|  |  |  |
| --- | --- | --- |
|  | **SRI KRISHNA INSTITUTIONS**  (An Autonomous Institution)  (Approved by AICTE and Affiliated to Anna University, Chennai)  ACCREDITED BY NAAC WITH “A” GRADE |  |
|  |  |  |

**Matrimony management system**

**TEAM NEME: STILL LOADING**

***Submitted by***

**D.GOBINATH (16EUEE037)**

**D.M.SARAVANA RAJ(17EUMT518)**

**H.ABDUL AZEEZ(16EUEE003)**

|  |  |
| --- | --- |
| **Date** | **Reviewed by** |
| 16-Oct-2018 | (Dr.senthil) |

|  |  |  |
| --- | --- | --- |
| S.No | Table of Contents | Page No. |
| 1. | About us & Contact us | 3 |
| 2. | Abstract | 4 |
| 3. | Objective | 4 |
| 4. | Existing Systems | 5 |
| 5. | Proposed Systems | 5 |
| 6. | Specifications | 6 |
| 7. | Budget | 6 |
| 8. | High level data documentation | 7 |
| 9. | Sample screen shots | 10 |
| 10. | Low level data documentation | 12 |
| 11. | Control flow diagram | 13 |
| 12. | Pseudocode | 13 |
| 13. | Algorithm | 16 |
| 14. | Flowchart | 19 |
| 15. | Code | 20 |
| 16. | Quality assurance-Test cases | 46 |
| 17. | Output screens | 49 |
| 18. | Complexity | 54 |
| 19. | Conclusion | 54 |
| 20. | Future enhancements | 54 |
| 21. | References | 55 |

**TABLE OF CONTENTSABOUT US:**

We are team of 3 members interested in application development of Matrimonial management system.

Till we have done two projects;

* Vehicle tracking using driver’s mobile GPS.
* Fingerprint based ATM system.

**CANTACT US:**

Gobinath.D (Developer)

Mail-id: [16euee037@skcet.ac.in](mailto:16euee037@skcet.ac.in)

9159494642

Saravana Raj.D.M (Tester)

Mail-id: [17eumt518@skcet.ac.in](mailto:17eumt518@skcet.ac.in)

9788678380

Abdul Azeez.H (Business analyst)

Mail-id: [16euee003@skcet.ac.in](mailto:16euee003@skcet.ac.in)

9629442984

**ABSTRACT:**

This project is about Matrimonial Management System. The system is about to conclude the matchmaking of the Grooms/Brides. From this system he/she can able to find perfect life parter for their life. The System consists of several modules to undergo, which covers the informations/details of all logged-in persons.This system allows the individuals to get his/her informations such as Name, Gender, Religion, Caste, Marital status, Salary, Occupation etc., after Registration has made.The person looking for marriage can register and search for a profile that matches their requirements.This system can also allows individuals to search their requirements by Gender, Age, Religion, Caste etc.,

**OBJECTIVE:**

Matrimonial website which will provide platform to a lot of Bride/Groom for finding perfect match. There are different sectors like Registration, Partner, Search, etc. So the Bride/Groom can get their interest for find their partner. Bride/Groom can directly search Partner according to their required criteria.The drawback of existing system is that searching a good life partner in this world is the times Consuming and cost effective. Online matrimonial system is used to overcome these drawbacks. This Secured Online Matrimonial System is a useful for a person of any religion who wishes to find a suitable life partner for himself or her.

**Existing system:**

The drawback of existing system is that searching a good life partner in this world is the times Consuming and cost effective. Online matrimonial system is used to overcome these drawbacks.Already existing system contains the details of Grooms/Brides as updated. eg., Name, religion, Salary etc.,In Existing system, there is possibilities of creating fake accounts.The fake account creation has been done mainly by skipping out the screen without entering the OTP which has been received by the given number.

**PROPOSED SYSTEM:**

The new thing we have proposed in the system was, If the candidate didn’t enter the correct OTP they received , they cannot able to create an account in the matrimony. In this manner,account creation by fake candidates has been reduces.Credit card details are also be expected from prime candidates for payment usage (But it is already existing) .

**SPECIFICATIONS:**

**Software Requirements:**

* + Pc with Windows Xp, Windows 7 with operating system
  + Processor: Intel Pentium or above
  + Hard disk: 512 GB or above
  + RAM: 1 GB or above DDR4 RAM
  + Bandwidth speed: 1024 mbps

**Client Specification:**

* Operating system independent/mobile
* Browser (Chrome,Firefox)
* Network speed 100kbps

|  |  |  |
| --- | --- | --- |
| S.No | Items | Cost per year |
| 1. | Domain cost | Rs.4,000 |
| 2. | Server Installation | Rs.35,000 |
| 3. | Server maintenance | Rs.70,000 |
| 4. | System Administrator | Rs.90,000 |
| 5. | Developing cost | Rs.50,000 |
|  | Total | Rs.2,49,000 |

**BUDGET:**

**HIGH LEVEL DESIGN DOCUMENTATION**

|  |
| --- |
| **BACKGROUND:** |

This document contains high level design of the project.

The project is matrimony management system.

**REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| SI | REQUIREMENTS | STATUS |
| 1 | Admin Module | Implemented |
| 2 | Candidate Module | Implemented |
| 3 | Service Module | Implemented |
| 4 | Search Module | Implemented |
| 5 | Expectation Module | Implemented |

The following are the requirements for this project.

**ADMIN MODULE:**

Main functions of Admin:

* Login
* Can able to view free members and paid members details
* Logout

**CANDIDATE MODULE:**

**For unregistered person, registration process:**

**Input:** Click on link for new registration.

**Output:** Registration is done for new user.

**Flow:** Control is redirected to registration form and after fill up the form and after click on the register button data of new user is stored into database and an email is sent to the user.

**For registered person, login process:**

**Input:** Member Id and password.

**Output:** Get the screen of verify Member Id and Password.

**Flow:** If Member Id and Password correct then user will get

his/her profile. user can do update profile, create album, change

photo, hide profile toure, apply for loan after correct login.

**If Password is forgotten then re-direct to retrieve page:**

**Input:** Click on link for forgotten password

**Output:** Get the screen for retrieving password and enter member id and email id on that page. An email will be sent to user with member id and password.

**SERVICE MODULE:**

**PAID CANDIDATE:**

* This module will contain service only for paid candidate.
* The services are as follows,
* Visibility of matching groom/bride contacts like, mobile number.

**EXPECTATION MODULE:**

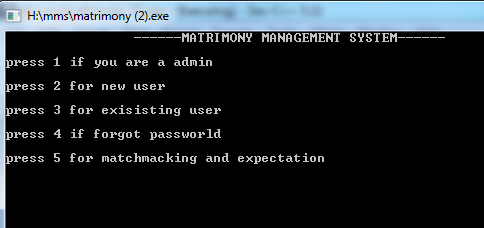
* It contains what are all the expectations that the Groom/Bride needs.
* Expectation module also acts as matching module and grooms/bride identify matching partner details.

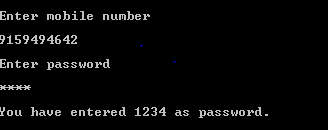
Eg. salary, qualification, etc..

**SEARCH MODULE:**

* The person looking for Marriage can register and search for a profile that matches their requirements by using Search module
* If the Groom/Bride know the partners name or id can able to view the profile by the help of search module.

**SAMPLE SCREEN SHORTS**

* admin module(1)
* candidate module and service module(2,3,4)
* search and expectation module(5)



**LOWER LEVEL DESIGN DOCUMENTATION**

**Modules:**

* Admin Module
* Candidate Module
* Service Module
* Expectation Module
* Search module

**Descriptions:**

**Admin Module:**

* The user need to Sign-in/login to create an user id in the matrimonial system.
* Admin module’s work is to publish the user id and to create a database for both the paid members and free members using User management.
* Paid members can able to contact with their required profile whereas free members cannot be able to contact with the required profiles but only they can able to view their profile and hence this process can be done by using the Admin module.

- By this manner the report for the individuals has been generated.

**Candidate Module:**

* Candidate module is simply used to direct the candidate as well as to verifying the datas/informations of the candidate.
* If the candidate is not registered it provides the registration link for registration to be done.
* If the candidate is already registered it checks whether the details are correct or not.
* If the candidate forgets the password, it provides a link for his/her mail-id to get logged-in back.

**Service Module:**

* Service module can provide service only for the paid candidates.
* The paid candidates are known to be a prime candidate, whereas the prime candidate can able to have the contact number as well as mail-id of their partner.

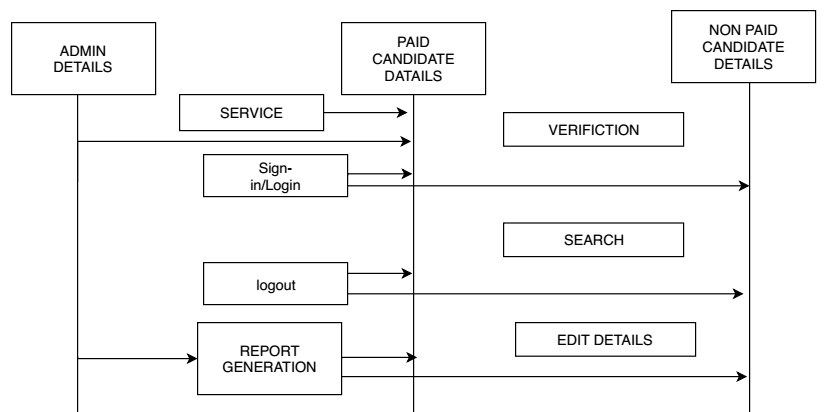
**Expectation Module:**

* Expectation modules contains the particulars that the Grooms/Brides expects from their partner such as qualification, salary etc.,

**Search Module:**

* Search module is used to search the Compatible pair for the Grooms/Brides.

**CONTROL FLOW DIAGRAM:**

****

**PSEUDO CODE:**

**78.P.1:ADMIN MODULE**

* Admin can login our account using email id
* After the login to identify the paid and non paid user datails
* If(paid user)
* Can able to view the partner detais
* Else
* Not ale to view the partner detail
* After the process log out the account

**78.P.2:CANDIDATE ENTRY**

* If(new user)
* get the all data from user with help of structure variables with the help of structure object
* else if
* login your account
* else
* forgotten your passworld request to send otp

**78.P.3:DATA ENTRY**

* Declare all variable in structure function
* Create structure object in main function
* Struct🡪object name
* Get the all the data in main fuction with the help of structure object
* objectname.variable

**78.P.4:FILE WRITE AND READ**

* Entered all the data stored in text file
* FILE\*pointer variable
* Use the file open( ) function to store entered data’s
* “w+”🡪read and write the data from user

**78.P.5:MATCHING MODULE**

* Verify if the user is paid or non paid
* If the user is paid call the matching module using callof( ) function
* After call( ) function they ask whether they need groom/bride details
* If(groom)
* Printf(print matching groom details)
* Else
* Printf(print bride details)

**78.P.5:SERVICE MODULE**

* If the user like to join paid member ship ifpaid( ) function is using
* Ifpaid( ) function get all required details like creadit card number,cvv number..,
* If(credit card number ==12&&cvv==3)
* The person is paid
* Else( )
* Re-enter card details
* Or else it will be exit

**ALGORITHM:**

INPUT: Registration/User-id.

OUTPUT: Matchmaking Profiles.

TERMINATION: Terminated when OTP is not entered.

**78.A.1 ADMIN MODULE:**

**Step1:** Click on admin login.

**Step2:** Enter the admin login-id.

**Step3:** Enter the password.

**Step4:** If password is correct, enter the Viewing report of the groom/bride informations.

**Step5:** Next Splitting the paid & unpaid candidate details.

**Step6:** Generating reports for both the candidates

.

**78.A.2 CANDIDATE MODULE:**

**Step1:** Check whether the candidate is existing user or new user.

**Step2:** If the candidate is existinguser , click on login candidate.

**Step3:** Enter the candidate user-id and enter the password.

**Step4:** If the password is correct, login to the candidate profile or else a link sends to registered mail-id.

**Step6:** If candidate is new user, open registration process.

**Step7:**  Enter the details of the groom/bride.

**Step8:** Create a login-id and generates the report**.**

**78.A.3 SERVICE MODULE:**

**Step1:** Check whether the candidate is paid user**.**

**Step2:** If paid**,** candidate can view the conduct details of matchmaking profile.

**Step3:** If not paid user, user can able to view the general details.

**78.A.4 EXPECTATION MODULE:**

**Step1:** Enter the expectation of the candidate

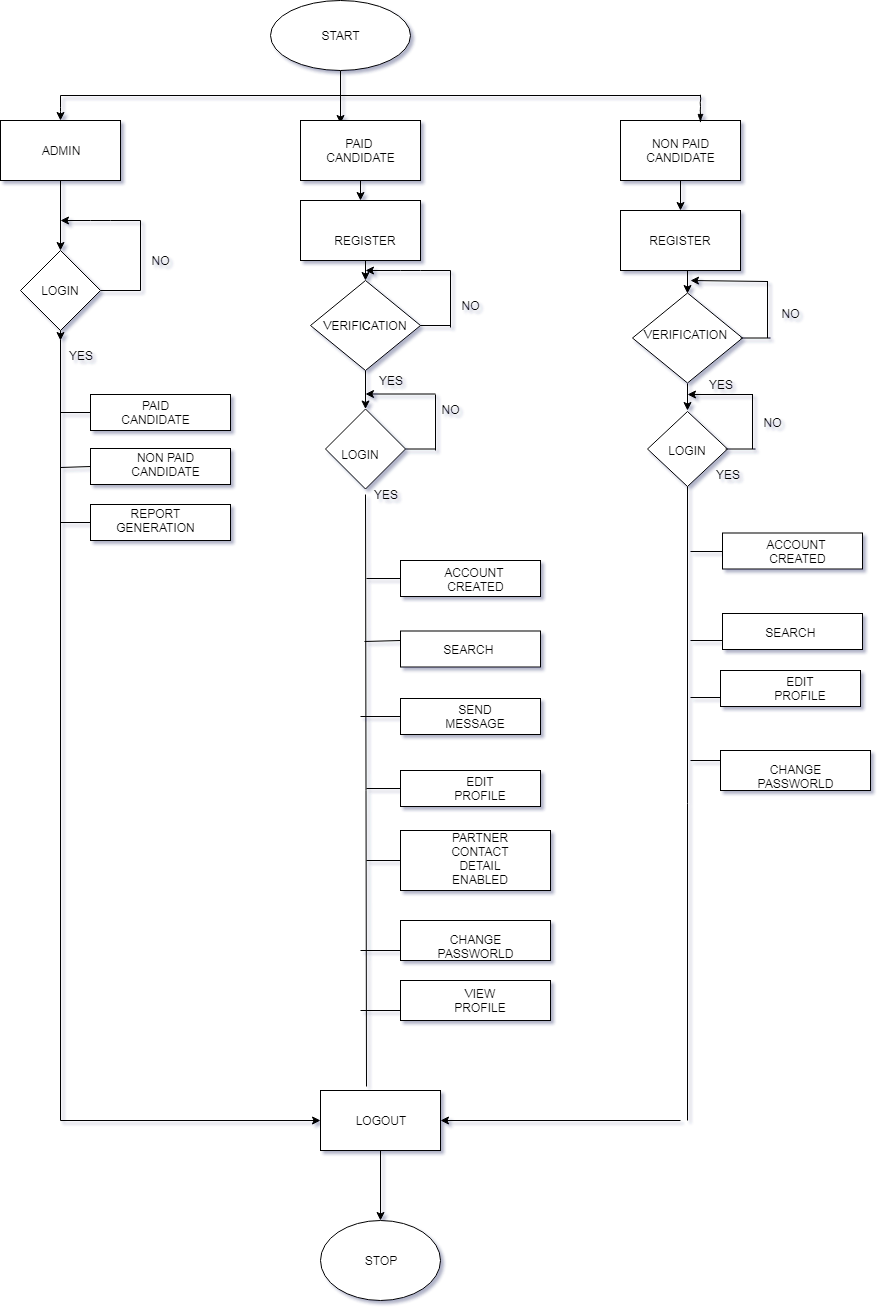
**Step2:** Details entered in the expectation block will be send directly to the user profile as **Expectation.**

**Step3**: Matchmaking of the groom/bride can be done by these expectation also.

**78**.**A.5 SEARCH MODULE:**

**Step1:** User can search the partner’s profile.

**Step2:** User have to enters the partner’s login id in search option and can able to view their profile details**.**

**FLOW CHART:**

**PROGRAM:**

#include<stdio.h>

#include<string.h>

#include<conio.h>

#include <stdlib.h>

#include<time.h>

struct mat

{

//variables

char mobile[20];

char pass[20];

char name[20];

int age;

char gender[20];

char dob[20];

char religion[20];

char motongue[20];

char caste[20];

char city[20];

char paiduser[20];

char qual[20];

char sal[20];

char creadit[20];

char cvv[5];

char card\_holdername[20];

char exdate[10];

char mobi[20];

};

void admin(struct mat);

void newuser(struct mat);

void exisistinguser(struct mat);

void forgotten(struct mat);

void match(struct mat);

void ifpaiduser(struct mat);

void callof(struct mat);

int i,c,otp,p=0,paid,var\_ad,var\_lenmob,var\_cmp;

char w;

//switch case n and array of structure element q

int n,q;

FILE \*nu;

FILE \*ex;

int main()

{

//file handling process

//exisisting user

struct mat m1;

printf("\t\t------MATRIMONY MANAGEMENT SYSTEM------\n\n");

printf("press 1 if you are a admin \n\npress 2 for new user \n\npress 3 for exisisting user \n\npress 4 if forgot passworld \n\npress 5 for matchmacking and expectation\n\n");

scanf("\n\n%d",&n);

printf("\n");

switch(n)

{//admin module

case 1:

{

admin(m1);

break;

}

//new use module

case 2:

{

newuser(m1);

break;

}

//exisisting user

case 3:

{

exisistinguser(m1);

break;

}

//forgot passworld

case 4:

{

forgotten(m1);

break;

}

case 5:

{

match(m1);

break;

}

default:printf("error!!!\n");

}

getch();

return 0;

}

void admin(struct mat m1)

{

printf("Enter mobile number\n\n");

scanf("%s",m1.mobile);

printf("\n");

var\_lenmob=strlen(m1.mobile);

if(var\_lenmob==10)

{

printf("Enter password\n\n");

do{

m1.pass[p]=getch();

if(m1.pass[p]!='\r')

{

printf("\*");

}

p++;

}while(m1.pass[p-1]!='\r');

1.pass[p-1]='\0';

printf("\n");

printf("\nYou have entered %s as password.\n\n",m1.pass);

printf("If you want to see candidate details press 1\n\n");

scanf("%d",&var\_ad);

printf("\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

if(var\_ad==1)

{char filepath[25]="d:\\test\\";

printf("Enter mobile number of a candidate\n\n") scanf("%s",m1.mobile);

printf("\n");

strcat(filepath,m1.mobile);

strcat(filepath,".txt");

printf(" %s\n\n ",filepath);

nu=fopen(filepath,"r");

fprintf(nu,"%s\n",m1.mobile);

/\*printf("\nEnter password\n");

//passworld hide process

do{

m1.pass[p]=getch();

if(m1.pass[p]!='\r')

{

printf("\*");

}p++; }while(m1.pass[p-1]!='\r');

m1.pass[p-1]='\0';

//printf("\nYou have entered %s as password.\n",m1.pass);\*/

while((w=fgetc(nu))!=EOF)

{printf("%c",w);}

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

else{

printf("Log out your account\n\n");

}}

else

{

printf("Re enter mobile number press 1\n\n");

scanf("%d",&var\_lenmob);

printf("\n");

if(var\_lenmob==1)

{admin(m1);}

else

{printf("Try again\n\n");}

}}

//new user module

void newuser(struct mat m1)

{char filepath[25]="d:\\test\\";

printf("Enter mobile number\n\n");

scanf("%s",m1.mobile);

printf("\n");

var\_lenmob=strlen(m1.mobile);

//printf("%d",var\_lenmob);

if(var\_lenmob==10)

{

strcat(filepath,m1.mobile);

strcat(filepath,".txt");

printf(" %s\n\n",filepath);

nu=fopen(filepath,"w+");

fprintf(nu,"\nmobile number=%s",m1.mobile);

printf("\n");

srand(time(NULL));

printf("Your OTP is: ");

for (i = 0; i < 9; ++i)

{c = '1' + rand() % ('1' - '9' + 1); }

printf("%d\n\n", c);

//otp generation processs

printf("Enter the Otp\n\n");

scanf("\n%d",&otp);

printf("\n");

if(otp == c)

{//password hide process

printf("Enter password\n\n");

do{

m1.pass[p]=getch();

if(m1.pass[p]!='\r')

{

printf("\*");

}

p++;

}while(m1.pass[p-1]!='\r');

m1.pass[p-1]='\0';

//printf("\nYou have entered %s as password.\n",m1.pass);

// fprintf(nu,"Password=%s\n",m1.pass);

printf("\n");

printf("\nEnter your name\n\n");

scanf("%s",m1.name);

printf("\n");

fprintf(nu,"\nName=%s",m1.name);

printf("Enter your age\n\n");

scanf("%d",&m1.age);

printf("\n");

fprintf(nu,"\nAge=%d",m1.age);

printf("Enter your gender\n\n");

scanf("%s",m1.gender);

printf("\n");

fprintf(nu,"\nGender=%s",m1.gender);

printf("Enter your date of birth\n\n");

scanf("%s",m1.dob);

printf("\n");

fprintf(nu,"\nDate of birth=%s",m1.dob);

printf("Enter your religion\n\n");

scanf("%s",m1.religion);

printf("\n");

fprintf(nu,"\nReligion=%s",m1.religion);

printf("Enter your mother tongue\n\n");

scanf("%s",m1.motongue);

printf("\n");

fprintf(nu,"\nMother tongue=%s",m1.motongue);

printf("Enter your caste\n\n");

scanf("%s",m1.caste);

printf("\n");

fprintf(nu,"\nCaste=%s",m1.caste);

printf("Enter your city\n\n");

scanf("%s",m1.city);

printf("\n");

fprintf(nu,"\nCountry=%s",m1.city);

printf("Enter your qualification\n\n");

scanf("%s",m1.qual);

printf("\n");

fprintf(nu,"\nqualification=%s",m1.qual);

printf("Enter your salary\n\n");

scanf("%s",m1.sal);

printf("\n");

fprintf(nu,"\nsalary=%s",m1.sal);

printf("if you want to join our paid membership enter 1 or else enter 0 \n\n");

scanf("%d",&paid);

printf("\n");

//service module

if(paid==1)

{

printf("partner detais are enabled and machmacking process enable\n\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

ifpaiduser(m1);

}

else

{

printf("you can view only partner profile\n\n");

fprintf(nu,"\n%s","not paid");

}

printf("Registered successfully!!!\n\n");

}

else

{

printf("Otp is incorrect Try again\n\n");

}

}

else

{ printf("Re enter mobile number press 1\n\n");

scanf("%d",&var\_lenmob);

printf("\n");

if(var\_lenmob==1)

{

newuser(m1);

}

else

{

printf("Try again\n\n");

}

}

fclose(nu);

}

//exisisting user module

void exisistinguser(struct mat m1)

{

char filepath[25]="d:\\test\\";

printf("Enter mobile number\n\n");

scanf("%s",m1.mobile);

printf("\n");

var\_lenmob=strlen(m1.mobile);

if(var\_lenmob==10)

{

strcat(filepath,m1.mobile);

strcat(filepath,".txt");

printf(" %s\n",filepath);

printf("\n");

nu=fopen(filepath,"r");

fprintf(nu,"\n%s",m1.mobile);

printf("\nEnter password\n\n");

//passworld hide process

do{

m1.pass[p]=getch();

if(m1.pass[p]!='\r')

{

printf("\*");

}

p++;

}while(m1.pass[p-1]!='\r');

m1.pass[p-1]='\0';

//printf("\nYou have entered %s as password.\n",m1.pass);

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

while((w=fgetc(nu))!=EOF)

{

printf("%c",w);

}

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

else

{

printf("Re enter mobile number press 1\n\n");

scanf("%d",&var\_lenmob);

printf("\n");

if(var\_lenmob==1)

{

exisistinguser(m1);

}

else

{

printf("Try again\n\n");

}

}

}

//forgotten password module

void forgotten(struct mat m1)

{

printf("Enter your registered mobile no\n\n");

scanf("%s",m1.mobile);

printf("\n");

var\_lenmob=strlen(m1.mobile);

if(var\_lenmob==10)

{

printf("The otp sended your mobile number\n\n");

srand(time(NULL));

printf("Your OTP is: ");

for (i = 0; i < 9; ++i)

{

c = '1' + rand() % ('1' - '9' + 1);

}

printf("%d\n\n", c);

scanf("\n%d",&otp);

printf("\n");

if(otp==c)

{

printf("Change your password\n\n");

do{

m1.pass[p]=getch();

if(m1.pass[p]!='\r')

{

printf("\*");

}

p++;

}while(m1.pass[p-1]!='\r');

m1.pass[p-1]='\0';

printf("\n");

//printf("\nYou have entered %s as password.\n",m1.pass);

printf("Your passworld changed successfully!!!\n\n");

}

else

{

printf("Otp is incorrect Try again\n\n");

}

}

else

{

printf("Re enter mobile number press 1\n\n");

scanf("%d",&var\_lenmob);

printf("\n");

if(var\_lenmob==1)

{

forgotten(m1);

}

else

{

printf("Try again\n\n");

}

}

}

//search & expectation module only for paid member

void match(struct mat m1)

{

char filepath[25]="d:\\test\\";

printf("Enter your mobile number \n\n");

scanf("%s",m1.mobile);

printf("\n");

var\_lenmob=strlen(m1.mobile);

if(var\_lenmob==10)

{

strcat(filepath,m1.mobile);

strcat(filepath,".txt");

printf(" %s\n\n",filepath);

nu=fopen(filepath,"r");

//matching variables

int loop, line=12;

char str[512];

char string[512];

if (nu == NULL) {

printf("Failed to open file\n");

}

printf("Enter if you are paid or not\n\n");

scanf("%s",string);

printf("\n");

for(loop = 0;loop<line;++loop){

fgets(str, sizeof(str), nu);

}

// printf("\nLine %d: %s\n", line, str);

if(strcmp(string,str) == 0 )

{

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("paid");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

callof(m1);

}

else

{

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("not match\n\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

fclose(nu);

}

else

{

printf("Re enter mobile number press 1\n\n");

scanf("%d",&var\_lenmob);

printf("\n");

if(var\_lenmob==1)

{

match(m1);

}

else

{

printf("Try again\n\n");

}

}

fclose(nu);

}

void callof(struct mat m1)

{

int o;

printf("Enter if you search groom press 1 or bride press 2...\n");

scanf("%d",&o);

if(o==1)

{

char filepath[25]="d:\\test\\8667790409";

strcat(filepath,".txt");

printf(" %s\n\n",filepath);

nu=fopen(filepath,"r");

//matching variables

char expectation[100];

int i=0,j=0,k=0,c=0;

int numProgs=0;

int numprogsa=0;

char\* programs[50];

char line[50];

// char var\_checkpaid[20];

// printf("%d",salary);

//fprintf(nu,"%s\n",m1.mobile);

/\* printf("\nEnter password\n");

//passworld hide process

do{

m1.pass[p]=getch();

if(m1.pass[p]!='\r')

{

printf("\*");

}

p++;

}while(m1.pass[p-1]!='\r');

m1.pass[p-1]='\0';

printf("\n");

//printf("\nYou have entered %s as password.\n",m1.pass);\*/

while(fgets(line, sizeof line, nu)!=NULL)

{

//check to be sure reading correctly

printf("%s", line);

//add each filename into array of programs

programs[i]=line;

programs[k]=line;

k++;

i++;

//count number of programs in file

numProgs++;

numprogsa++;

}

for (j=0 ; j<numProgs+1; j++)

{

if(strcmp(programs[j],expectation)==0);

{

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nIt is your matching partner profile\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

break;

}

}

}

else if(o==2)

{

char filepath[25]="d:\\test\\1234567890";

strcat(filepath,".txt");

printf(" %s\n\n",filepath);

nu=fopen(filepath,"r");

//matching variables

char expectation[100];

int i=0,j=0,k=0,c=0;

int numProgs=0;

int numprogsa=0;

char\* programs[50];

char line[50];

// char var\_checkpaid[20];

// printf("%d",salary);

//fprintf(nu,"%s\n",m1.mobile);

/\* printf("\nEnter password\n");

//passworld hide process

do{

m1.pass[p]=getch();

if(m1.pass[p]!='\r')

{

printf("\*");

}

p++;

}while(m1.pass[p-1]!='\r');

m1.pass[p-1]='\0';

printf("\n");

//printf("\nYou have entered %s as password.\n",m1.pass);\*/

while(fgets(line, sizeof line, nu)!=NULL)

{

//check to be sure reading correctly

printf("%s", line);

//add each filename into array of programs

programs[i]=line;

programs[k]=line;

k++;

i++;

//count number of programs in file

numProgs++;

numprogsa++;

}

for (j=0 ; j<numProgs+1; j++)

{

if(strcmp(programs[j],expectation)==0);

{

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nIt is your matching partner profile\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

break;

}

}

}

fclose(nu);

}

void ifpaiduser(struct mat m1)

{

int l1=0,l2=0,l3;

printf("Enter your creadit card number\n\n");

scanf("%s",m1.creadit);

printf("\n");

printf("Enter the card holder name\n\n");

scanf("%s",m1.card\_holdername);

printf("\n");

printf("Enter expiry date\n\n");

scanf("%s",m1.exdate);

printf("\n");

printf("Enter cvv number\n\n");

scanf("%s",m1.cvv);

printf("\n");

l1=strlen(m1.creadit);

//printf("%d\n%d",m1.creadit,m1.cvv);

l2=strlen(m1.cvv);

if(l1==12&&l2==3)

{

printf("Card verified\n\n");

printf("you are paid successfully\n\n");

// scanf("%s",m1.paiduser);

printf("\n");

fprintf(nu,"\nyour are paid=%s","paid user");

}

else

{

printf("Re enter your card details press 1\n\n");

scanf("%d",&l3);

printf("\n");

if(l3==1)

{

ifpaiduser(m1);

}

else

{

printf("payment failed\n\n");

}

}

}

**QUALITY ASSURANCE (TEST CASES):**

**TITLE BOX:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SERIAL NO** | **INPUT** | **OUTPUT** | **STATUS** |
| 1 | Press 1 if you are an admin | Enter Mb.no & password | Pass |
| 2 | Press 2 if you are new user | Registration process | Pass |
| 3 | Press 3 if you are an existing user | Profile is displayed | Pass |
| 4 | Press 4 if you have forgotton password | Generate otp | Pass |
| 5 | Press 5 for matchmaking and expectation | Matchmaking profile | Pass |

**ADMIN MODULE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SERIAL NO** | **INPUT** | **OUTPUT** | **STATUS** |
| 1 | Press 1 for admin | Enter your number | pass |
| 2 | Enter password | Password is visible | pass |
| 3 | Enter candidate mobile number | The candidate details are visible | pass |

**REGISTRATION MODULE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SERIAL NO** | **INPUT** | **OUTPUT** | **STATUS** |
| 1 | Enter your mobile no | A file is created.A OTP is generated | Pass |
| 2 | Enter your OTP | Create password | Pass |
| 3 | If OTP is wrong | OTP is incorrect try again | Pass |
| 4 | Registration process | Details are stored as record | Pass |
| 5 | If candidate is willing to be paid member | Partners contact details are visible | Pass |
| 6 | If not a paid member | Visible only general details | Pass |

**EXISTING USER:**

|  |  |  |  |
| --- | --- | --- | --- |
| **SERIAL NO** | **INPUT** | **OUTPUT** | **STATUS** |
| 1 | Press 3 for existing user | Enter your mobile no | Pass |
| 2 | Enter password | Entered password is hidden | Pass |
| 3 | After login process | Records are displayed | Pass |

**FORGOT PASSWORD:**

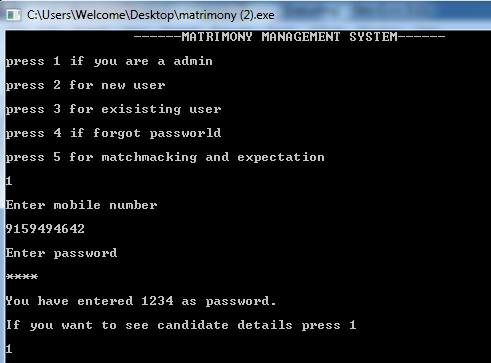
|  |  |  |  |
| --- | --- | --- | --- |
| **SERIAL NO** | **INPUT** | **OUTPUT** | **STATUS** |
| 1 | Enter your registered mobile no | Otp is generated | Pass |
| 2 | Create new password | New password is generated | Pass |

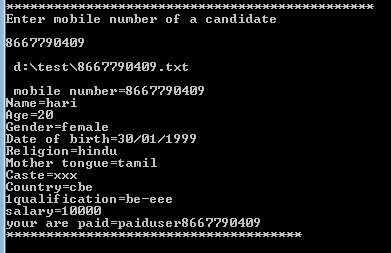
**MATCHMAKING AND EXPECTATION:**

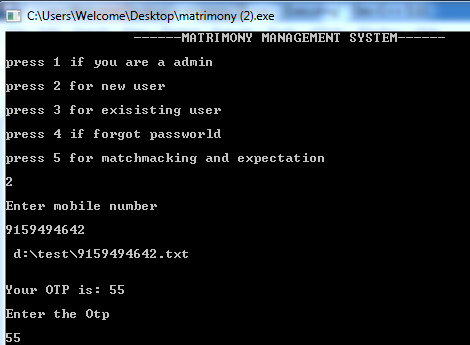
|  |  |  |  |
| --- | --- | --- | --- |
| **SERIAL NO** | **INPUT** | **OUTPUT** | **STATUS** |
| 1 | Enter paid candidate  Mobile number | Mobile number displayed | Pass |
| 2 | Find if they person is paid/not | If paid proceed  Next step | Pass |
| 3 | Enter if yousearch groom/bride | Matching groom/bride details visible | Pass |

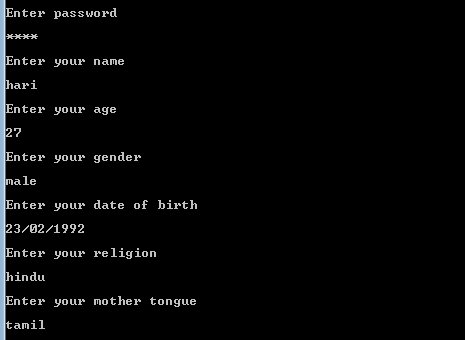
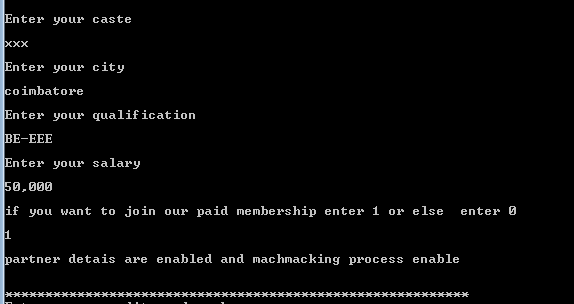
**OUTPUT SCREENS:**

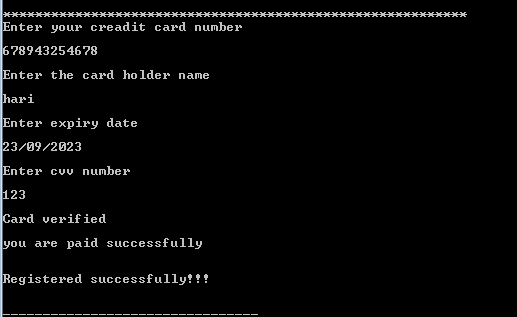
**ADMIN MODULE**

****

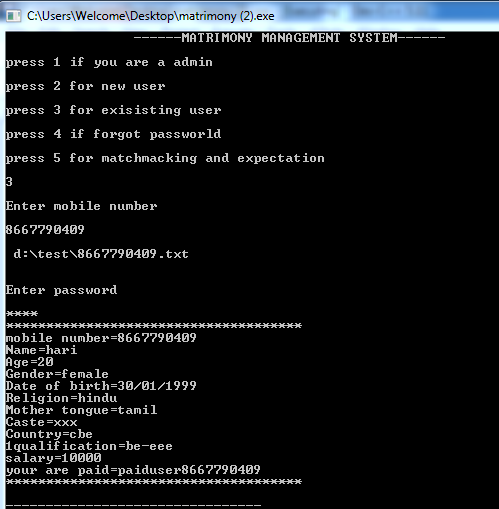
****

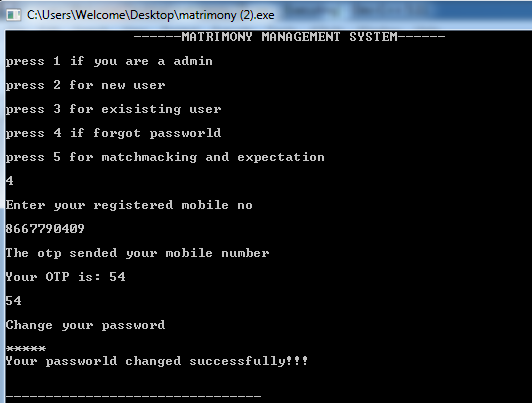
**CANDIDATE MODULE**

****

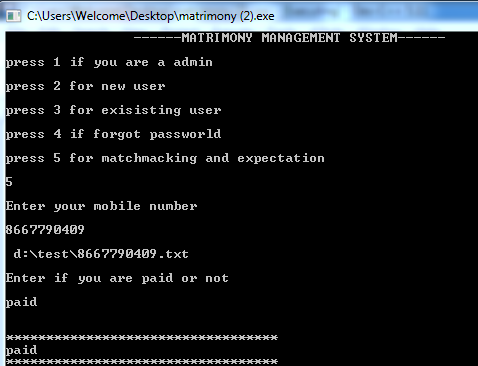
****

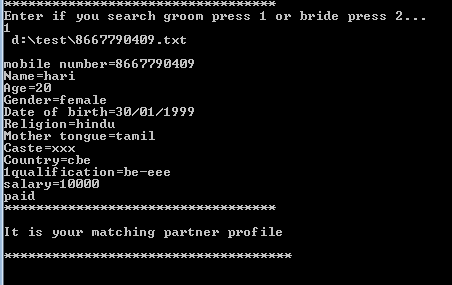
**EXSISTING USER**

****

**FORGOT PASSWORD**

**MATCHING MODULE**

****



**COMPLEXITY:**

**No of lines:**619

**Execution time:**16.2 seconds

**CONCLUSION:**

Matrimonial management system is to provide Grooms and Brides with excellent matchmaking experience by exploring the opportunities and resources to meet true potential Life Partners .This system provides a platform to a lot of Grooms/Brides for finding perfect match.

**FUTURE ENHANCEMENT:**

In future my project looks useful as today. If user necessities increase with a little effort it can updated.This system maintains the proper levels of information , privacy and security around all the activities.

**REFERENCES:**

<https://www.scribd.com/doc/29584953/Matrimonial-Project-Report>

<https://www.scribd.com/document/60785625/Online-Matrimonial-Website>

<https://ieeexplore.ieee.org/document/6781349>