DATING PLATFORM PROJECT DOCUMENTATION

1. Introduction

The Dating Platform is an interactive web application designed to facilitate meaningful connections between users based on shared interests, preferences, and location. The platform allows users to create profiles, browse matches, and engage in conversations. It leverages core web technologies such as HTML, CSS, MySQL, Java Servlets, JSP, Maven, and JavaScript for validation.

Key Features:

● User-friendly profile creation and management.

● Matchmaking based on user preferences and algorithms.

● User authentication and session management.

● Modular and scalable design using Maven.

---

2. System Requirements

Hardware Requirements:

● Processor: Minimum dual-core processor.

● RAM: 4GB or higher.

● Storage: At least 500MB free space.

Software Requirements:

● Operating System: Windows, macOS, or Linux.

● JDK: Version 8 or higher.

● Apache Tomcat: Version 9.0 or higher.

● Browser: Latest versions of Chrome, Firefox, or Edge.

---

3. Technology Stack

Frontend:

● HTML and JSP: HTML provides the structure for web pages, while JSP generates dynamic content on the server-side before rendering in the browser.

● CSS: Enables responsive and visually appealing layouts.

● JavaScript: Facilitates dynamic client-side interactions and form validations.

Backend:

● Java Servlets: Manage core business logic and handle HTTP requests and responses.

● DAO Pattern: Manages database operations to ensure a clean separation between application logic and database interaction.

Database:

● MySQL: Secure and scalable relational database for storing user profiles and matches.

Build Tool:

● Maven: Manages dependencies, automates builds, and maintains project structure.

Testing:

● JUnit: Ensures reliable server-side logic through unit and integration testing.

---

4. Installation Guide

Step 1: Clone the Repository

git clone https://github.com/Gaurav-K-Github/Dating-Platform.git

cd Dating-Platform

Step 2: Configure MySQL Database

1. Create a new database named dating_platform.
2. Run the SQL script in the db directory to set up tables.

Step 3: Configure Environment Variables

Update the db.properties file with your MySQL credentials:

db.url=jdbc:mysql://localhost:3306/dating_platform

db.username=<admin>

db.password=<admin@123>

Step 4: Build and Deploy the Application

Use Maven to build the project:

mvn clean install

Deploy the generated WAR file to Apache Tomcat:

1. Copy the WAR file from the target directory to the Tomcat webapps folder.

2. Start Tomcat and access the application at http://localhost:8080/Dating-Platform.

---

5. Project Architecture

High-Level Architecture:

● Frontend: HTML, JSP, CSS for UI.

● Backend: Java Servlets handling core logic and business processes.

● Database: MySQL for data persistence.

Components:

1. User Interface: HTML, JSP, CSS, and JavaScript.
2. Servlets: Handle user requests and manage sessions.
3. DAO: Manages database interactions.
4. Database: Stores user profiles and matches.

---

6. Frontend Design

Overview:

The frontend offers a seamless user experience with interactive pages for profile creation, browsing matches, and messaging.

Key Components:

● Profile Page: Allows users to create and manage profiles.

● Match Page: Displays potential matches.

---

7. Backend Implementation

Servlets:

● UserController: Manages user registrations and logins.

● MatchController: Handles matchmaking logic.

Business Logic:

● Java Servlets process HTTP requests and execute application logic.

● DAO classes interact with MySQL to fetch and update data.

---

8. Database Design

Data Models:

1. User:

○ id: Integer (Primary Key)

○ username: String

○ email: String

○ password: String (hashed)

2. Match:

○ match_id: Integer (Primary Key)

○ user1_id: Foreign key referencing User table

○ user2_id: Foreign key referencing User table

---

9. Core Functionalities

Matching Algorithm:

● Match users based on shared interests, age, and location.

Profile Management:

● Users can update personal information, photos, and preferences.

---

10. User Authentication

Features:

● Secure registration and login.

● Password hashing and session management.

---

11. Error Handling

Common Issues:

● Invalid login credentials.

● Database connection issues.

Solutions:

● Display user-friendly error messages.

● Log errors server-side for debugging.

---

12. Scalability

Strategies:

● Optimize SQL queries for faster performance.

● Use caching and database indexing.

---

13. Testing

Types of Testing:

1. Unit Testing: Test Servlets and DAO classes.
2. Integration Testing: Ensure proper interaction between components.
3. UI Testing: Validate frontend functionality.

---

14. Deployment

Hosting:

● Deploy on Apache Tomcat.

Deployment Process:

1. Build using Maven.
2. Deploy the WAR file to Tomcat webapps.

---

15. Future Enhancements

Planned Features:

● Profile verification.

● Video call functionality.

16. Team Members:

- GAURAV KUMAR PANDEY (Team Lead)
- AMIT TIWARI