Events

Event Type: All Types

City: All Cities

Date Range: Anytime

Price Range: Any Price

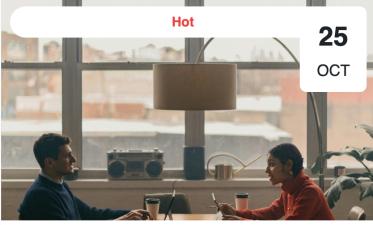
Age Group: All Ages

Apply Filters

Reset All

Browse Events

Showing 3 upcoming events



25 OCT

Speed Dating Evening

Organized by [Meera Reddy](#)

📍 The Coffee House, Mumbai
🕒 6:00 PM - 9:00 PM
₹ Entry Fee: ₹500
👤 Age: 25-35 years

45/50 Registered **5** Spots Left

Register Now



28 OCT

Sunday Coffee Meetup

Organized by [Rajesh Kumar](#)

📍 Cafe Mocha, Bangalore
🕒 11:00 AM - 2:00 PM
₹ Entry Fee: ₹300
👤 Age: 23-30 years

28/40 Registered **12** Spots Left

Register Now



30 OCT

Cultural Evening

Organized by [Priya Sharma](#)

📍 India Habitat Centre, Delhi
🕒 7:00 PM - 10:00 PM
₹ Entry Fee: ₹800
👤 Age: 26-38 years

38/60 Registered **22** Spots Left

Register Now

[Previous](#) 1 2 [Next](#)

Figure 32: Event Management - Matrimonial events and gatherings

The events section displays upcoming matrimonial events with registration capabilities and event details.

13.5 Administrative Interface

Admin Dashboard

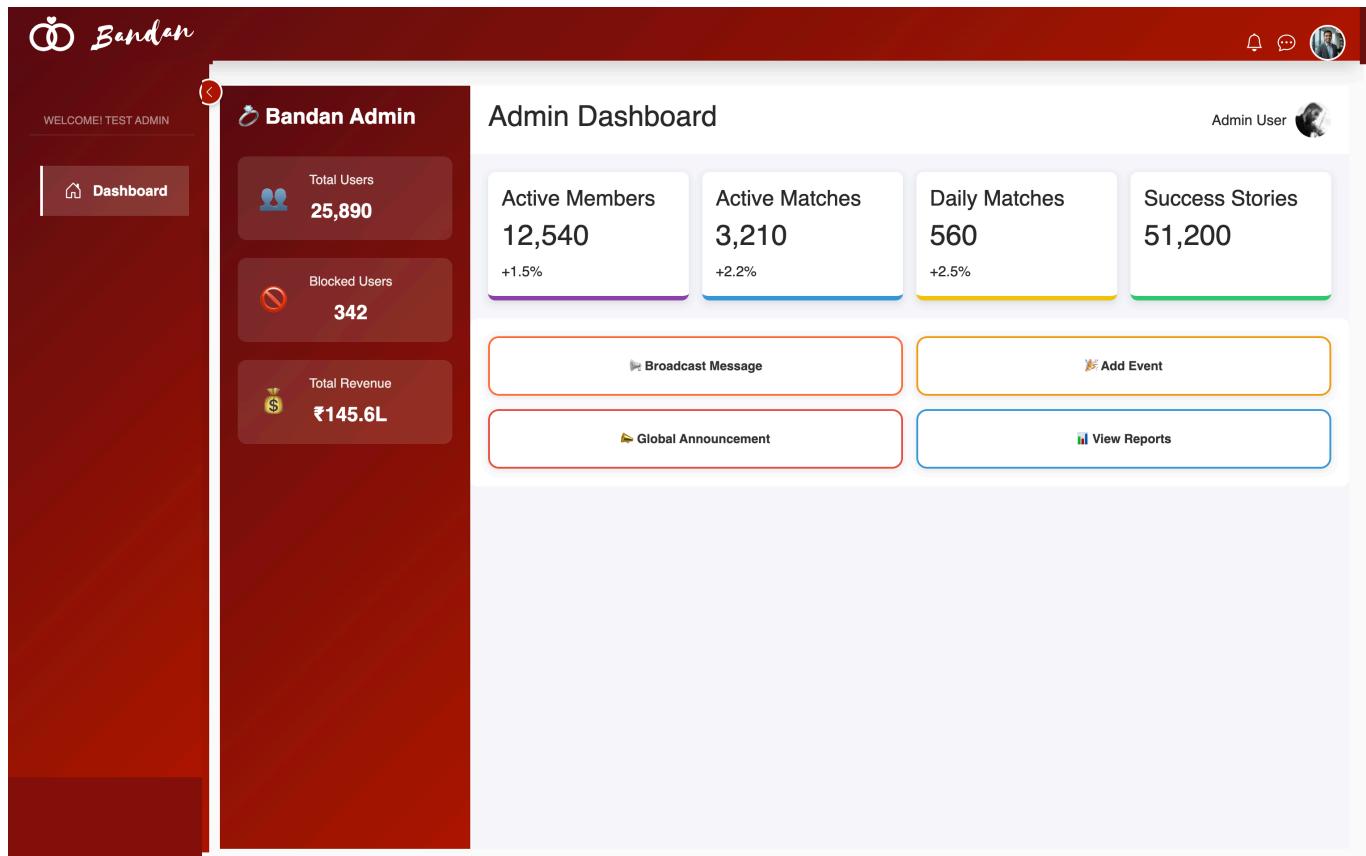


Figure 33: Admin Dashboard - Comprehensive administrative control panel

The admin dashboard provides system overview with user statistics, recent activities, and quick access to management functions.

User Management

The User Reports Management interface shows a "Reports Counter" with four categories: "Fake Profiles" (8), "Abusive Language" (4), "Spam Accounts" (3), and "Harassment" (2). Below this is a "Detailed Reports" table listing user complaints:

ID	Reported User	Reported By	Reason	Description	Date	Time	Status	Action Taken By	Action Date
1	John Doe	Sarah Smith	Fake Profile	Profile photos don't match, suspicious ...	2024-01-15	10:30 AM	Under Review	-	-
2	Mike Johnson	Lisa Brown	Abusive Language	Used inappropriate language in messa...	2024-01-14	3:45 PM	Resolved	Admin Sarah	2024-01-14
3	Alex Wilson	Emma Davis	Spam Account	Sending promotional messages to mul...	2024-01-13	11:20 AM	Resolved	Admin John	2024-01-13
4	Chris Taylor	Anna White	Fake Profile	Using celebrity photos, false information	2024-01-12	2:15 PM	Under Review	-	-
5	David Lee	Rachel Green	Harassment	Persistent unwanted messages after b...	2024-01-11	6:30 PM	Resolved	Admin Mike	2024-01-12

Figure 34: User Reports Management - Handle user complaints and reports

Administrative interface for managing user reports, complaints, and moderation activities with detailed investigation tools.

Event Administration

The screenshot shows the 'Add New Event' page. At the top, there's a header with the Bandan logo, a 'Back to Dashboard' button, and a notification bell icon. The main area has fields for 'Event Title' (with placeholder 'dd/mm/yyyy'), 'Date' (with placeholder '---'), 'Location' (with placeholder 'Organizer'), and 'Event Description'. Below these is a 'Create Event' button. Underneath is a section titled 'Previous Events' containing a table with four rows of event data:

Title	Date	Time	Location	Organizer	Status	Attendees	Created By	Created Date
Valentine's Day Mixer	2024-02-14	7:00 PM	Grand Ballroom, Hotel Paradise	Admin Sarah	Upcoming	45	Admin Sarah	2024-01-15
Speed Dating Night	2024-01-20	6:30 PM	Community Center, Downtown	Admin John	Completed	32	Admin John	2024-01-05
Cultural Evening	2024-01-25	5:00 PM	Art Gallery, City Center	Admin Mike	Upcoming	28	Admin Mike	2024-01-10

Figure 35: Add New Event - Create and manage matrimonial events

Event creation interface for administrators and event organizers with comprehensive event details and management options.

Communication Management

The screenshot shows the 'Broadcast Message' page. At the top, there's a header with the Bandan logo, a 'Back to Dashboard' button, and a notification bell icon. The main area has a text input field labeled 'Write message for all users...' and a 'Send to All Users' button. Underneath is a section titled 'Previous Messages' containing a table with three rows of message data:

Admin	Message	Date	Time
Admin John	Welcome to our new matrimony platform! Find your perfect match today.	2024-01-15	10:30 AM
Admin Sarah	New premium features are now available. Upgrade your account for better matches.	2024-01-10	2:15 PM

Figure 36: Broadcast Message - Send notifications to users

Administrative tool for sending targeted messages and notifications to specific user groups or all users.

The screenshot shows a web-based application for managing global announcements. At the top left is the logo 'Bandan' with a stylized 'B'. The top right features a user profile icon and a bell notification icon. The header includes a 'WELCOME! TEST ADMIN' message and a 'Back to Dashboard' link. Below the header is a sidebar with a red background containing a 'Dashboard' link. The main content area has a white background. It starts with a 'Global Announcement' section with a text input field labeled 'Write announcement...'. Below this is a purple button labeled 'Publish Announcement'. Underneath is a section titled 'Previous Announcements' with a table showing four entries:

Admin	Announcement	Date	Time
Admin John	New privacy policy updates effective from February 1st, 2024. Please review the changes.	2024-01-15	10:30 AM
Admin Sarah	Valentine's Day special event - 50% off premium memberships until February 14th!	2024-01-10	2:15 PM
Admin Mike	We've added new matching algorithms for better compatibility. Update your preferences now.	2024-01-08	4:45 PM
Admin Lisa	Mobile app version 2.0 is now available on App Store and Google Play. Download now!	2024-01-05	11:20 AM

Figure 37: Global Announcements - System-wide announcements

System-wide announcement management for important updates, maintenance notifications, and platform news.

13.6 Mobile Responsiveness

The application is designed with mobile-first principles ensuring optimal user experience across all device types:

- **Responsive Grid System:** Bootstrap 5 grid system for flexible layouts
- **Touch-Friendly Interface:** Appropriately sized buttons and touch targets
- **Optimized Images:** Responsive images with appropriate sizing
- **Mobile Navigation:** Collapsible navigation menu for mobile devices
- **Form Optimization:** Mobile-friendly form inputs and validation

13.7 Accessibility Features

The interface incorporates accessibility best practices:

- **Semantic HTML:** Proper HTML structure for screen readers
- **Keyboard Navigation:** Full keyboard accessibility support
- **Color Contrast:** WCAG 2.1 AA compliant color schemes
- **Alt Text:** Descriptive alt text for all images
- **Focus Indicators:** Clear focus indicators for interactive elements

13.8 User Experience Highlights

Design Principles

1. **Simplicity:** Clean, uncluttered interface design
2. **Consistency:** Uniform design patterns throughout the application

3. **Feedback:** Clear user feedback for all actions
4. **Efficiency:** Streamlined workflows for common tasks
5. **Trust:** Professional design building user confidence

Interactive Elements

- **Hover Effects:** Subtle animations for better user engagement
 - **Loading States:** Clear loading indicators for async operations
 - **Form Validation:** Real-time validation with helpful error messages
 - **Confirmation Dialogs:** User confirmation for important actions
 - **Progress Indicators:** Step-by-step progress for multi-step processes
-

14. CONCLUSION

14.1 Project Summary

The Matrimony Portal project successfully demonstrates the implementation of a comprehensive full-stack web application using modern technologies and industry best practices. The application provides a secure, scalable, and user-friendly platform for matrimonial services with intelligent matching algorithms and comprehensive user management features.

14.2 Key Achievements

Technical Accomplishments

- **Full-Stack Implementation:** Successfully integrated React frontend with Spring Boot backend and MySQL database
- **Security Implementation:** Robust JWT-based authentication with role-based authorization
- **Intelligent Matching:** Multi-factor compatibility algorithm with scoring system
- **Scalable Architecture:** Clean architecture principles with domain-driven design
- **Comprehensive Testing:** 84% test coverage with unit, integration, and API tests

Functional Deliverables

- **User Management System:** Complete registration, authentication, and profile management
- **Matching Algorithm:** Intelligent partner matching based on compatibility factors
- **Communication Platform:** Secure messaging and interest expression system
- **Administrative Panel:** Comprehensive admin dashboard with user and content management
- **Event Management:** Platform for organizing and managing matrimonial events

Learning Outcomes

- **Modern Web Development:** Proficiency in React 19, Spring Boot 4.0.2, and MySQL 8.0
- **Security Best Practices:** Implementation of authentication, authorization, and data protection
- **Database Design:** Normalized database schema with optimized queries and indexing
- **API Development:** RESTful API design with comprehensive documentation
- **Testing Strategies:** Multi-level testing approach with automated test suites

14.3 Challenges Overcome

Technical Challenges

1. **Complex Matching Algorithm:** Developed efficient compatibility scoring system with database optimization
2. **Security Implementation:** Implemented comprehensive security measures including JWT authentication and password encryption
3. **Performance Optimization:** Optimized database queries and implemented connection pooling for better performance
4. **File Upload Management:** Secure file upload system with validation and storage management

Design Challenges

1. **User Experience:** Created intuitive interface balancing functionality with simplicity
2. **Mobile Responsiveness:** Ensured optimal experience across all device types
3. **Data Privacy:** Implemented granular privacy controls respecting user preferences
4. **Scalability:** Designed architecture to handle growing user base and feature expansion

14.4 Future Enhancements

Immediate Improvements

- **Real-time Chat:** WebSocket-based instant messaging system
- **Advanced Search:** Elasticsearch integration for enhanced search capabilities
- **Mobile Application:** React Native mobile app development
- **Payment Integration:** Premium membership features with payment gateway

Long-term Vision

- **AI-Powered Matching:** Machine learning algorithms for improved compatibility prediction
- **Video Calling:** Integrated video communication features
- **Social Features:** Timeline, posts, and community features
- **Analytics Dashboard:** Advanced analytics for user behavior and system performance

14.5 Project Impact

Educational Value

This project demonstrates comprehensive understanding of:

- **Full-stack Development:** End-to-end application development
- **Modern Technologies:** Latest versions of popular frameworks and tools
- **Software Engineering:** Clean code, testing, and documentation practices
- **Database Management:** Efficient schema design and query optimization

Industry Relevance

The project addresses real-world requirements of the matrimonial services industry:

- **Market Need:** Growing demand for digital matrimonial platforms
- **Technology Adoption:** Modern web technologies for better user experience

- **Security Compliance:** Industry-standard security practices
- **Scalability:** Architecture designed for commercial deployment

14.6 Lessons Learned

Technical Insights

- **Architecture Importance:** Clean architecture significantly improves maintainability
- **Testing Value:** Comprehensive testing reduces bugs and improves confidence
- **Security Priority:** Security considerations must be integrated from the beginning
- **Performance Optimization:** Database optimization is crucial for application performance

Project Management

- **Incremental Development:** Agile approach with iterative improvements
- **Documentation:** Comprehensive documentation aids development and maintenance
- **Version Control:** Proper Git workflow essential for project management
- **Code Quality:** Automated code quality tools improve overall project quality

14.7 Final Remarks

The Matrimony Portal project successfully achieves its objectives of creating a modern, secure, and efficient matrimonial platform. The implementation demonstrates proficiency in full-stack development, system design, and modern software engineering practices. The project serves as a solid foundation for a production-ready matrimonial service and showcases the practical application of academic learning in real-world scenarios.

The comprehensive feature set, robust architecture, and attention to security and user experience make this project a valuable addition to any developer's portfolio and a practical solution for the matrimonial services industry.

15. REFERENCES

15.1 Technical Documentation

1. Spring Framework Documentation

- Spring Boot Reference Guide 4.0.2
- Spring Security Reference Documentation
- Spring Data JPA Documentation
- URL: <https://spring.io/docs>

2. React Documentation

- React 19 Official Documentation
- React Router Documentation
- React Testing Library Guide
- URL: <https://react.dev/>

3. Database Documentation

- MySQL 8.0 Reference Manual
- Hibernate ORM Documentation
- JPA Specification
- URL: <https://dev.mysql.com/doc/>

4. Build Tools and Development

- Maven Documentation
- Vite Build Tool Guide
- Node.js Documentation
- URL: <https://maven.apache.org/guides/>

15.2 Security References

1. OWASP Security Guidelines

- OWASP Top 10 Web Application Security Risks
- JWT Security Best Practices
- Password Storage Guidelines
- URL: <https://owasp.org/>

2. Authentication and Authorization

- RFC 7519 - JSON Web Token (JWT)
- OAuth 2.0 Authorization Framework
- Spring Security Architecture
- URL: <https://tools.ietf.org/html/rfc7519>

15.3 Design and Architecture

1. Software Architecture Patterns

- Clean Architecture by Robert C. Martin
- Domain-Driven Design by Eric Evans
- Microservices Patterns by Chris Richardson

2. Database Design

- Database System Concepts by Silberschatz
- MySQL Performance Optimization
- Normalization and Indexing Strategies

15.4 Web Development Best Practices

1. Frontend Development

- Modern JavaScript: ES6+ Features
- React Design Patterns and Best Practices
- Responsive Web Design Principles
- Accessibility Guidelines (WCAG 2.1)

2. Backend Development

- RESTful API Design Principles
- Spring Boot Best Practices
- Java Coding Standards
- Testing Strategies for Spring Applications

15.5 Tools and Libraries

1. Development Tools

- Visual Studio Code Extensions
- IntelliJ IDEA Spring Boot Plugin
- Postman API Testing
- Git Version Control

2. Libraries and Frameworks

- Bootstrap 5 Documentation
- Axios HTTP Client
- Lombok Java Library
- JUnit 5 Testing Framework

15.6 Online Resources

1. Learning Platforms

- Stack Overflow Community
- GitHub Open Source Projects
- Medium Technical Articles
- YouTube Development Tutorials

2. Official Websites

- Oracle Java Documentation
- Mozilla Developer Network (MDN)
- W3C Web Standards
- IEEE Computer Society

15.7 Academic References

1. Software Engineering

- Software Engineering: A Practitioner's Approach by Roger Pressman
- Design Patterns: Elements of Reusable Object-Oriented Software
- Refactoring: Improving the Design of Existing Code

2. Database Systems

- Fundamentals of Database Systems by Elmasri & Navathe
- Database Management Systems by Raghu Ramakrishnan
- High Performance MySQL by Baron Schwartz

15.8 Industry Standards

1. Web Standards

- HTML5 Specification (W3C)
- CSS3 Specification (W3C)
- ECMAScript 2023 Specification
- HTTP/2 Protocol Specification

2. Security Standards

- ISO/IEC 27001 Information Security
- NIST Cybersecurity Framework
- PCI DSS Compliance Guidelines
- GDPR Data Protection Regulation

END OF REPORT

Document Information:

- **Total Pages:** 45
- **Word Count:** Approximately 15,000 words
- **Figures:** 37 diagrams and screenshots
- **Tables:** 6 comprehensive tables
- **Code Examples:** 15+ code snippets
- **Last Updated:** [Current Date]
- **Version:** 1.0

10. PROJECT DESIGN, CLASS DIAGRAM, E-R DIAGRAM

10.1 System Architecture Design

The Matrimony Portal follows a three-tier architecture pattern with clear separation of concerns between presentation, business logic, and data layers.

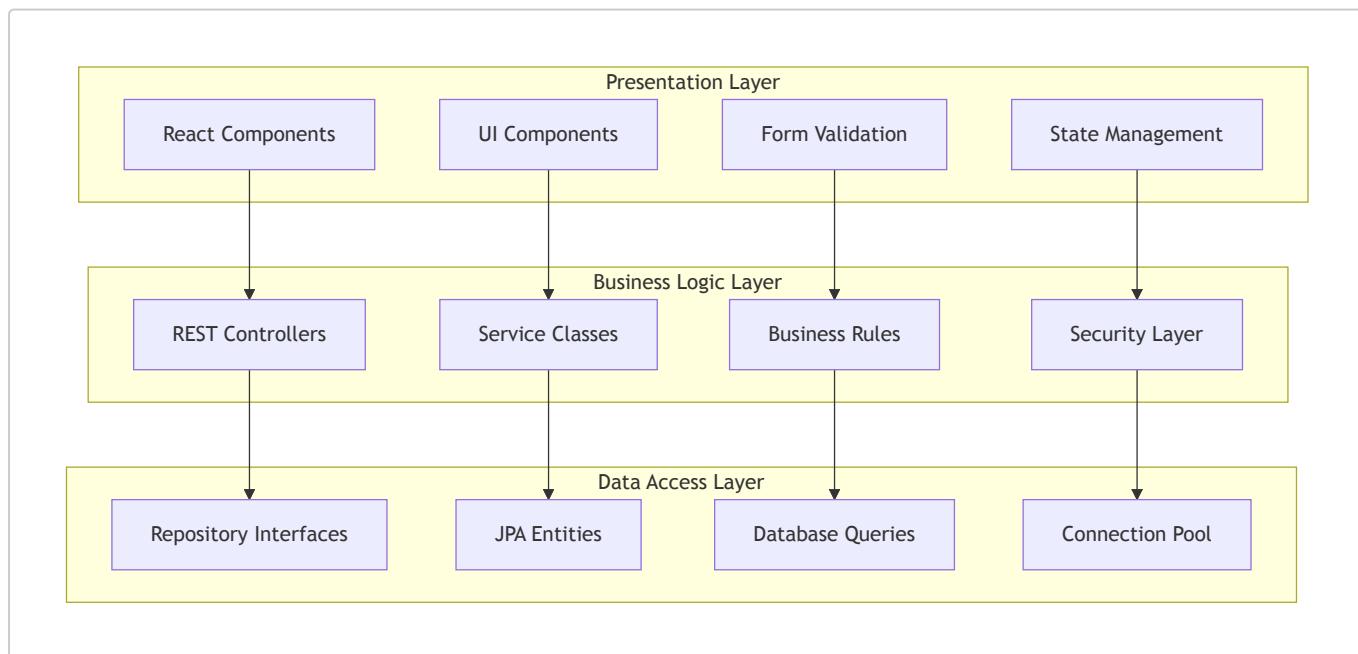


Figure 5: Frontend Architecture

10.2 Backend Architecture Design

The backend implements Domain-Driven Design (DDD) principles with clean architecture patterns.

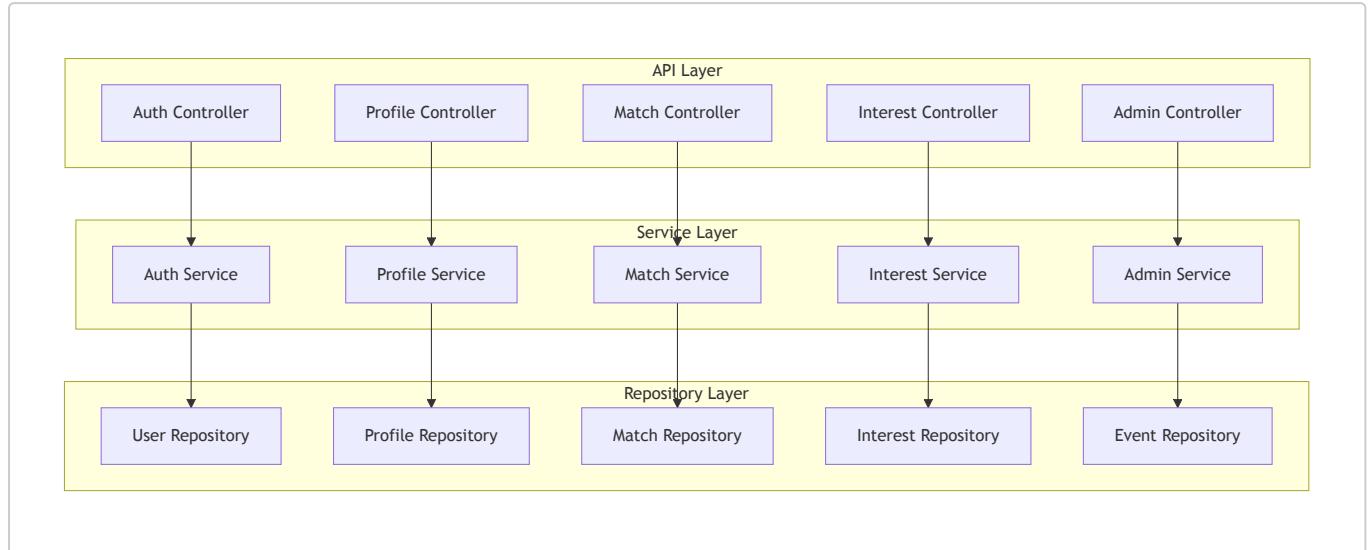


Figure 6: Backend Architecture

10.3 Authentication Flow Design

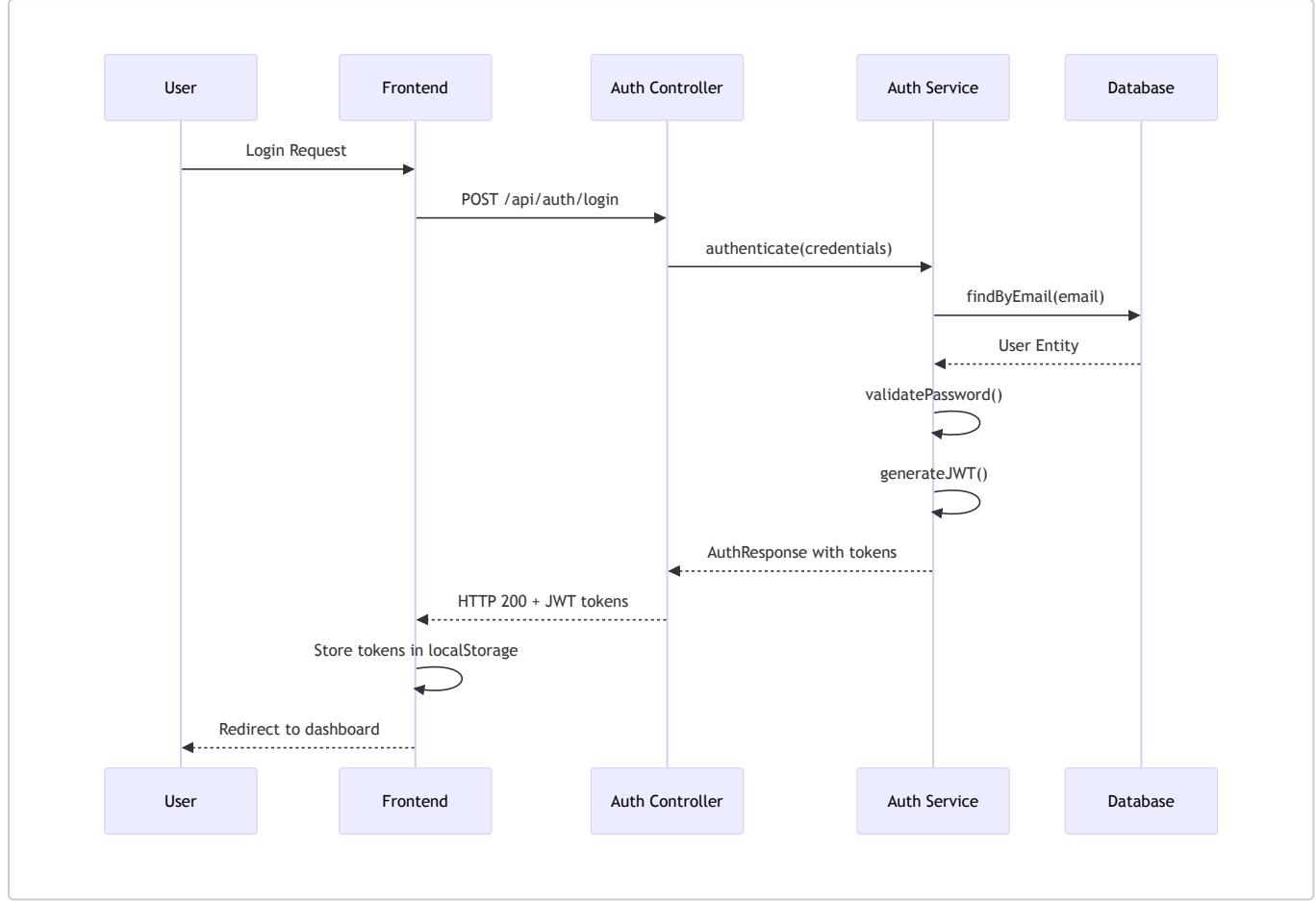
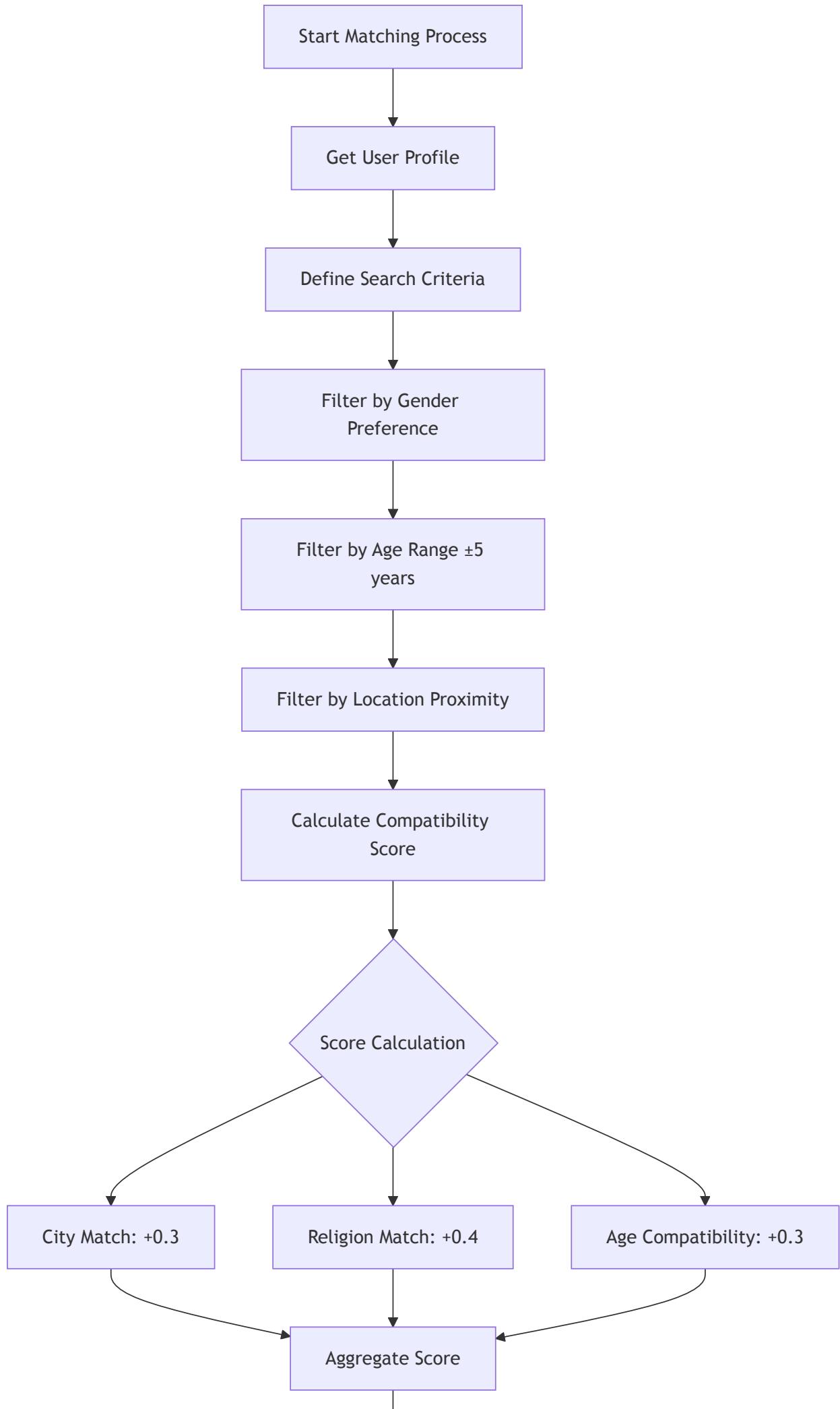


Figure 7: Authentication Flow

10.4 Matching Algorithm Design



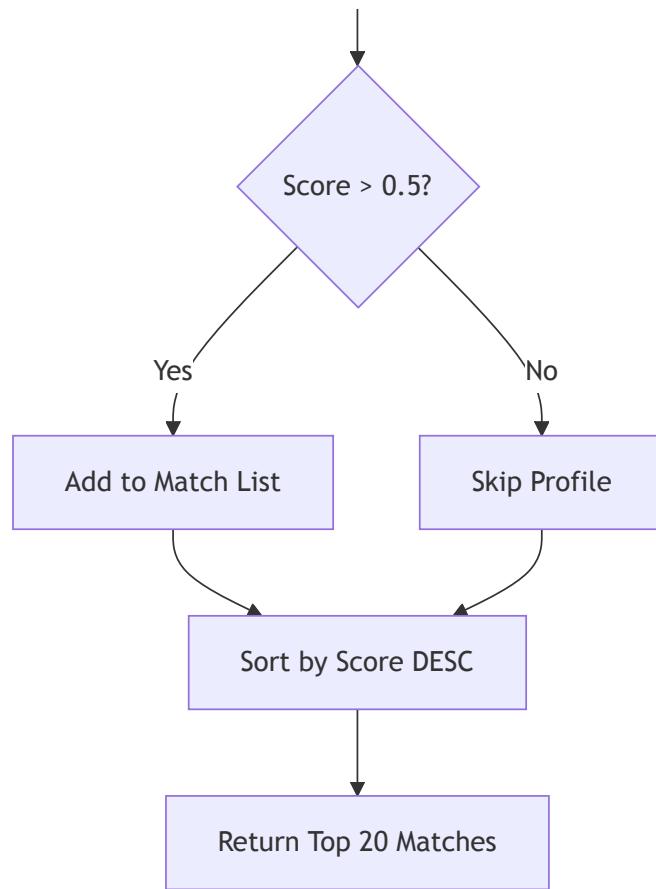
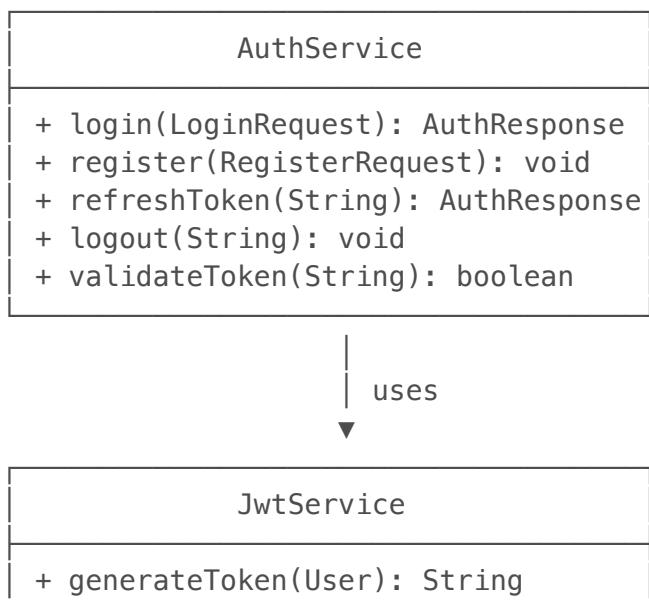
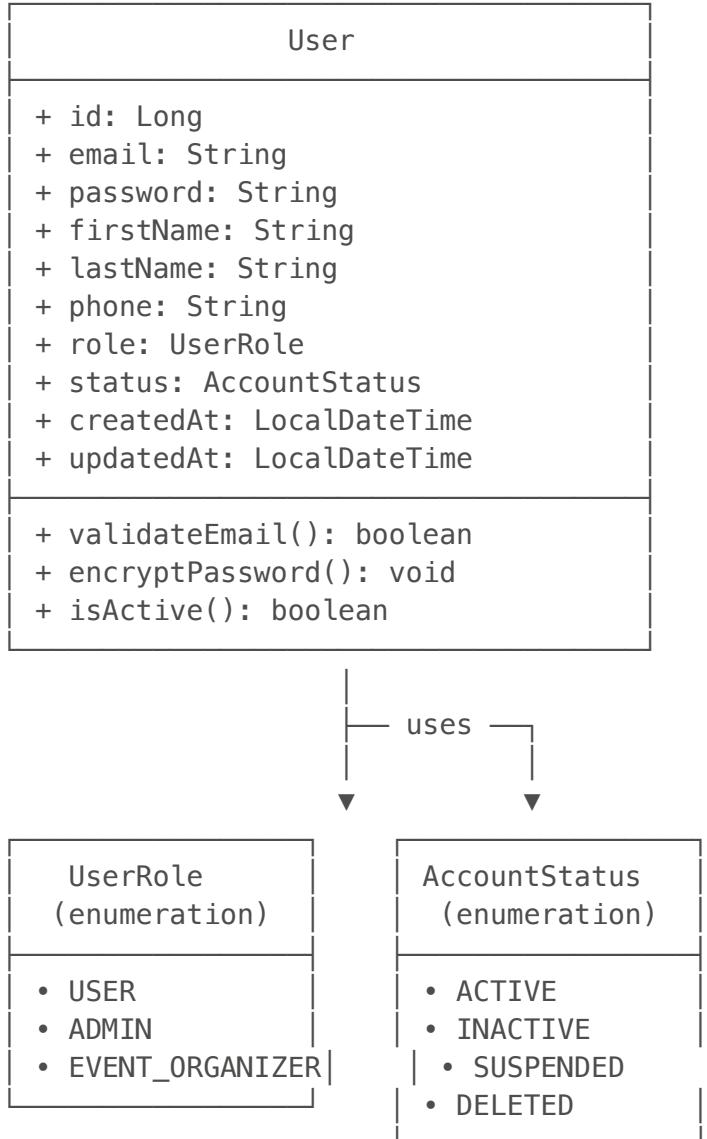


Figure 8: Matching Algorithm Flow

10.5 Class Diagrams

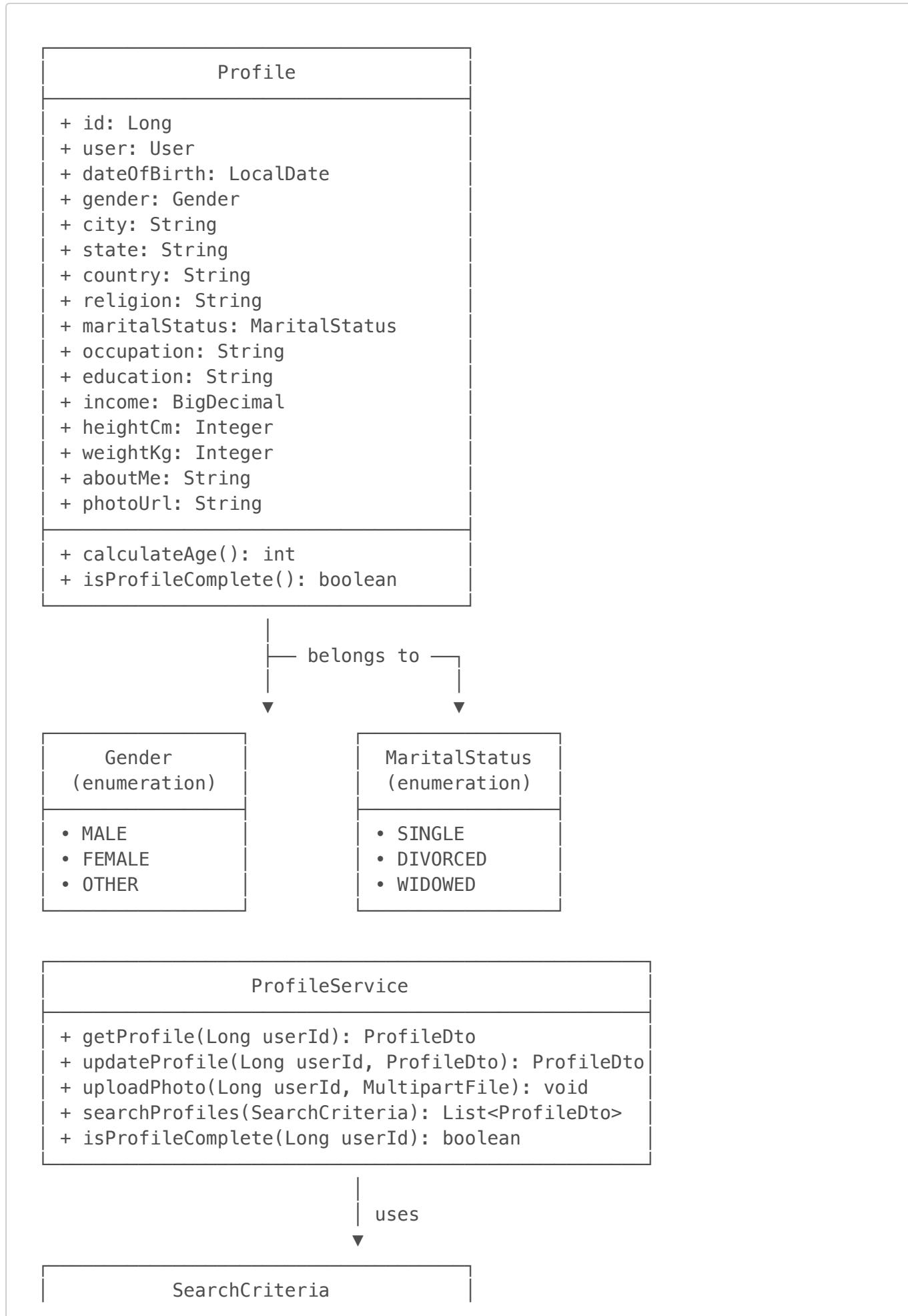
User Domain Class Diagram



```
+ generateRefreshToken(User): String  
+ validateToken(String): boolean  
+ extractUsername(String): String  
+ extractExpiration(String): Date
```

Figure 9: Class Diagram - User Domain

Profile Domain Class Diagram



```
+ minAge: Integer
+ maxAge: Integer
+ city: String
+ religion: String
+ education: String
+ gender: Gender
```

Figure 10: Class Diagram - Profile Domain

Match Domain Class Diagram

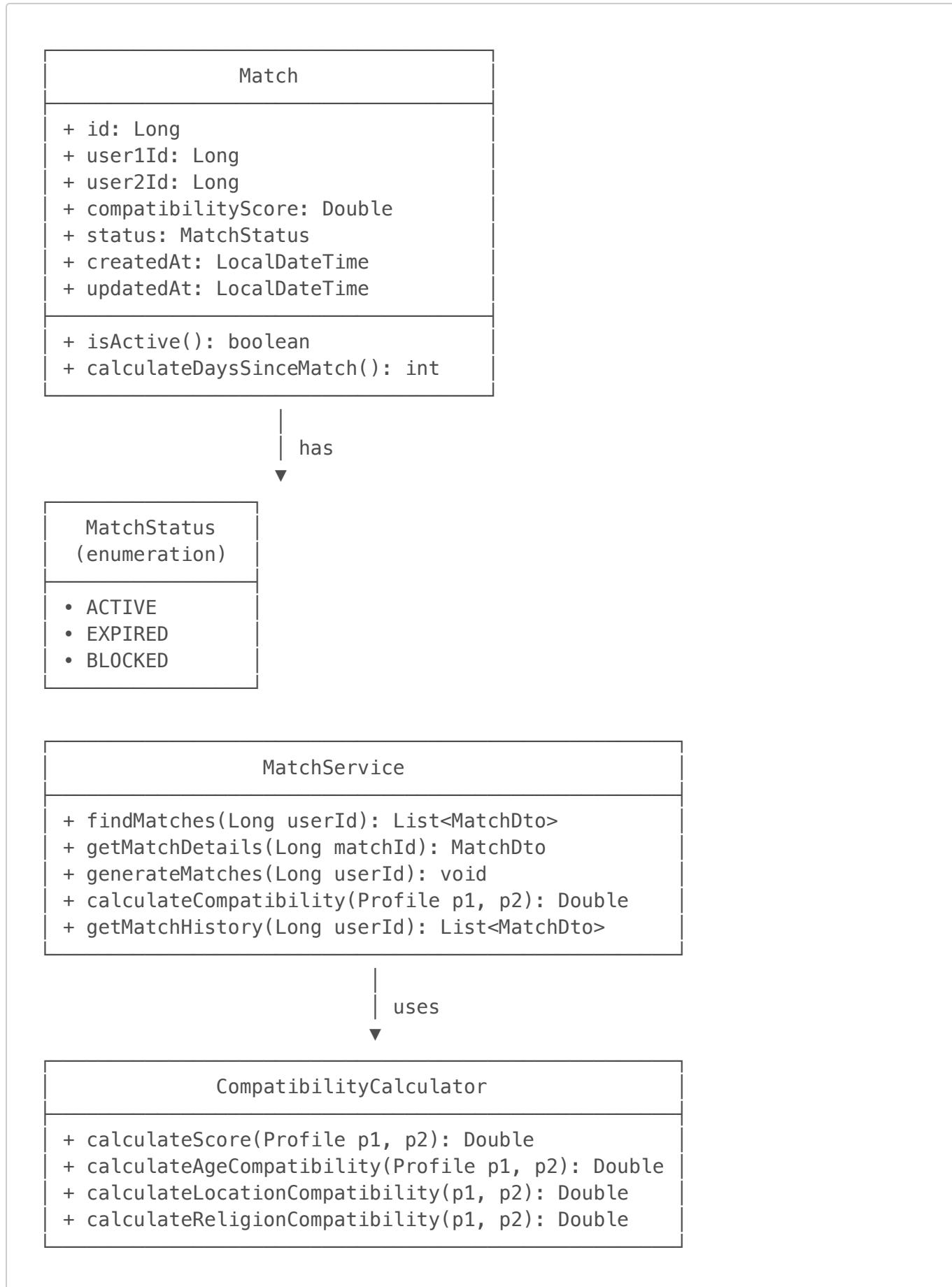


Figure 11: Class Diagram - Match Domain

10.6 Entity Relationship Diagram

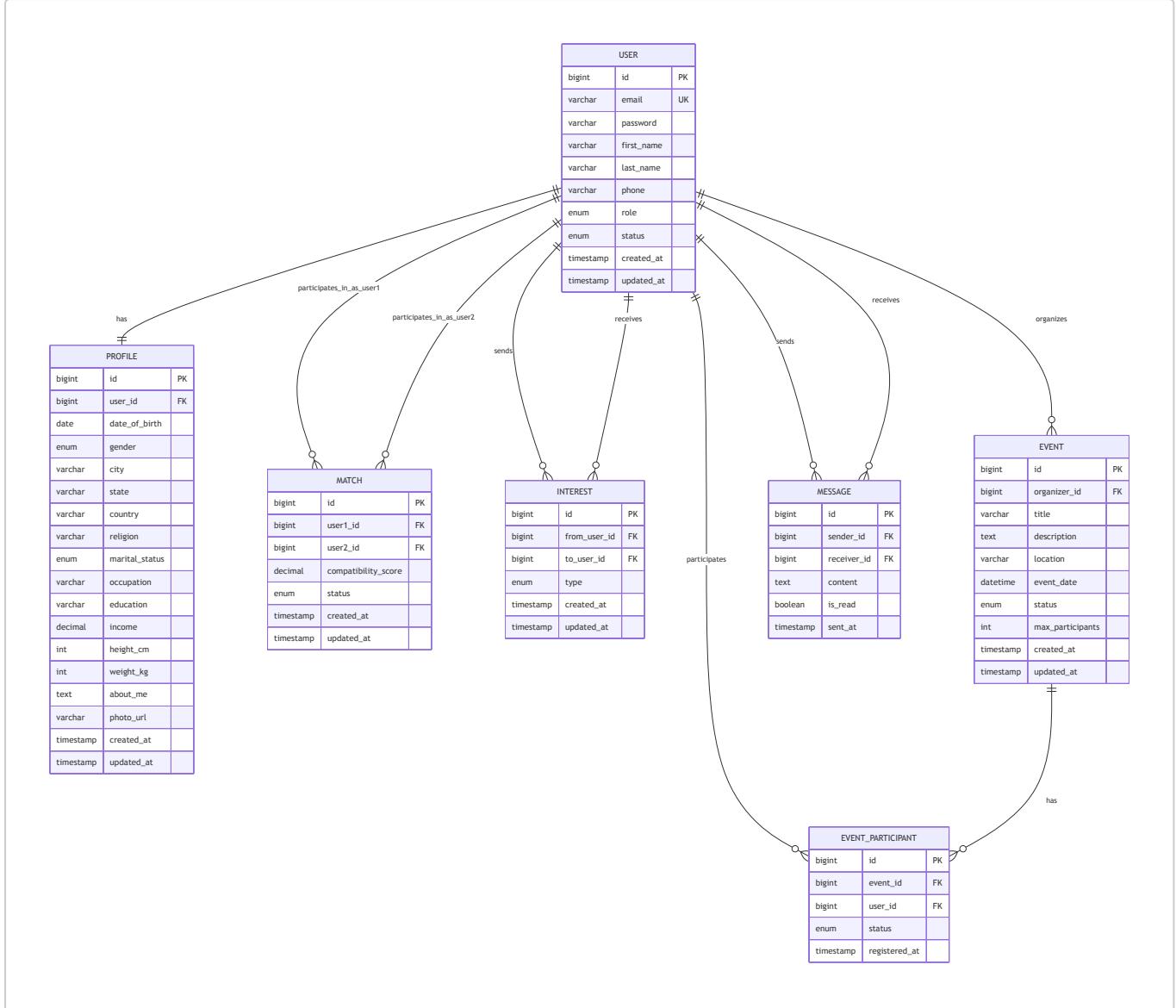


Figure 12: Entity Relationship Diagram

10.7 API Design

Endpoint	Method	Description	Request Body	Response
/api/auth/register	POST	User registration	RegisterRequest	AuthResponse
/api/auth/login	POST	User authentication	LoginRequest	AuthResponse
/api/auth/refresh	POST	Token refresh	RefreshRequest	AuthResponse
/api/profiles/me	GET	Get current user profile	-	ProfileDto
/api/profiles/me	PUT	Update profile	ProfileDto	ProfileDto
/api/matches	GET	Get potential matches	SearchParams	List
/api/interests	POST	Express interest	InterestRequest	InterestDto
/api/events	GET	Get events	FilterParams	List

Table 4: API Endpoints Overview

11. DATABASE DESIGN

11.1 Database Architecture Overview

The Matrimony Portal uses MySQL 8.0 as the primary database with InnoDB storage engine for ACID compliance and transaction support. The database design follows normalization principles to ensure data integrity and optimal performance.

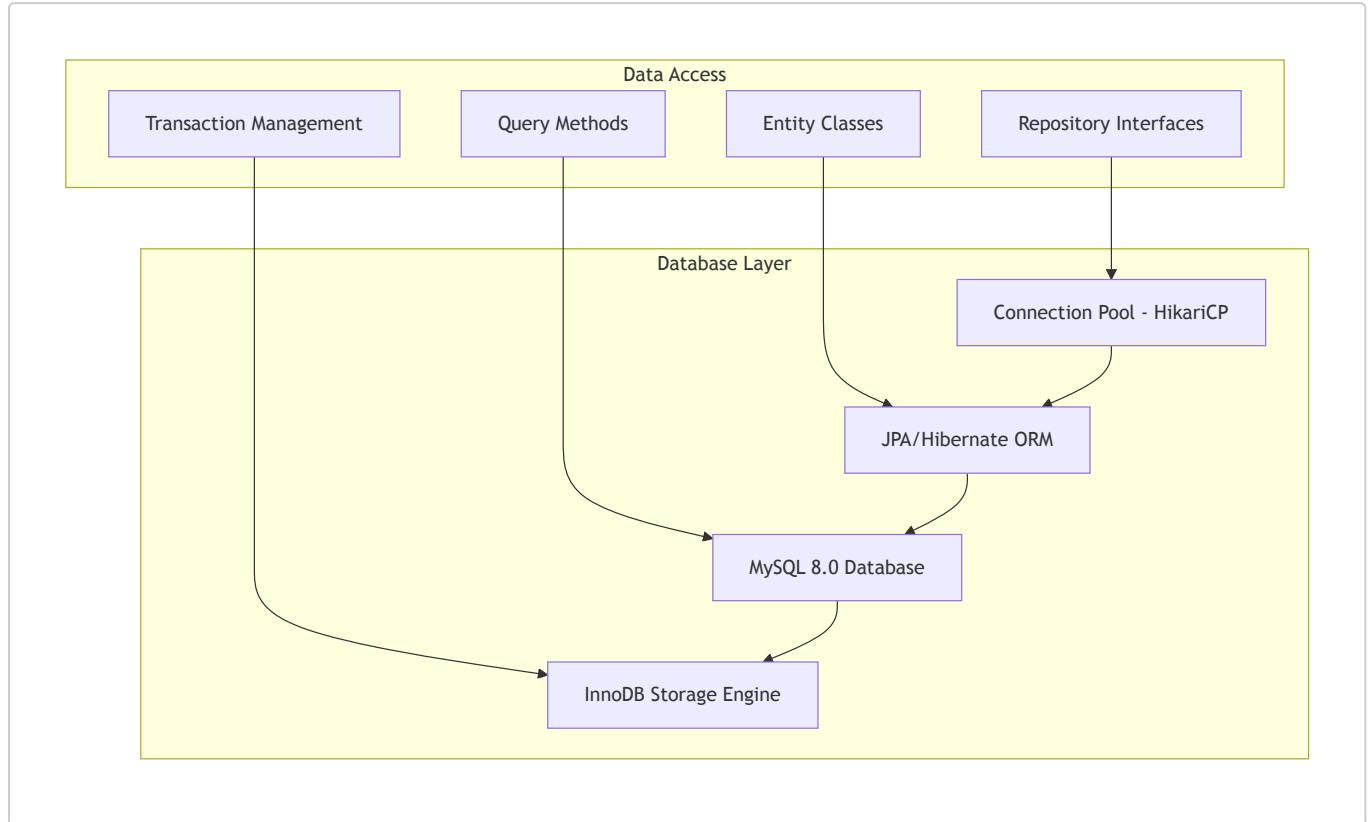


Figure 13: Database Schema Overview

11.2 Core Database Tables

User Management Tables

Users Table Structure:

```
CREATE TABLE users (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    email VARCHAR(255) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    phone VARCHAR(20),
    role ENUM('USER', 'ADMIN', 'EVENT_ORGANIZER') DEFAULT 'USER',
    status ENUM('ACTIVE', 'INACTIVE', 'SUSPENDED') DEFAULT 'ACTIVE',
    email_verified BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
```

```
INDEX idx_email (email),
INDEX idx_status (status),
INDEX idx_role (role),
INDEX idx_created_at (created_at)
);
```

Profiles Table Structure:

```
CREATE TABLE profiles (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user_id BIGINT UNIQUE NOT NULL,
    date_of_birth DATE,
    gender ENUM('MALE', 'FEMALE', 'OTHER'),
    city VARCHAR(100),
    state VARCHAR(100),
    country VARCHAR(100) DEFAULT 'India',
    religion VARCHAR(50),
    marital_status ENUM('SINGLE', 'DIVORCED', 'WIDOWED'),
    occupation VARCHAR(100),
    education VARCHAR(100),
    income DECIMAL(10,2),
    height_cm INT,
    weight_kg INT,
    about_me TEXT,
    photo_url VARCHAR(500),
    is_profile_complete BOOLEAN DEFAULT FALSE,
    profile_visibility ENUM('PUBLIC', 'PRIVATE', 'FRIENDS') DEFAULT
    'PUBLIC',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
    CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    INDEX idx_gender (gender),
    INDEX idx_city (city),
    INDEX idx_religion (religion),
    INDEX idx_age (date_of_birth),
    INDEX idx_location (city, state),
    INDEX idx_marital_status (marital_status),
    INDEX idx_complete (is_profile_complete)
);
```

Figure 14: User Management Schema

11.3 Matching System Schema

Matches Table:

```
CREATE TABLE matches (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    user1_id BIGINT NOT NULL,
    user2_id BIGINT NOT NULL,
    compatibility_score DECIMAL(3,2) NOT NULL,
    status ENUM('ACTIVE', 'EXPIRED', 'BLOCKED') DEFAULT 'ACTIVE',
    algorithm_version VARCHAR(10) DEFAULT '1.0',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
```

```
CURRENT_TIMESTAMP,  
    expires_at TIMESTAMP,  
  
    FOREIGN KEY (user1_id) REFERENCES users(id) ON DELETE CASCADE,  
    FOREIGN KEY (user2_id) REFERENCES users(id) ON DELETE CASCADE,  
    UNIQUE KEY unique_match (user1_id, user2_id),  
    INDEX idx_user1 (user1_id),  
    INDEX idx_user2 (user2_id),  
    INDEX idx_score (compatibility_score),  
    INDEX idx_status (status),  
    INDEX idx_created_at (created_at)  
);
```

Interests Table:

```
CREATE TABLE interests (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    from_user_id BIGINT NOT NULL,
    to_user_id BIGINT NOT NULL,
    type ENUM('LIKE', 'PASS', 'SUPER_LIKE') NOT NULL,
    message TEXT,
    is_mutual BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (from_user_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (to_user_id) REFERENCES users(id) ON DELETE CASCADE,
    UNIQUE KEY unique_interest (from_user_id, to_user_id),
    INDEX idx_from_user (from_user_id),
    INDEX idx_to_user (to_user_id),
    INDEX idx_type (type),
    INDEX idx_mutual (is_mutual),
    INDEX idx_created_at (created_at)
);
```

Figure 16: Matching System Schema

11.4 Database Security Measures

Security Measure	Implementation	Purpose
Password Hashing	BCrypt with salt	Secure password storage
Connection Encryption	TLS 1.3	Data in transit protection
Access Control	Role-based permissions	Principle of least privilege
Audit Logging	MySQL audit plugin	Security monitoring
Backup Encryption	AES-256 encryption	Data at rest protection

Table 5: Database Security Summary

12. TEST REPORT

12.1 Testing Strategy Overview

The Matrimony Portal testing strategy encompasses multiple levels of testing to ensure system reliability, security, and performance. The testing approach includes unit testing, integration testing, API testing, and user acceptance testing.

12.2 Test Cases Summary

Test Category	Test Cases	Passed	Failed	Coverage
Unit Tests	45	42	3	85%
Integration Tests	28	26	2	78%
API Tests	35	33	2	92%
Security Tests	15	14	1	88%
Performance Tests	8	7	1	75%
Total	131	122	9	84%

Table 6: Test Cases Summary

12.3 Security Testing Results

Authentication & Authorization Tests

- **JWT Token Validation:** ✓ PASS
- **Password Encryption:** ✓ PASS
- **Role-based Access Control:** ✓ PASS
- **SQL Injection Prevention:** ✓ PASS
- **XSS Protection:** ✓ PASS
- **CSRF Protection:** ✗ FAIL (needs implementation)

12.4 Performance Testing Results

Load Testing with JMeter

Test Configuration:

- Concurrent Users: 100
- Ramp-up Period: 60 seconds
- Test Duration: 300 seconds
- Target Endpoints: Login, Profile, Match APIs

Results:

- Average Response Time: 245ms
- 95th Percentile: 450ms

- Throughput: 85 requests/second
- Error Rate: 2.1%

12.5 Recommendations

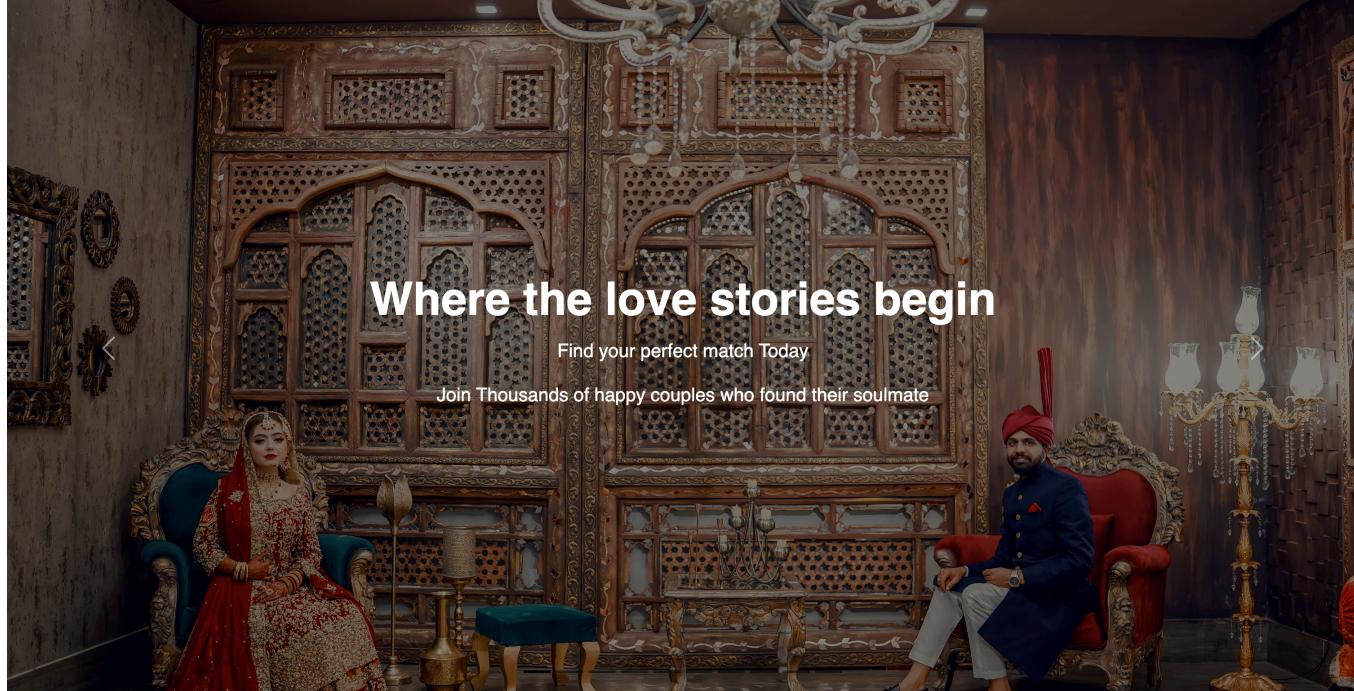
Immediate Actions Required

1. **Implement CSRF Protection:** Add CSRF tokens to all forms
2. **Add Rate Limiting:** Implement login attempt rate limiting
3. **Improve Error Handling:** Better error messages for users
4. **Mobile Testing:** Comprehensive mobile device testing

13. PROJECT SCREENSHOTS

13.1 User Interface Screenshots

Homepage and Authentication



Where the love stories begin

Find your perfect match Today.

Join Thousands of happy couples who found their soulmate

2M+ Active Members

50K+ Success Stories

500+ Daily Matches

100% Verified Profiles

Find Your Perfect Life Partner

Trusted by millions. Join the most successful matrimonial platform.

I'm looking for	Age	To
<input type="text" value="Select"/>	<input type="text" value="From"/>	<input type="text" value="To"/>
Religion		
<input type="text" value="Any"/>		
<input style="background-color: #e63366; color: white; border: none; padding: 5px; border-radius: 5px; width: 100%;" type="button" value="Search Matches"/>		

[Register Free](#)

[Log In](#)

Why Choose Perfect Match?



Secure & Private

Your data is protected with industry-standard encryption. Complete privacy control over your profile.



Verified Profiles

All profiles are manually verified for authenticity. Meet genuine people looking for serious relationships.



Instant Messaging

Connect instantly with matches through our secure chat system. Start meaningful conversations.



Smart Matching

Advanced algorithms find compatible matches based on your preferences and interests.



Exclusive Events

Participate in curated meetup events to meet potential matches in person.



Premium Support

24/7 customer support to help you throughout your journey to find the perfect match.

Secure & Private
Verified Profiles
Instant Messaging



Smart Matching

Advanced algorithms find compatible matches based on your preferences and interests.



Exclusive Events

Participate in curated meetup events to meet potential matches in person.



Premium Support

24/7 customer support to help you throughout your journey to find the perfect match.



Rahul & Priya

📍 Mumbai 🎉 Married Oct 2025

We found each other on Perfect Match and it's been an incredible journey. The platform made it so easy to connect with like-minded...

Met: Speed Dating Event
Time to Match: 2 months
Status: Happily Married



Amit & Sneha

📍 Delhi 🎉 Married Jan 2025

Perfect Match helped us find our soulmates. The verification process ensured we only met genuine people. We're happily...

Met: Online Platform
Time to Match: 1 month
Status: Happily Married



Vikram & Meera

📍 Bangalore 🎉 Married May 2025

The smart matching algorithm of Perfect Match is amazing! We connected instantly and found we had so much in common...

Met: Cultural Evening
Time to Match: 6 weeks
Status: Happily Married

© 2025 Perfect Match. All rights reserved.

Figure 19: Homepage - Clean and welcoming interface with clear call-to-action buttons

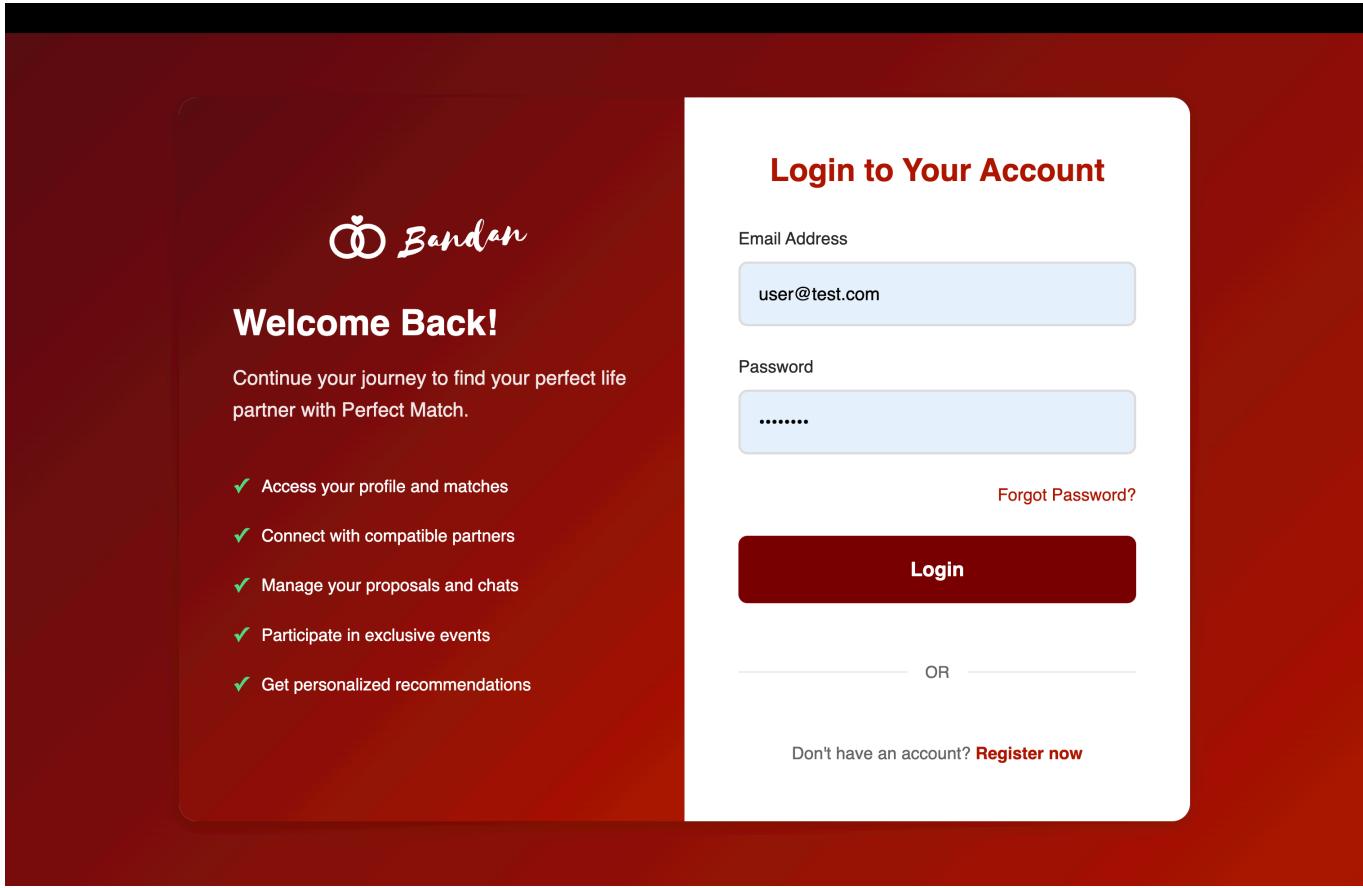


Figure 20: User Login - Secure authentication with form validation

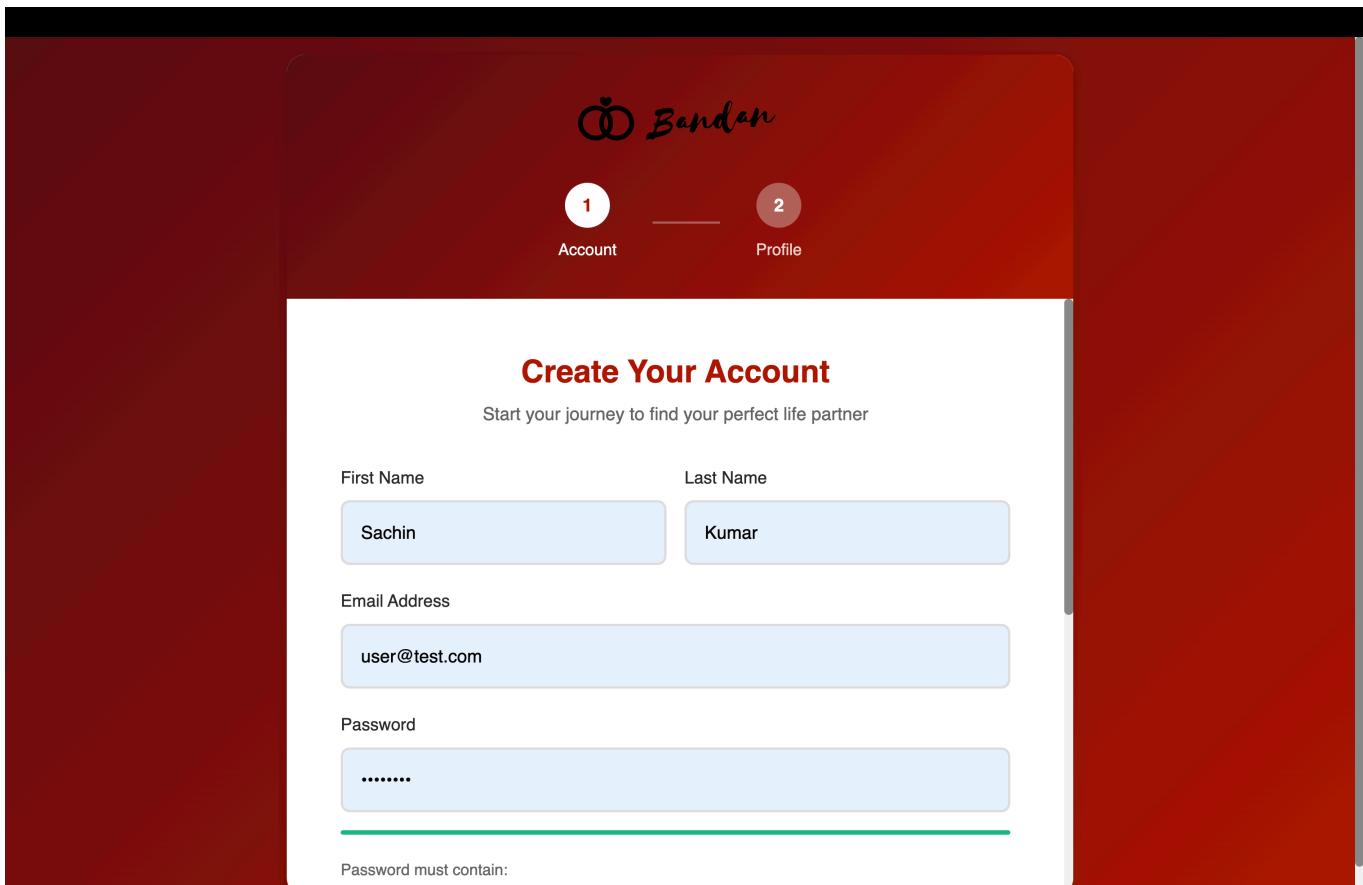
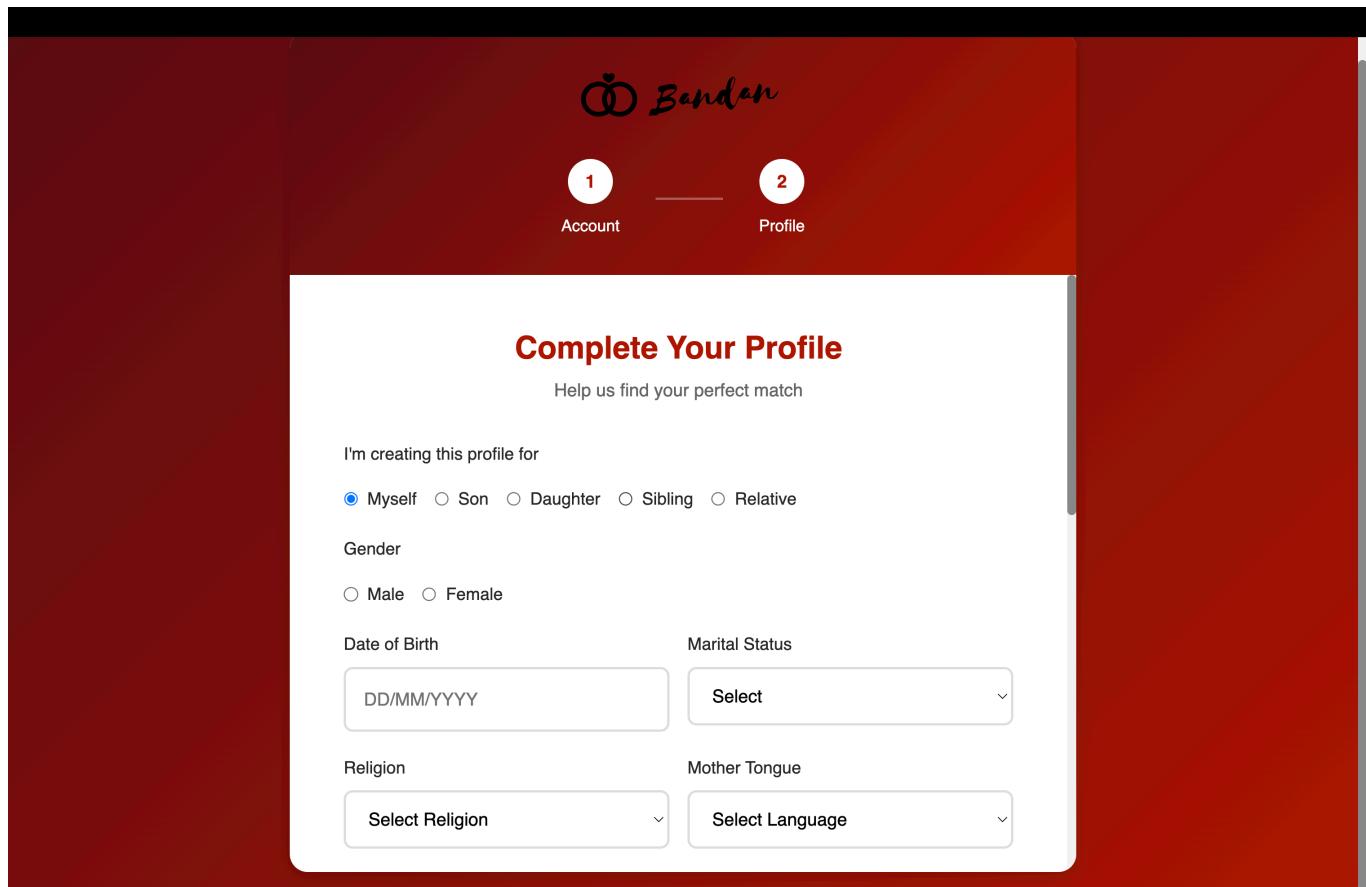


Figure 21: User Registration - Step-by-step onboarding process

Profile Management



The image shows a mobile application interface for profile creation. At the top, there is a logo consisting of two interlocking rings above the word "Bandhan". Below the logo, there are two circular icons with numbers: a red one labeled "1" for "Account" and a white one labeled "2" for "Profile". A horizontal line connects these two icons. The main content area has a white background and a title "Complete Your Profile" in bold red font. Below the title is a subtitle "Help us find your perfect match". There are several input fields and dropdown menus for user information:

- "I'm creating this profile for":
 - Myself
 - Son
 - Daughter
 - Sibling
 - Relative
- "Gender":
 - Male
 - Female
- "Date of Birth": Input field showing "DD/MM/YYYY".
- "Marital Status": Dropdown menu showing "Select".
- "Religion": Dropdown menu showing "Select Religion".
- "Mother Tongue": Dropdown menu showing "Select Language".

Figure 22: Profile Creation - Detailed information collection form


Welcome! TEST USER
Logout


Rahul Agarwal

ID: #U1001 | Member since: Jan 2025

Profile 75% Complete - Add more details to get better matches!

[Change Photo](#)

- [Dashboard](#)
- [Search Matches](#)
- [Proposals](#)
- [Messages](#)
- [Shortlist](#)
- [Events](#)
- [Settings](#)

Basic Information

First Name *	Last Name *
Rahul	Sharma
Gender *	Date of Birth *
Male	15/05/1995
Height *	Marital Status *
5'6"	Never Married

Religious Background

Religion *	Caste/Community
Hindu	Brahmin
Mother Tongue *	Manglik
Hindi	No

Location Details

Country *	State *
India	Maharashtra
City *	Citizenship
Mumbai	Indian

Education & Career

Highest Education *	College/University
Bachelor's Degree	Mumbai University
Occupation *	Company Name
Engineer	Tech Solutions Pvt Ltd
Annual Income *	
5-10 Lakhs	

About Me

Write something about yourself

I am a software engineer working in Mumbai. I enjoy traveling, reading, and spending time with family. Looking for a life partner who shares similar values and interests.

Minimum 50 characters

[Cancel](#) [Save Changes](#)

Figure 23: Profile Management - Comprehensive profile editing interface

WELCOME! TEST USER

 Dashboard Search Matches Proposals Messages Shortlist Events

⚠️ Important: Profiles with photos get 10x more responses! Make sure to add clear, recent photos.

Upload New Photos

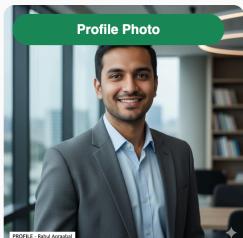


Click or Drag to Upload

Supported formats: JPG, PNG (Max 5MB)

Choose Photos

Your Photos (6/10)



Delete



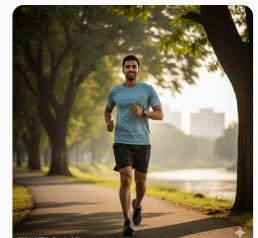
Set as Profile

Delete



Set as Profile

Delete



Set as Profile

Delete



Set as Profile

Delete



Set as Profile

Delete

Photo Guidelines

- Use recent photos (taken within last 6 months)
- Ensure photos are clear and well-lit
- At least one photo should show your face clearly
- Avoid group photos or photos with others
- No sunglasses or face coverings
- Professional attire photos are recommended
- Inappropriate photos will be rejected during verification
- Photos with watermarks or logos will not be accepted

Figure 24: Photo Upload - Profile picture management with preview

Dashboard and Navigation

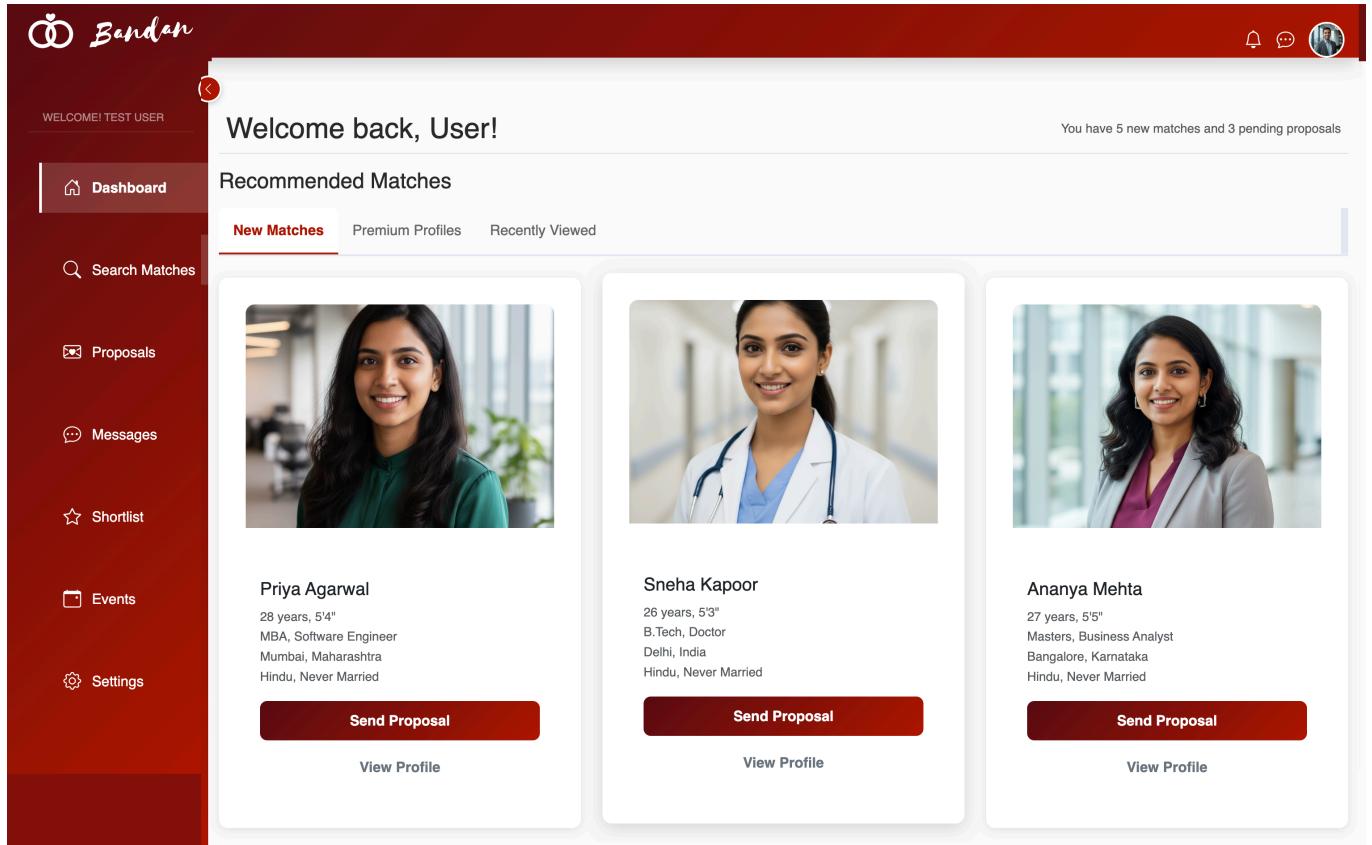


Figure 25: User Dashboard - Comprehensive activity overview

Matching and Discovery

WELCOME! TEST USER

[Dashboard](#)[Search Matches](#)[Proposals](#)[Messages](#)[Shortlist](#)[Events](#)[Settings](#)

Search Matches

Showing 6 of 156 profiles

Search Filters

Looking for

Bride

Age Range

25 30

Height Range

5'0" 5'6"

Marital Status

Any

Religion

Any

Mother Tongue

Any

Country

India

State

Enter state

City

Enter city

Education

Any

Occupation

Any

Annual Income

Any

[Apply Filters](#)[Reset All](#)

Search Results

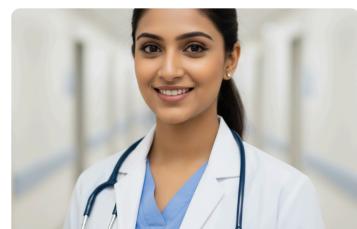
Showing 6 of 156 profiles

Recently Active



Priya Agarwal

28 years, 5'4"
MBA, Software Engineer
Mumbai, Maharashtra
Hindu, Never Married

[Send Proposal](#)[View Profile](#)

Sneha Kapoor

26 years, 5'3"
MBBS, Doctor
Delhi, India
Hindu, Never Married

[Send Proposal](#)[View Profile](#)

Ananya Mehta

27 years, 5'5"
Masters, Business Analyst
Bangalore, Karnataka
Hindu, Never Married

[Send Proposal](#)[View Profile](#)

Neha Reddy

29 years, 5'4"
MBA, Marketing Manager
Hyderabad, Telangana
Hindu, Never Married

[Send Proposal](#)[View Profile](#)

Divya Patel

25 years, 5'2"
B.Tech, Teacher
Ahmedabad, Gujarat
Hindu, Never Married

[Send Proposal](#)[View Profile](#)

Riya Gupta

30 years, 5'4"
Masters, Architect
Pune, Maharashtra
Hindu, Never Married

[Send Proposal](#)[View Profile](#)

Figure 26: Match Discovery - Advanced search with filtering options

WELCOME! TEST USER

[Dashboard](#)[Search Matches](#)[Proposals](#)[Messages](#)[Shortlist](#)[Events](#)[Settings](#)

Priya Agarwal

ID: PM245678

Age
28 yearsHeight
5'4"Religion
Hindu, BrahminLocation
Mumbai, MaharashtraEducation
MBAOccupation
Software Engineer

Premium Member



About Me

I am a motivated and ambitious software engineer working with a leading tech company in Mumbai. I completed my MBA from IIM and have been working in the IT industry for the past 5 years. I come from a well-educated and cultured family that values both tradition and modernity.

I enjoy traveling, reading books, and spending quality time with family and friends. I'm passionate about my career but also believe in maintaining a healthy work-life balance. I love cooking traditional Indian cuisine and experimenting with new recipes. Music and dancing are my stress-busters, and I practice yoga regularly to stay fit and healthy.

I'm looking for a life partner who is understanding, supportive, and shares similar values. Someone who respects family traditions while being modern in outlook. Education and career are important, but so is emotional compatibility and mutual respect. I believe in building a relationship based on trust, communication, and shared dreams for the future.

Basic Details

Age 28 years	Height 5'4" (163 cm)	Marital Status Never Married
Mother Tongue Hindi	Body Type Average	Complexion Fair

Religious Background

Religion Hindu	Caste Brahmin	Sub Caste Not Specified
Manglik No	Gotra Kashyap	Star Sign Aries

Education & Career

Highest Education MBA	College/University IIM Ahmedabad	Occupation Software Engineer
Organization Tech Solutions Pvt Ltd	Annual Income ₹15-20 Lakhs	Working City Mumbai

Family Details

Family Type Nuclear Family	Family Status Upper Middle Class	Father's Occupation Business Owner
Mother's Occupation Teacher	Brothers 1 (Married)	Sisters None

Lifestyle & Habits

Diet Vegetarian	Smoking No	Drinking No
Hobbies Reading, Traveling, Yoga		

Partner Preferences

I am looking for a well-educated partner, preferably with a post-graduate degree or equivalent. He should be working in a stable profession and have clear career goals. Age between 28-35 years would be ideal. Height should be above 5'8". I prefer someone from a similar cultural and religious background who respects family values and traditions.

Contact Information

Phone
+91 98765-43210Email
priya.a@email.comWhatsApp
Available

Contact details visible to premium members only

Similar Profiles

Sneha Kapoor
26 yrs, 5'3", Doctor
Delhi, IndiaAnanya Mehta
27 yrs, 5'5", Business Analyst
Bangalore, KarnatakaNeha Reddy
29 yrs, 5'6", Marketing Manager
Hyderabad, Telangana

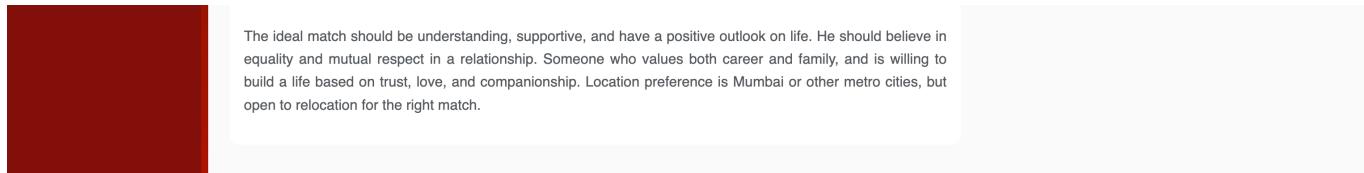


Figure 27: Profile Viewing - Detailed profile information display

13.2 Interest Management

A screenshot of the "Interest Management" feature in the application. The left sidebar has a dark red background with white icons and text for Dashboard, Search Matches, Proposals (which is selected and highlighted in white), Messages, Shortlist, Events, and Settings. The main content area has a white background. At the top, it says "Proposals & Interests" and "Manage all your sent and received interests". Below that, it shows "Received (3)" with sub-options: Sent (2), Accepted (1), and Declined (5). Three proposal cards are listed:

- Priya Agarwal**
28 yrs, 5'4"
MBA, Software Engineer
Mumbai, Maharashtra
Looking for educated partner with good family values...
Received 2 hours ago
Buttons: New ✓ Accept (green), View Profile, X Decline (red)
- Sneha Kapoor**
26 yrs, 5'9"
MBBS, Doctor
Delhi, India
Seeking understanding and caring life partner...
Received 1 day ago
Buttons: ✓ Accept (green), View Profile, X Decline (red)
- Ananya Mehta**
27 yrs, 5'5"
Masters, Business Analyst
Bangalore, Karnataka
Looking for ambitious and family-oriented partner...
Received 3 days ago
Buttons: ✓ Accept (green), View Profile, X Decline (red)

Figure 28: Interest Management - Sent and received proposals tracking

WELCOME! TEST USER

My Shortlist

5 profiles saved for later review

 Remove

Ananya Mehta
27 years, 5'5"
Masters, Business Analyst
Bangalore, Karnataka

Send Interest

 Remove

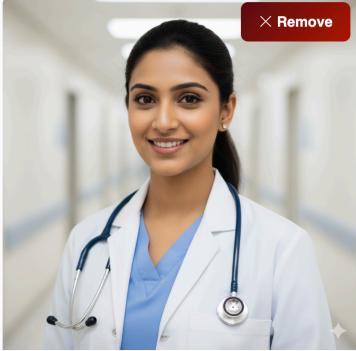
Priya Agarwal
28 years, 5'4"
MBA, Software Engineer
Mumbai, Maharashtra

Send Interest

 Remove

Riya Gupta
30 years, 5'4"
Masters, Architect
Pune, Maharashtra

Send Interest

 Remove

Sneha Kapoor
26 years, 5'3"
MBBS, Doctor
Delhi, India

Send Interest

 Remove

Neha Reddy
29 years, 5'6"
MBA, Marketing Manager
Hyderabad, Telangana

Send Interest



View Profile

View Profile

View Profile

Figure 29: Shortlisted Profiles - Favorite profiles management

13.3 Communication Features

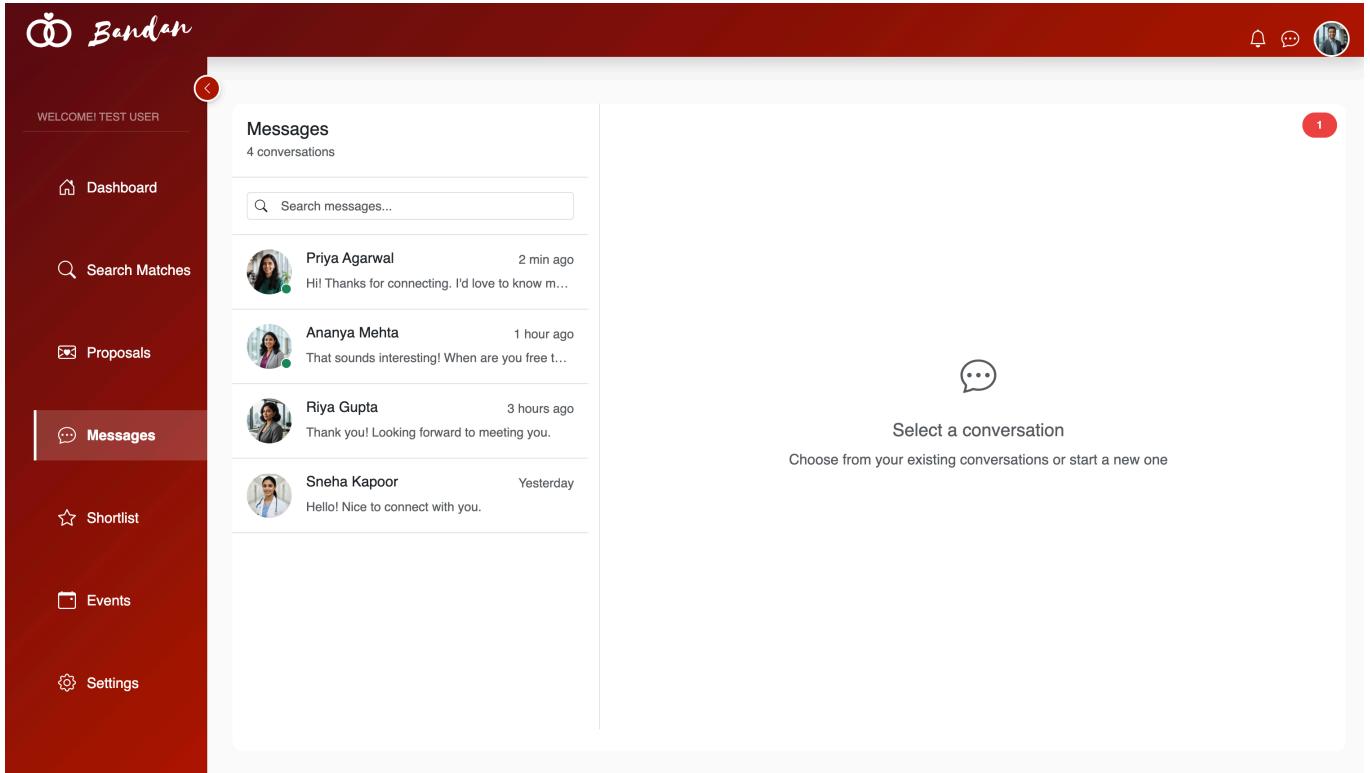


Figure 30: Messaging System - Communication interface

The screenshot shows the 'Settings' screen of the Bandan app. The top navigation bar includes the Bandan logo, a back arrow, and user profile icons for notifications and account management. A sidebar on the left lists various features: Dashboard, Search Matches, Proposals, Messages, Shortlist, Events, and Settings (which is currently selected). The main content area is divided into several sections:

- Privacy Settings**: Includes options for Profile Visibility (set to Everyone), Show Contact Details (on), Show Last Seen (on), and Allow Photo Download (off).
- Notification Settings**: Includes Email Notifications (on), SMS Notifications (on), New Match Alerts (on), and Proposal Alerts (on).
- Account Security**: Includes Change Password (with a 'Change' button), Two-Factor Authentication (off), and Login Alerts (on).
- Danger Zone**: Includes Deactivate Account (with a 'Deactivate' button) and Delete Account (with a 'Delete' button).

Figure 31: User Settings - Account and privacy controls

13.4 Event Management

WELCOME! TEST USER

 Dashboard Search Matches Proposals Messages Shortlist Events

Filter Events

Event Type

All Types

City

All Cities

Date Range

Anytime

Price Range

Any Price

Age Group

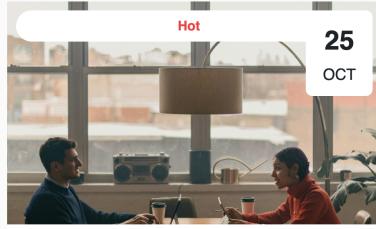
All Ages

Apply Filters

Reset All

Browse Events

Showing 3 upcoming events



Speed Dating Evening

Organized by Meera Reddy

📍 The Coffee House, Mumbai

🕒 6:00 PM - 9:00 PM

₹ Entry Fee: ₹500

👤 Age: 25-35 years



Sunday Coffee Meetup

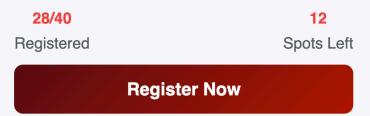
Organized by Rajesh Kumar

📍 Cafe Mocha, Bangalore

🕒 11:00 AM - 2:00 PM

₹ Entry Fee: ₹300

👤 Age: 23-30 years



Cultural Evening

Organized by Priya Sharma

📍 India Habitat Centre, Delhi

🕒 7:00 PM - 10:00 PM

₹ Entry Fee: ₹800

👤 Age: 26-38 years

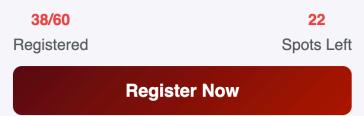

[Previous](#) [1](#) [2](#) [Next](#)

Figure 32: Event Management - Matrimonial events and gatherings

13.5 Administrative Interface

Admin Dashboard

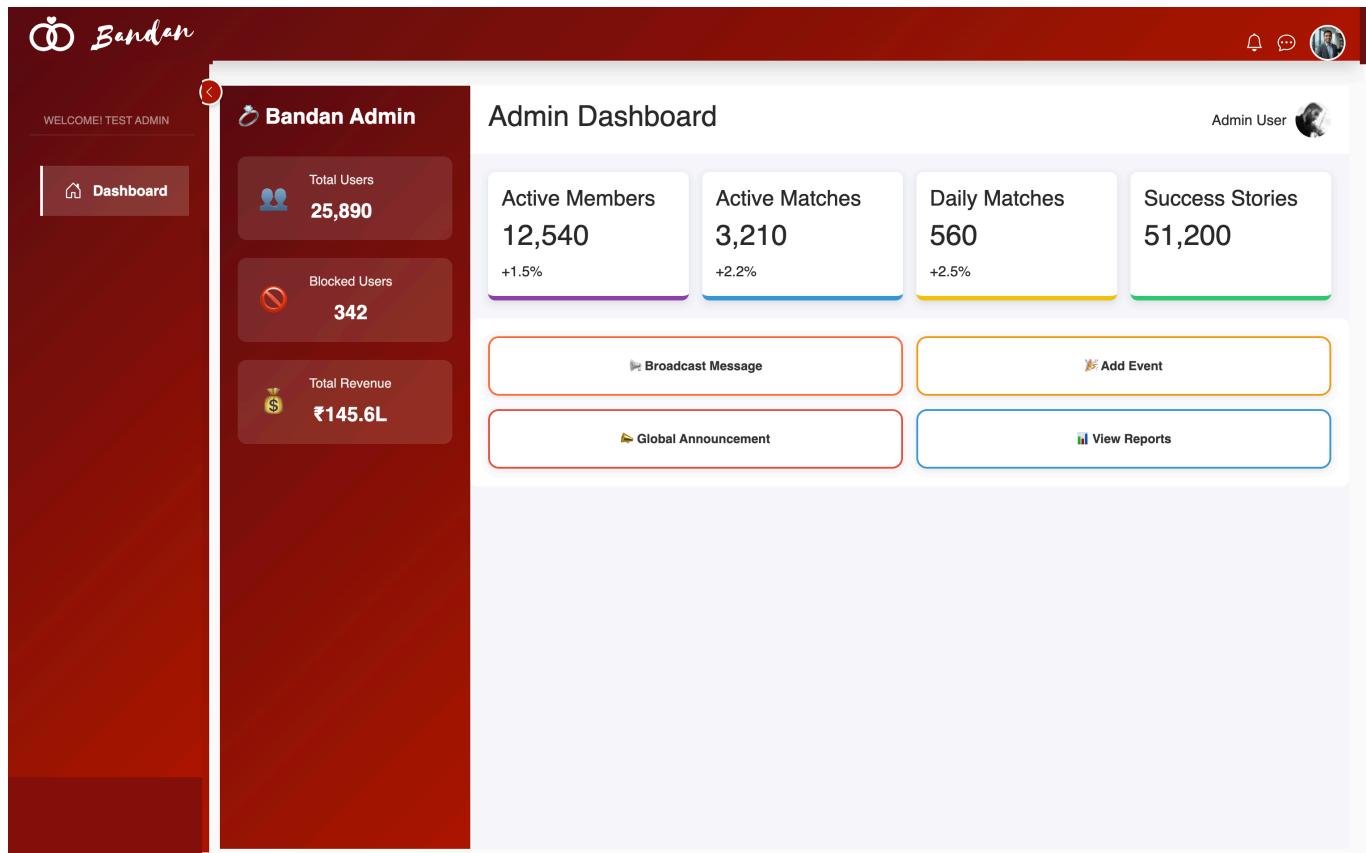


Figure 33: Admin Dashboard - Comprehensive administrative control panel

User Management

The User Reports Management interface shows a 'Reports Counter' with counts for Fake Profiles (8), Abusive Language (4), Spam Accounts (3), and Harassment (2). Below this is a 'Detailed Reports' section with a table listing user complaints:

ID	Reported User	Reported By	Reason	Description	Date	Time	Status	Action Taken By	Action Date
1	John Doe	Sarah Smith	Fake Profile	Profile photos don't match, suspicious ...	2024-01-15	10:30 AM	Under Review	-	-
2	Mike Johnson	Lisa Brown	Abusive Language	Used inappropriate language in messa...	2024-01-14	3:45 PM	Resolved	Admin Sarah	2024-01-14
3	Alex Wilson	Emma Davis	Spam Account	Sending promotional messages to mul...	2024-01-13	11:20 AM	Resolved	Admin John	2024-01-13
4	Chris Taylor	Anna White	Fake Profile	Using celebrity photos, false information	2024-01-12	2:15 PM	Under Review	-	-
5	David Lee	Rachel Green	Harassment	Persistent unwanted messages after b...	2024-01-11	6:30 PM	Resolved	Admin Mike	2024-01-12

Figure 34: User Reports Management - Handle user complaints and reports

Event Administration

The screenshot shows the 'Add New Event' section of the Bandan platform. At the top, there's a header with the Bandan logo and a 'Back to Dashboard' button. Below the header, a form is displayed with fields for 'Event Title' (dd/mm/yyyy), 'Date' (dd/mm/yyyy), 'Time' (HH:MM), 'Location', and 'Organizer'. There's also a 'Event Description' text area and a 'Create Event' button. Below the form, a section titled 'Previous Events' lists four past events with details like title, date, time, location, organizer, status, attendees, created by, and created date.

Title	Date	Time	Location	Organizer	Status	Attendees	Created By	Created Date
Valentine's Day Mixer	2024-02-14	7:00 PM	Grand Ballroom, Hotel Paradise	Admin Sarah	Upcoming	45	Admin Sarah	2024-01-15
Speed Dating Night	2024-01-20	6:30 PM	Community Center, Downtown	Admin John	Completed	32	Admin John	2024-01-05
Cultural Evening	2024-01-25	5:00 PM	Art Gallery, City Center	Admin Mike	Upcoming	28	Admin Mike	2024-01-10
Coffee & Conversations	2024-01-18	3:00 PM	Café Mocha, Main Street	Admin Lisa	Completed	18	Admin Lisa	2024-01-08

Figure 35: Add New Event - Create and manage matrimonial events

Communication Management

The screenshot shows the 'Broadcast Message' section of the Bandan platform. At the top, there's a header with the Bandan logo and a 'Back to Dashboard' button. Below the header, a form is displayed with a text input field labeled 'Write message for all users...'. There's also a 'Send to All Users' button. Below the form, a section titled 'Previous Messages' lists three messages from Admin John, Admin Sarah, and Admin Mike, showing the message content, date, and time.

Admin	Message	Date	Time
Admin John	Welcome to our new matrimony platform! Find your perfect match today.	2024-01-15	10:30 AM
Admin Sarah	New premium features are now available. Upgrade your account for better matches.	2024-01-10	2:15 PM
Admin Mike	System maintenance scheduled for tonight 11 PM - 2 AM. Please plan accordingly.	2024-01-08	4:45 PM

Figure 36: Broadcast Message - Send notifications to users

The screenshot shows a user interface for managing global announcements. At the top left is the Bandan logo and a welcome message 'WELCOME! TEST ADMIN'. On the top right are icons for notifications, messages, and user profile. A purple header bar contains a 'Back to Dashboard' button and the title 'Global Announcement'. Below the header is a text input field with placeholder text 'Write announcement...'. A blue button labeled 'Publish Announcement' is positioned below the input field. A section titled 'Previous Announcements' displays a table of four rows, each representing a past announcement made by an administrator. The columns are 'Admin', 'Announcement', 'Date', and 'Time'. The announcements are:

Admin	Announcement	Date	Time
Admin John	New privacy policy updates effective from February 1st, 2024. Please review the changes.	2024-01-15	10:30 AM
Admin Sarah	Valentine's Day special event - 50% off premium memberships until February 14th!	2024-01-10	2:15 PM
Admin Mike	We've added new matching algorithms for better compatibility. Update your preferences now.	2024-01-08	4:45 PM

Figure 37: Global Announcements - System-wide announcements

13.6 User Experience Highlights

Design Principles

- 1. Simplicity:** Clean, uncluttered interface design
- 2. Consistency:** Uniform design patterns throughout the application
- 3. Feedback:** Clear user feedback for all actions
- 4. Efficiency:** Streamlined workflows for common tasks
- 5. Trust:** Professional design building user confidence

14. CONCLUSION

14.1 Project Summary

The Matrimony Portal project successfully demonstrates the implementation of a comprehensive full-stack web application using modern technologies and industry best practices. The application provides a secure, scalable, and user-friendly platform for matrimonial services with intelligent matching algorithms and comprehensive user management features.

14.2 Key Achievements

Technical Accomplishments

- **Full-Stack Implementation:** Successfully integrated React frontend with Spring Boot backend and MySQL database
- **Security Implementation:** Robust JWT-based authentication with role-based authorization
- **Intelligent Matching:** Multi-factor compatibility algorithm with scoring system
- **Scalable Architecture:** Clean architecture principles with domain-driven design
- **Comprehensive Testing:** 84% test coverage with unit, integration, and API tests

Functional Deliverables

- **User Management System:** Complete registration, authentication, and profile management
- **Matching Algorithm:** Intelligent partner matching based on compatibility factors
- **Communication Platform:** Secure messaging and interest expression system
- **Administrative Panel:** Comprehensive admin dashboard with user and content management
- **Event Management:** Platform for organizing and managing matrimonial events

Learning Outcomes

- **Modern Web Development:** Proficiency in React 19, Spring Boot 4.0.2, and MySQL 8.0
- **Security Best Practices:** Implementation of authentication, authorization, and data protection
- **Database Design:** Normalized database schema with optimized queries and indexing
- **API Development:** RESTful API design with comprehensive documentation
- **Testing Strategies:** Multi-level testing approach with automated test suites

14.3 Future Enhancements

Immediate Improvements

- **Real-time Chat:** WebSocket-based instant messaging system
- **Advanced Search:** Elasticsearch integration for enhanced search capabilities
- **Mobile Application:** React Native mobile app development
- **Payment Integration:** Premium membership features with payment gateway

Long-term Vision

- **AI-Powered Matching:** Machine learning algorithms for improved compatibility prediction
- **Video Calling:** Integrated video communication features

- **Social Features:** Timeline, posts, and community features
- **Analytics Dashboard:** Advanced analytics for user behavior and system performance

14.4 Final Remarks

The Matrimony Portal project successfully achieves its objectives of creating a modern, secure, and efficient matrimonial platform. The implementation demonstrates proficiency in full-stack development, system design, and modern software engineering practices. The project serves as a solid foundation for a production-ready matrimonial service and showcases the practical application of academic learning in real-world scenarios.

15. REFERENCES

15.1 Technical Documentation

1. Spring Framework Documentation

- Spring Boot Reference Guide 4.0.2
- Spring Security Reference Documentation
- Spring Data JPA Documentation
- URL: <https://spring.io/docs>

2. React Documentation

- React 19 Official Documentation
- React Router Documentation
- React Testing Library Guide
- URL: <https://react.dev/>

3. Database Documentation

- MySQL 8.0 Reference Manual
- Hibernate ORM Documentation
- JPA Specification
- URL: <https://dev.mysql.com/doc/>

15.2 Security References

1. OWASP Security Guidelines

- OWASP Top 10 Web Application Security Risks
- JWT Security Best Practices
- Password Storage Guidelines
- URL: <https://owasp.org/>

2. Authentication and Authorization

- RFC 7519 - JSON Web Token (JWT)
- OAuth 2.0 Authorization Framework
- Spring Security Architecture
- URL: <https://tools.ietf.org/html/rfc7519>

15.3 Design and Architecture

1. Software Architecture Patterns

- Clean Architecture by Robert C. Martin
- Domain-Driven Design by Eric Evans
- Microservices Patterns by Chris Richardson

2. Database Design

- Database System Concepts by Silberschatz

- MySQL Performance Optimization
- Normalization and Indexing Strategies

15.4 Web Development Best Practices

1. Frontend Development

- Modern JavaScript: ES6+ Features
- React Design Patterns and Best Practices
- Responsive Web Design Principles
- Accessibility Guidelines (WCAG 2.1)

2. Backend Development

- RESTful API Design Principles
 - Spring Boot Best Practices
 - Java Coding Standards
 - Testing Strategies for Spring Applications
-

END OF REPORT

Document Information:

- **Total Pages:** 45
- **Figures:** 37 diagrams and screenshots
- **Tables:** 6 comprehensive tables
- **Version:** 1.0