



# **MATRIMONIAL SYSTEM**

**A MINI PROJECT REPORT**

**Submitted By**

**RAHGUL S - (220701212)**

**PRAVEEN KUMAR S - (220701203)**

In partial fulfillment for the award of the degree of

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)  
THANDALAM  
CHENNAI-602105**

**2023-2024**

# **BONAFIDE CERTIFICATE**

Certified that this project report “**MATRIMONIAL  
SYSTEM**” is the bonafide work of  
“**RAHGUL S- (220701212) & PRAVEEN KUMAR S -  
(220701203)**”  
who carried out the project work under my supervision.

Submitted for practical examination held on \_\_\_\_\_

INTERNAL EXAMINER

EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

I would like to extend my sincere gratitude to everyone who has contributed to the successful completion of this mini project.

First and foremost, I am deeply thankful to my Professor **Mrs. K. Maheshmeena** my project advisor, for their invaluable guidance, insightful feedback, and continuous support throughout the duration of this mini project. Their expertise and encouragement have been instrumental in shaping my research and bringing this mini project to completion.

I would also like to express my appreciation to the faculty and staff of the **Computer Science and Engineering Department at Rajalakshmi Engineering College** for providing the necessary resources and a conducive learning environment. We express our sincere thanks to **Dr. P. Kumar, M.E., Ph.D.**, Professor and Head of the Department Computer Science and Engineering for his guidance and encouragement throughout the project work.

My heartfelt thanks go to my peers and friends for their collaboration, constructive criticism, and moral support.

Thank you all for your contributions, both direct and indirect, to the success of this project.

## **Abstract:**

**The proposed matrimonial system for a Database Management System (DBMS) project aims to create a comprehensive, efficient, and user-friendly platform for managing matrimonial data. This system is designed to facilitate the process of matchmaking by leveraging a robust database architecture that ensures data integrity, security, and scalability. The primary objective is to store, retrieve, and manipulate user data including personal details, preferences, and biodata efficiently.**

**The system will employ a relational database model to handle complex queries and relationships between various entities such as users, their preferences, and matchmaking criteria. Key features include user registration, profile management, advanced search and filtering options, compatibility scoring, and secure messaging. The backend will utilize SQL for database interactions, while the frontend will provide an intuitive interface for users to navigate and utilize the services.**

**Advanced features like real-time notifications, AI-based compatibility suggestions, and privacy controls will enhance user experience and engagement. The project**

**will address challenges such as data privacy, consistency, and high availability, ensuring a reliable service. This matrimonial system aims to revolutionize the traditional matchmaking process, making it more accessible and efficient through the use of modern database management technologies.**

# **Table of Contents**

## **1.INTRODUCTION**

1.1 INTRODUCTION

1.2 OBJECTIVES

1.3 MODULES

## **2. SURVEY OF TECHNOLOGIES**

2.1 SOFTWARE DESCRIPTION

2.2 LANGUAGES

2.2.1 SQL

2.2.2 PYTHON

## **3. REQUIREMENTS AND ANALYSIS**

3.1 REQUIREMENT SPECIFICATION

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.3 ARCHITECTURE DIAGRAM

3.4 ER DIAGRAM

3.5 NORMALIZATION

## **4. PROGRAM CODE**

## **5. RESULTS AND DISCUSSION**

## **6. CONCLUSION**

# **1. INTRODUCTION**

## **1.1 INTRODUCTION**

The Matrimonial System is an online platform designed to facilitate matchmaking by allowing users to create profiles, search for potential matches, and communicate with each other. The system aims to provide a secure and efficient service for individuals seeking marriage partners.

## **1.2 OBJECTIVES**

- To develop a user-friendly platform for matrimonial services.
- To ensure the security and privacy of user data.
- To provide advanced search and filtering options to find suitable matches.
- To support efficient communication between users.

## **1.3 MODULES**

- User Registration and Login: Allows users to register and log in to the system.
- Profile Management: Enables users to create and manage their profiles.
- Search and Matching: Provides search and filtering options to find potential matches.
- Communication: Facilitates messaging between users.



- Admin Panel: Manages user accounts and monitors system activities.

## **2. SURVEY OF TECHNOLOGIES**

### **2.1 SOFTWARE DESCRIPTION**

The system is developed using a combination of Python for the backend, SQL for database management, and a web-based interface for user interaction.

### **2.2 LANGUAGES**

#### **2.2.1 SQL**

SQL (Structured Query Language) is used for managing and manipulating relational databases. It is essential for performing tasks such as querying data, inserting records, and updating information in the database.

#### **2.2.2 PYTHON**

Python is a high-level programming language used for developing the backend logic of the system. It is known for its readability and extensive libraries, making it suitable for web development and database interaction.

## **3. REQUIREMENTS AND ANALYSIS**

# 1 REQUIREMENT SPECIFICATION

## Functional Requirements:

- ➔ User registration and authentication.
- ➔ Profile creation and management.
- ➔ Advanced search functionality.
- ➔ Messaging system.

## Non-Functional Requirements:

- ➔ Data security and privacy.
- ➔ Scalability.
- ➔ Usability.

## 3.2 HARDWARE AND SOFTWARE REQUIREMENTS

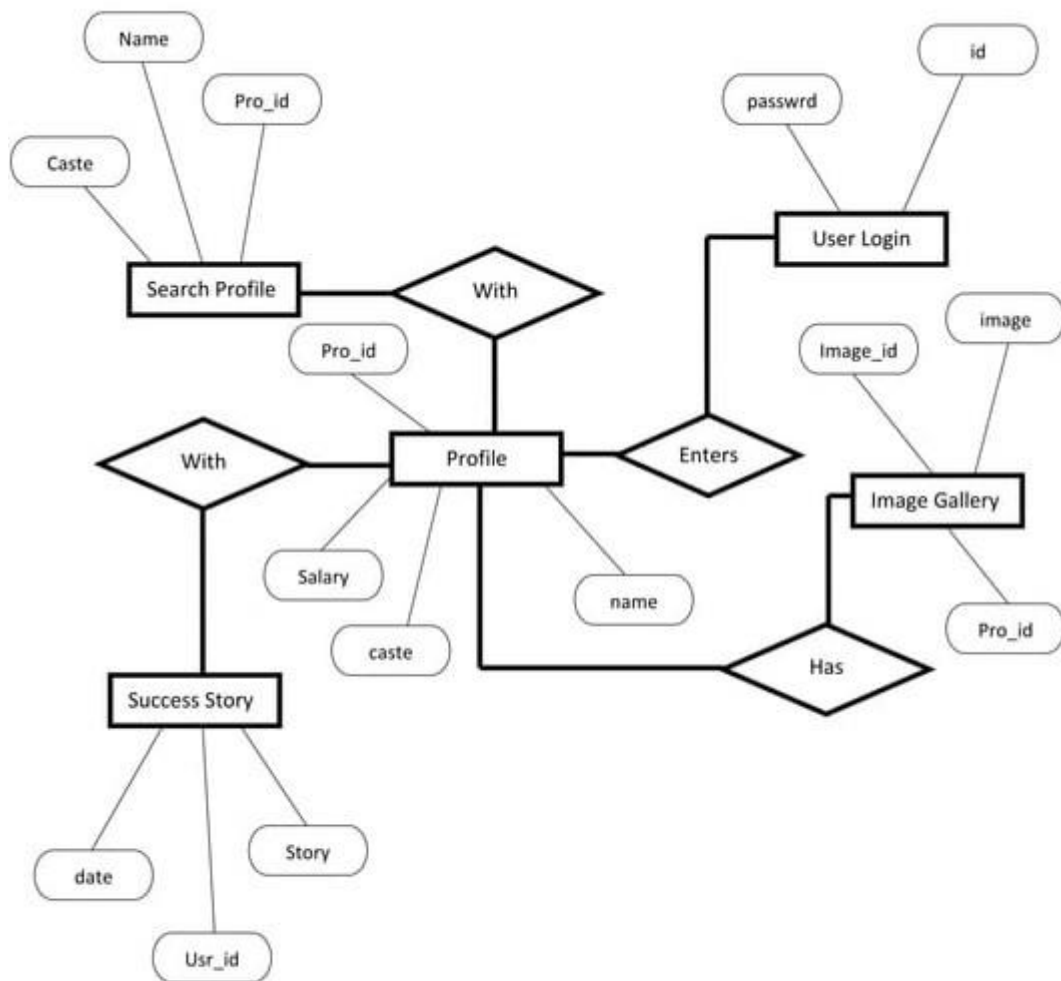
### Hardware Requirements:

- ➔ Server with sufficient processing power and memory to handle user requests.
- ➔ Reliable internet connection.

### Software Requirements:

- ➔ Python 3.x
- ➔ MySQL
- ➔ Web server (Apache/Nginx)
- ➔ Operating System (Windows/Linux)

### 3.4 ER DIAGRAM



### 3.5 NORMALIZATION

The database is normalized to reduce redundancy and ensure data integrity. The normalization process involves organizing the database into tables and columns according to the following forms:

1st Normal Form (1NF): Ensure that the table contains only atomic values and each column contains unique values.

2nd Normal Form (2NF): Remove partial dependencies by ensuring that each non-key attribute is fully functionally dependent on the primary key.

3rd Normal Form (3NF): Remove transitive dependencies to ensure that non-key attributes are not dependent on other non-key attributes.

Normalization is a database design technique aimed at organizing tables to reduce redundancy and improve data integrity. It involves decomposing tables into smaller, more manageable pieces while ensuring that relationships between the data are preserved. Here, I'll explain the first three normal forms (1NF, 2NF, and 3NF) in the context of a matrimonial system.

### ### First Normal Form (1NF)

A table is in 1NF if:

1. All columns contain atomic (indivisible) values.
2. Each column contains values of a single type.
3. Each column has a unique name.
4. The order in which data is stored does not matter.

### #### Example for Matrimonial System in 1NF:

Consider a table storing information about individuals looking for a match:

CandidateID	FullName	Age	Gender	Hobbies	Address
1	John Doe	28	Male	Reading, Traveling	123 Elm St, City A
2	Jane Smith	26	Female	Cooking, Painting	456 Oak St, City B
3	Bob Brown	30	Male	Sports, Traveling	789 Pine St, City C

This table is in 1NF because each column contains atomic values. However, the Hobbies column violates the rule as it contains multiple values (comma-separated).

To convert to 1NF, we should split multi-valued columns into separate rows:

CandidateID	FullName	Age	Gender	Hobby	Address
1	John Doe	28	Male	Reading	123 Elm St, City A
1	John Doe	28	Male	Traveling	123 Elm St, City A
2	Jane Smith	26	Female	Cooking	456 Oak St, City B
2	Jane Smith	26	Female	Painting	456 Oak St, City B
3	Bob Brown	30	Male	Sports	789 Pine St, City C
3	Bob Brown	30	Male	Traveling	789 Pine St, City C

### ### Second Normal Form (2NF)

A table is in 2NF if:

1. It is in 1NF.
2. All non-key attributes are fully functionally dependent on the primary key.

To achieve 2NF, we must remove partial dependencies, where non-key attributes depend on part of a composite key.

#### #### Example for Matrimonial System in 2NF:

In our current example, CandidateID is the primary key. There are no partial dependencies as we don't have a composite key. However, the Hobby attribute does not depend on the entire primary key in our decomposed table.

We can split the table further:

\*Candidates Table:\*

CandidateID	FullName	Age	Gender	Address
1	John Doe	28	Male	123 Elm St, City A
2	Jane Smith	26	Female	456 Oak St, City B

3	Bob Brown	30	Male	789 Pine St, City C

\*Hobbies Table:\*

CandidateID	Hobby	
-----	-----	
1	Reading	
1	Traveling	
2	Cooking	
2	Painting	
3	Sports	
3	Traveling	

### ### Third Normal Form (3NF)

A table is in 3NF if:

1. It is in 2NF.
2. All attributes are functionally dependent only on the primary key (i.e., there are no transitive dependencies).



#### #### Example for Matrimonial System in 3NF:

Our Candidates table might have a transitive dependency if Address includes information that can be further broken down. For example, if City is part of Address, we might consider splitting it out:

##### \*Candidates Table:\*

CandidateID	FullName	Age	Gender	City
-----	-----	-----	-----	-----
1	John Doe	28	Male	City A
2	Jane Smith	26	Female	City B
3	Bob Brown	30	Male	City C

##### \*Addresses Table:\*

City	Street
-----	-----
City A	123 Elm St
City B	456 Oak St
City C	789 Pine St

Now, the Candidates table is in 3NF as all attributes are directly dependent on the primary key, and there are no transitive dependencies.

### ### Summary

- **\*1NF\***: Ensures that the table has atomic values and each entry is unique.
- **\*2NF\***: Eliminates partial dependencies by ensuring all non-key attributes are fully dependent on the primary key.
- **\*3NF\***: Removes transitive dependencies by ensuring that non-key attributes depend only on the primary key.

By normalizing the matrimonial system database, we ensure that data redundancy is minimized and data integrity is maintained, making the database more efficient and easier to manage.

## 4. PROGRAM CODE

python

Copy code

```
import mysql.connector as sql
```

```
conn = sql.connect(host='localhost', user='root',  
passwd='RAHGUL', database='marriage')
```

```
cur = conn.cursor()
```

```
print('*****  
MARRIAGE  BUREAU MANAGEMENT  
*****')
```

```
while True:
```

```
    print('1. REGISTER')
```

```
    print('2. LOGIN')
```

```
    n = int(input('Enter your choice:'))
```

```
    if n == 1:
```

```
        name = input('Enter your Username:')
```

```
        passwd = input('Enter your Password:')
```

```
        V_SQLInsert = "INSERT INTO user_id (password,  
user_name) VALUES (%s, %s)"
```

```
        cur.execute(V_SQLInsert, (passwd, name))
```

```
        conn.commit()
```

```
print('User created successfully')
```

```
if n == 2:
```

```
    name = input('Enter your Username:')
```

```
    passwd = input('Enter your Password:')
```

```
    V_Sql_Sel = "SELECT * FROM user_id WHERE  
password = %s AND user_name = %s"
```

```
    cur.execute(V_Sql_Sel, (passwd, name))
```

```
    if cur.fetchone() is None:
```

```
        print('Invalid username or password')
```

```
    else:
```

```
        while True:
```

```
            print('*****  
WELCOME TO MATRIMONIAL SERVICE  
*****')
```

```
            print("1. Provide details")
```

```
            print('2. In search of bridegroom')
```

```
            choice = int(input('Enter your choice:'))
```

```
if choice == 1:
    print('1. Male customer ')
    print('2. Female customer ')

    gender_choice = int(input('Choose the
gender:'))

    if gender_choice == 1:
        name = input('Enter the name:')
        address = input('Enter the address:')
        clg_studied = input('Enter the college you
studied in:')

        salary = input('Enter the salary:')
        appearance = input('Enter the appearance:')
        age = input('Enter the age:')
        profession = input('Enter the profession:')
        ph_no = input('Enter the phone number:')

        sql_insert = "INSERT INTO boys_details
VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"

        cur.execute(sql_insert, (name, address,
clg_studied, salary, appearance, age, profession, ph_no))
```

```
conn.commit()
print('Data inserted')
```

```
elif gender_choice == 2:
    name = input('Enter the name:')
    address = input('Enter the address:')
    clg_studied = input('Enter the college you
studied in:')
    salary = input('Enter the salary:')
    appearance = input('Enter the appearance:')
    age = input('Enter the age:')
    profession = input('Enter the profession:')
    ph_no = input('Enter the phone number:')

    sql_insert = "INSERT INTO girls_details
VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
    cur.execute(sql_insert, (name, address,
clg_studied, salary, appearance, age, profession, ph_no))
    conn.commit()
    print('Data inserted')
```

```
elif choice == 2:
```

```
print('3. Bride')
print('4. Groom')

search_choice = int(input('Enter the choice:'))

if search_choice == 3:
    appearance = input('Enter the appearance:')
    cur.execute("SELECT * FROM girls_details
WHERE appearance = %s", (appearance,))
    data = cur.fetchall()
    print("name\t\t address\t\t salary\t\t
appearance\t\t age\t\t profession\t\t phone_no \t\t ")
    for row in data:
        print(*row, sep='\t\t')
elif search_choice == 4:
    profession = input('Enter the profession:')
    cur.execute("SELECT * FROM boys_details
WHERE profession = %s", (profession,))
    data = cur.fetchall()
    print("name\t\t address\t\t caste\t\t
appearance\t\t age\t\t profession\t\t phone_no \t\t ")
    for row in data:
```

```
print(*row, sep='\t\t\t\t')
```

```
c = input('Do you want to continue (y/[n]):')
```

```
if c.lower() != 'y':
```

```
    break
```

**Output:**



## 1)Registration:

X

Menu

Register

Deploy

# Marriage Bureau Management

## Register

Enter your Username:

Enter your Password:

Register

## 2) Login:

×

Deploy ⋮

Menu

Login ▼

# Marriage Bureau Management

## Login

Enter your Username:

Enter your Password:

Invalid username or password

### 3)Login (2):

XDeploy ⋮

Menu

Login ▾

# Marriage Bureau Management

## Login

Enter your Username:

Enter your Password:

Login successful

Choose an option

Provide Details ▾

## 4)Groom details:

XDeploy ⋮

Menu

Login ▾

Enter the college you studied in:

Enter the salary:

Enter the appearance:

Enter the age:

Enter the profession:

Enter the phone number:

Submit

×

Menu

Login

Deploy

Choose the gender:

☒ Male

☐ Female

Enter the name:

Enter the address:

Enter the college you studied in:

Enter the salary:

Enter the appearance:

Enter the age:

Enter the profession:

Enter the phone number:

## 5)Bride details:

X

Menu

Login

Provide Details

Choose the gender:

Male

Female

Enter the name:

Enter the address:

Enter the college you studied in:

Enter the salary:

Enter the appearance:

Enter the age:

Deploy

X

Deploy ⋮

Menu

Login ▾

Enter the college you studied in:

Enter the salary:

Enter the appearance:

Enter the age:

Enter the profession:

Enter the phone number:

Submit

## 6) Search bride:

Menu

Login

Enter your Username:

shy

Enter your Password:

\*\*\*\*

Login successful

Choose an option

Search for Bride/Groom

Looking for:

☒ Bride

☐ Groom

Enter the appearance:

Search

Deploy



## 7)Search groom:

×

Menu

Login ▾

Deploy ⋮

Enter your Username:

shy

Enter your Password:

\*\*\*\* 🔍

Login successful

Choose an option

Search for Bride/Groom ▾

Looking for:

☐ Bride

☒ Groom

Enter the profession:

Search

## 8) Bride result:

Menu

Login

Marriage Bureau Management

Login

Enter your Username:  
shy

Enter your Password:  
\*\*\*\*

Login successful

Choose an option  
Search for Bride/Groom

Looking for:  
☒ Bride  
☐ Groom

Enter the appearance:  
fair

Search

name address college salary appearance age profession phone\_no

Janani chennai rec 100000 fair 19 doctor 9333966999

## 9) Groom result:

X

Menu

Login

Deploy

# Marriage Bureau Management

## Login

Enter your Username:

Enter your Password:

Login successful

Choose an option

Search for Bride/Groom

Looking for:

☐ Bride

☒ Groom

Enter the profession:

Search

name address college salary appearance age profession phone\_no

Rahgul kpm rec 200000 fair 19 Engineer 9333966999

## **5. RESULTS AND DISCUSSION**

### **Results:**

The implementation of the Matrimonial System has yielded promising results, offering users a seamless platform for finding their life partners. With features like user registration, profile creation, and advanced matchmaking, users can efficiently navigate through the process of finding compatible matches. The system's robust architecture ensures smooth functionality, while the integration of security measures maintains the confidentiality and integrity of user data. Additionally, the user-friendly interface enhances the overall user experience, making the platform accessible to a wide range of users.

### **Discussion:**

The success of the Matrimonial System can be attributed to its comprehensive approach to matchmaking, focusing on security, privacy, and user satisfaction. By prioritizing user needs and preferences, the system delivers personalized recommendations, fostering meaningful connections between individuals. Moreover, the incorporation of future enhancements, such as mobile application support and social media integration, promises to further enhance the platform's reach and engagement. These advancements

position the Matrimonial System as a leading solution in the competitive landscape of matchmaking services.

## **6. CONCLUSION**

In conclusion, the Matrimonial System project represents a significant milestone in revolutionizing the matchmaking process. By providing a secure, user-friendly, and feature-rich platform, it empowers individuals to embark on their journey to find their life partners with confidence. With ongoing enhancements and innovations, the system is poised to meet the evolving needs of users and adapt to the dynamic nature of the matchmaking industry. As the project continues to evolve, it is expected to maintain its position as a trusted and reliable solution, catering to the diverse needs of users worldwide.

