

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

This online matrimonial site is mainly developed to let individual find their potential matches for marriage according to their priorities set. This project allows the phrase 'Marriages are made in heaven' to be rephrased as 'Marriages are now made online'. This application allows browsing profiles of those who have registered themselves on this site. This allows individual to give their information such as Name, Gender, Religion, Caste, Marital status, Current salary, Occupation etc. Thus this application allows individual to let others view its profile and also allows it to view others profile depending on the priority set.

1.2 PROJECT SPECIFICATION

This is a website which sell products at a low price. The system includes 2 modules. They are:

- **Registration**-This module helps individual who are looking for marriage to register themselves in order to use the services of this application.
- **Create profile**-This application allows individual to create profile which includes all the details that he/she wishes to let other registered users to know.
- **Prioritize profile**-This application allows individual to set priorities according to which the profiles will be displayed to the user.
- **Search profile**-The individual can search profiles on this site which matches their requirement.
- **Chatting**-This allows an interested individual to send chats by both of their permission.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analysing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

The existing system is all type of service entries are registered to directly by people. And all records are keeping manually. So it's very difficult and time consuming. Since all these services are carried out by the normal people they are suffered from long distance journey. Client side validations are performed to check entry of invalid fields or incorrect format. The user id and password are checked thoroughly for existing users to ensure security. Stored procedures and classes are used to increase the speed of processing.

2.3 DRAWBACKS OF EXISTING SYSTEM

The limitations of current system are mentioned below:

- The people should enquire directly about the details of the products and grounds..
- Require more space to keep the paper documents.
- More time consuming

2.4 PROPOSED SYSTEM

This system provides good and efficient information and thus making the service smarter. Some of the Features,

- High Speed response for users.
- Easy to handle and flexible.
- Fast and convenient
- High accuracy and saving.
- Fully online service

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is a process to check possibilities of system development. It is a method to check various different requirements and availability of financial & technical resources. Before starting the process various parameters must be checked like:

- Estimated finance is there or not?
- The man power to operate the system is there or not?
- The man power is trained or not?

All the above conditions must be satisfied to start the project. This is why in depth analysis of feasibility is carried out.

There are three different ways feasibility can be tested

- 1) Economical Feasibility
- 2) Technical Feasibility
- 3) Operational Feasibility.

3.1.1 Economical Feasibility

In economical feasibility, analysis of the cost of the system is carried out. The system should be only developed if it is going to give returned the current manual system user can get the price only by purchasing the newspapers. In addition if he/she wants to see archives of particular equity then he has to refer to all the old newspapers. For research reports he has to buy another magazine. So Instead of buying no of magazines user has to just go online and with a single click he can get whatever information he wants. So our project of online share news passes the test of economical feasibility.

3.1.2 Technical Feasibility

It is basically used to see existing computer, hardware and software etc, weather it is sufficient or additional equipments are required? Minimum System Requirement is such that it can be affordable by of the user who is having computer. All the user

requires is compatible browser and .net framework installed so our system is fully technical feasible.

3.1.3 Operational Feasibility.

Once the system is designed there must be trained and expert operator. If there are not trained they should given training according to the needs of the system. From the user's perspective our system fully operational feasible as it just requires some knowledge of computer. Operators only need add daily prices of various equities and there are enough validations available so operator does not require any special technical knowledge. So our system also passes the test of operational feasibility.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Pentium IV/AMD Dual core
RAM	-	4 GB
Hard disk	-	500 GB

3.2.2 Software Specification

Front End	-	PHP
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by RasmusLedorf in 1995, the

reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:Hypertext Preprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data.

It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

MySQL is a database management system.

database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL databases are relational.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming

environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

The MySQL Database Server is very fast, reliable, scalable, and easy to use.

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It's a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

The purpose of the design is to create architecture for the evolving implementation and to establish the common tactical policies that must be used by desperate elements of the system. We begin the design process as soon as we have some reasonably completed model of the behaviour of the system. It is important to avoid premature designs, wherein develop designs before analysis reaches closer. It is important to avoid delayed designing where in the organization crashes while trying to complete an unachievable analysis model.

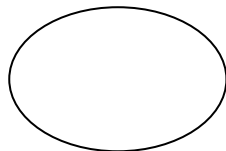
Throughout the project, the context flow diagrams, data flow diagrams and flow charts have been extensively used to achieve the successful design of the system. In our opinion, “efficient design of the data flow and context flow diagrams helps to design the system successfully without much major flaws within the scheduled time”. This is the most complicated part in a project. In the designing process, my project took more than the activities in the software life cycle. If we design a system efficiently with all the future enhancements, the project will never become junk and it will be operational.

The data flow diagrams were first developed by Larry Constantine as way for expressing system requirements in graphical form. A data flow diagram also known as “bubble” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. It functionally decomposes the requirement specification down to the lowest level. DFD depicts the information flow, the transformation flow and the transformations that are applied as data move from input to output. Data Flow Diagram is quite effective, especially when the required design is unclear and the user and analyst need a notational language for communication. It is used to model the system components such as the system process, the data used by the process, any external entities that interact with the system and information flows in the system.

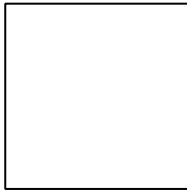
Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

Main symbols used in the data flow diagram are:

Circle represents a process that transforms incoming data flows in to outgoing data flows.



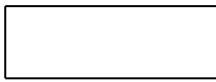
A square defines a source and destination of system data.



Arrow identifies data in motion.



An open rectangle defines a data store, data at rest or temporary repository of data.



Steps to Construct Data Flow Diagrams:-

Four steps are commonly used to construct a DFD:

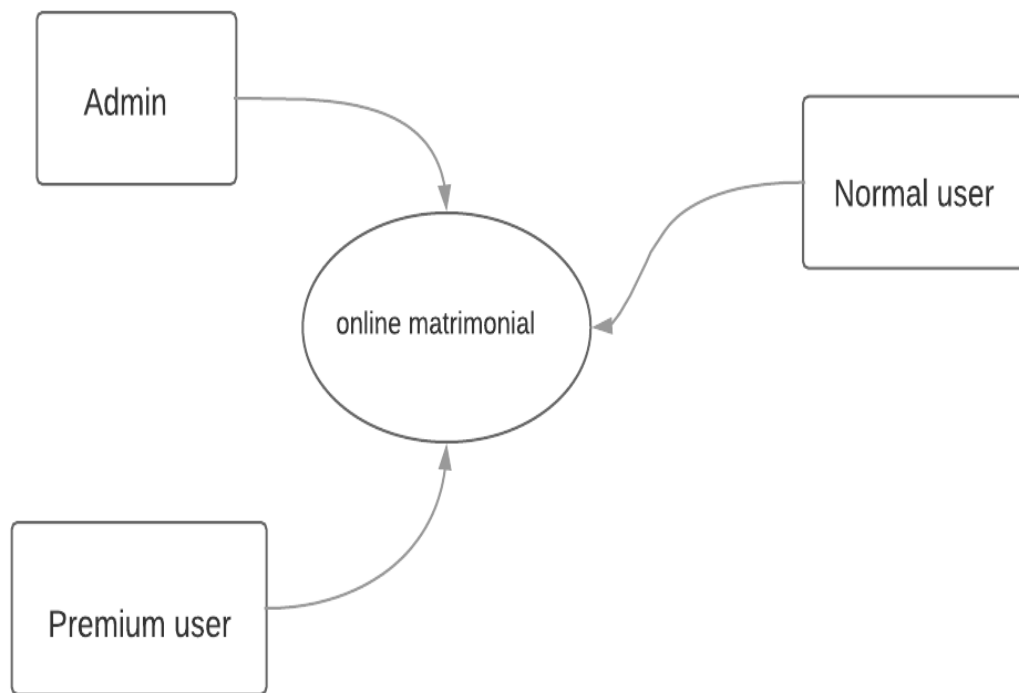
1. Process should be named and numbered for easy reference. Each name should be representative of the process.
2. The destination of flow is from top to bottom and from left to right.
3. When a process is exploded in to lower level details they are numbered.
4. The names of data stores, sources and destinations are written in capital letters.

Rules for constructing a Data Flow Diagram

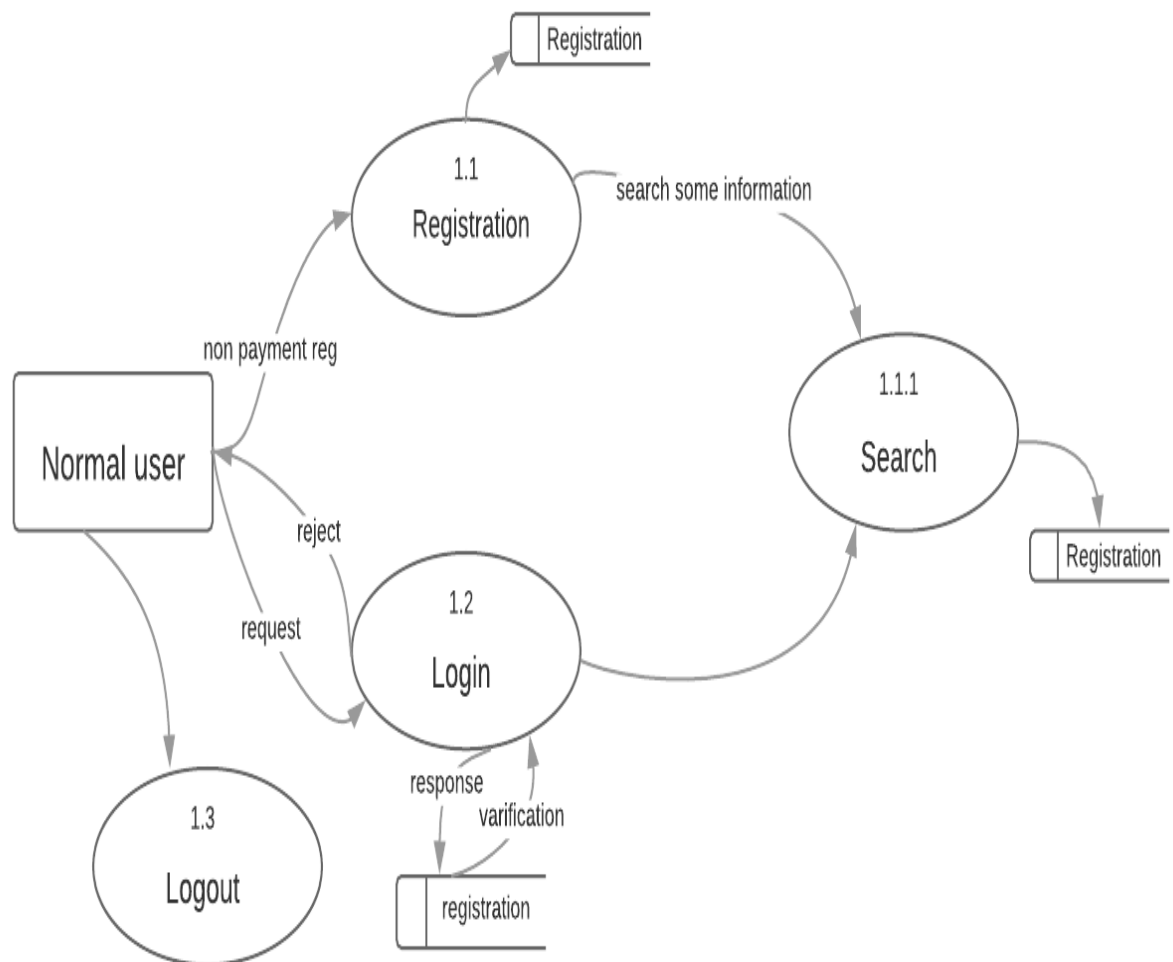
- Arrows should not cross each other.
- Squares, circles and files must bear names.
- Decomposed data flow squares and circles can have same names.
- Draw all data flow around the outside of the diagram.

Data Flow Diagrams of Catering Management System

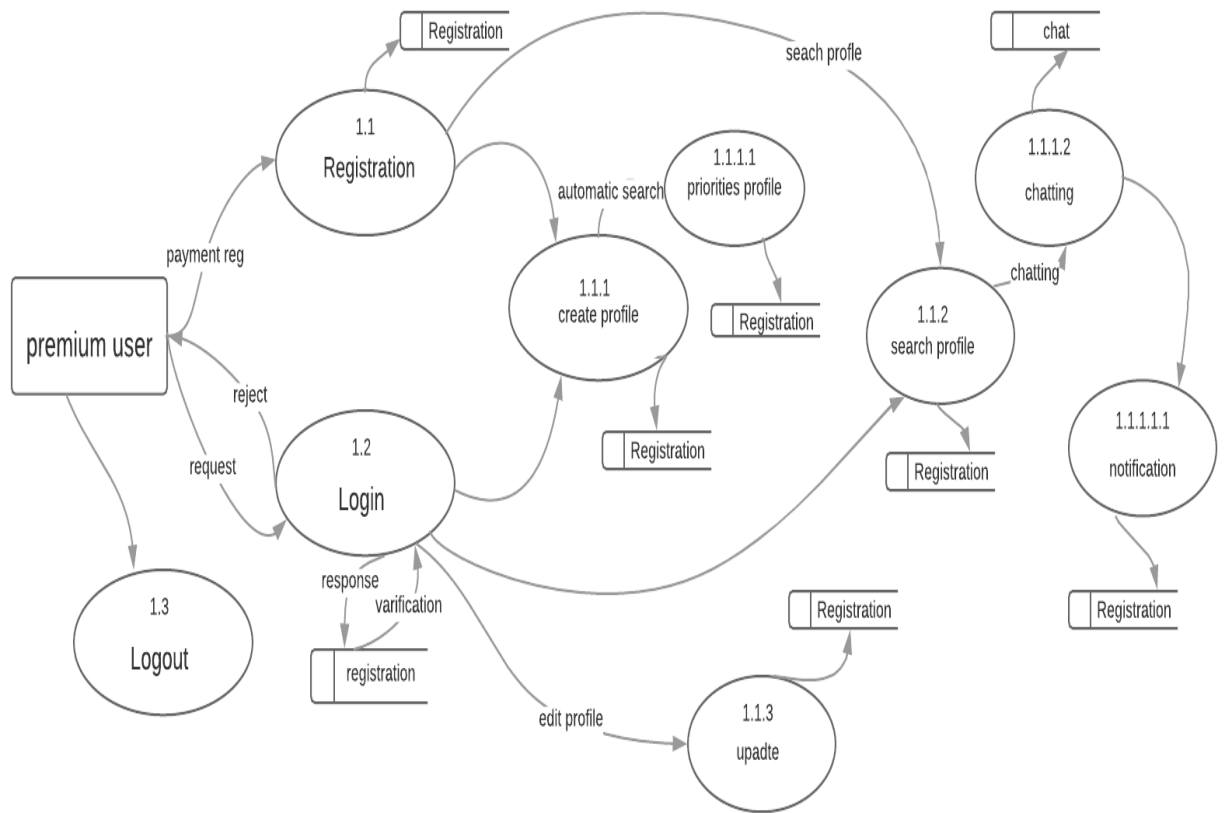
Level -0



Level 0 data flow diagram

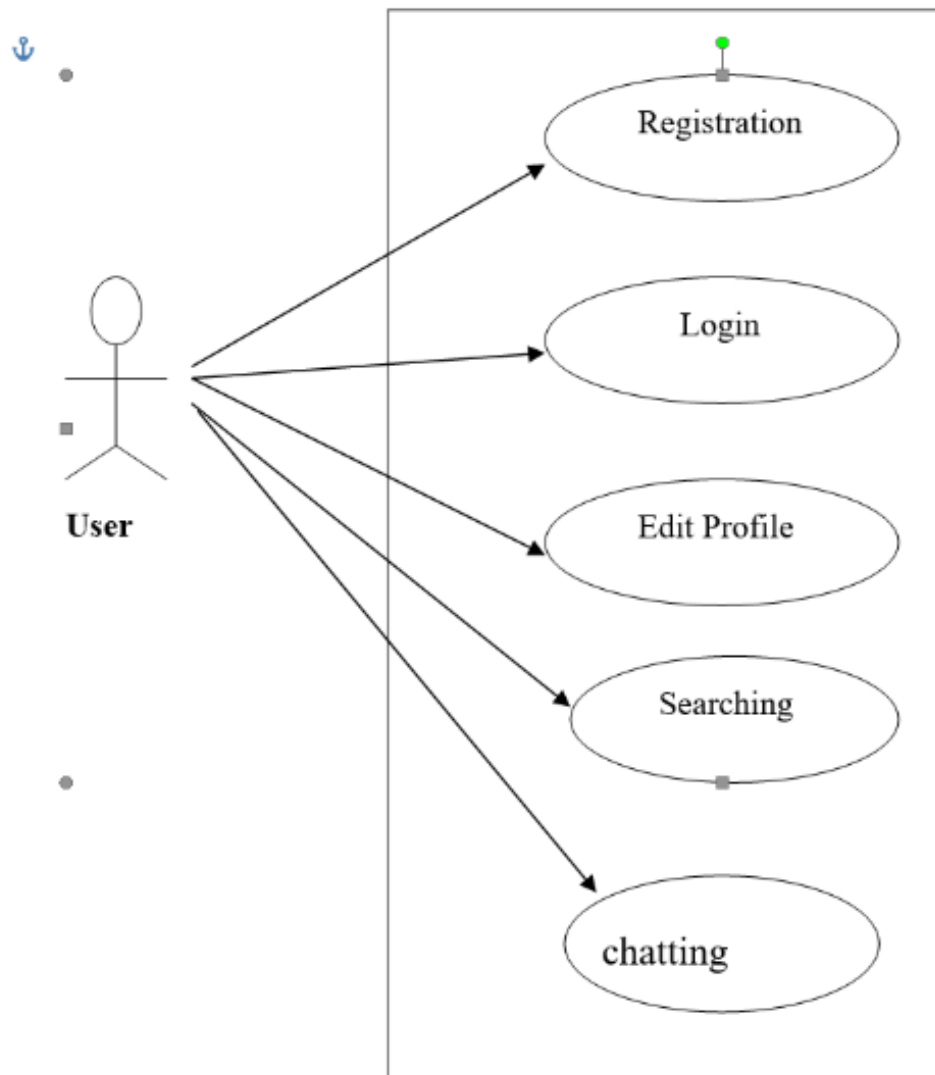
LEVEL 1**NORMAL USER**

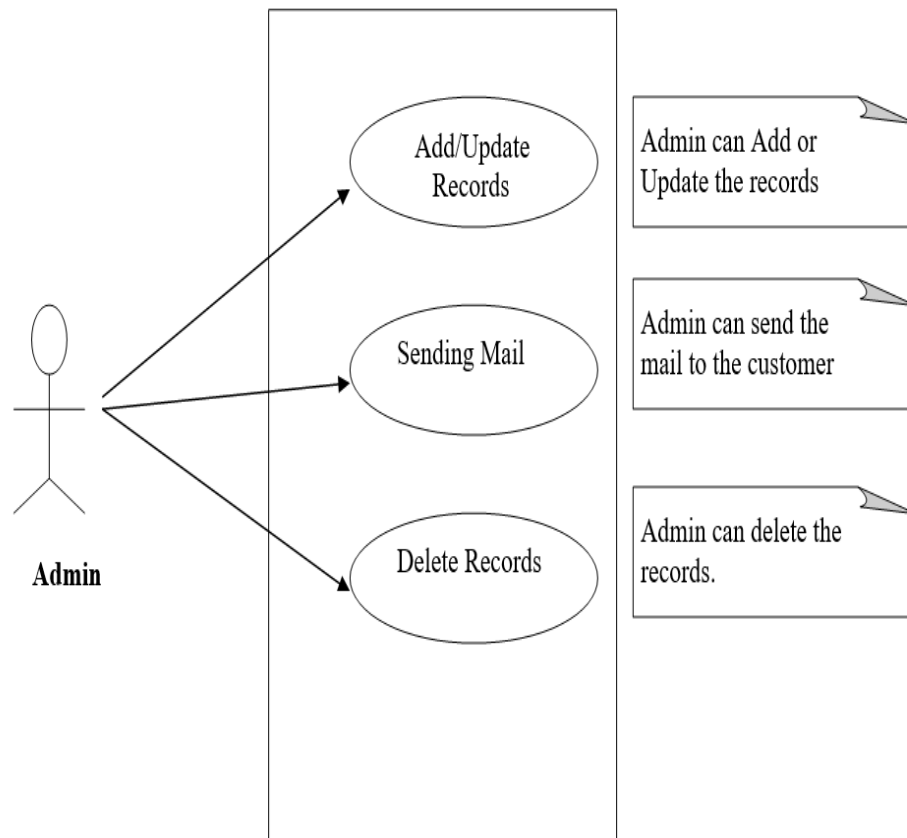
PREMIUM USER



4.3 UML DIAGRAM

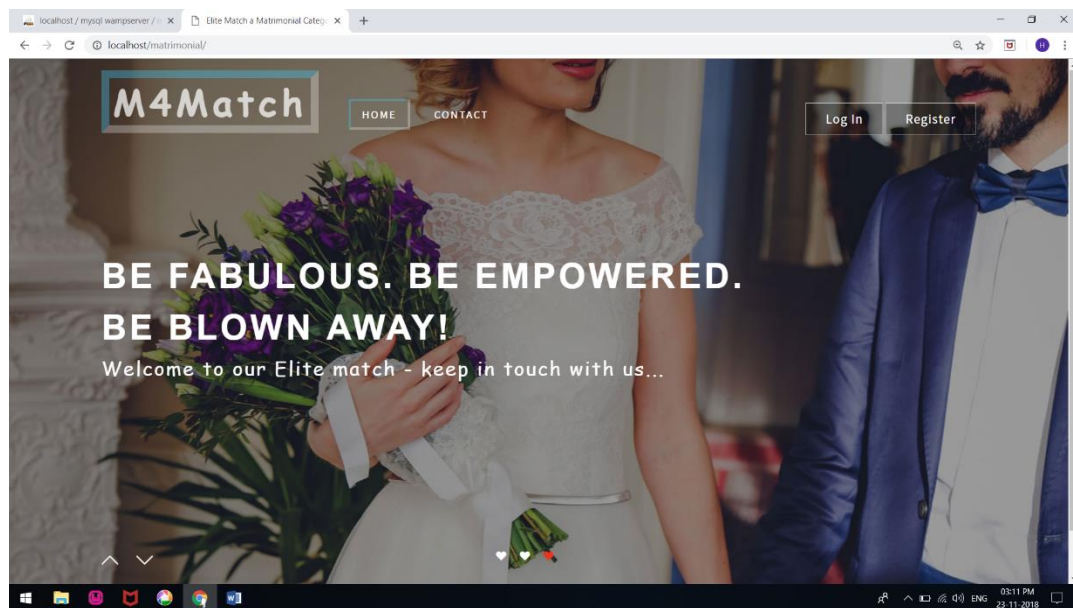
User



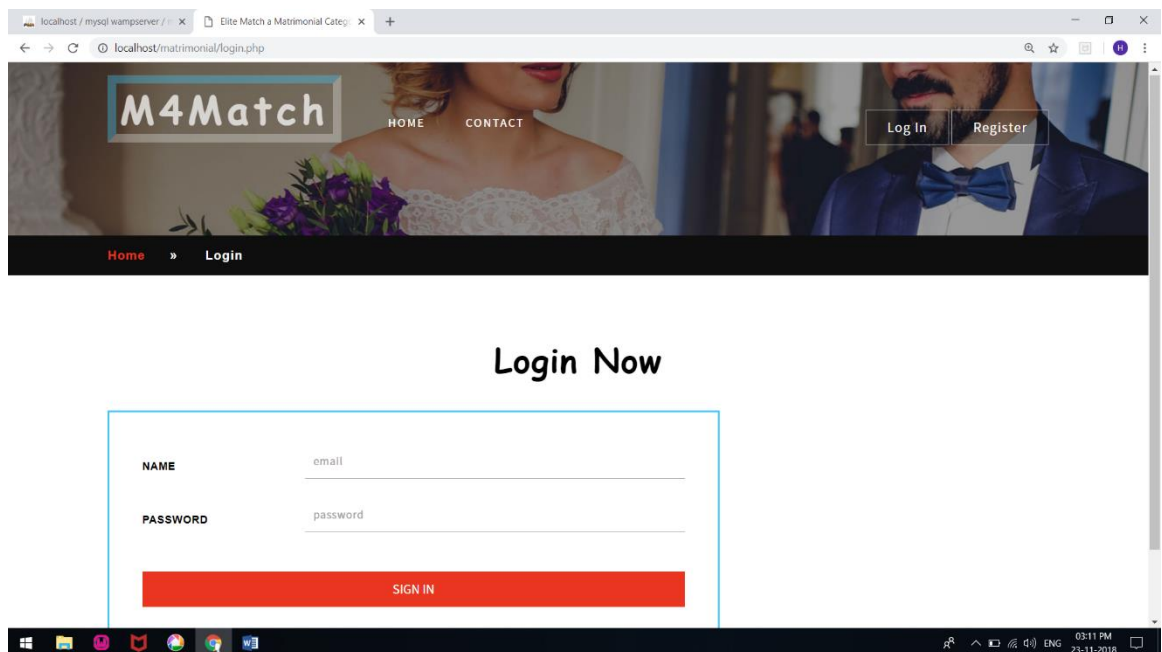
Admin

4.4 USER INTERFACE DESIGN

Home



Login



Registration

localhost / mysql wampserver / ... x Elite Match a Matrimonial Cate... x +

localhost/matrimonial/register.php

M4Match HOME CONTACT Log In Register

Home » Register

Register Now

PROFILE FOR Select

NAME Name

GENDER ☐ Male ☐ Female

03:12 PM 23-11-2018

4.5. Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.5.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.5.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two

types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it

satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this, I decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attributes

TABLES

Tbl_login

L_id :pk

F_Name	Data Type	Constrains	Description
L_id	Int	Not null	Identify login users
E_mail	Varchar (50)	Not null	Refers from tbl_reg
Password	Varchar (20)	Not null	Check password for success/failure

Tbl_registration**R_id :pk , p_id:ck**

F_Name	Data Type	Constrains	Description
R_id	Int	Not null	Identify normal users
Name	Varchar (50)	Not null	Identify users name
Age	Int	Not null	Identify age
Cast	Varchar (20)	Not null	Identify cast
Gender	Varchar (20)	Not null	Identify male or female
P_id	Int	Null	Identify premium users
Address	Varchar (50)	Null	Identify address
occupation	Varchar (20)	Null	Identify job
Education	Varchar (20)	Null	Identify graduation details
Phone no	Int	Null	Contact number

Tbl_cast**C_id : pk**

F_Name	Data Type	Constrains	Description
C_id	Int	Not null	Identify cast
Religion	Varchar (20)	Not null	Add new religion
cast	Varchar (20)	Not null	Add new cast

Tbl_payment**B_id :pk , p_id : fk**

F_Name	Data Type	Constrains	Description
B_id	Int	Not null	Identify bank id
B_Name	Varchar (50)	Not null	Identify bank name
Account no	Int	Not null	Confirm account
Cvv	Int	Not null	Card verification value
Card no	Int	Not null	Verify card holder
P_id	Int	Null	Refers from tbl_reg

Tbl_chat**P_ide :pk , p_id :fk**

F_Name	Data Type	Constrains	Description
P_ide	Int	Not null	Identify sender
Date	Date	Null	Identify chat date
Time	Time	Null	Identify chat time
Description	Varchar(200)	Not null	Message
P_id	Int	Null	Refers from tbl_reg and identify receiver

Tbl_family**f_id :pk**

F_Name	Data Type	Constrains	Description
f_id	Int	Not null	Identify family id
f_Name	Varchar (50)	Not null	Identify father name
f_occupation	Varchar(50)	Not null	Identify father job
M_name	Varchar(50)	Not null	Identify mother name
M_occupation	Varchar(50)	Not null	Identify mother job
brothers	Int	Not Null	No of brothers
sisters	Int	Not null	No of sisters

Tbl_request**Re_id :pk**

F_Name	Data Type	Constrains	Description
R_id	Int	Not null	Identify id
Date	Date	Not null	Identify chat date
email	Varcha(50)	Not null	Identify sender
V_name	Varchar(200)	Not null	Receiver

Tbl_looking**L_id :pk**

F_Name	Data Type	Constrains	Description
l_id	Int	Not null	Identify looking id
L_age	Int	Not null	Looking age
L_height	Int	Not null	Looking height
L_weight	Int	Not null	Looking weight
L_cast	Varchar(50)	Not null	Looking cast
L_religion	Varchar (50)	Not Null	Looking religion
Diet	Varchar(50)	Not null	Non/veg

Tbl_image**I_id : pk**

F_Name	Data Type	Constrains	Description
I_id	Int	Not null	Identify Image
Images	Varchar (20)	Not null	Add new image

Tbl_personal**p_id :pk**

F_Name	Data Type	Constrains	Description
p_id	Int	Not null	Identify personal id
M_toung	Varchar(50)	Not null	Identify language
B_group	Varchar(50)	Not null	Identify blood
weight	Int	Not null	Identify weight
height	Int	Not null	Identify height
Diet	Varchar(50)	Not null	Non/veg

Tbl_contact**co_id:pk**

F_Name	Data Type	Constrains	Description
co_id	Int	Not null	Identify contact id
Name	Varchar(50)	Not null	Identify name
Email	Varchar(50)	Not null	Identify email
sub	Varchar(200)	Not null	Identify subject
Message	Varchar(200)	Not null	Identify message

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Software testing should not be confused with debugging. Debugging is the process of analysing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components.

The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover phase.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Matrimonial Web Application is to provide Grooms and Brides with excellent matchmaking experience by exploring the opportunities and resources to meet true potential partner.

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 Bride/Groom can use match By Email functionality so he/she can get directly E-mail alert for the match which fulfil their required criteria. It helps the user by providing profiles of perspective “Bride” or “Groom” and other information regarding them online.

Matrimonial web application provide facility like quick tour.this is a module that contains the flow of the website .Here user can have a idea how he can commit himself in the website.

Matrimonial web application provide facility to change preference about partner.

This application provide facility like edit profile, update photo and delete photo, hide profile, create album, send express interest, send personal message, apply for loan to the user.

7.2 FUTURE SCOPE

- It is possible to provide the web space to the users for creating his portal.
- It is possible to create our own mail server.
- It is possible to create chat server so that user can communicate with each other.
- It is possible to provide facility like create video album.

CHAPTER 8

BIBLIOGRAPHY

BOOKS/REFERENCES:

- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “*Software engineering: a precise approach*”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- www.agilemodeling.com/artifacts/useCaseDiagram.htm

CHAPTER 9

APPENDIX

SAMPLE CODE

register.php

<!--

Author: W3layouts

Author URL: <http://w3layouts.com>

License: Creative Commons Attribution 3.0 Unported

License URL: <http://creativecommons.org/licenses/by/3.0/>

-->

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Elite Match a Matrimonial Category Bootstrap Responsive Website Template | Register :: W3layouts</title>

<!-- for-mobile-apps -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="Elite Match Responsive web template, Bootstrap Web Templates, Flat Web Templates, Android Compatible web template,

Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, Sony Ericsson, Motorola web design" />

<script type="application/x-javascript"> addEventListener("load", function() { setTimeout(hideURLbar, 0); }, false);

function hideURLbar(){ window.scrollTo(0,1); } </script>

<!-- //for-mobile-apps -->

```
<link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />

<link href="css/font-awesome.css" rel="stylesheet">

<link rel="stylesheet" href="css/jquery-ui.css" />

<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />

<!--fonts-->

<!--<link href="//fonts.googleapis.com/css?family=Raleway:400,500,600,700,800,900"
rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700"
rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Dancing+Script:400,700" rel="stylesheet">

<!--//fonts-->-->

</head>

<body>

<!--banner start here-->

<div class="banner-w3ls inner-banner-agileits" id="home">

<div class="container">

<!-- header -->

<div class="header-inner">

<h1 class="logo">

<a href="index.php">M4Match</a></h1>

<nav class="navbar navbar-default">

<div class="navbar-header">
```

```
<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
```

```
<span class="sr-only">Toggle navigation</span>
```

```
<span class="icon-bar"></span>
```

```
<span class="icon-bar"></span>
```

```
<span class="icon-bar"></span>
```

```
</button>
```

```
</div>
```

```
<!-- navbar-header -->
```

```
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
```

```
<ul class="nav navbar-nav navbar-right">
```

```
<li><a href="index.php">Home</a></li>
```

```
</li>
```

```
<li><a href="contact.php">Contact</a></li>
```

```
</ul>
```

```
</div>
```

```
<div class="clearfix"> </div>
```

```
</nav>
```

```
<div class="header-right-w3ls">
```

```
<a href="login.php">Log In</a>
```

```
<a href="register.php">Register</a>
```

```
</div>
```

```
<div class="clearfix"></div>

</div>

</div>

</div>

<!--/home-->

<!--banner end here-->

<!-- breadcrumbs -->

<div class="w3l_agileits_breadcrumbs">

<div class="container">

<div class="w3l_agileits_breadcrumbs_inner">

<ul>

<li><a href="index.html">Home</a><span>#187;</span></li>

<li>Register</li>

</ul>

</div>

</div>

</div>

<!-- //breadcrumbs -->

<div class="login elite-app">

<div class="container">

<div class="tittle-agileinfo">

<h3>Register Now</h3>
```


</div>

<div class="col-md-8 login-form-w3-agile">

<form action="regaction.php" method="post" name="registration" enctype="multipart/form-data">

<div class="w3_form_body_grid">

Profile For

<select id="w3_country" name="pf" onChange="change_country(this.value)" class="frm-field required">

<option value="Select">Select</option>

<option value="Myself">Myself</option>

<option value="Son">Son</option>

<option value="Daughter">Daughter</option>

<option value="Brother">Brother</option>

<option value="Sister">Sister</option>

<option value="Relative">Relative</option>

<option value="Friend">Friend</option>

</select>

</div>

<div class="w3_form_body_grid">

Name

<input type="text" name="r_name" pattern="^[a-zA-Z]+\$" placeholder="Name" required="">

</div>

```
<div class="w3_form_body_grid">

<span>Gender</span>

<div class="w3_gender">

<div class="colr ert">

<label class="radio"><input type="radio" name="gender" value="male"
checked=""><i></i>Male</label>

</div>

<div class="colr">

<label class="radio"><input type="radio" value="female"
name="gender"><i></i>Female</label>

</div>

<div class="clearfix"> </div>

</div>

<div class="clearfix"> </div>

</div>

<div class="w3_form_body_grid w3_form_body_grid1">

<span>Date Of Birth</span>

<input class="date" id="datepicker" name="dob" type="text" value="mm/dd/yyyy"
onFocus="this.value = '";" onBlur="if (this.value == ") {this.value = '2/08/2013';}"
required="">

</div>

<div class="w3_form_body_grid">

<span>religion</span>

<select id="w3_country1" onChange="change_country(this.value)" name="religion"
class="frm-field required">
```

```
<?php

session_start();

include ("dbconnect.php");

$sql="select relegion from cast ";

$result=mysql_query($sql,$con);

$rowcount=mysql_num_rows($result);

echo $rowcount;

if($rowcount !=0 )

{

while($row=mysql_fetch_array($result))

{

$relegion=$row['relegion'];

echo"<option value='$relegion'$>$relegion</option>";

}

}

?>

</select>

</div>

<div class="w3_form_body_grid">

<span>cast</span>

<select id="w3_country1" onChange="change_country(this.value)" name="cast" class="frm-
field required">
```

```
<?php

session_start();

include ("dbconnect.php");

$sql="select c_cast from cast ";

$result=mysql_query($sql,$con);

$rowcount=mysql_num_rows($result);

echo $rowcount;

if($rowcount !=0 )

{

while($row=mysql_fetch_array($result))

{

$c_cast=$row['c_cast'];

echo"<option value='$c_cast'$c_cast</option>";

}

}

?>

</select>

</div>

<div class="w3_form_body_grid w3_form_body_grid1">

<span>Address</span>

<textarea rows="5" name="address" cols="40"></textarea>

</div>
```

```
<div class="w3_form_body_grid">

<span>occupation</span>

<input type="text" name="occupation">

</div>

<div class="w3_form_body_grid">

<span>education</span>

<select id="education" onChange="change_country(this.value)" name="education"
class="frm-field required">

<option value="phd">phd</option>

<option value="pg">pg</option>

<option value="ug">ug</option>

<option value="+2">+2</option>

<option value="10">10</option>

<option value="below 10th">below 10th</option>

<option value="no education">No education</option>

</select>

</div>

<div class="w3_form_body_grid">

<span>Email</span>

<input type="email" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,4}$" name="email"
placeholder="Email" required="">

</div>

<div class="w3_form_body_grid">
```

```
<span>Phone_no</span>

<input type="text" pattern="^\d{10}$" name="ph_no" required="">

</div>

<div class="w3_form_body_grid w3_form_body_grid1">

<span>Password</span>

<input type="password" pattern="^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?!.*\s).*$"
name="password" placeholder="Password" required="" title="caps,small,number">

</div>

<div class="w3_form_body_grid w3_form_body_grid1">

<span>Photo</span>

<input type="file" name="photo" placeholder="photo" required="">

</div>

<input type="submit" value="Sign Up">

</form>

<!--/footer -->

<!-- js -->

<script type='text/javascript' src='js/jquery-2.2.3.min.js'></script>

<script src="js/SmoothScroll.min.js"></script>

<!-- smooth scrolling -->

<script type="text/javascript">
```

```
$(document).ready(function() {

/*

var defaults = {

containerID: 'toTop', // fading element id

containerHoverID: 'toTopHover', // fading element hover id

scrollSpeed: 1200,

easingType: 'linear'

};

*/

$.UItoTop({ easingType: 'easeOutQuart' });

});

</script>

<a href="#home" class="scroll" id="toTop" style="display: block;"> <span
id="toTopHover" style="opacity: 1;"> </span></a>

<!-- //smooth scrolling -->

<!-- //for-Clients -->

<!-- start-smoth-scrolling -->

<script type="text/javascript" src="js/move-top.js"></script>

<script type="text/javascript" src="js/easing.js"></script>

<script type="text/javascript">

jQuery(document).ready(function($) {

$(".scroll").click(function(event){
```

```
event.preventDefault();

$('html,body').animate({ scrollTop:$(this.hash).offset().top },1000);

});

});

</script>

<!-- start-smoth-scrolling -->

<!-- Calendar -->

<script src="js/jquery-ui.js"></script>

<script>

$(function() {

$( "#datepicker" ).datepicker();

});

</script>

<!-- //Calendar -->

<script type="text/javascript" src="js/bootstrap-3.1.1.min.js"></script>

</body>

</html>
```

Login.php

```
<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Elite Match a Matrimonial Category Bootstrap Responsive Website Template | Login
:: W3layouts</title>
```



```
<!-- for-mobile-apps -->
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<meta name="keywords" content="Elite Match Responsive web template, Bootstrap Web  
Templates, Flat Web Templates, Android Compatible web template,
```

```
Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, Sony  
Ericsson, Motorola web design" />
```

```
<script type="application/x-javascript"> addEventListener("load", function() {  
setTimeout(hideURLbar, 0); }, false);
```

```
function hideURLbar(){ window.scrollTo(0,1); } </script>
```

```
<!-- //for-mobile-apps -->
```

```
<link href="css/bootstrap.css" rel="stylesheet" type="text/css" media="all" />
```

```
<link href="css/font-awesome.css" rel="stylesheet">
```

```
<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
```

```
<!--fonts-->
```

```
<link href="//fonts.googleapis.com/css?family=Raleway:400,500,600,700,800,900"  
rel="stylesheet">
```

```
<link href="//fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700"  
rel="stylesheet">
```

```
<link href="//fonts.googleapis.com/css?family=Dancing+Script:400,700" rel="stylesheet">
```

```
<!--//fonts-->
```

```
</head>
```

```
<body>
```

```
<!--banner start here-->
```

```
<div class="banner-w3ls inner-banner-agileits" id="home">
```

```
<div class="container">

<!-- header -->

<div class="header-inner">

<h1 class="logo">

<a href="index.php">M4Match</a></h1>

<nav class="navbar navbar-default">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#bs-
example-navbar-collapse-1">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

</div>

<!-- navbar-header -->

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav navbar-right">

<li><a href="index.php">Home</a></li>

<li><a href="contact.php">Contact</a></li>

</ul>

</div>

<div class="clearfix"> </div>
```

```
</nav>

<div class="header-right-w3ls">

<a href="login.php">Log In</a>

<a href="register.php">Register</a>

</div>

<div class="clearfix"></div>

</div>

</div>

</div>

<!--//home-->

<!--banner end here-->

<!-- breadcrumbs -->

<div class="w3l_agileits_breadcrumbs">

<div class="container">

<div class="w3l_agileits_breadcrumbs_inner">

<ul>

<li><a href="index.php">Home</a><span>&#187;</span></li>

<li>Login</li>

</ul>

</div>

</div>

</div>
```

```
<!-- //breadcrumbs -->

<div class="login elite-app">

<div class="container">

<div class="tittle-agileinfo">

<h3>Login Now</h3>

</div>

<div class="col-md-8 login-form-w3-agile">

<form action="loginaction.php" method="post" name="login">

<div class="w3_form_body_grid">

<span>Name</span>

<input type="email" name="email" placeholder="email" required="">

</div>

<div class="w3_form_body_grid">

<span>Password</span>

<input type="password" name="password" placeholder="password" required="">

</div>

<div class="agile_remember">

<div class="agile_remember_left">

</div>

<div class="clearfix"> </div>

</div>

<input type="submit" value="Sign In">
```

```
</form>

</div>

</div>

</div>

<!--login-inner-->

<!-- footer -->

<!--/footer -->

<!-- js -->

<script type='text/javascript' src='js/jquery-2.2.3.min.js'></script>

<script src="js/SmoothScroll.min.js"></script>

<!-- smooth scrolling -->

<script type="text/javascript">

$(document).ready(function() {

/*

var defaults = {

containerID: 'toTop', // fading element id

containerHoverID: 'toTopHover', // fading element hover id

scrollSpeed: 1200,

easingType: 'linear'

};

*/

$.UItoTop({ easingType: 'easeOutQuart' });
```

```
});

</script>

<a href="#home" class="scroll" id="toTop" style="display: block;"> <span
id="toTopHover" style="opacity: 1;"> </span></a>

<!-- //smooth scrolling -->

<!-- //for-Clients -->

<!-- start-smoth-scrolling -->

<script type="text/javascript" src="js/move-top.js"></script>

<script type="text/javascript" src="js/easing.js"></script>

<script type="text/javascript">

jQuery(document).ready(function($) {

$(".scroll").click(function(event){

event.preventDefault();

$('html,body').animate({ scrollTop:$(this.hash).offset().top },1000);

});

});

</script>

<!-- start-smoth-scrolling -->

<script type="text/javascript" src="js/bootstrap-3.1.1.min.js"></script>

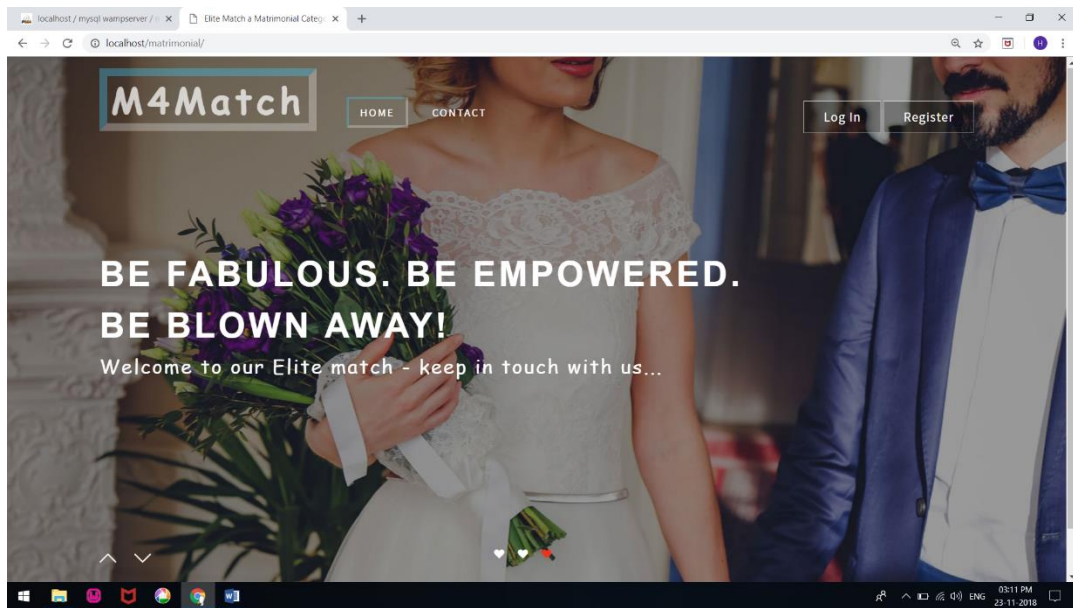
</body>

</html>
```

SCREEN SHOTS

USER

Home



Login

localhost / mysql wampserver / x Elite Match a Matrimonial Categori x +

localhost/matrimonial/login.php

M4Match HOME CONTACT Log In Register

Home » Login

Login Now

NAME email

PASSWORD password

SIGN IN

03:11 PM 23-11-2018

Registration

localhost / mysql wampserver / x Elite Match a Matrimonial Categori x +

localhost/matrimonial/register.php

M4Match HOME CONTACT Log In Register

Home » Register

Register Now

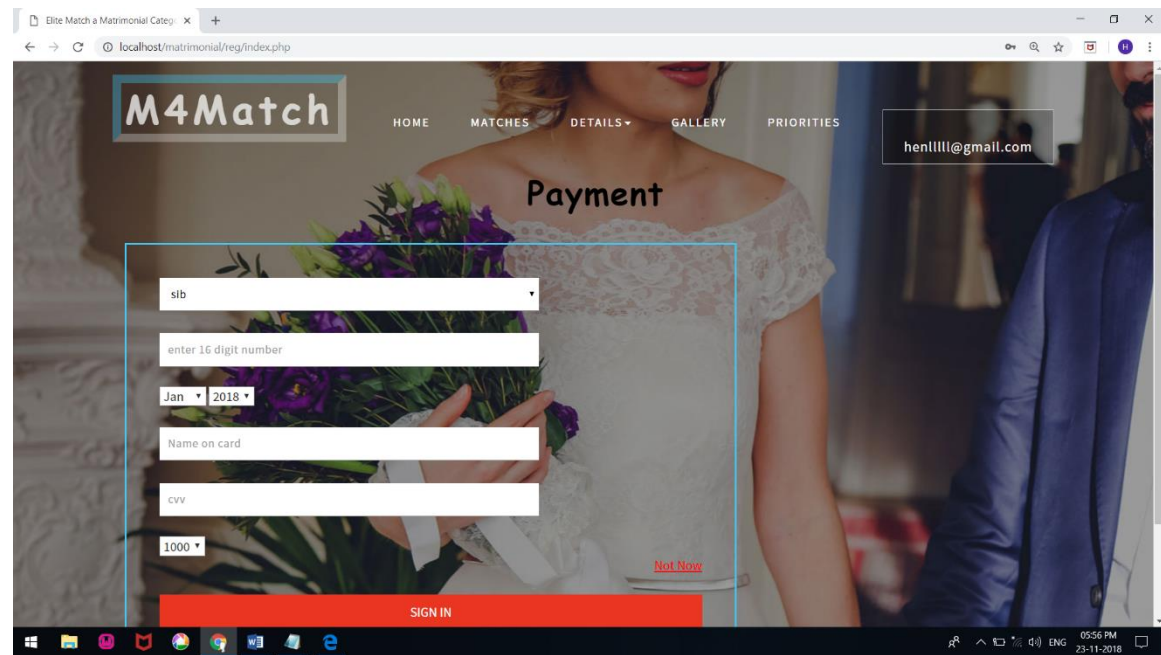
PROFILE FOR Select

NAME Name

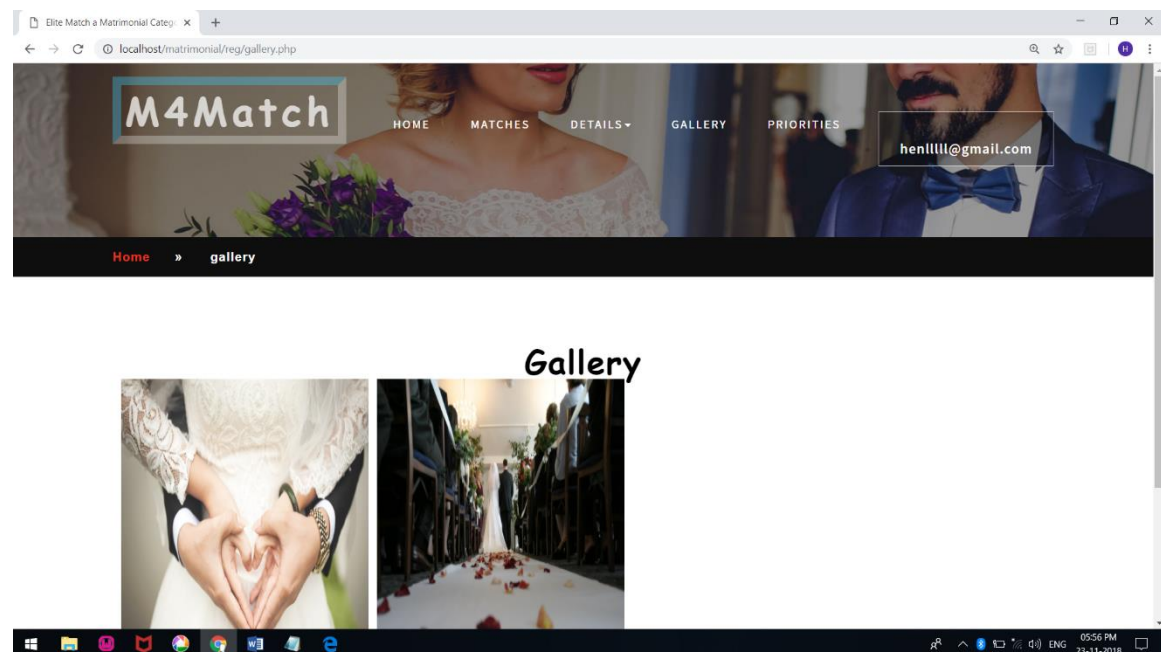
GENDER ☐ Male ☐ Female

03:12 PM 23-11-2018

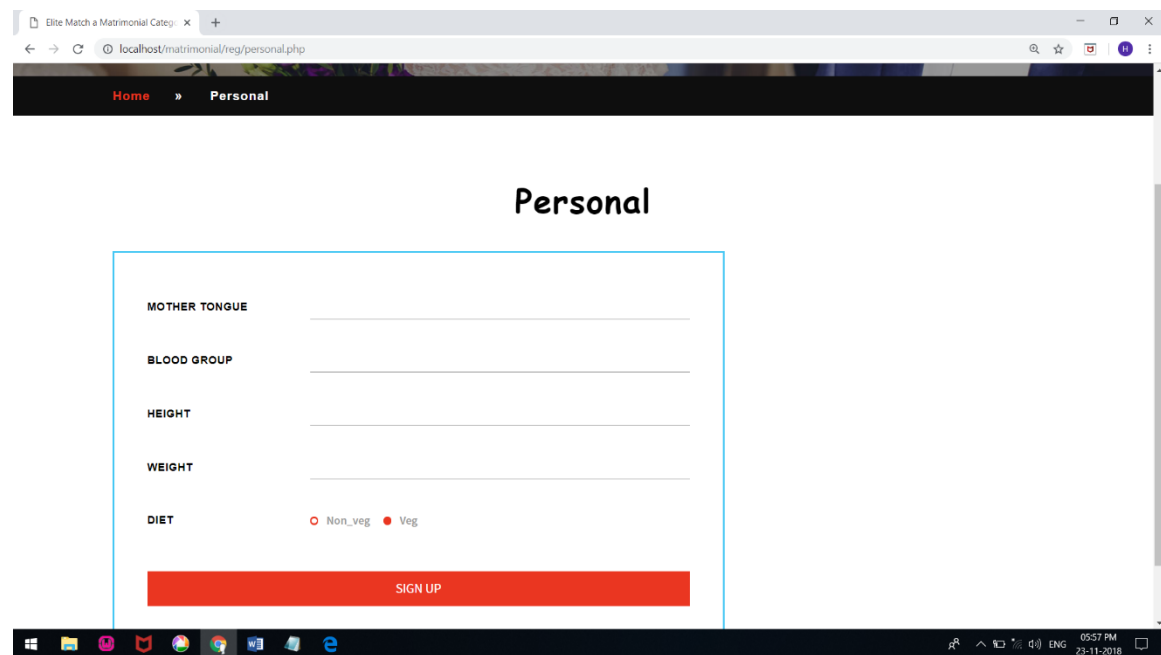
Payment



Gallery



Details adding



Elite Match a Matrimonial Category: X

localhost/matrimonial/reg/personal.php

Home » Personal

Personal

MOTHER TONGUE _____

BLOOD GROUP _____

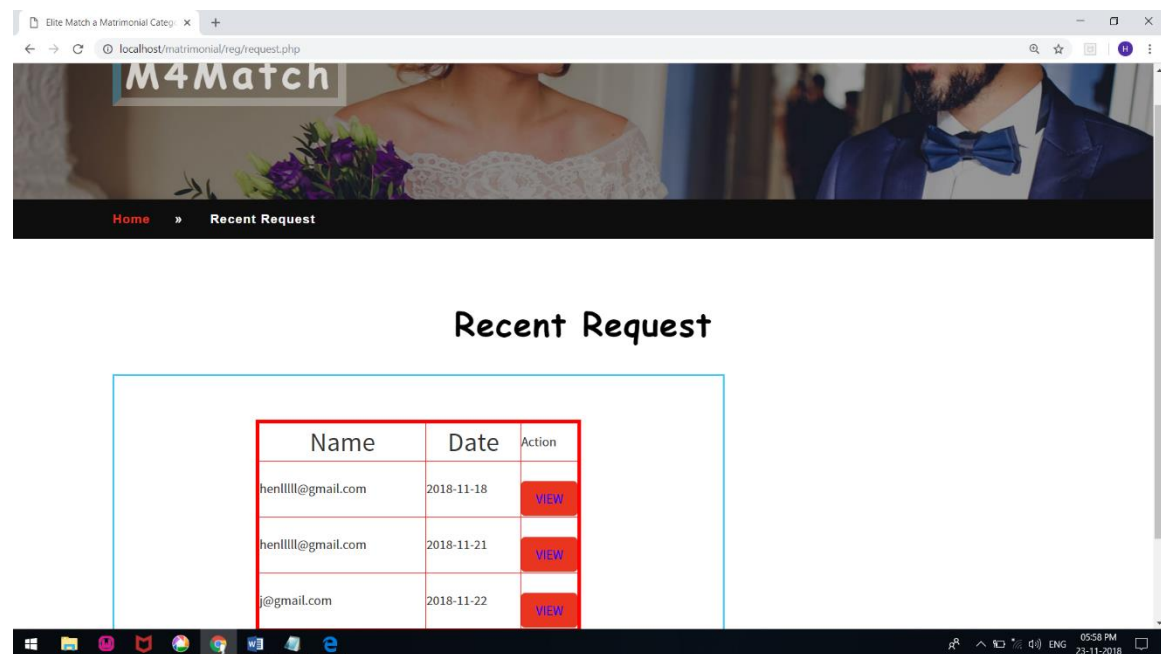
HEIGHT _____

WEIGHT _____

DIET ☐ Non_veg ☐ Veg

SIGN UP

Request



Elite Match a Matrimonial Category: X

localhost/matrimonial/reg/request.php

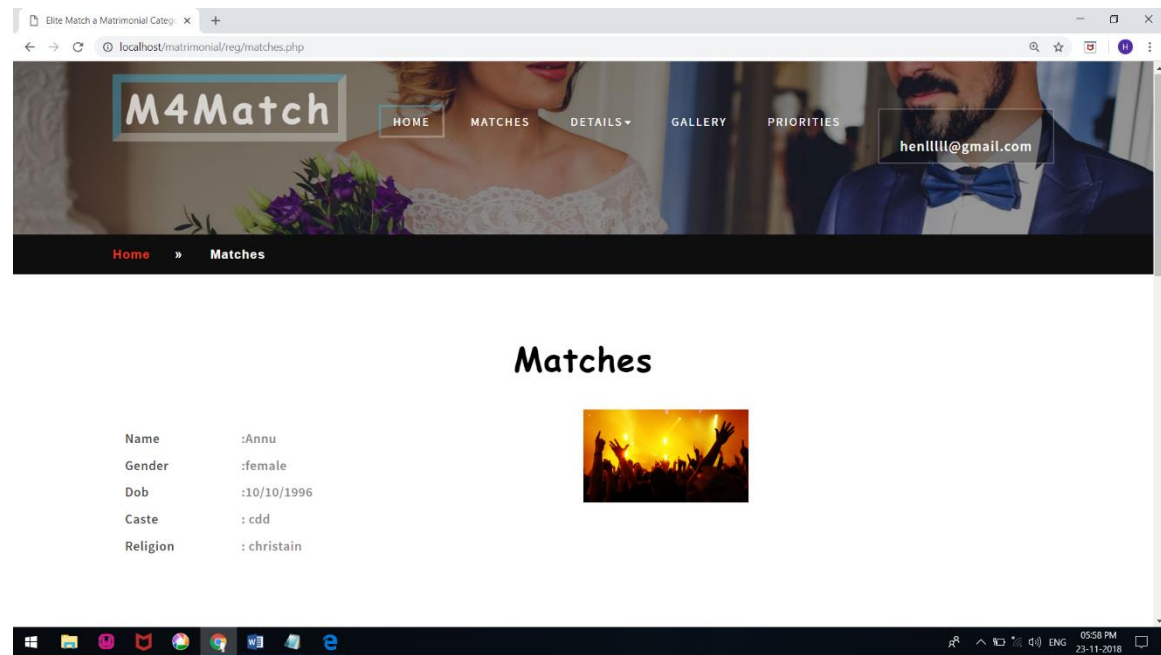
M4Match

Home » Recent Request

Recent Request

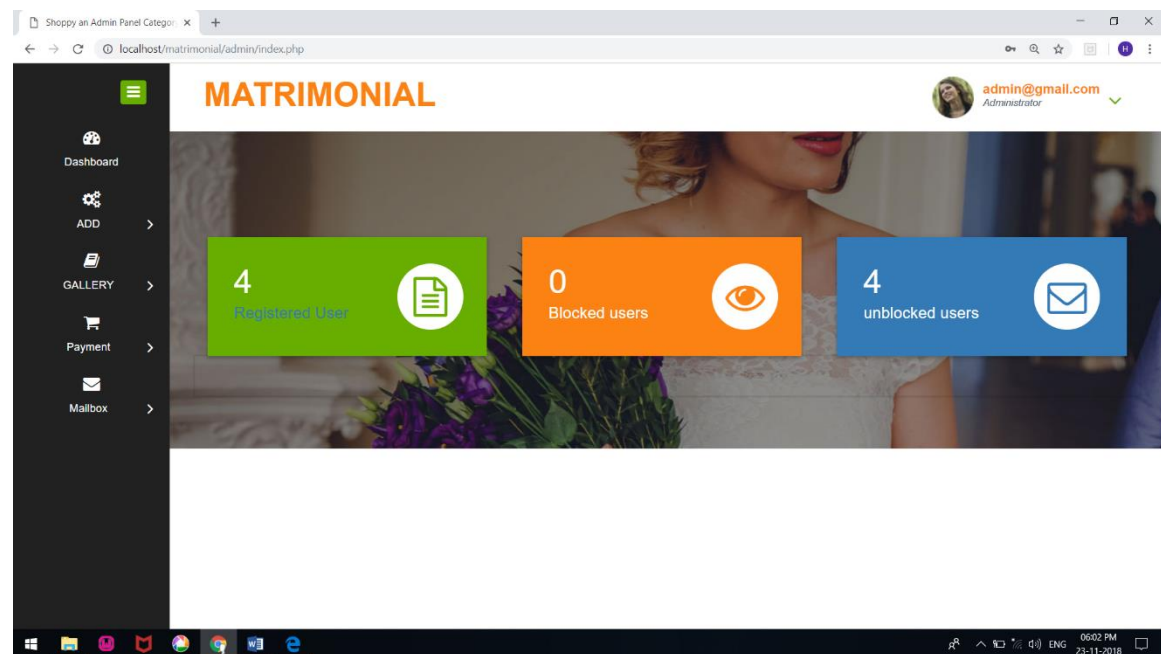
Name	Date	Action
henlllll@gmail.com	2018-11-18	VIEW
henlllll@gmail.com	2018-11-21	VIEW
jj@gmail.com	2018-11-22	VIEW

Matches



ADMIN

Home



Profile

NAME HENEL PJ

GENDER Male

DATE OF BIRTH 01 APRIL 1996

RELIGION CHRISTIAN

ADDRESS PALAYOOR(H), PO VENKITANGU, THRISSUR ,KERALA-680510

OCCUPATION STILL STUDY

EDUCATION MASTER OF COMPUTER APPLICATION

EMAIL henel@mca.ajce.in

Payment view

Name	Bank	Amount
sail	sib	1000
Henel	sib	1000
Henel	sib	1000
Henel	sib	1000

Add new cast/religion

