

# **MATRIMONIAL WEBSITE**

# **Index**

1. Title of the Project

2. Introduction

3. Objective

4. Project Category

5. Analysis

- Modules and Description
- Database Design
- ER Diagram
- Data Flow Diagram (DFD)

6. Complete Structure

- Process Logical Diagram

7. Platform Used

- Hardware Requirement
- Software Requirement

8. Future Scope

9. Bibliography

## **1. Introduction**

Marriage matchmaking has evolved significantly from traditional offline practices to modern digital platforms. With increasing internet adoption, online matrimonial portals have become one of the most effective ways to connect individuals with compatible partners.

This project aims to develop a full-stack matrimonial web application using the MERN stack, which provides scalability, security, and efficient data handling.

Users can register, log in, create profiles, search matches, and express interest securely through a user-friendly interface.

## **2. Objective**

The key objectives of this project are:

- To design and develop a dynamic and interactive matrimonial website.
- To implement secure user authentication using JWT.
- To allow users to create, view, and edit their personal profiles.
- To provide smart search and filtering of potential matches.
- To support interest-sending functionality for matchmaking.
- To utilize MERN technologies for fast, scalable, and modern web development.

### **3. Project Category**

This project falls under the following categories:

#### **1. Web-Based Application**

- A browser-accessible online matrimonial platform developed using modern web technologies.

#### **2. Full-Stack Development Project**

- Involves client-side development with React.js and server-side development using Node.js and Express.js, along with database integration.

#### **3. Database Management System Project (DBMS)**

- Uses MongoDB to store user profiles, matchmaking data, and interest interactions.

#### **4. Online Matrimonial / Matchmaking System**

- Focuses on connecting users with suitable partners through smart filtering and interest-based interactions.

#### **5. Security-Enabled Application**

- Implements JWT authentication and encrypted password management to ensure data privacy and safety.

#### **6. Cloud-Deployable Application**

- The structure is scalable and deployable on online cloud platforms like MongoDB Atlas, Render, Vercel, or Netlify.

## **4. Analysis**

### **A. Modules and Description**

#### **1. User Registration & Login**

- Users create an account with basic details.
- Secure authentication using JWT.
- Prevents unauthorized access.

#### **2. Profile Management**

- Users update personal details like age, location, religion, caste, occupation, bio, partner preferences.
- Data stored securely in MongoDB.

#### **3. Match Browsing Module**

Users browse other profiles.

Filters available:

Gender

Age range

Location

Religion / caste

#### 4. Interest Management

Users can “Send Interest” to other profiles.

Stored as likedProfiles and receivedInterests.

#### 5. Search & Filter Module

Smart searching based on user-entered filters.

Delivers accurate and relevant matchmaking results.

### B. Database Design (Overview)

Main collection:

Users

User Fields

fullName

email

password

gender

age

religion, caste, location

bio, occupation, incomeRange

partnerPreferences (age, location, religion)

likedProfiles (list of Object IDs)

receivedInterests (list of Object IDs)

MongoDB ensures fast queries and flexible schema management.

### C. ER Diagram (Textual Description)

Entities:

1. User

user\_id (Primary Key)

name, age, gender, email

profile details

## 2. Interest

interest\_id

sender\_user\_id (FK)

receiver\_user\_id (FK)

Relationships:

One User can send many interests → (1:M)

One User can receive many interests → (1:M)

## D. Data Flow Diagram (DFD)

DFD Level 0 (Context Level)

User → Matrimony System → Database

DFD Level 1

1. User Registration Process

2. Login Process

3. Profile Management

4. Search Match Process

5. Send Interest Process

## **6. Complete Structure**

Process Logical Diagram (Flow Explanation)

1. User lands on the home page

2. Registration → Data saved in database

3. Login → Token generated

4. User Dashboard opens

5. User updates profile

6. User searches/browses matches

7. User sends interest to another profile

8. System notifies receiver

9. Both parties can continue interaction

## **7. Platform Used**

### **A. Hardware Requirements**

- a) Processor: Intel i3 or above
- b) RAM: Minimum 4 GB
- c) Storage: 5 GB required for project setup
- d) Device: Laptop/Desktop

### **B. Software Requirements**

- a) OS: Windows / Linux / macOS
- b) Node.js
- c) MongoDB
- d) Visual Studio Code
- e) Postman (for API testing)
- f) Browser (Chrome/Firefox)

g) React.js library

h) Express.js framework

## **8. Future Scope**

a) Profile picture upload

b) Real-time chat system

c) AI-based match recommendations

d) Mobile app version

e) Admin dashboard for user verification

f) Paid membership and premium filters

g) OTP/Email verification for security

## **9. Bibliography**

- a) MERN Stack Documentation
- b) MongoDB Documentation
- c) React Official Docs
- d) Node.js & Express.js Guide
- e) W3Schools / GeeksforGeeks (Web References)
- f) Online Matrimonial Website Research Papers