# CHAPTER 1

# INTRODUCTION

## 1.1 PROJECT OVERVIEW

**Maternal Health Activist System** allows to store ashaworker’s and user’s details in the desired ward, details of new born in that ward, all the details from maternity to child birth and further health details of child until an infant . This software package allows public health department to store all the details related to the child birth and take care of materns whenever necessary. This project will also be able to provide reports of child birth, mortal child birth, maternity death, gender ratio, details about pregnancy in the desired ward, health issues of infants and mothers.

## 1.2 PROJECT SPECIFICATION

The Maternal Health Activists System is user-friendly software. The main objective of the system is to create an open relation between the ashaworkers and mothers.
The system includes 5 modules. They are:

### 1. The administrator module

The admin here indicates one of the employee of MoHFW who has control over the entire activity of ashaworkers.
The duties are:

- Add
    - state
    - district under each state
    - Panchayaths under each district
    - wards
- View registered asha workers
- Approve asha workers
- Delete asha workers
- View monthly report send by ashaworkers
- Send feedbacks on report

### 2. The Asha worker module

The main working people in this project are these people. And the functionalities of this module are:

- View registered users under their ward
- Approve users
- Send feedbacks for users after checking their details
- Collect and keep details of every newborn
- Send reports for the admin collected each month respectively
- View notifications/feedbacks sent by the admin
- Store immunization details of child

3. **The User Module**

   Here the users refer to the pregnant ladies. They need to fill up the required details asked by ashaworkers. They can also get notifications from their ashaworkers. Users can also view the immunization details of their child.

4. **The Registration module**

   The ashaworkers as well as users need to register officially, with prefed user name and password.

5. **The Login module**

   Admin, asha worker and user can login to their on account to perform certain functions. Only the approved asha workers and users are allowed to login.

# CHAPTER 2

# SYSTEM STUDY

## 2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems andtheinformationtorecommendimprovementsonthesystem.Itisaproblemsolving activity that requires intensive communication between the system users and system developers.Systemanalysisorstudyisanimportantphaseofanysystemdevelopment 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 o 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, analyzing 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 systemiscalledtheexistingsystem.Nowtheexistingsystemissubjectedtoclosestudy 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 EXISTINGSYSTEM

In present system all work is done on paper. We will not be generated reports on time and human efforts are very high in a paper based system.

## 2.3 DRAWBACKS OF EXISTINGSYSTEM

- Manual work.
- Consumes large volume of paper work.
- Time consuming.
- It needs experts.
- Accurately it needs to be computerized.

## 2.4 PROPOSEDSYSTEM

The Maternal Health Activists System is user-friendly software. The main objective of the system is to create an open relation between the ashaworkers and mothers. It reduces the paper works of asha workers. The system is very simple in its design and implementation. The system requires very low system resources and the system will work in almost all configurations. The main objective of the proposed system can be enumerated as follows:
The users can also take part in this Maternal Health Activists System online.

## 2.5 ADVANTAGES OF PROPOSEDSYSTEM

**Advantages of proposed system**

The system is very simple in its design and implementation. The system requires very low system resources and the system will work in almost all configurations.

- Security of data.
- Ensure the data.
- It is easy to use.
- Minimize manual data entry.

- Greater efficiency.
- User friendly and interactive.
- Minimum time required.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## 3.1 FEASIBILITYSTUDY

A feasibility analysis evaluates the candidate systems and determines the best system that meets performance requirements. The purpose of feasibility study is to investigate the present system, evaluate the possible application of computer based methods, select a tentative system, evaluate the cost and effectiveness of the proposed system, evaluate the impact of proposed system on existing personnel and ascertain the need for new personnel. Feasibility is carried out to see if the system is technically, economically and operationally feasible. All projects are feasible when given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time.

A feasibility study is not warranted for systems in which economic justification is obvious, technical risk is low, few legal problems are expected and no reasonable alternative exists. An estimate is made of whether the identified user may be satisfied using current hardware and software technologies. The study will decide if the proposed system will be cost effective from the business point of view and if it can be developed in the existing budgetary constraints. The result should inform the decision of whether to go ahead with a more detailed analysis.

### 3.1.1 Technical Feasibility

Technical feasibility is the most important of all types of feasibility analysis. Technical feasibility deals with hardware as well as software requirements. An idea from the outline design to system requirements in terms of input/output files and procedures is drawn and types of hardware and software and the methods required for running the system are analysed.

The Maternal health Activist System is implemented in PHP with WAMP server as front end and Microsoft SQL server as back end with forms running under Windows XP (Vista, 7, 8, 10) environment. All the required software and hardware are available in market domain. The implementation does not require more hardware or software implementations. Existing computer software and hardware can itself be used for implementing this. Hence the system is technically feasible.

### 3.1.2 Economic Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of software, more commonly known as the cost/benefit analysis. The procedure is to determine the benefits and savings that are expected from a candidate and compare them with costs. If the benefits outweigh cost, the decision is made to design and implement the system; otherwise further alternatives have to be made. Here it is seen that no new hardware or software is needed for the development of the system.

Here the Maternal Health Activist System is feasible, as the organization possesses the hardware and software resources required for the functioning of the system. Any additional resources, if required, can also be easily acquired. This system has a lot of features at a minimum cost so it is feasible to implement and it will be very much beneficial to the programmers in the reduced cost.

### 3.1.3 Behavioral Feasibility

Behavioral Feasibility is connected with human organizational and political aspects. The issues considered are the job changes that will be brought about, the organizational structures that will be distributed and the new skills that will be required.

Maternal Health Activist System is user friendly. It is easier to implement in an organization. No need of more number of staffs to implement it in an organization. The employees in the organization will not have to alter the work patterns and work in an atmosphere of change. Thus, the application is operationally feasible.

## 3.2 SYSTEMSPECIFICATION

### 3.2.1 Hardware Specification

| | | |
|---|---|---|
| Processor | - | Intel core i3 |
| RAM | - | 4 GB |
| Hard disk | - | 1 TB |

### 3.2.2 Software Specification

| | | |
|---|---|---|
| Front End | - | HTML, CSS |
| Back end | - | MYSQL |
| Client on PC | - | Windows 7 and above. |
| Technologies used | - | JS, HTML5, J Query, PHP, CSS |

## 3.3 SOFTWAREDESCRIPTION

### 3.3.1 PHP

PHP is a server side scripting language designed for web development but alsoused as a general purpose programming language.PHP is now installed on more than 244 million websites and2. 1million webs ervers. Originally created by Rasmus Ledorf in1995, 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 Website provides the latest information about MySQLsoftware.

- **MySQL is a database managementsystem.**

  A 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, youneed 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 otherapplications.

- **MySQL databases arerelational.**

  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 setup rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between differentables. The database enforces these rules, so that with a well-designed database, your application never sees in consistent, duplicate, orphan, out-of-date, ormissing 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-specificAPIthathidestheSQLsyntax.SQLisdefinedbytheANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist.Inthismanual,“SQL92”refers to the standard released in1992,

  “SQL: 1999” refers to the standard released in1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at anytime.

- **MySQL software is OpenSource.**

  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 touse.**

  If that is what you are looking for, you should give itatry. 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 Server works in client/server or embeddedsystems.**

  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 in to 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 processor 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 PhysicalDesign.

## 4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**.UML is different from the other common programming languages such as C++, Java, COBOL, etc.UML is a pictorial language used to make software blueprints.UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system.Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc.UML is not a programming language but tools can be used to generate code in various languages using UML diagrams.

UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete.UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

### 4.2.1 USE CASEDIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. A usecase diagram contains four components.

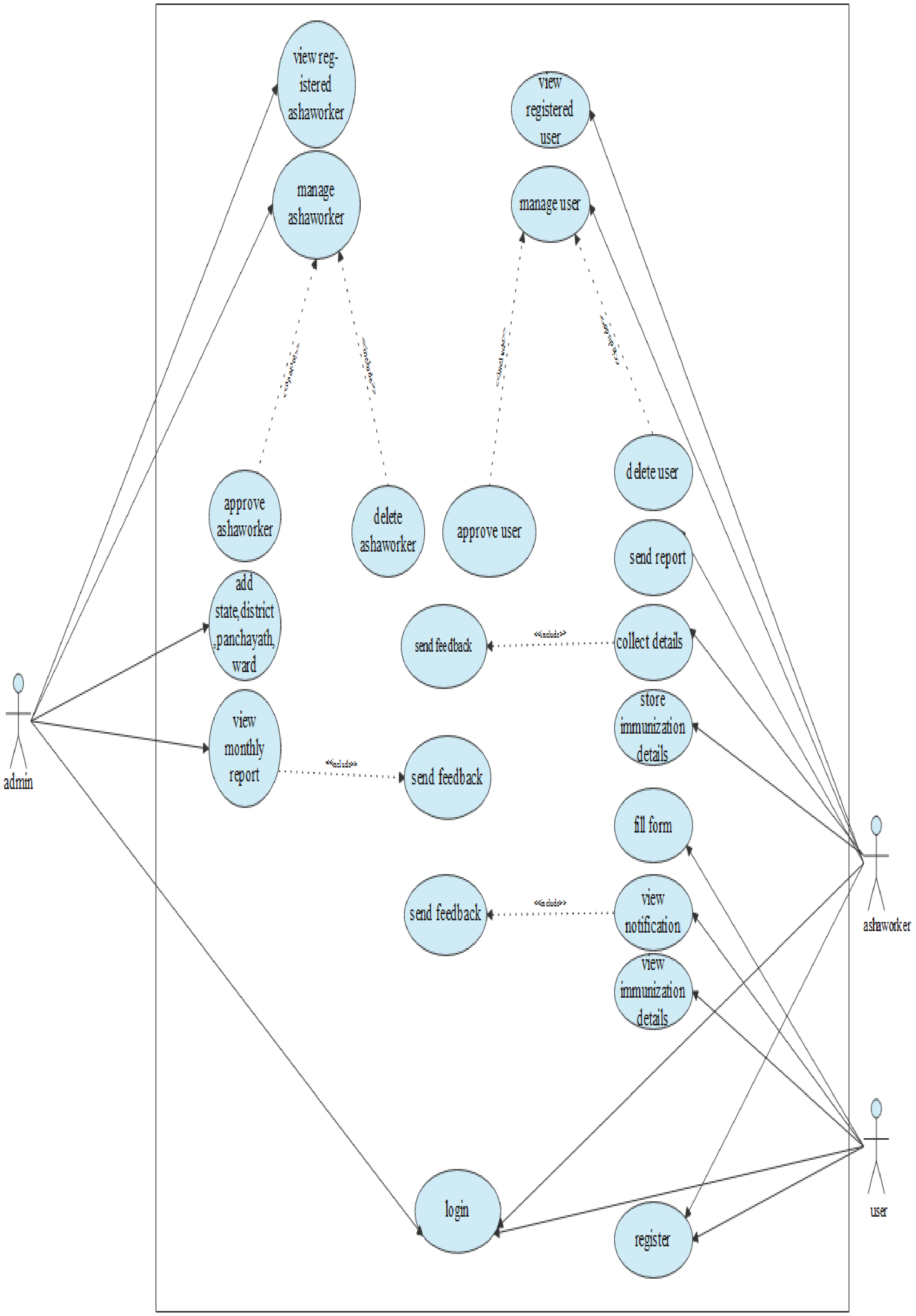
- The boundary, which defines the system of interest in relation to the world aroundit.

- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the usecases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

## Usecase diagram



## 4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

**Sequence Diagram Notations –**

i. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram

iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

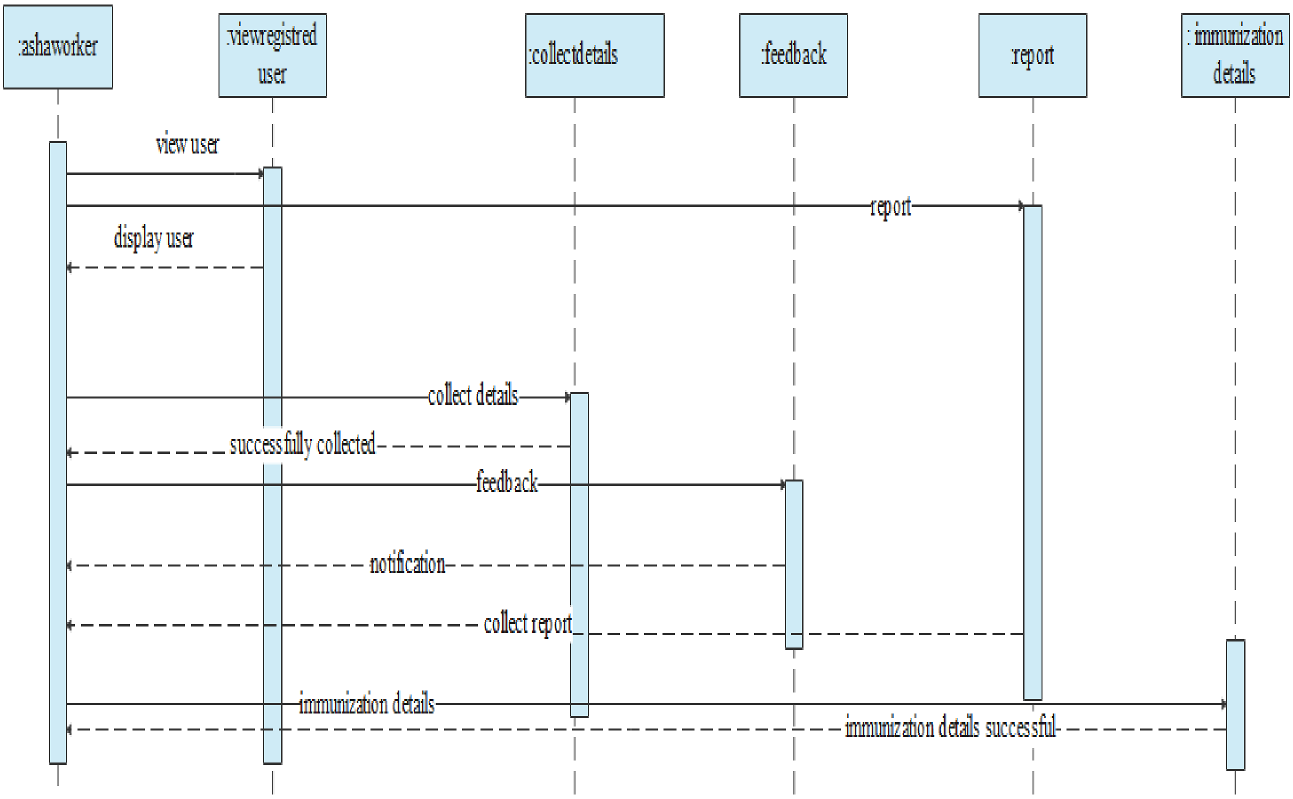
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message

- Reply Message
- Found Message
- Lost Message

**iv. Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

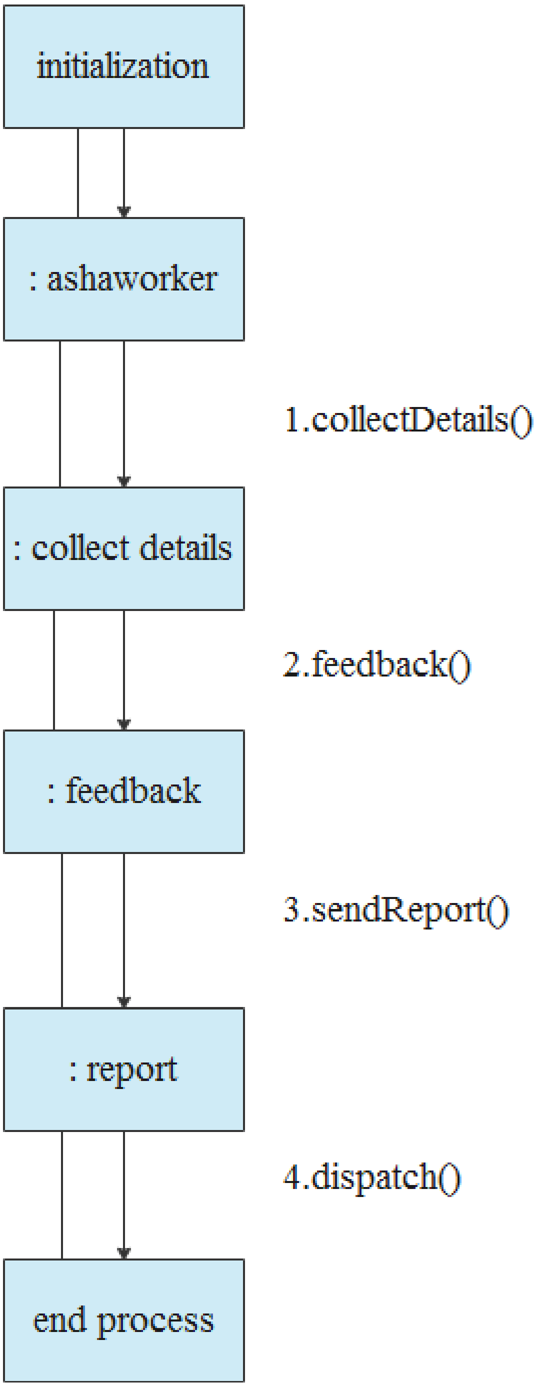
**Uses of sequence diagrams –**

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.



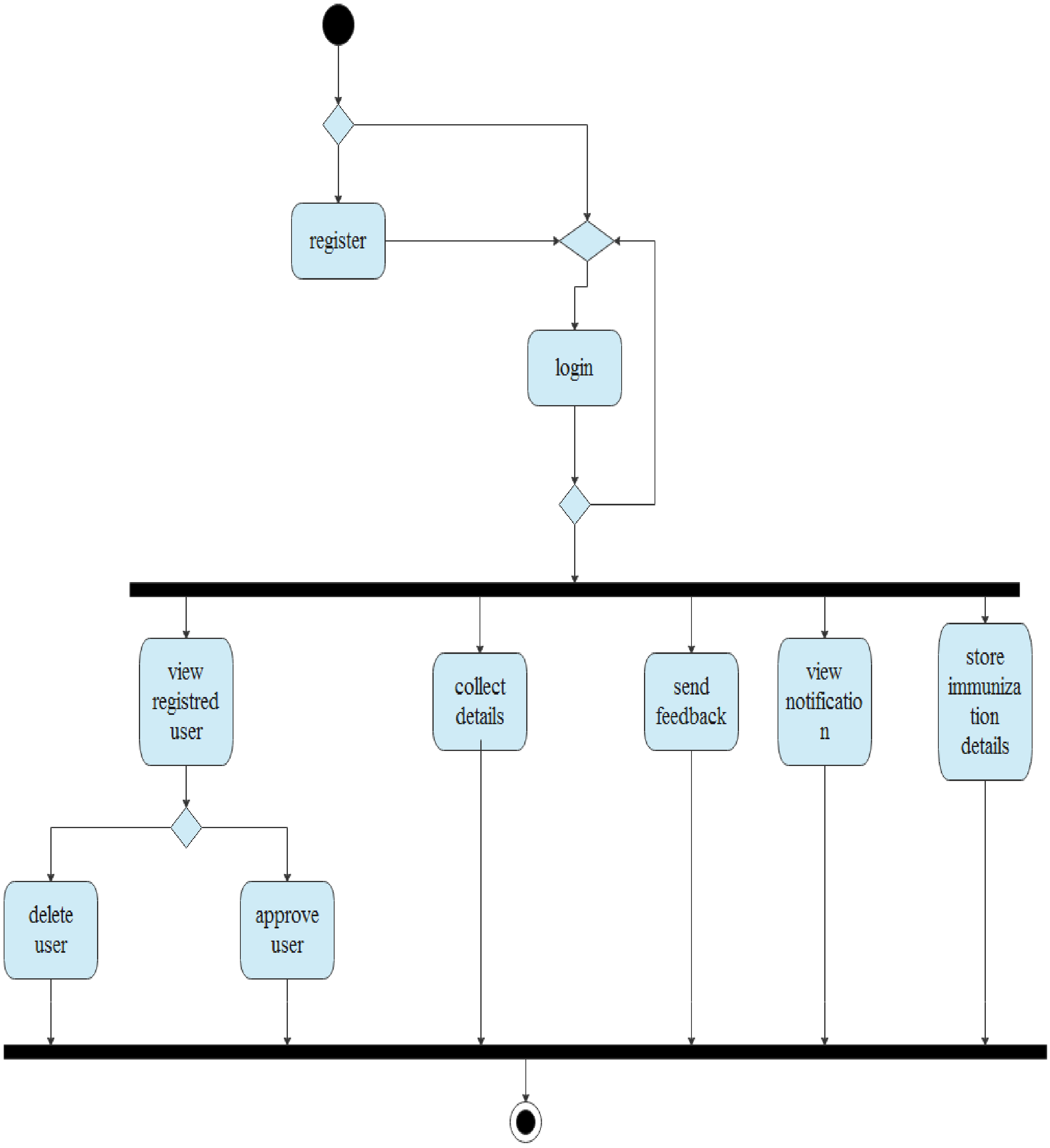
## 4.2.3 COLLABORATION DIAGRAM

In the collaboration diagram, the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another. Emphasizes on the structural organization of the objects that send and receive messages.



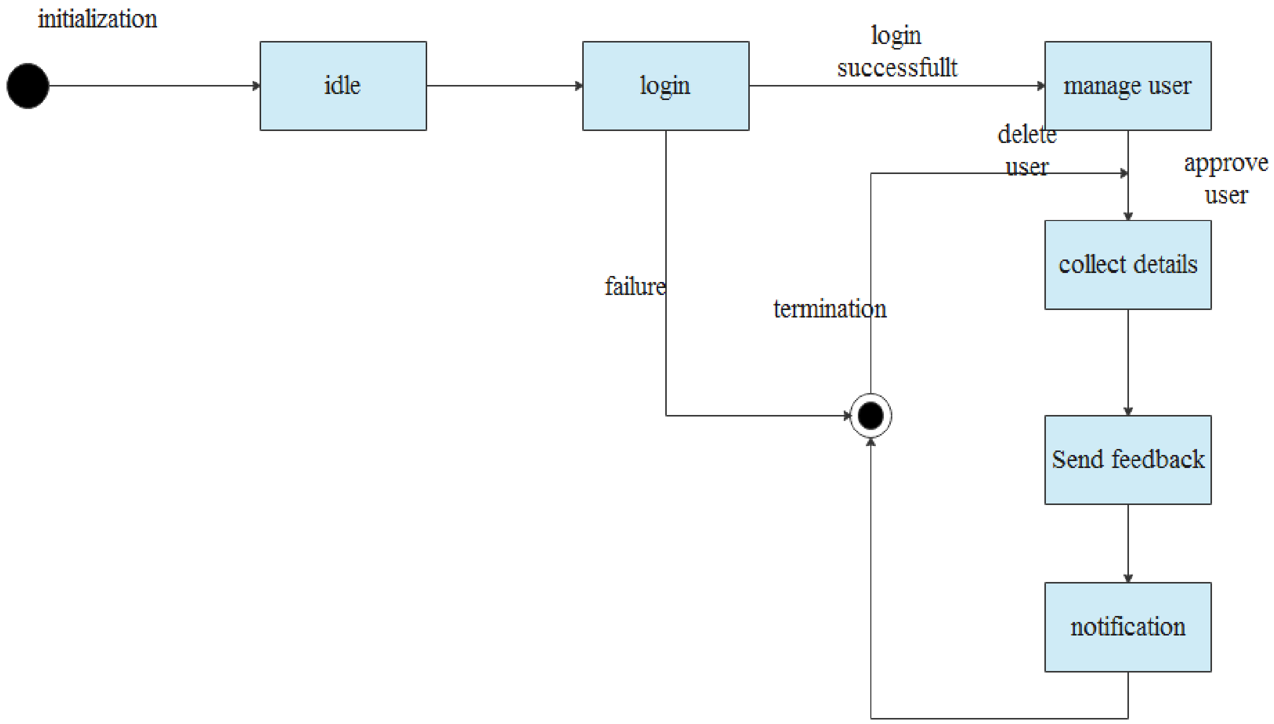
## 4.2.4 ACTIVITY DIAGRAM

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The control flow is drawn from one operation to another



## 4.2.5 STATE CHART DIAGRAM

A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

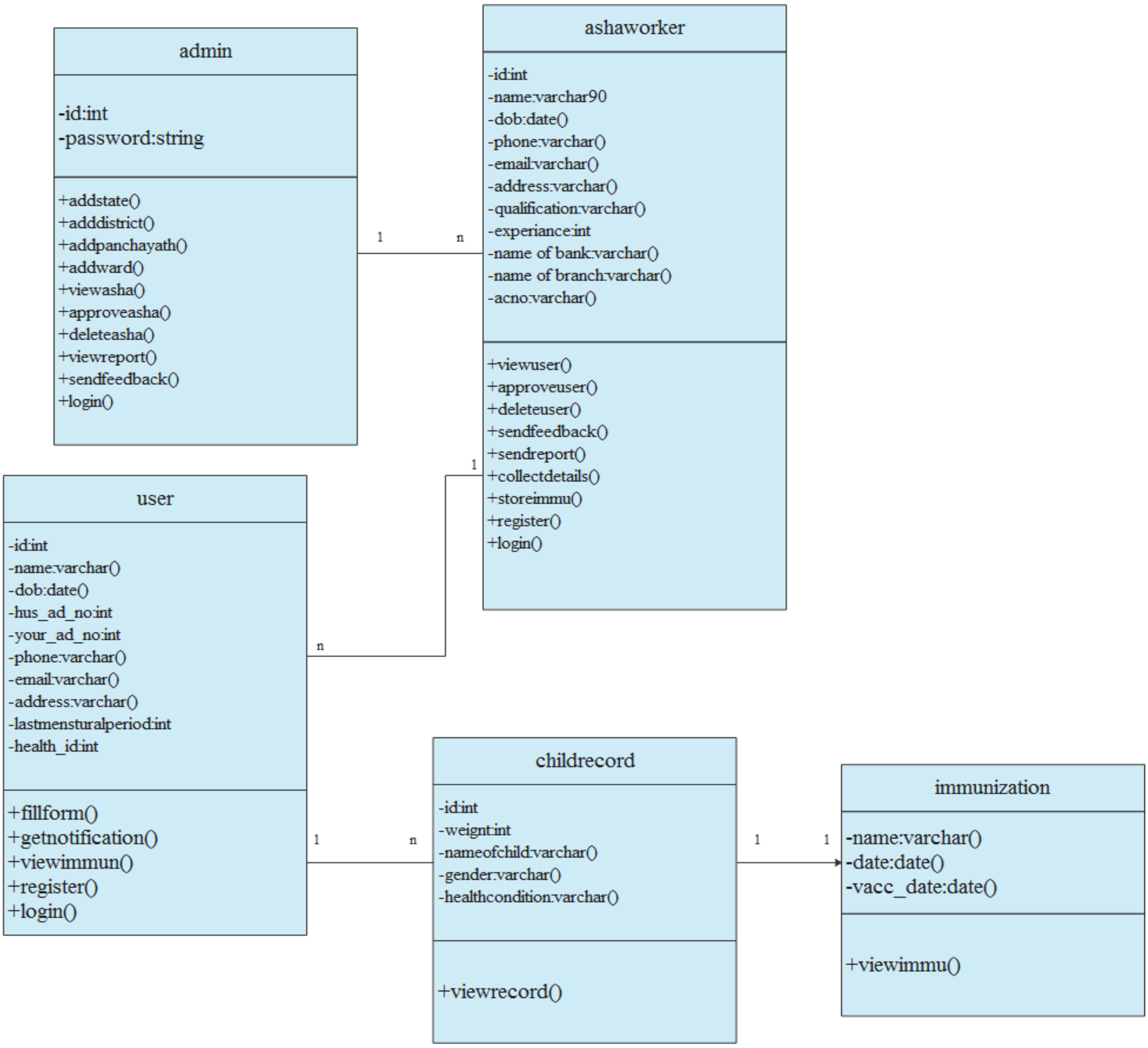


## 4.2.6 CLASS DIAGRAM

UML class diagram gives an overview of a software system by displaying classes, attributes, operations, and their relationships. This Diagram includes the class name, attributes, and operation in separate designated compartments.Class Diagram defines the types of objects in the system and the different types of relationships that exist among them.

### Benefits of Class Diagram

Class Diagram Illustrates data models for even very complex information systems. It provides an overview of how the application is structured before studying the actual code. This can easily reduce the maintenance time. Helpful for developers and other stakeholders.



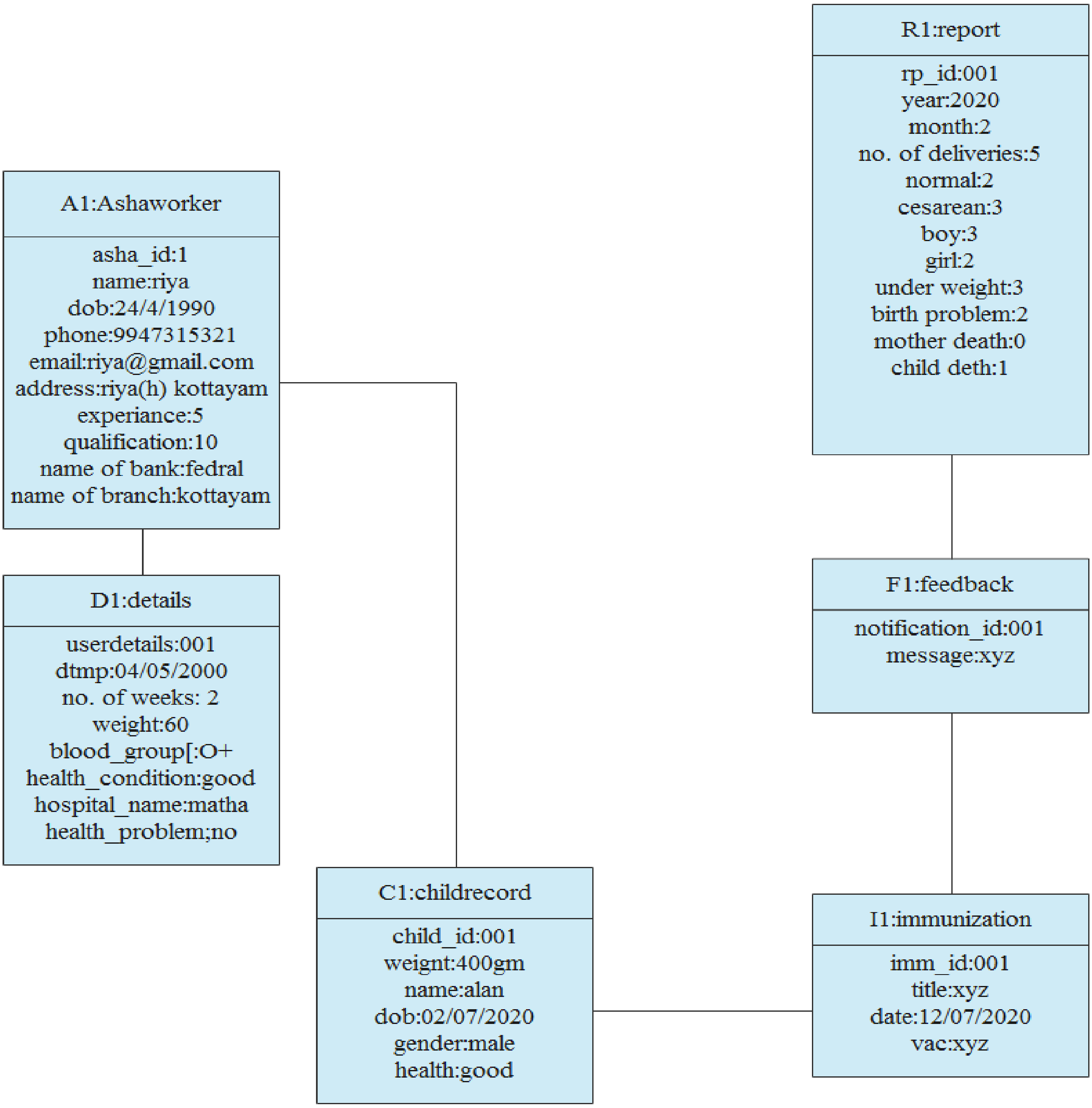
## 4.2.7 OBJECT DIAGRAM

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

### Purpose of Object Diagrams

A class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature.

It means the object diagram is closer to the actual system behaviour. The purpose is to capture the static view of a system at a particular moment.

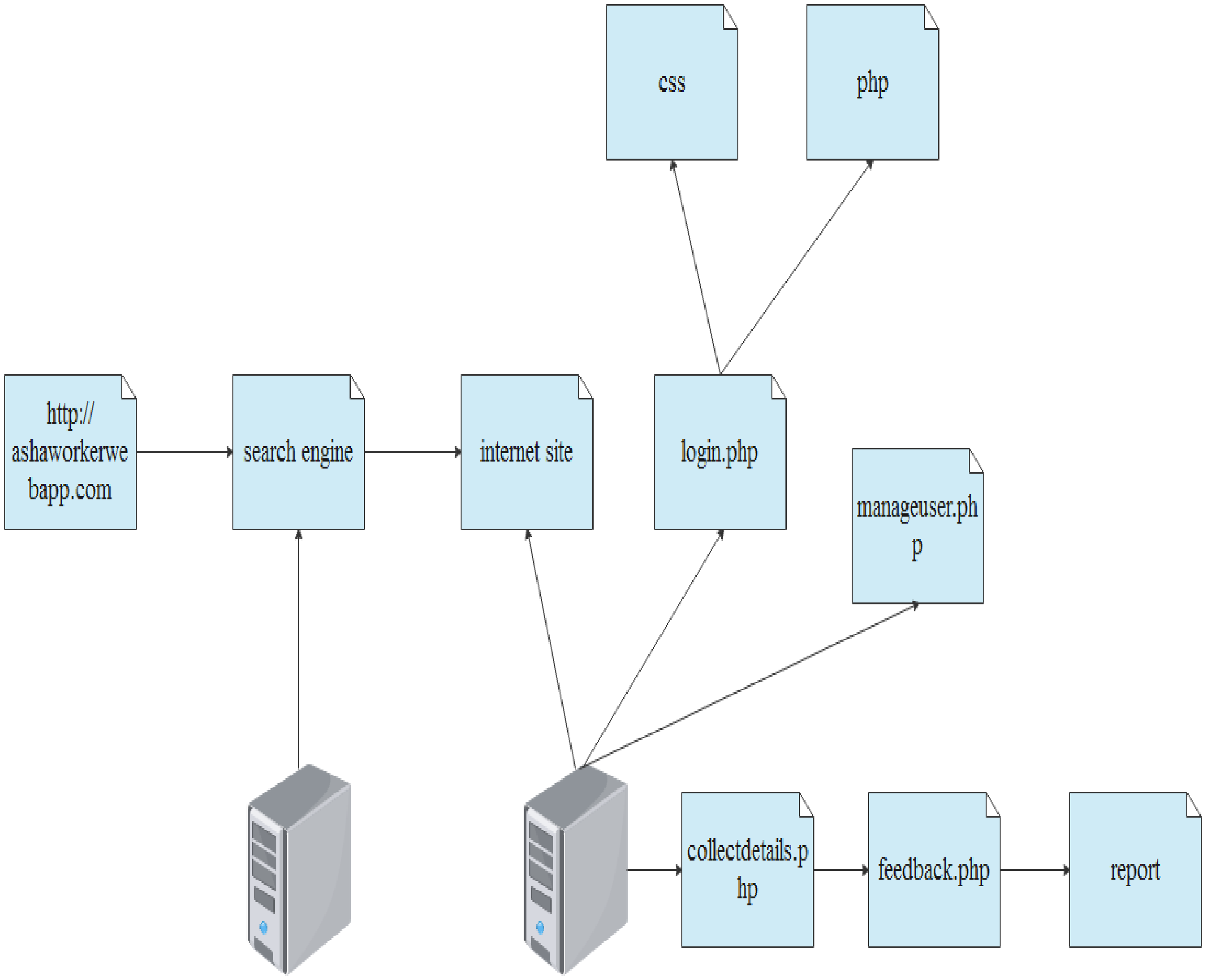


## 4.2.8 COMPONENT DIAGRAM

Component diagrams are used to model the physical aspects of a system. Component diagrams are used during the implementation phase of an application.

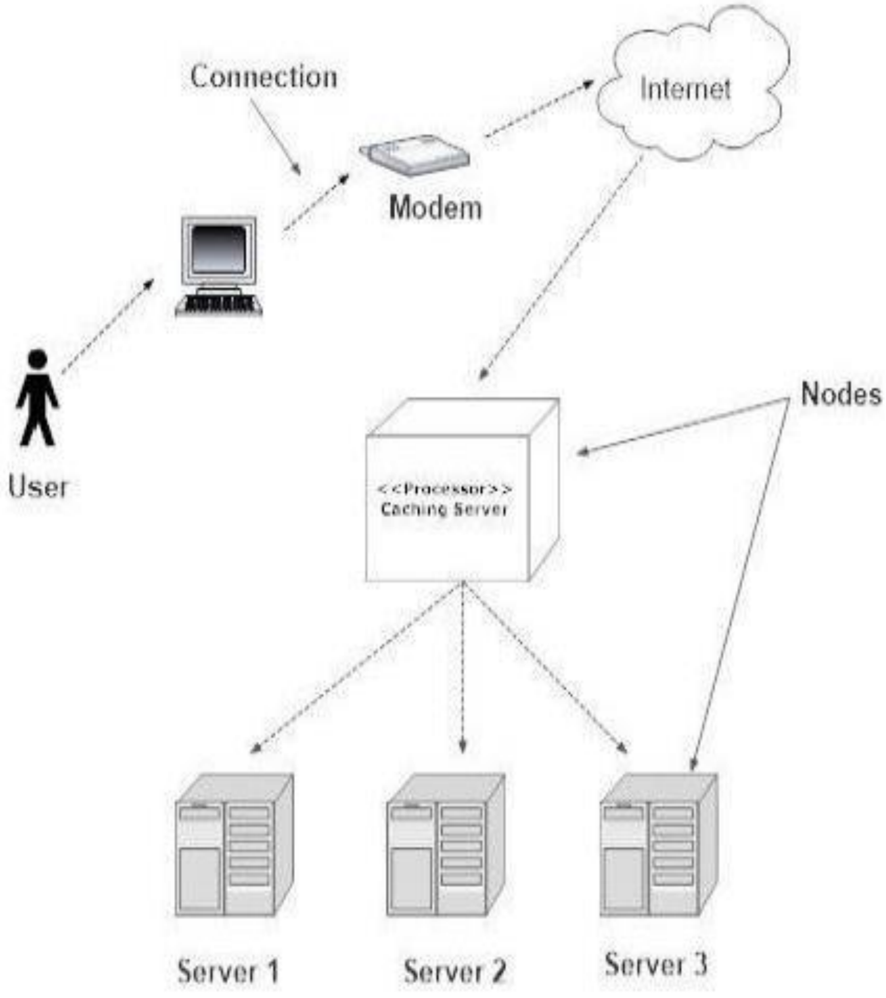
**Purpose of Component Diagrams**

It does not describe the functionality of the system but it describes the components used to make those functionalities.It Visualize the components of a system.



## 4.2.9 DEPLOYMENT DIAGRAM

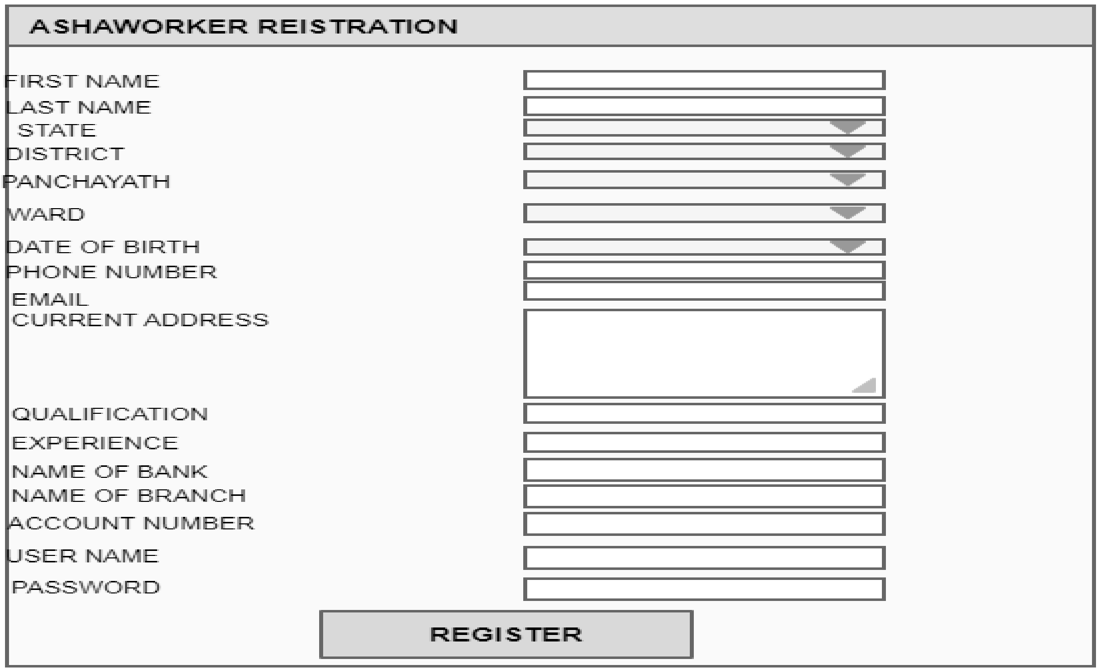
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.



## 4.3 USER INTERFACE DESIGN

### 4.3.1-INPUT DESIGN

Form Name : Ashaworker Registration

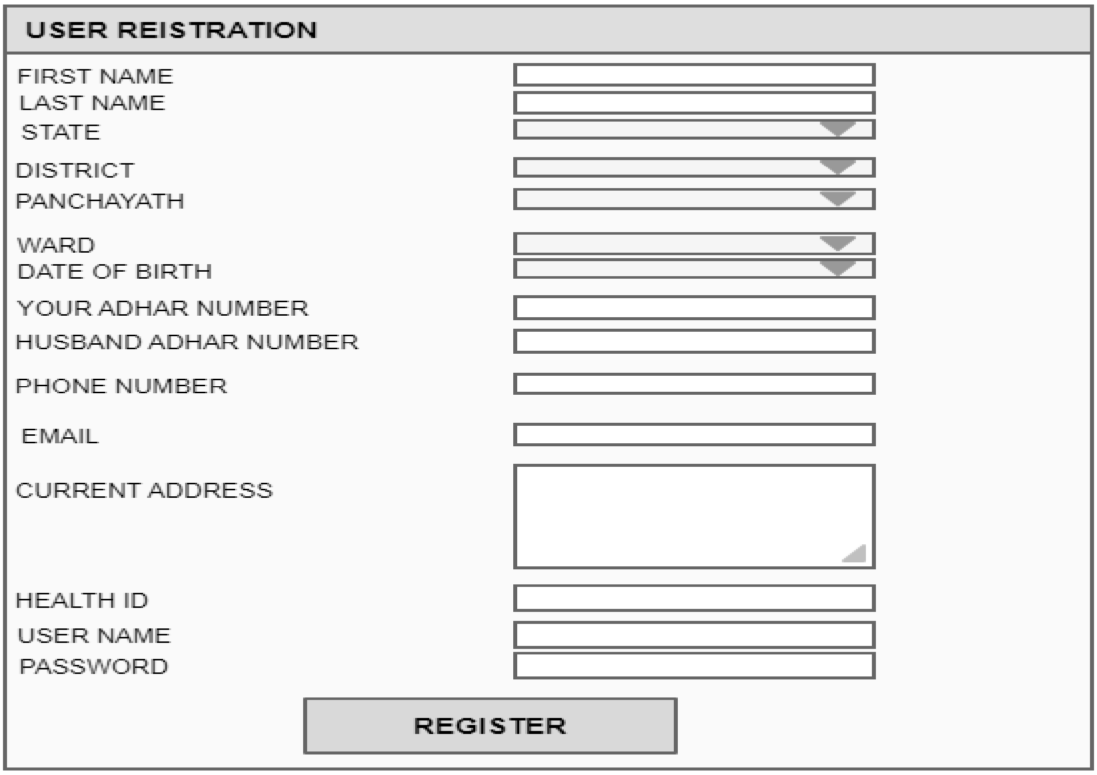


ASHAWORKER REISTRATION

FIRST NAME
LAST NAME
STATE
DISTRICT
PANCHAYATH
WARD
DATE OF BIRTH
PHONE NUMBER
EMAIL
CURRENT ADDRESS
QUALIFICATION
EXPERIENCE
NAME OF BANK
NAME OF BRANCH
ACCOUNT NUMBER
USER NAME
PASSWORD

REGISTER

Form Name : User Registration

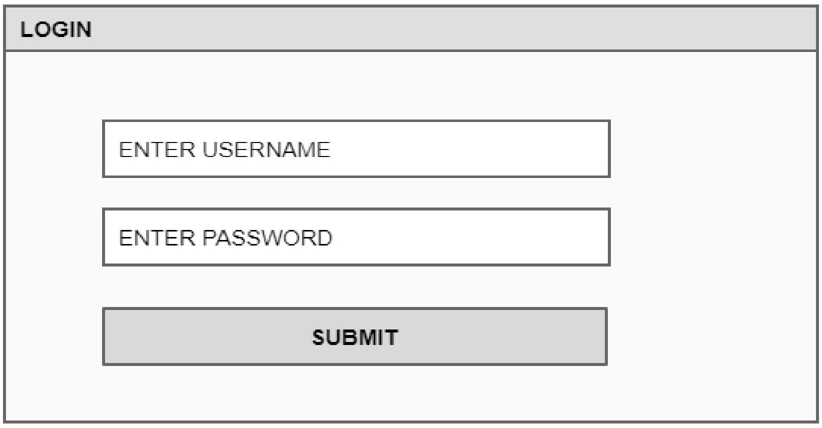


USER REISTRATION

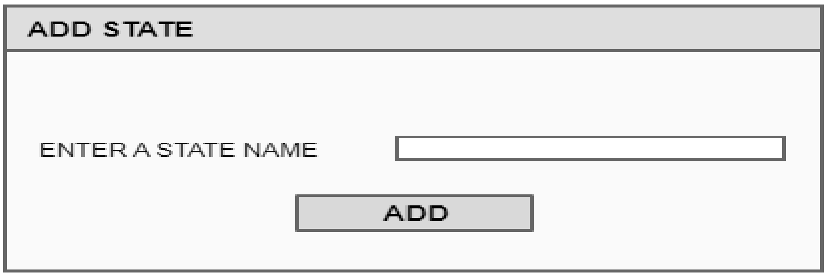
FIRST NAME
LAST NAME
STATE
DISTRICT
PANCHAYATH
WARD
DATE OF BIRTH
YOUR ADHAR NUMBER
HUSBAND ADHAR NUMBER
PHONE NUMBER
EMAIL
CURRENT ADDRESS
HEALTH ID
USER NAME
PASSWORD

REGISTER

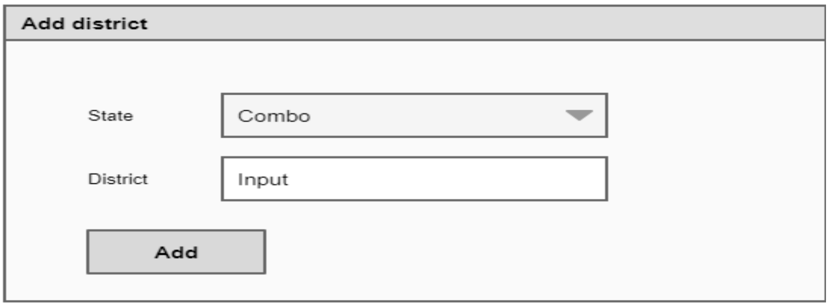
Form Name : Login



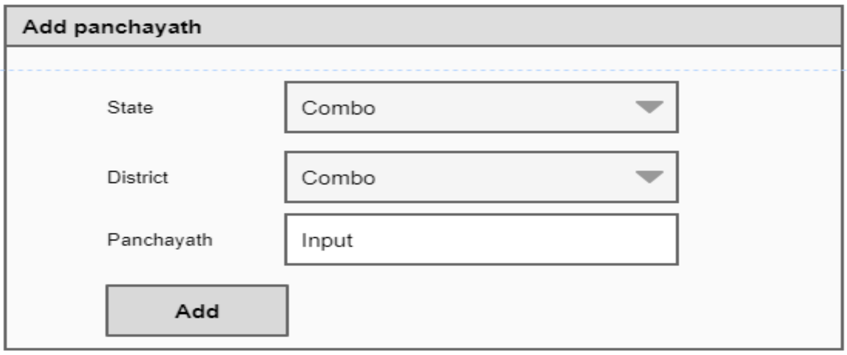
Form Name : Add state



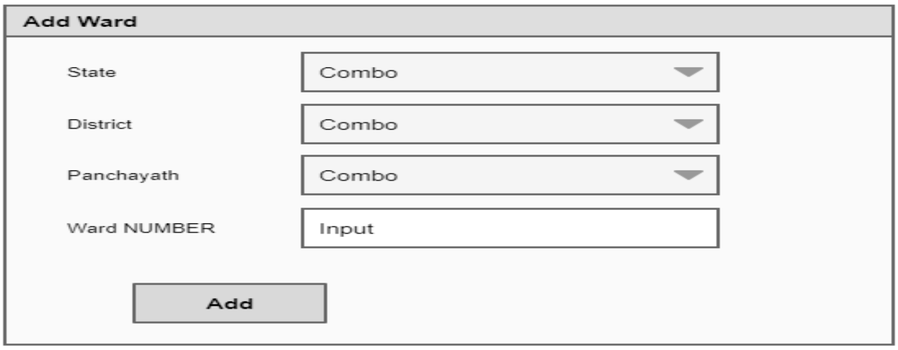
Form Name : Add district



Form Name : Add panchayath



Form Name : Add ward



Add Ward

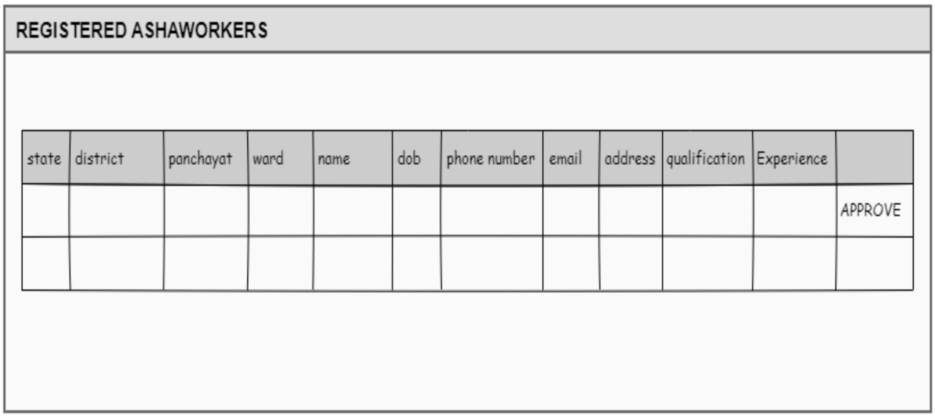
State Combo

District Combo

Panchayath Combo

Ward NUMBER Input

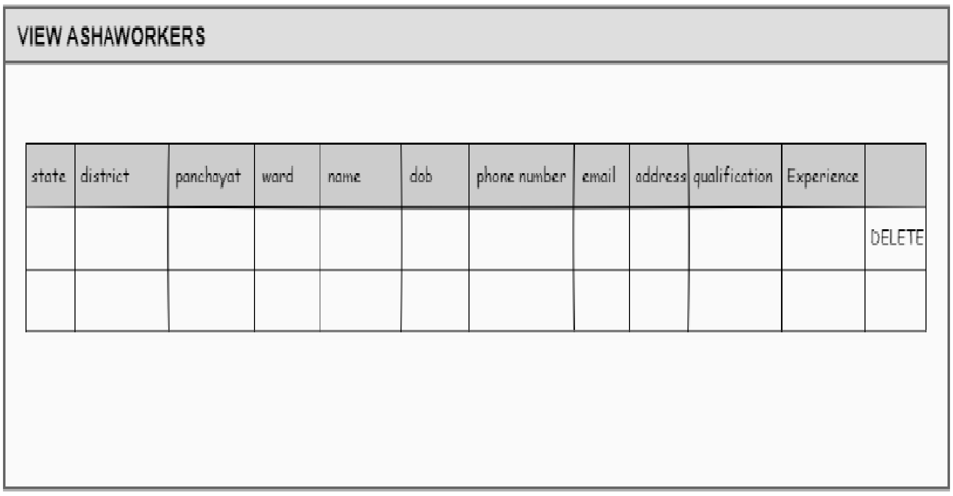
Form Name :Approve ashaworker



REGISTERED ASHAWORKERS

| state | district | panchayat | ward | name | dob | phone number | email | address | qualification | Experience | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | APPROVE |
| | | | | | | | | | | | |

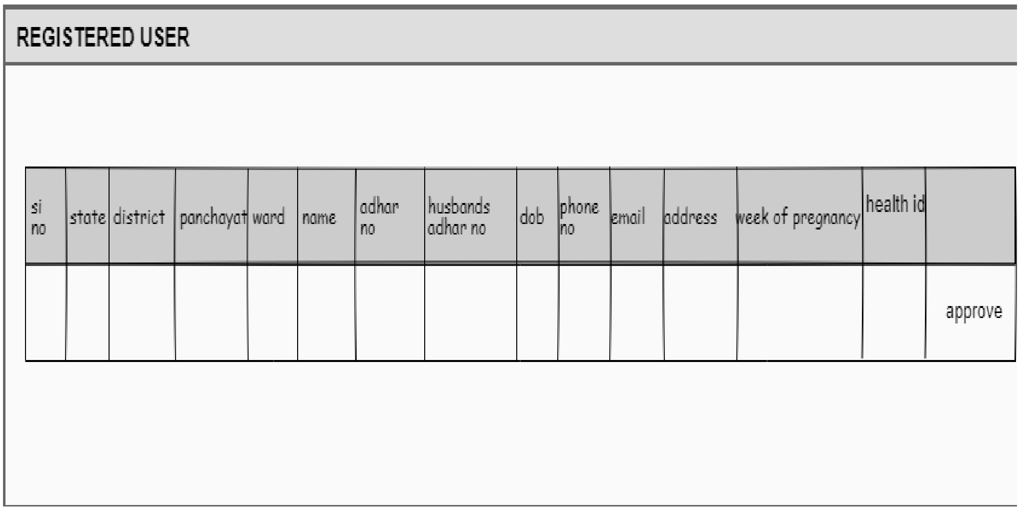
Form Name : view ashaworker



VIEW ASHAWORKERS

| state | district | panchayat | ward | name | dob | phone number | email | address | qualification | Experience | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | DELETE |
| | | | | | | | | | | | |

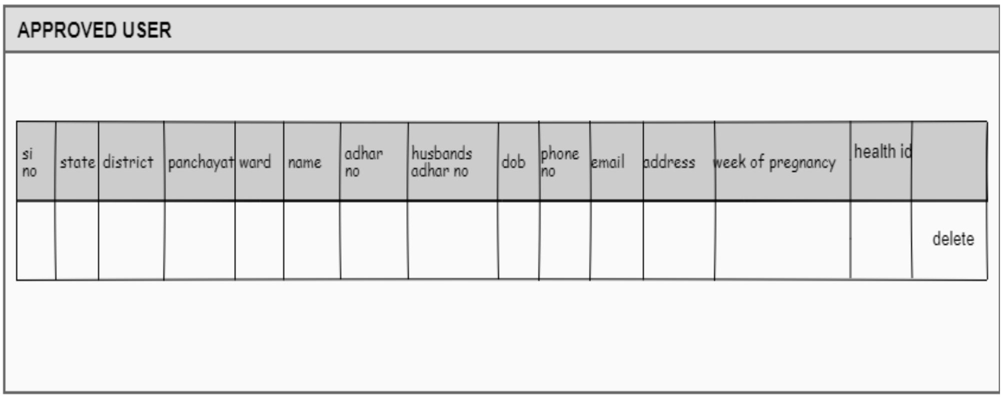
Form Name: Approve user



REGISTERED USER

| si no | state | district | panchayat | ward | name | adhar no | husbands adhar no | dob | phone no | email | address | week of pregnancy | health id | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | approve |

Form Name : view user

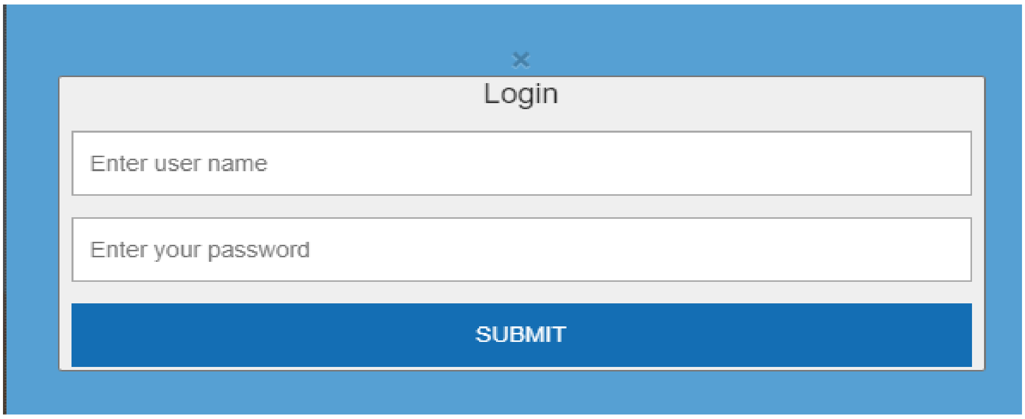


APPROVED USER

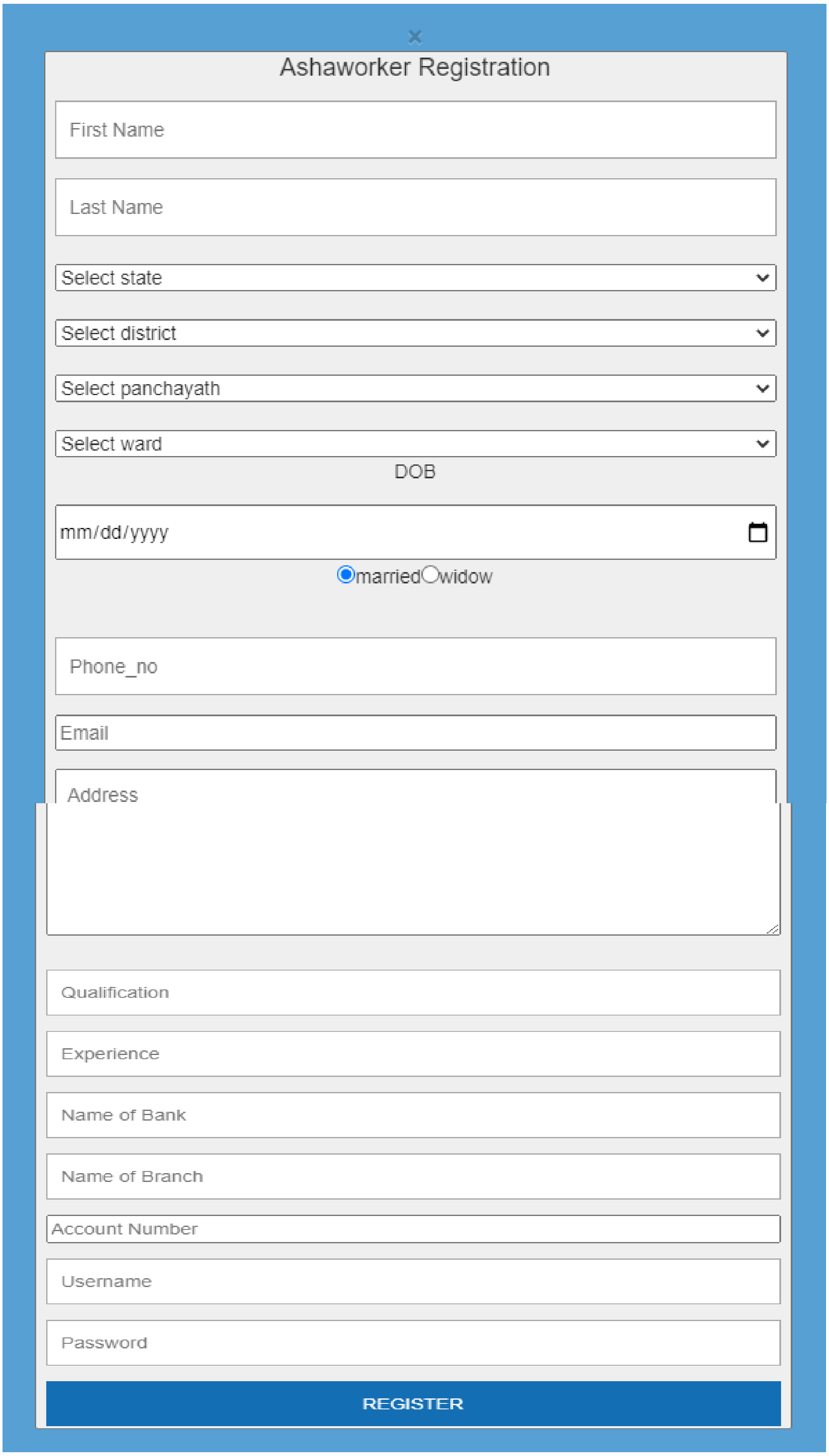
| si no | state | district | panchayat | ward | name | adhar no | husbands adhar no | dob | phone no | email | address | week of pregnancy | health id | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | delete |

### 4.3.2 OUTPUT DESIGN

**Login**



## Ashaworker Registration



Ashaworker Registration

First Name

Last Name

Select state

Select district

Select panchayath

Select ward

DOB

mm/dd/yyyy

● married ○ widow

Phone_no

Email

Address

Qualification

Experience

Name of Bank

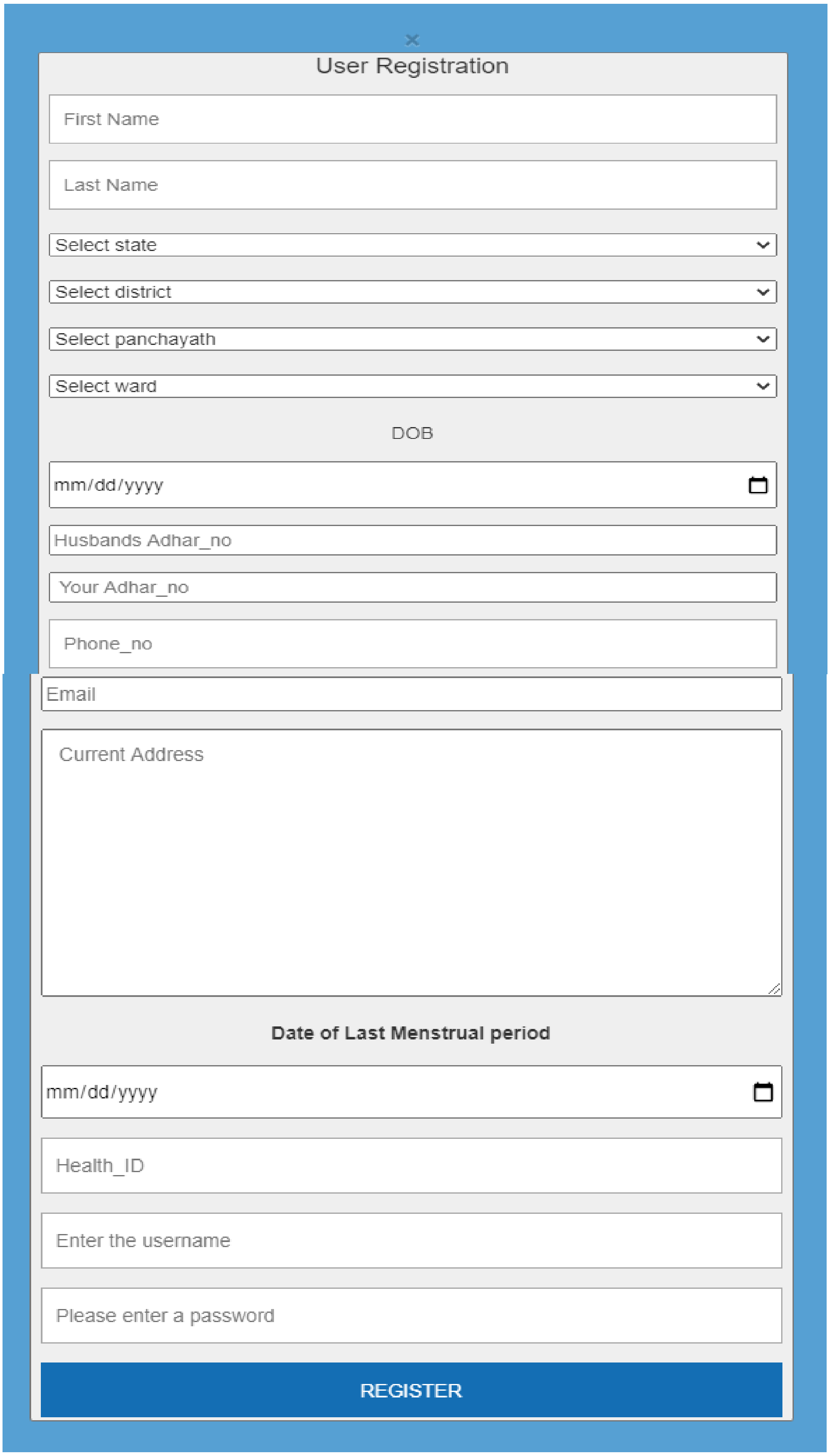
Name of Branch

Account Number

Username

Password

## User Registration



User Registration

First Name

Last Name

Select state

Select district

Select panchayath

Select ward

DOB

mm/dd/yyyy

Husbands Adhar_no

Your Adhar_no

Phone_no

Email

Current Address

**Date of Last Menstrual period**

mm/dd/yyyy

Health_ID

Enter the username

Please enter a password

## 4.4 DATABASEDESIGN

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 specificDBMSthatwillbeusedtoimplementthesysteminquestion.Thisstepiscalled 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 majorobjectives.

- Data Integrity
- Dataindependence

**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 tale 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.

### 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 donor 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 itsdependentattributes.Makesuretokeeparelationwiththeoriginalprimarykeyand any attributes that are fully functionally dependent on it. This step helps in taking out datathatisonlydependentonapartofthekey.Arelationissaidtobeinsecondnormal 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 we 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 attribute.

## TABLE DESIGN

**Table No : 01**

**Table Name : tbl_login**

**Primary Key : login_id**

**Foreign Key :**

**Table Description : To store login information**

| Sl no. | Column Name | Data Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **login_id** | Int | Primary key | Login Id |
| 2 | **Username** | varchar(20) | | Username |
| 3 | **Password** | varchar(20) | | Password |
| 4 | **Type** | varchar(20) | | Type |
| 5 | **Status** | Int | | Status |

**Table No : 02**

**Table Name : tbl_asharegister**

**Primary Key : asha_id**

**Foreign Key : login_id, ward_id**

**Table Description : To store ashaworker registration information**

| | | | | |
|---|---|---|---|---|
| 1 | **asha_id** | Int | Primary key | Ashaworker Id |
| 2 | **login_id** | Int | Foreign key | Login Id |
| 3 | **First_name** | varchar(20) | | First name |
| 4 | **last_name** | varchar(20) | | Last name |
| 5 | **ward_id** | Int | Foreign key | Ward Id |
| 6 | **Dob** | Date | | Date of Birth |
| 7 | **marital_status** | varchar(20) | | Marital Status |
| 8 | **ph_no** | varchar(20) | | Phone Number |
| 9 | **Email** | varchar(50) | | Email Id |
| 10 | **Address** | varchar(100) | | Address |
| 11 | **Qualification** | varchar(50) | | Qualification |

| | | | | |
|---|---|---|---|---|
| 12 | **Experience** | varchar(50) | | Experience |
| 13 | **Bank** | varchar(50) | | Bank Name |
| 14 | **Branch** | varchar(50) | | Bank Branch name |
| 15 | **Acno** | varchar(50) | | Bank Account number |

**Table No : 03**

**Table Name : tbl_user**

**Primary Key : user_id**

**Foreign Key : login_id, ward_id**

**Table Description : To store user registration information**

| **Sl no.** | **Column Name** | **Data Type** | **Constraint** | **Description** |
|---|---|---|---|---|
| 1 | **user_id** | Int | Primary key | User Id |
| 2 | **login_id** | Int | Foreign key | Login Id |
| 3 | **First_name** | varchar(20) | | First name |
| 4 | **Last_name** | varchar(20) | | Last name |
| 5 | **ward_id** | Int | Foreign key | Ward Id |
| 6 | **ad_num** | Bigint | | Aadhaar Number |
| 7 | **Hus_Ad_num** | Bigint | | Husband’s Aadhaar Number |
| 8 | **phone_no** | Varchar(12) | | Phone number |
| 9 | **Dob** | Date | | Date of Birth |
| 10 | **Email** | Varchar(50) | | Email Id |
| 11 | **Address** | Varchar(100) | | Address |
| 12 | **health_id** | Varchar(20) | | Health Id |

**Table No : 04**

**Table Name : tbl_childrecord**

**Primary Key : child_id**

**Foreign Key : user_id**

**Table Description : To store child information**

| Sl no. | Name | Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **child_id** | Int | Primary key | Child id |
| 2 | **user_id** | Int | Foreign key | User id |
| 3 | **Weight** | Varchar(20) | | Weight |
| 4 | **Name** | Varchar(20) | | Name |
| 5 | **Dob** | Date | | Date of birth |
| 6 | **Gender** | Varchar(10) | | Gender |
| 7 | **Health** | Varchar(10) | | Health condition of the child |

**Table No : 05**

**Table Name : tbl_ Immunization**

**Primary Key : imm_id**

**Foreign Key : child_id**

**Table Description : To store immunization details.**

| Sl no | Column Name | Data Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **imm_id** | Int | Primary key | Immunization id |
| 2 | **child_id** | Int | Foreign key | Child id |
| 3 | **Title** | Varchar(20) | | Title |
| 4 | **Date** | Date | | Date |
| 5 | **Vac** | Date | | Vaccination date |

**Table No : 06**

**Table Name : tbl_ notification**

**Primary Key : un_id**

**Foreign Key : notification_id**

**Table Description : To store notification**

| Sl no. | Column Name | Data Type | constraint | Description |
|---|---|---|---|---|
| 1 | **unid** | Int | Primary key | User Notification Id |
| 2 | **notification_id** | Int | Foreign key | User Id |
| 3 | **Message** | Varchar(500) | | Message |

**Table No : 07**

**Table Name : tbl_ state**

**Primary Key : state_id**

**Foreign Key :**

**Table Description: To store state information**

| Sl no. | Column Name | Data Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **state_id** | Int | Primary key | State Id |
| 2 | **state_name** | Varchar(30) | | State Name |

**Table No : 08**

**Table Name : tbl_ district**

**Primary Key : district_id**

**Foreign Key : State_id**

**Table Description : To store district information**

| Sl no. | Column Name | Data Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **district_id** | Int | Primary key | District Id |
| 2 | **district** | Varchar(30) | | District |
| 3 | **State_id** | int | Foreign key | State id |

**Table No : 08**

**Table Name : tbl_ panchayath**

**Primary Key : panchayath_id**

**Foreign Key : district_id**

**Table Description : To store panchayath information**

| Sl no. | Column Name | Data Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **panchayath_id** | Int | Primary key | Panchayath Id |
| 2 | **panchayath** | Varchar(30) | | pnchayath |
| 3 | **district_id** | int | Foreign key | district id |

**Table No : 09**

**Table Name : tbl_ ward**

**Primary Key : ward_id**

**Foreign Key : panchayath_id**

**Table Description : To store ward information**

| Sl no. | Column Name | Data Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **ward_id** | Int | Primary key | Ward Id |
| 2 | **Ward** | Varchar(30) | | Ward |
| 3 | **panchayath_id** | int | Foreign key | Panchayath id |

**Table No : 10**

**Table Name : tbl_ userdetails**

**Primary Key : Userdetails_id**

**Foreign Key : User_id**

**Table Description : To store user details**

| Sl no. | Column Name | ıta Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **Userdetails_id** | Int | Primary key | User details id |
| 2 | **User_id** | Int | Foreign key | User id |
| 3 | **Dlmp** | date | | Date of last menstrual period |
| 4 | **Num_of_weeks** | Int | | Number of weeks |
| 5 | **Weight** | Varchar(30) | | Weight |
| 6 | **Blood_group** | Varchar(30) | | Blood group |
| 7 | **Health_condition** | Varchar(30) | | Health condition |
| 8 | **Hospital_name** | Varchar(30) | | Hospital name |
| 9 | **Gynec_name** | Varchar(30) | | Gynecologist name |
| 10 | **First_child** | Varchar(30) | | First child |
| 11 | **no_of_children** | Int | | Number of children |
| 12 | **Delivery_type** | Varchar(30) | | Previous delivery type |
| 13 | **Hus_name** | Varchar(30) | | Husband name |
| 14 | **Hus_job** | Varchar(30) | | Husband job |
| 15 | **Job** | Varchar(30) | | Job |
| 16 | **Health_problems** | Varchar(30) | | Health problems |

| | |
|---|---|
| **Table No** | **: 11** |
| **Table Name** | **: tbl_Ashaworker_report** |
| **Primary Key** | **: Userdetails_id** |
| **Foreign Key** | **: User_id** |
| **Table Description** | **: To store ashaworker report** |

| Sl no. | Column Name | ıta Type | Constraint | Description |
|---|---|---|---|---|
| 1 | **Rp_id** | Int | Primary key | Report id |
| 2 | **Asha_id** | Int | Foreign key | Ashaworker id |
| 3 | **Year** | Int | | Year of report |
| 4 | **Month** | | | Month of report |
| 5 | **no_of_deliveries** | Int | | Number of deliveries |
| 6 | **normal** | Int | | Number of normal deliveries |

| | | | | |
|---|---|---|---|---|
| 7 | **cesarean** | Int | | Number of cesarean deliveries |
| 8 | **boy** | Int | | Number of boys |
| 9 | **girl** | Int | | Number of girls |
| 10 | **underweight** | Int | | Number of under weight children |
| 11 | **Birth_problems** | Int | | Number of birth problems found |
| 12 | **mother_death** | Int | | Number of mother death during delivery |
| 13 | **child_death** | Int | | Number of child death during delivery |

# 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 actuallywanted.

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 behavior 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 undiscoverederror.
- A successful test is one that uncovers an undiscoverederror.

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 engineer screate 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 best ated in measurable terms. So that the meantime 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 testplan.

The levels of testing include:

- Unit testing
- IntegrationTesting
- Data validationTesting
- Output Testing

### 5.2.1 UnitTesting

Unit testing focuses verification effort on the smallest unit of software design–the software component or module. Using the component level design description as aguide, 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 aretested.

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 setup to rerouteor 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 where 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 thatwill 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 new system design in to operation, which is the process of converting a new revised system design into an operation alone.

At this stage the main workload, 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 are liable 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.

## 6.2 IMPLEMENTATIONPROCEDURES

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 thenew system.
-  Their confidence in the software is builtup.
-  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 lifecycle. 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

The Maternal Health Activist System is developed mainly focusing on the struggle of asha workers walking house to house to collect and store the details of pregnant ladies in their specified area. To reduce the bulk of paper work, this system marks its success. This is an efficient and secure system. As well as the system has flexibility and ease of use. The proposed system has a lot of benefits including time conservation, less effort and user friendly. More over the system is transparent with good security concern.

The system is found to be work efficiently and effectively. New features can be added with slight modification of the software, which make it easy to expand the scope of the Maternal Health Activist System.

### 7.2 FUTURESCOPE

In this Maternal Health Activist System, there is a great scope for future development. Asha workers has limitations to to guide pregnant ladies in their health condition because they are not qualified medical representatives. So for that we can add a direct consultancy of a doctor for users. So that the doctor can give necessary guidelines about the health status of pregnant women.

The project is currently implemented based on a particular panchayath and wards under the gramapanchayath . We can further implement this system in an all India level where the admin will be a head from the health commission department of India and we have to add another module called the JPHN(Junior public health nurse) , the current admin to coordinate all the JPHNs in our country.

# CHAPTER 8

# BIBLIOGRAPHY

## REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”,2009.
- Roger S Pressman, “*Software Engineering*”,1994.
- PankajJalote, “So*ftware engineering*: a precise approach”,2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

## WEBSITES:

- http://www.codeproject.com
- http://www.codesource.com
- http://www.w3schools.com
- http://www.stackoverflow.com

# CHAPTER 9

# APPENDIX

## 9.1 Sample Code

**asharegister.php**

```
<html><body>
<div class="modal fade" id="areg" tabindex="-1" role="dialog">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header" style="  overflow-x: hidden; overflow-y: auto;">
<button type="button" class="close" data-dismiss="modal">&times;<button>
<h4>Ashaworker Registration</h4>
<div class="login">
<form name="myform" method="post" action="ashareg_action.php" onsubmit="return validateform()">
<?php
include('db.php');
?>
<input type="text" name="first_name" id="first_name" placeholder="First Name" required >
<input type="text" name="last_name" id="last_name" placeholder="Last Name" required ><br><br>
<select name="state" required id="state1">
State
<option value="">Select state</option>
<?php
$select="select *from state";
$ex=mysqli_query($con,$select);
while($array=mysqli_fetch_array($ex))
{
?>
<option value="<?php echo $array['state_id'];?>"><?php echo $array['state_name'];?></option>
<?php
}
```

```
?>
</select><br><br>
<select name="district" required id="dist1">
<option value="">Select district</option>
</select><br><br>
<select name="panchayath" required id="panchayath1">
<option value="">Select panchayath</option>
</select><br><br>
<select name="ward" required id="ward1">
<option value="">Select ward</option>
</select>
<right>DOB</right>
<input type="date" max="2000-01-01" name="dob" required>
<!-<input type="text" onfocus="(this.type='date')" onblur="(this.type='text')" name="dob"
 max="2001-01-01" required  placeholder="DOB">-->
<input type="radio" name="marital_status" value="married" checked>married<input type
="radio name="marital_status" value="widow">widow<br><br>
<input type="text" name="ph_no"   maxlength="10" pattern="^[6789]\d{9}$" placeholder
="Phone_no" required>
<input type="email" name="email" placeholder="Email" required>
<textarea name="address" rows="4" required  placeholder="Address"></textarea>
<input type="text" name="qualification" required  placeholder="Qualification">
<input type="text" name="experience" required  placeholder="Experience">
<input type="text" name="nob" required placeholder="Name of Bank">
<input type="text" name="nobr" required placeholder="Name of Branch">
<input type="number" name="acno" maxlength="14" required placeholder="Account Nu
mber">
<input type="text" name="username" required  placeholder="Username">
<input type="password" name="password" required  placeholder="Password">
<input type="submit" value="REGISTER" name="add">
</form></div></div></div></div></div>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
```

```
$("#state").change(function(){
var x=$(this).val();
$("#dist").load('get_dist.php',{sid:x});
});
$("#dist").change(function(){
var x=$(this).val();
$("#panchayath").load('get_pan.php',{did:x});
});
 $("#panchayath").change(function(){
 var x=$(this).val();
 $("#ward").load('get_ward.php',{pid:x});
 });
 });
 </script></body></html>
```

**Asha_reg_action.php**

```
<?php
include 'db.php';
if(isset($_POST['add']))
{
$st=$_POST['state'];
$disct=$_POST['district'];
$pan=$_POST['panchayath'];
$ward=$_POST['ward'];
$first_name=$_POST['first_name'];
$last_name=$_POST['last_name'];
$dob=$_POST['dob'];
$marital_status=$_POST['marital_status'];
$ph=$_POST['ph_no'];
$email=$_POST['email'];
 $add=$_POST['address'];
 $qualification=$_POST['qualification'];
 $experience=$_POST['experience'];
 $nob=$_POST['nob'];
 $nobr=$_POST['nobr'];
 $acno=$_POST['acno'];
 $user=$_POST['username'];
 $pw=$_POST['password'];
 $ins=mysqli_query($con,"insert into login(`username`,`password`,`type`,`status`)
 values('$us er','$pw','ashaworker','0')");
 $id=mysqli_insert_id($con);
 $insert=mysqli_query($con,"insert into asharegister(`login_id`,`state_id`, `district_id`,
 `panchayath_id`, `ward_id`, `first_name`, `last_name`, `dob`, `marital_status`, `ph_no`,
 `email`, `address`, `qualification`,`experience`,`bank`,`branch`,`acno`)values('$id','$st',
'$disct','$pan','$ward','$first_name','$last_name','$dob','$marital_status','$ph','$email',
'$add','$qualification','$experience','$nob','$nobr','$acno')");

 if($insert)
 {
 header('location:index1.php');
 //echo "<script type='text/javascript'>alert('Data Successfully Inserted');
 //window.location='index.php';
 //</script>";
 }
 }

//header('location:index.php');
?>
```

**userregister.php**

```
<html><body>
 <div class="modal fade" id="reg" tabindex="-1" role="dialog">
 <div class="modal-dialog">
 <!-- Modal content-->
 <div class="modal-content">
  <div class="modal-header" style="  overflow-x: hidden;   overflow-y: auto;">
 <button type="button" class="close" data-dismiss="modal">&times;<button>
  <h4>User Registration</h4>
  <div class="login">
 <form name="form" method="post" action="userreg_action.php" onsubmit
 ="return validate()">
 <?php
 include('db.php');
  ?>
 <input type="text" name="firstname"  placeholder="First Name" required>
 <input type="text" name="lastname"  placeholder="Last Name" required><br><br>
 <select name="state" required id="state">
 <option value="">Select state</option>
 <?php
  $select="select *from state";
  $ex=mysqli_query($con,$select);
  while($array=mysqli_fetch_array($ex))
 {
 ?>
 <option value="<?php echo $array['state_id'];?>"><?php echo $array['state_name'];?></o
 ption>
 <?php
 }
 ?>
 </select><br><br>
 <select name="district" required id="dist">
 <option value="">Select district</option>
 </select><br><br>
 <select name="panchayath" required id="panchayath">
 <option value="">Select panchayath</option>
 </select><br><br>
 <select name="ward"  id="ward">
 <option value="">Select ward</option>
 <!--<option value="1">1</option>
 <option value="2">2</option>
 <option value="3">3</option>-->
 </select><br><br>
```

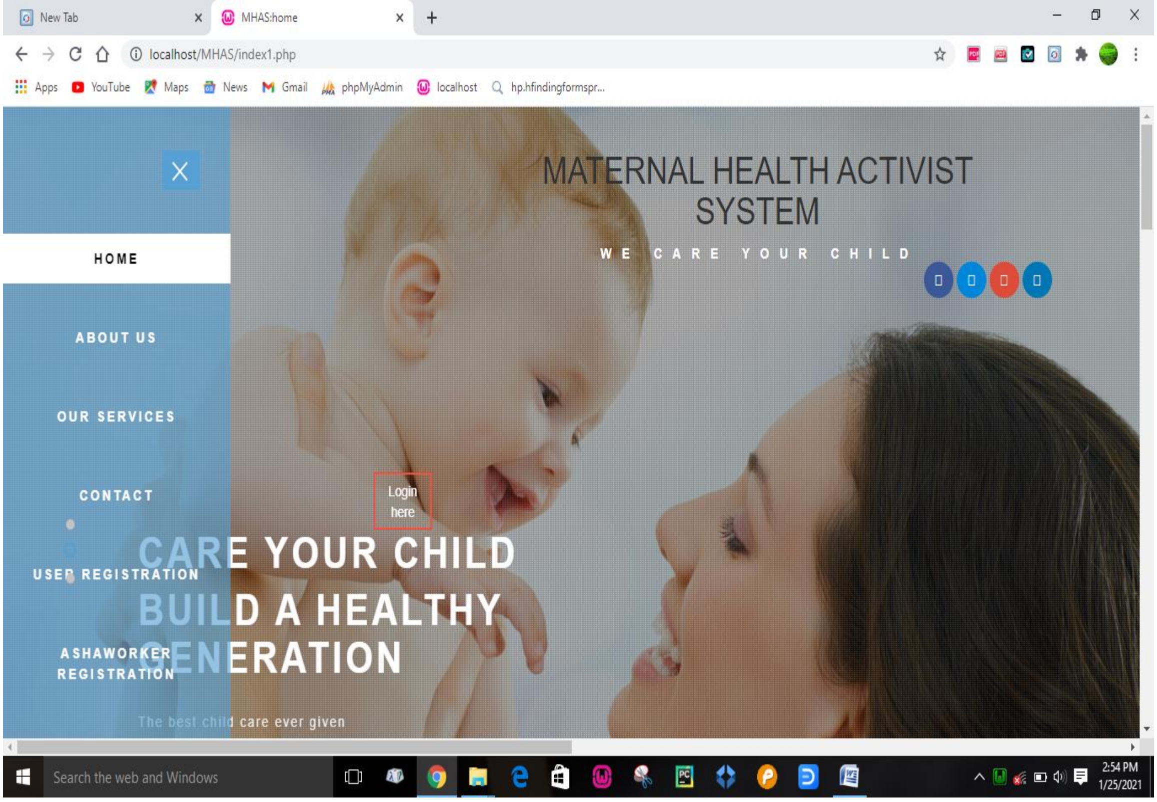
```
<right>DOB</right>
<input type="date" max="2000-01-01" name="dob">
<input name="han" oninput="javascript: if (this.value.length > this.maxLength) this.value
= this.value
slice(0, this.maxLength);"type = "number"  maxlength = "12" placeholder="Husbands Ad
har_no"
required />
<input type="number" name="an" oninput="javascript: if (this.value.length > this.maxLen
gth) this.value = this.value.slice(0, this.maxLength);"  maxlength = "12" required  placehol
der=" Your Adhar_no">
<input type="text" name="ph_no"   maxlength="10" pattern="^[6789]\d{9}$" placeholder
="Phone_no" required>
<input type="email" name="email" required placeholder="Email">
<textarea name="address" rows="6" required placeholder=" Current Address"></textarea
>
<label>Date of Last Menstrual period</label>
<input type="date"  name="wop"  max="<?php echo $date;?>" >
<input type="text" name="health_id" required  placeholder="Health_ID">
<input type="text" name="username" required  placeholder="Enter the username">
<input type="password" name="password" required  placeholder="Please enter a passwor
d">
<input type="submit" value="REGISTER" name="add">
</form></div></div></div></div></div></body></html>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#state").change(function(){
var x=$(this).val();
$("#dist").load('get_dist.php',{sid:x});
});
 $("#dist").change(function(){
 var x=$(this).val();
 $("#panchayath").load('get_pan.php',{did:x});
 });
 $("#panchayath").change(function(){
 var x=$(this).val();
 $("#ward").load('get_ward.php',{pid:x});
 });
 });
```

**Reg_action.php**

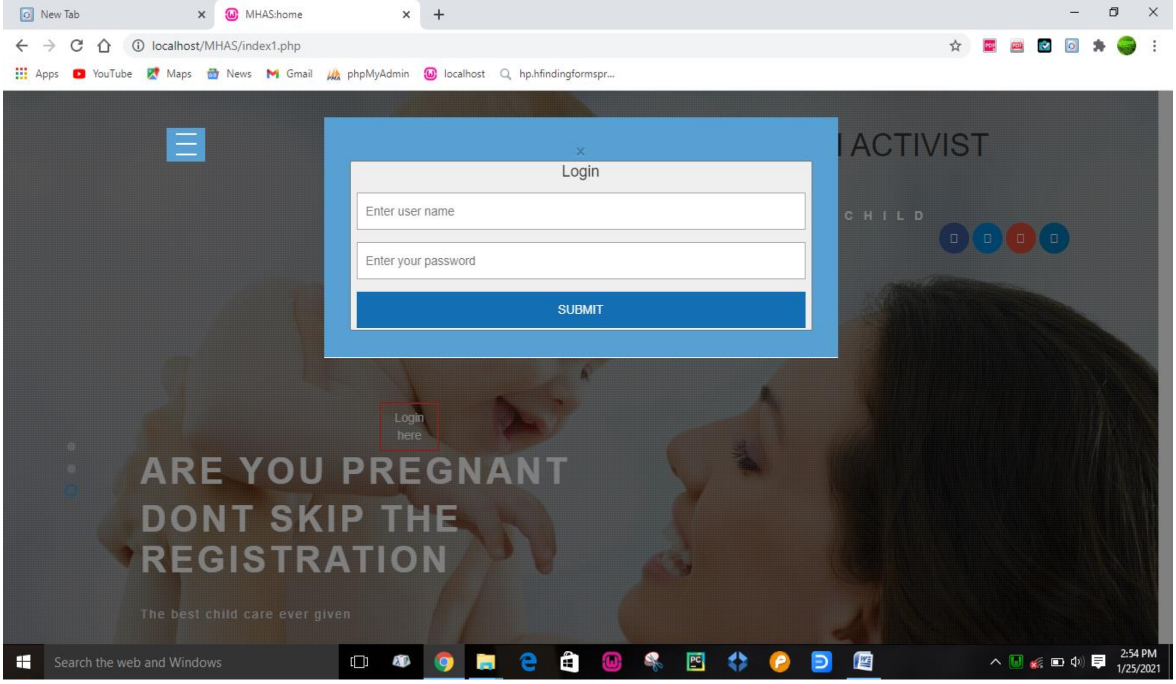
```
<?php
include 'db.php';
if(isset($_POST['add']))
{
 $st=$_POST['state'];
 $disct=$_POST['district'];
 $pan=$_POST['panchayath'];
 $ward=$_POST['ward'];
 $firstname=$_POST['firstname'];
 $lastname=$_POST['lastname'];
 $dob=$_POST['dob'];
 $han=$_POST['han'];
 $an=$_POST['an'];
 $ph=$_POST['ph_no'];
 $email=$_POST['email'];
 $add=$_POST['address'];
 $wop=$_POST['wop'];
 $hid=$_POST['health_id'];
 $user=$_POST['username'];
 $pw=$_POST['password'];
 $ins=mysqli_query($con,"insert into login(`username`,`password`,`type`,`status`)
 values('$user','$pw','user','0')");
 $id=mysqli_insert_id($con);
 $ins="INSERT INTO `user`(`login_id`, `state_id`, `district_id`, `panchayath_id`, `ward_i
d`,
 `firstname`, `lastname`, `dob`, `han`, `an`, `phone_no`, `email`, `address`, `wop`, `health_i
d`)
v
alues('$id','$st','$disct','$pan','$ward','$firstname','$lastname','$dob','$han','$an','$ph','$emai
l',
'$add','$wop','$hid')";
$insert=mysqli_query($con,$ins);
if($insert)
{
header('location:index1.php');
//echo "<script type='text/javascript'>alert('added succesfully');
//window.location='index.php';
//</script>";
}
}
?>
```

## 9.2 SCREEN SHOTS

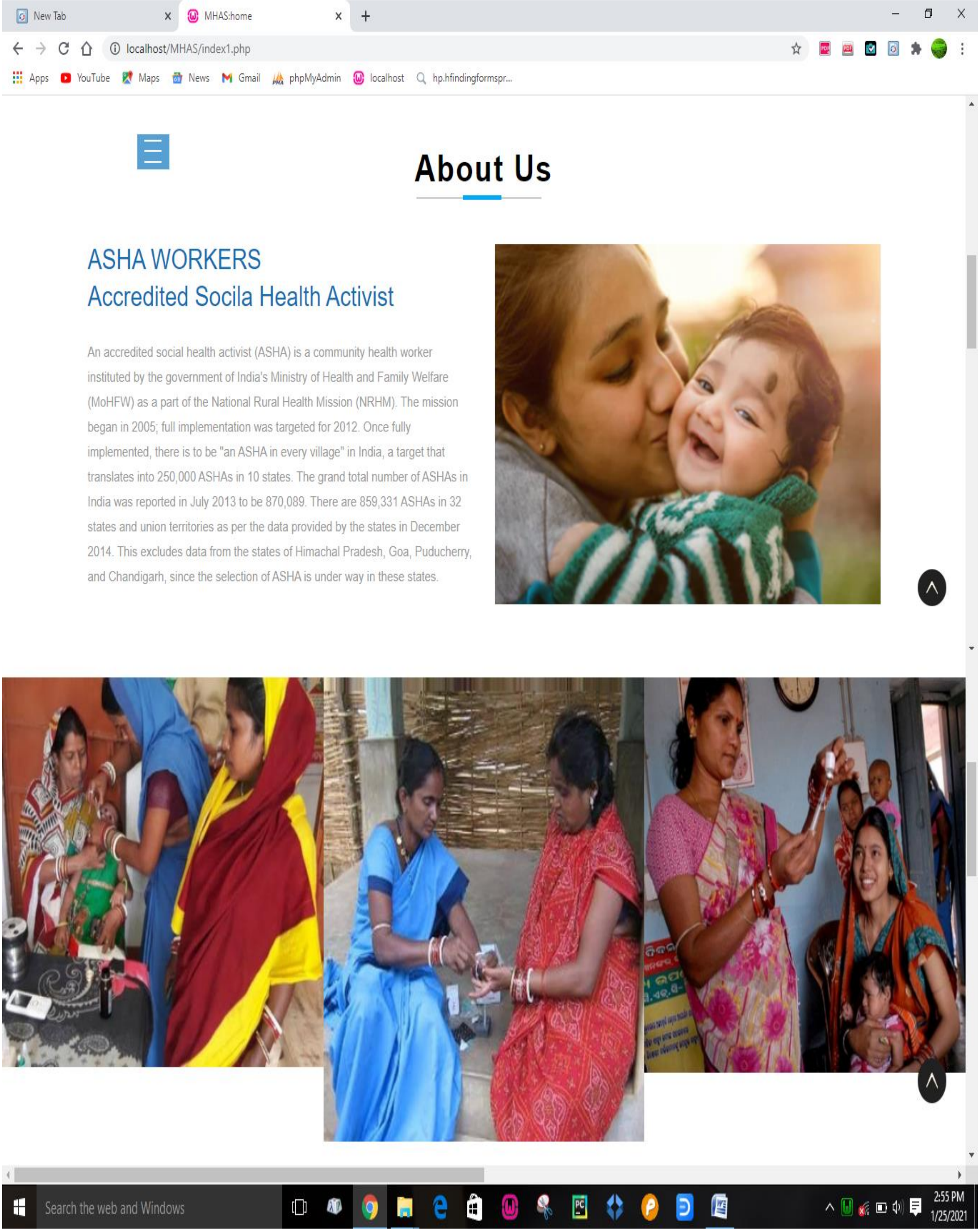
### HOME PAGE



### LOGIN PAGE



## ABOUT PAGE



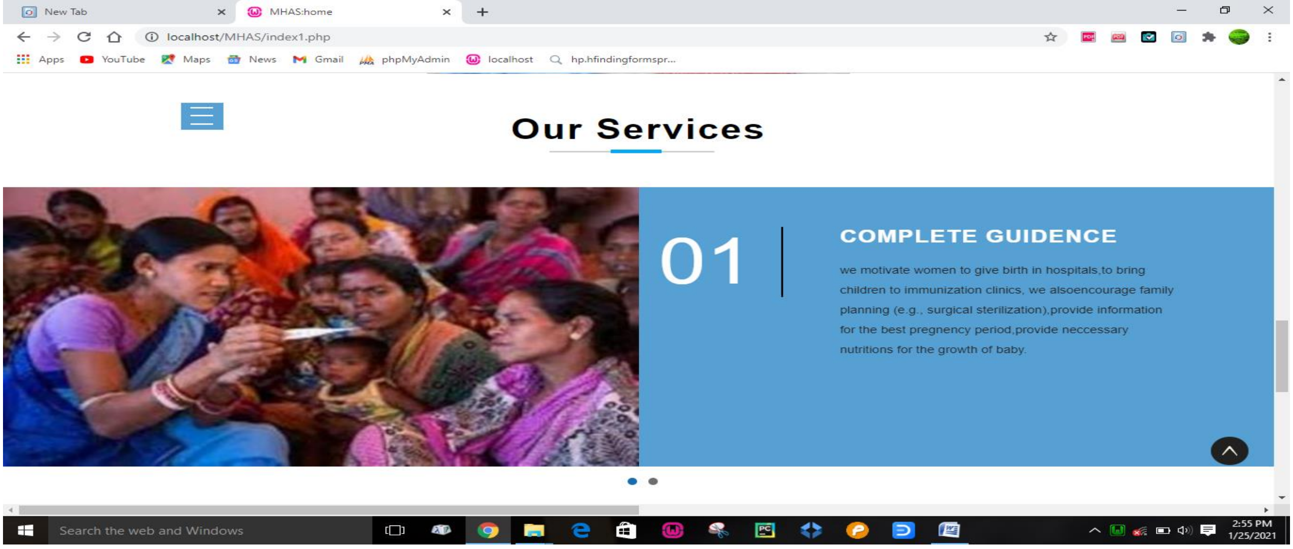
About Us

ASHA WORKERS

Accredited Socila Health Activist

An accredited social health activist (ASHA) is a community health worker instituted by the government of India's Ministry of Health and Family Welfare (MoHFW) as a part of the National Rural Health Mission (NRHM). The mission began in 2005; full implementation was targeted for 2012. Once fully implemented, there is to be "an ASHA in every village" in India, a target that translates into 250,000 ASHAs in 10 states. The grand total number of ASHAs in India was reported in July 2013 to be 870,089. There are 859,331 ASHAs in 32 states and union territories as per the data provided by the states in December 2014. This excludes data from the states of Himachal Pradesh, Goa, Puducherry, and Chandigarh, since the selection of ASHA is under way in these states.

## OUR SERVICES

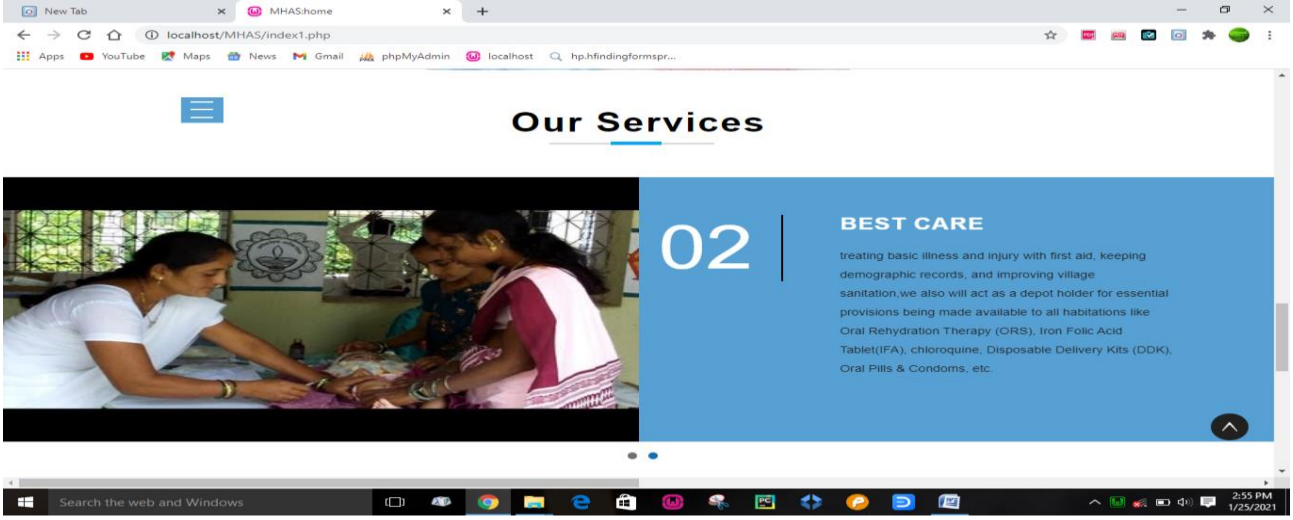


Our Services

01

COMPLETE GUIDENCE

we motivate women to give birth in hospitals,to bring children to immunization clinics, we alsoencourage family planning (e.g., surgical sterilization),provide information for the best pregnency period,provide neccessary nutritions for the growth of baby.



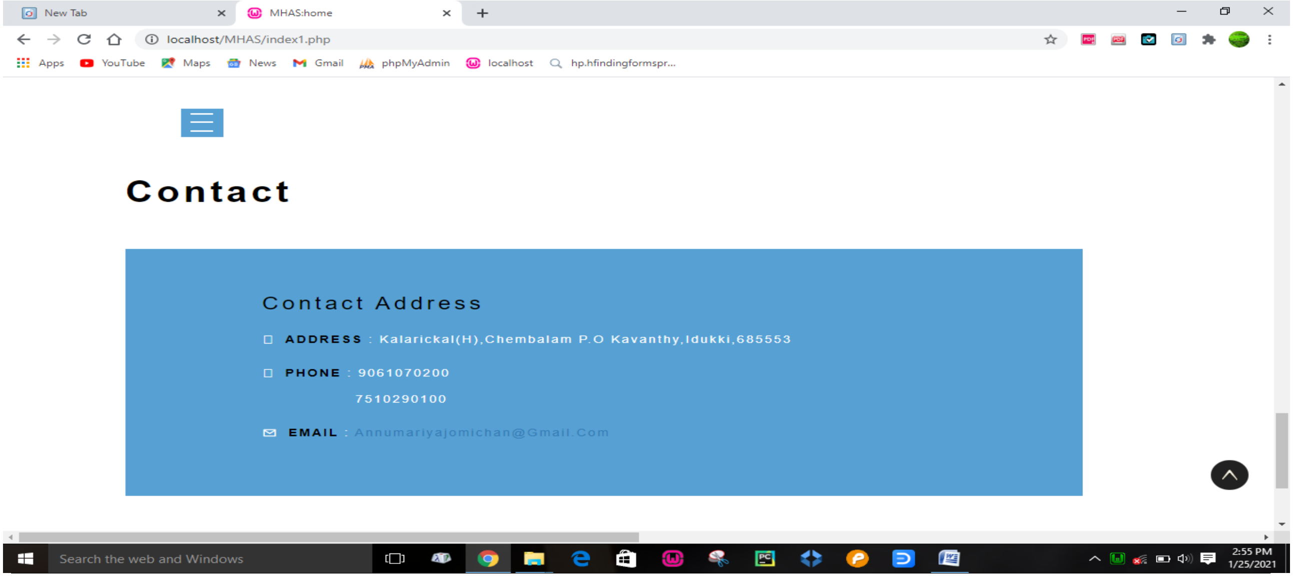
Our Services

02

BEST CARE

treating basic illness and injury with first aid, keeping demographic records, and improving village sanitation,we also will act as a depot holder for essential provisions being made available to all habitations like Oral Rehydration Therapy (ORS), Iron Folic Acid Tablet(IFA), chloroquine, Disposable Delivery Kits (DDK), Oral Pills & Condoms, etc.

## CONTACT PAGE



Contact

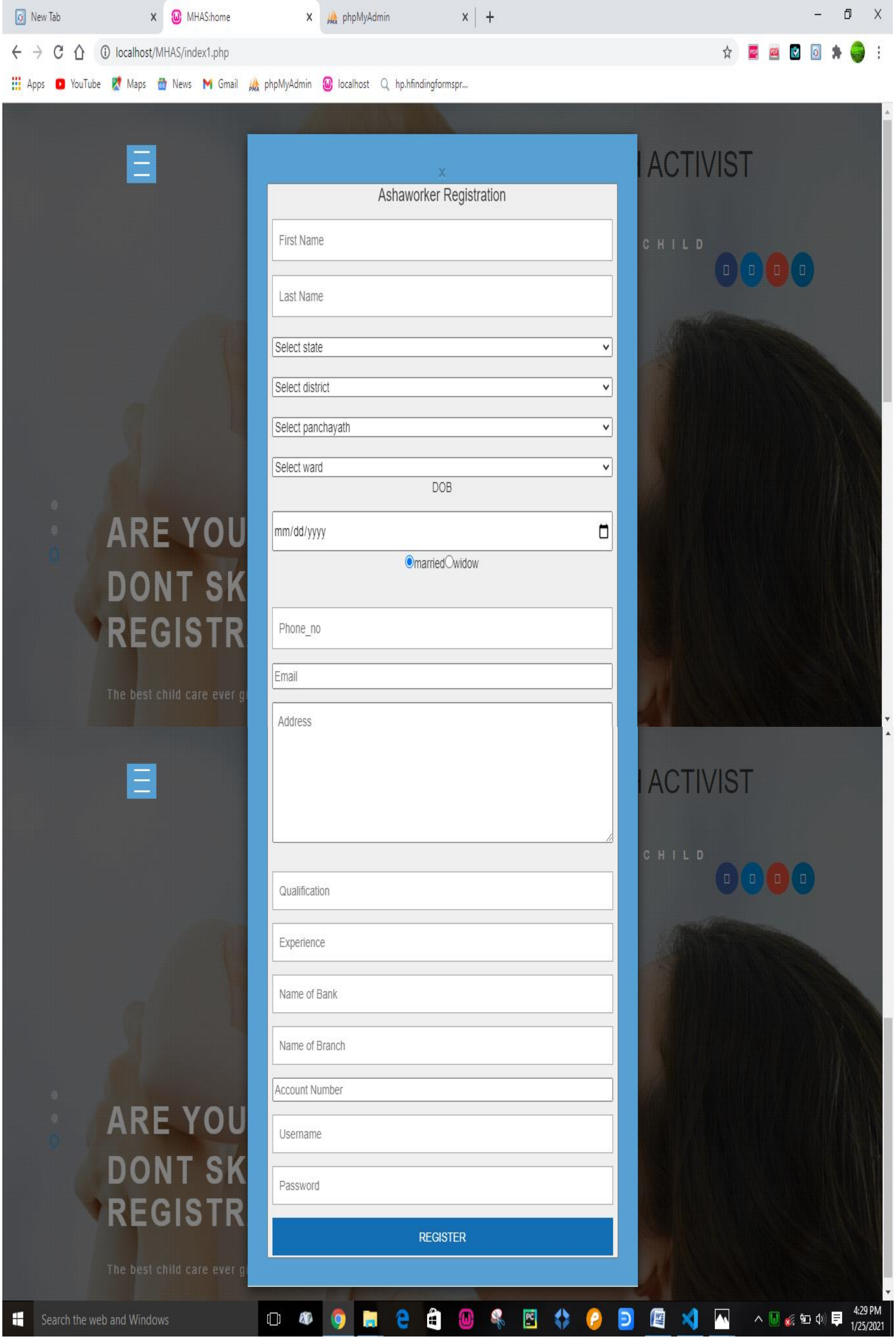
Contact Address

□ ADDRESS : Kalarickal(H),Chembalam P.O Kavanthy,Idukki,685553

□ PHONE : 9061070200
7510290100

✉ EMAIL : Annumariyajomichan@Gmail.Com

## ASHA REGISTER



Ashaworker Registration

First Name

Last Name

Select state

Select district

Select panchayath

Select ward

DOB

mm/dd/yyyy

● married ○ widow

Phone_no

Email

Address

Qualification

Experience

Name of Bank

Name of Branch

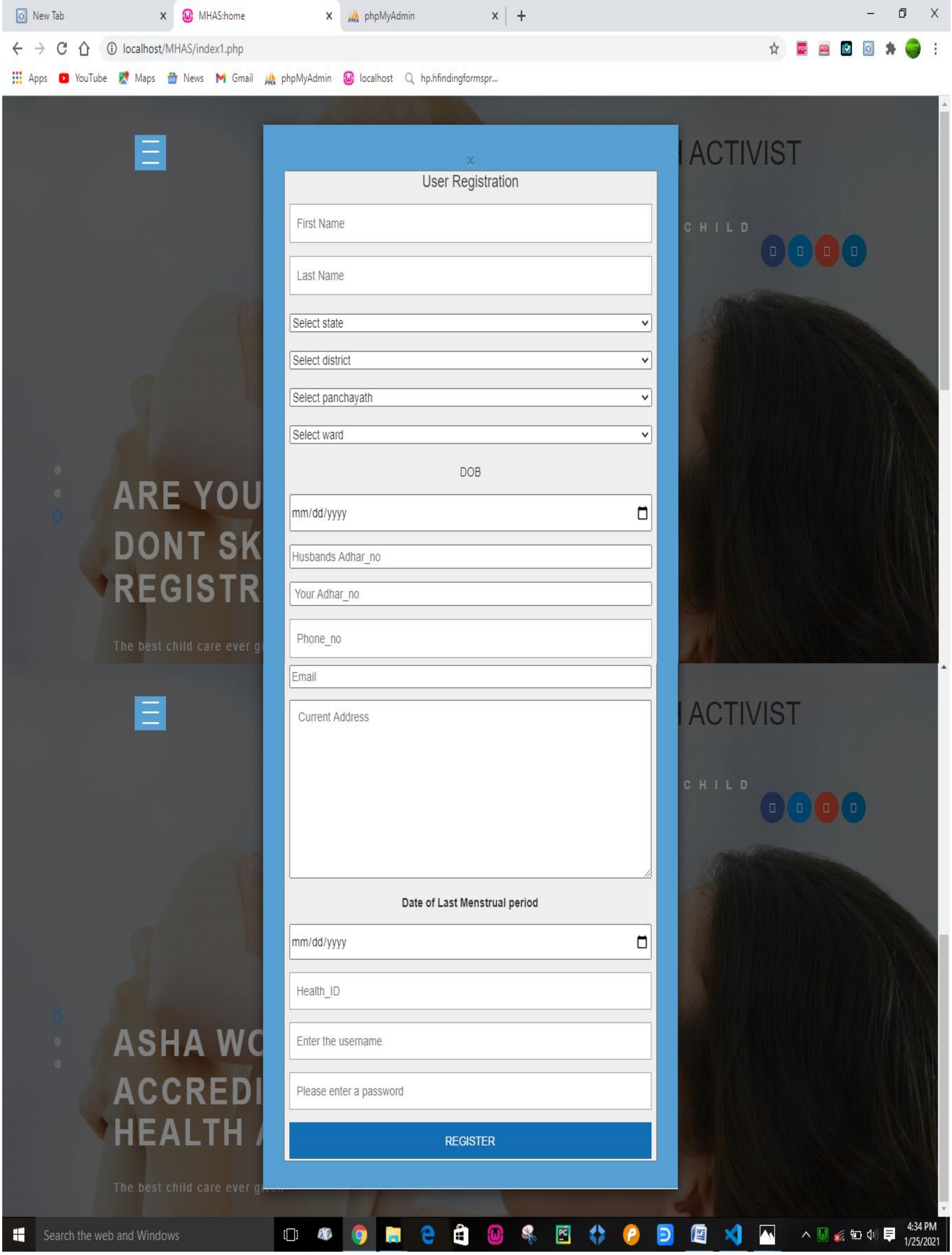
Account Number

Username

Password

REGISTER

## USER REGISTRATION



New Tab
MHAS:home
phpMyAdmin
localhost/MHAS/index1.php
Apps YouTube Maps News Gmail phpMyAdmin localhost hp.hfindingformspr...
ACTIVIST
CHILD
ARE YOU
DONT SK
REGISTR
The best child care ever g
ASHA WO
ACCREDI
HEALTH A
Search the web and Windows
4:34 PM
1/25/2021

User Registration

First Name

Last Name

Select state

Select district

Select panchayath

Select ward

DOB

mm/dd/yyyy

Husbands Adhar_no

Your Adhar_no

Phone_no

Email

Current Address

Date of Last Menstrual period

mm/dd/yyyy

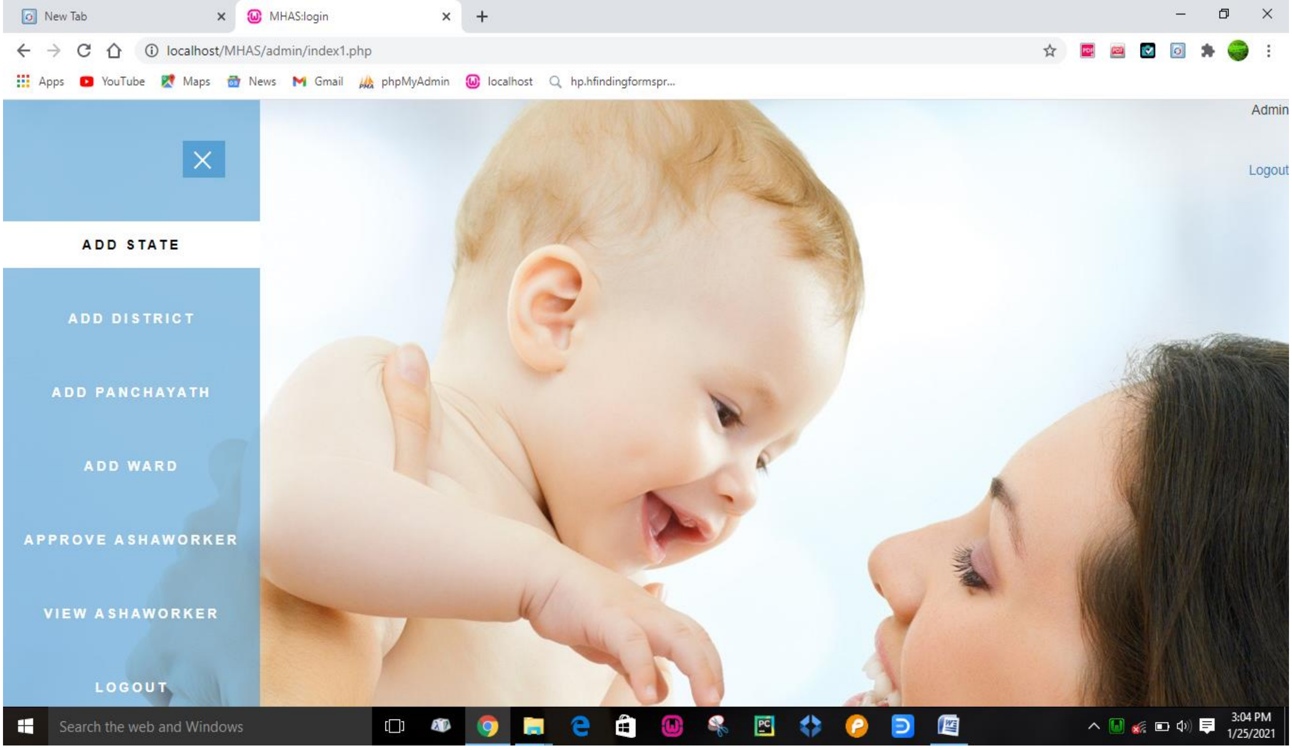
Health_ID

Enter the username

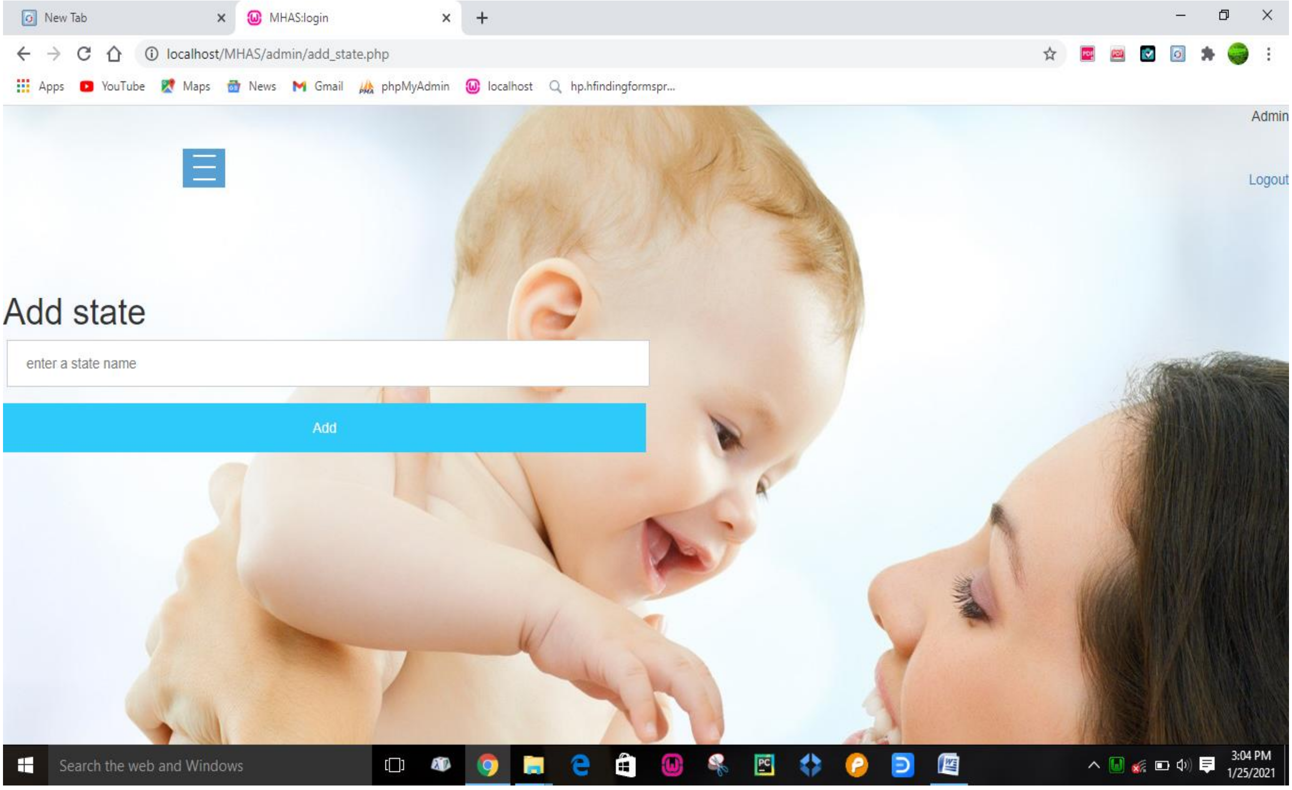
Please enter a password

REGISTER

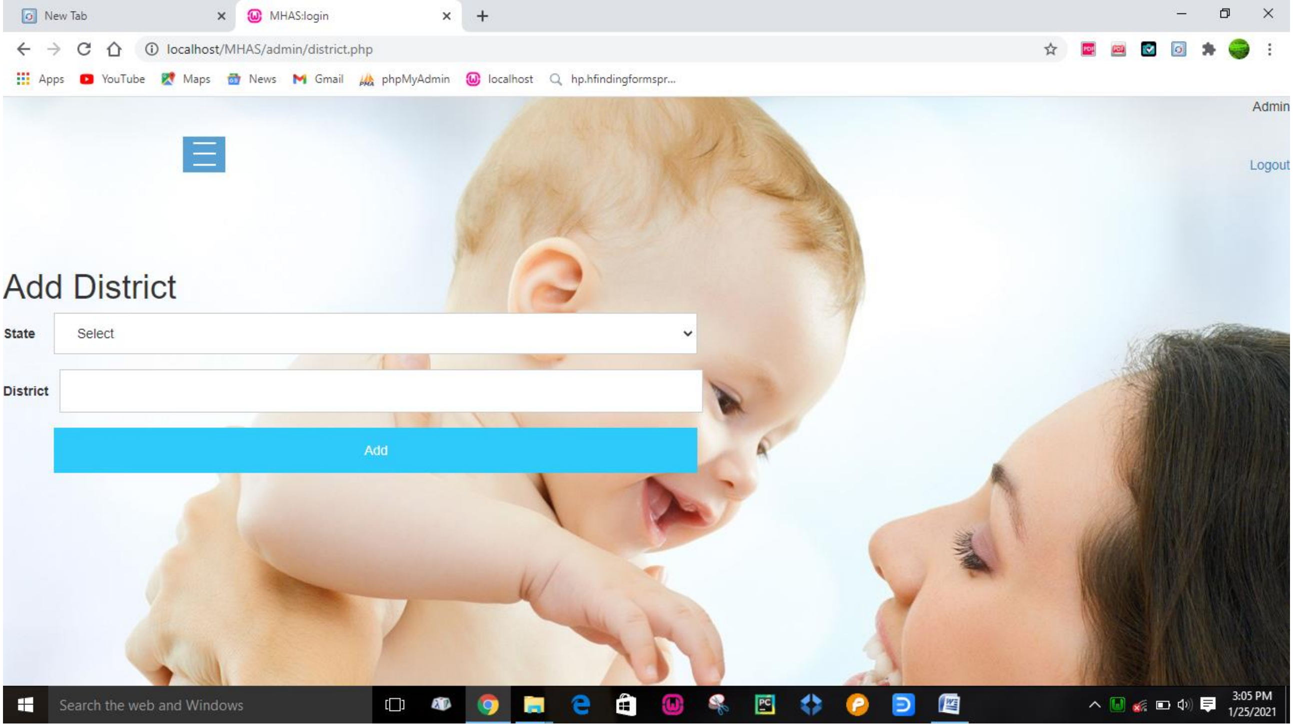
## ADMIN PAGE



## ADD STATE



## ADD DISTRICT



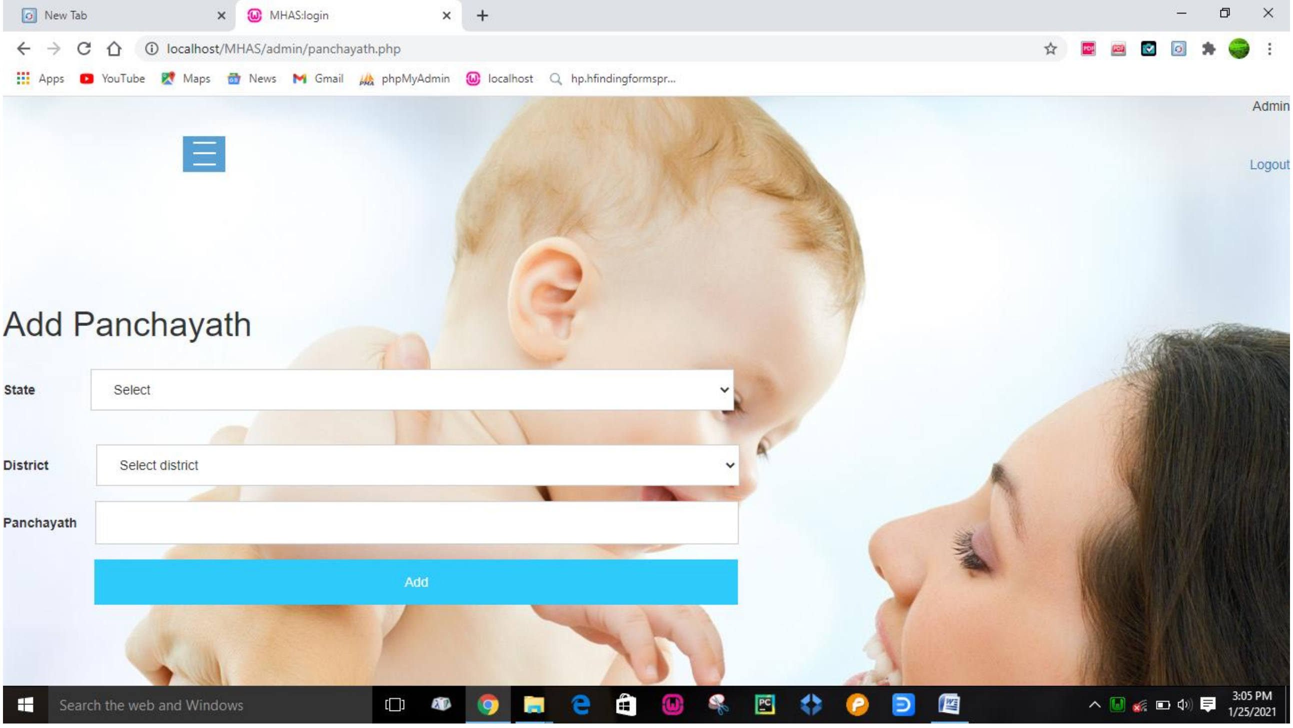
localhost/MHAS/admin/district.php

Add District

State: Select

District:

## ADD PANCHAYATH



localhost/MHAS/admin/panchayath.php

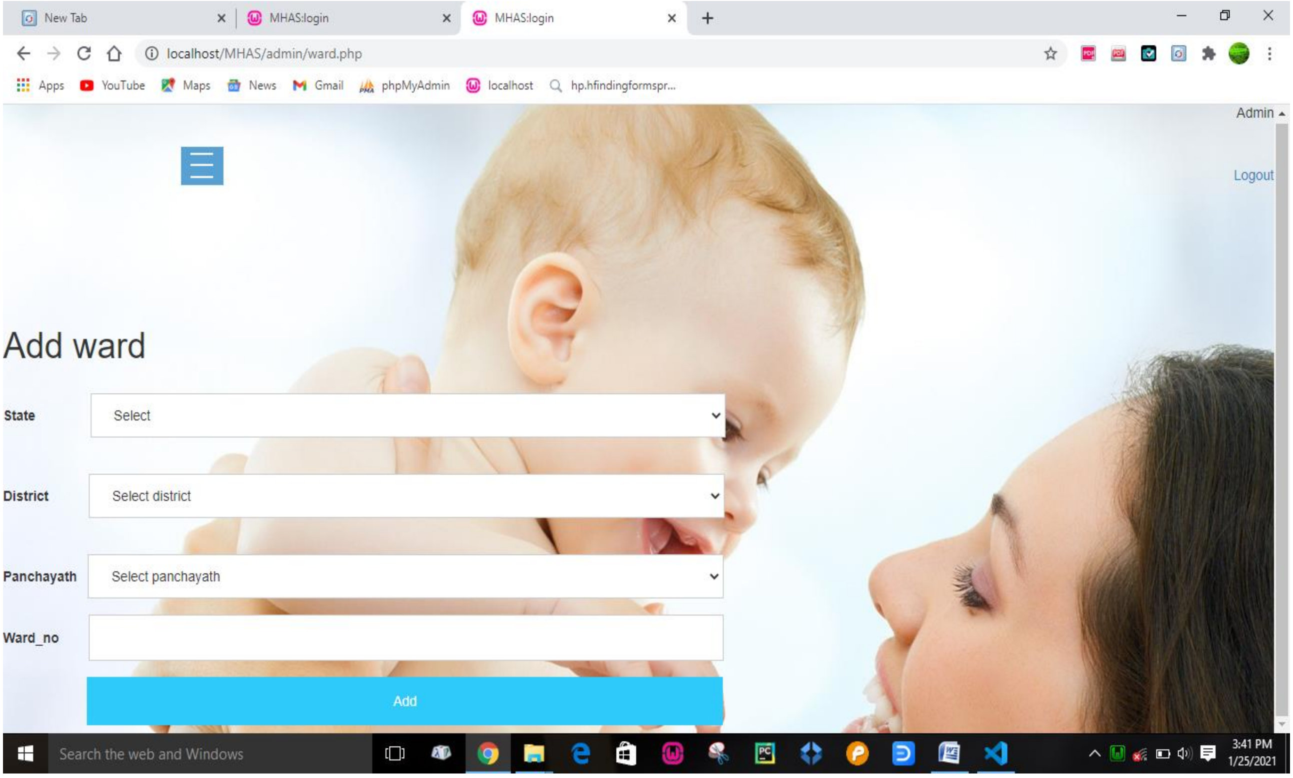
Add Panchayath

State: Select

District: Select district

Panchayath:

## ADD WARD

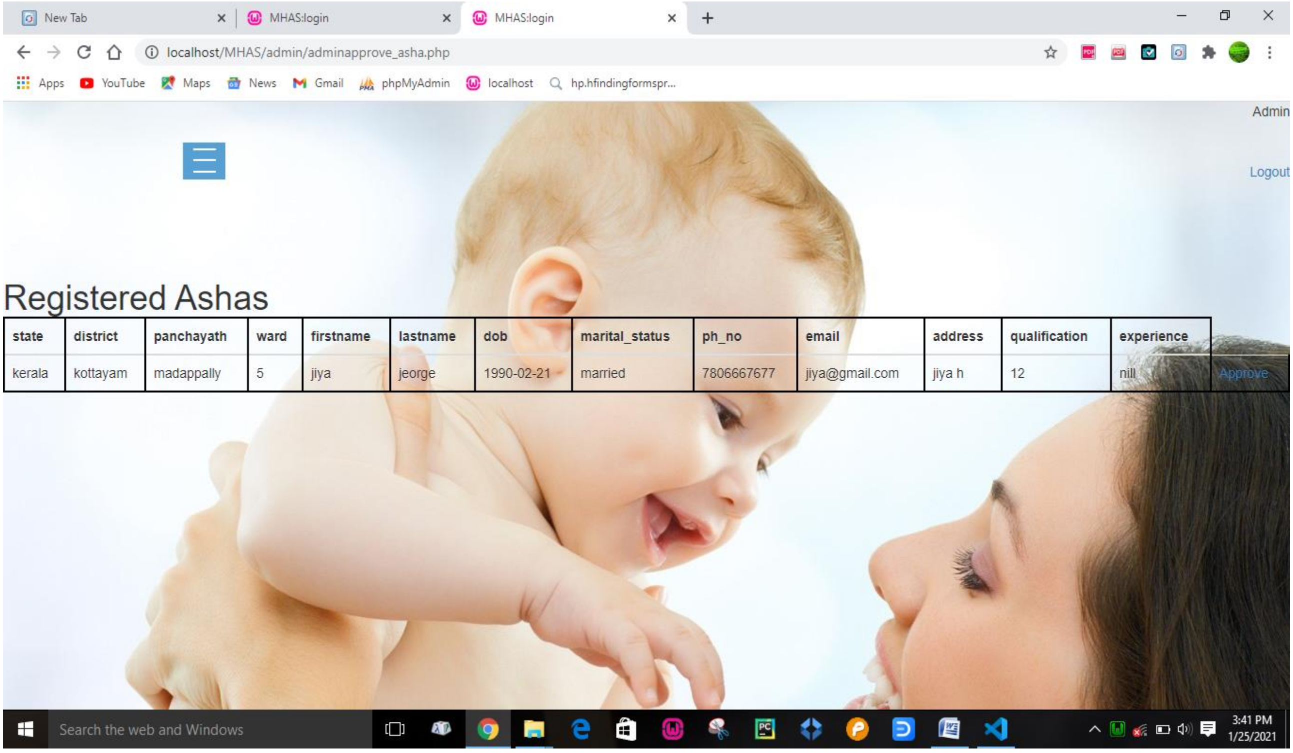


Add ward

| | |
|---|---|
| State | Select |
| District | Select district |
| Panchayath | Select panchayath |
| Ward_no | |

Add

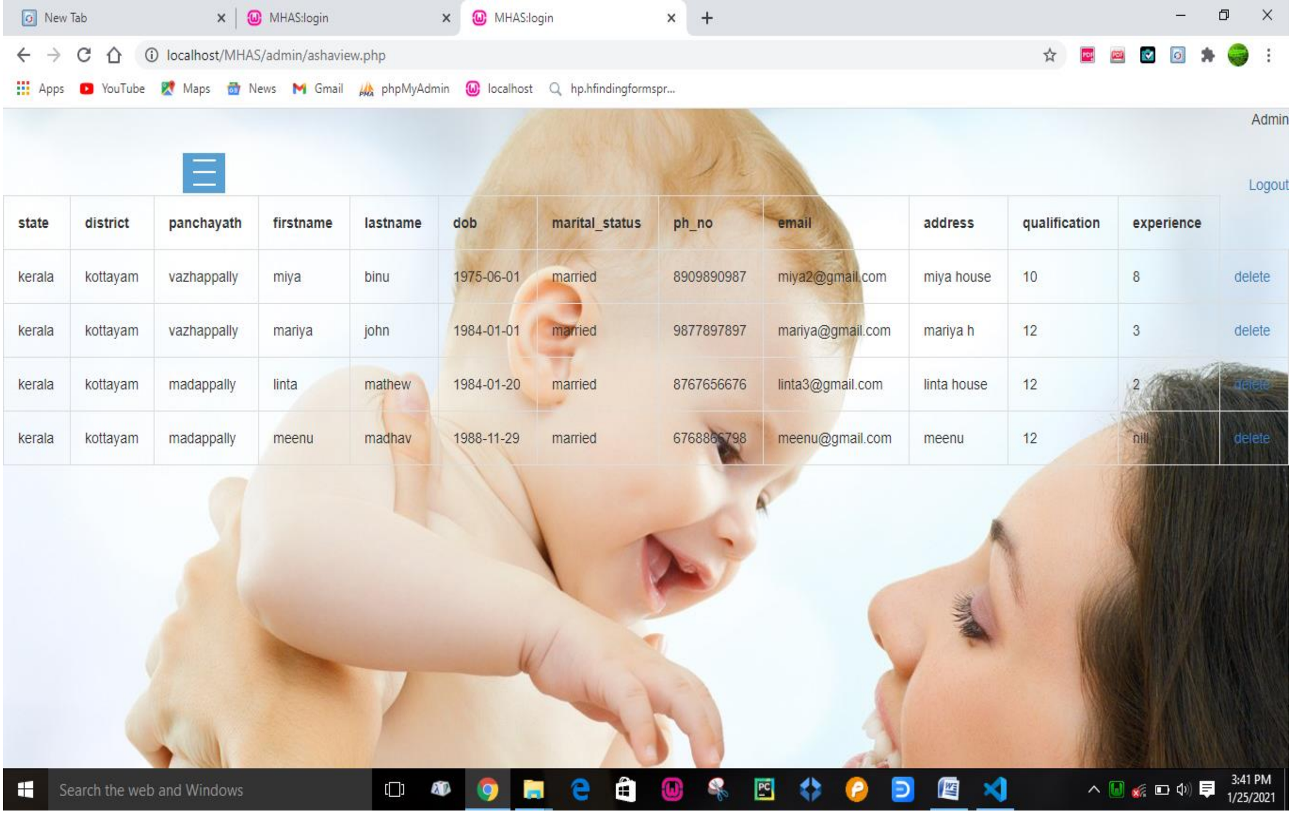
## APPROVE ASHAWORKER



Registered Ashas

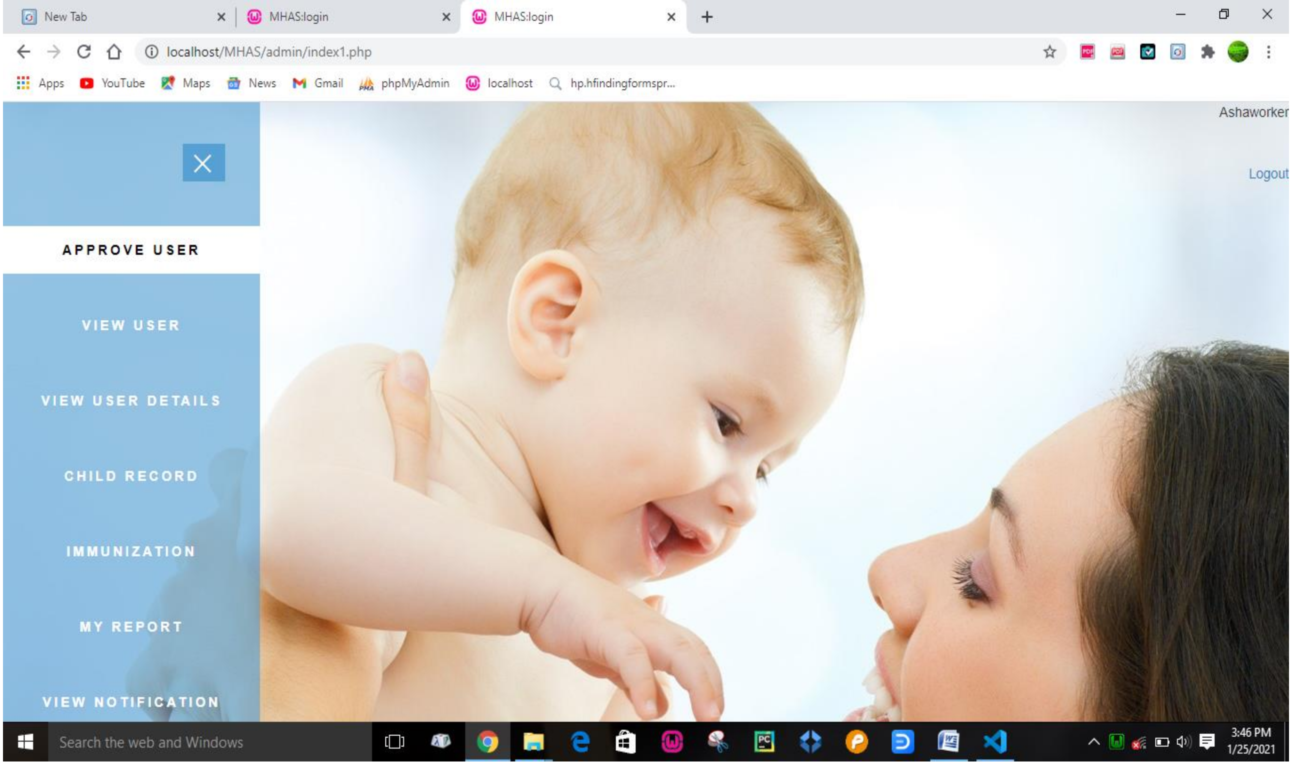
| state | district | panchayath | ward | firstname | lastname | dob | marital_status | ph_no | email | address | qualification | experience | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| kerala | kottayam | madappally | 5 | jiya | jeorge | 1990-02-21 | married | 7806667677 | jiya@gmail.com | jiya h | 12 | nill | Approve |

## VIEW ASHAWORKER

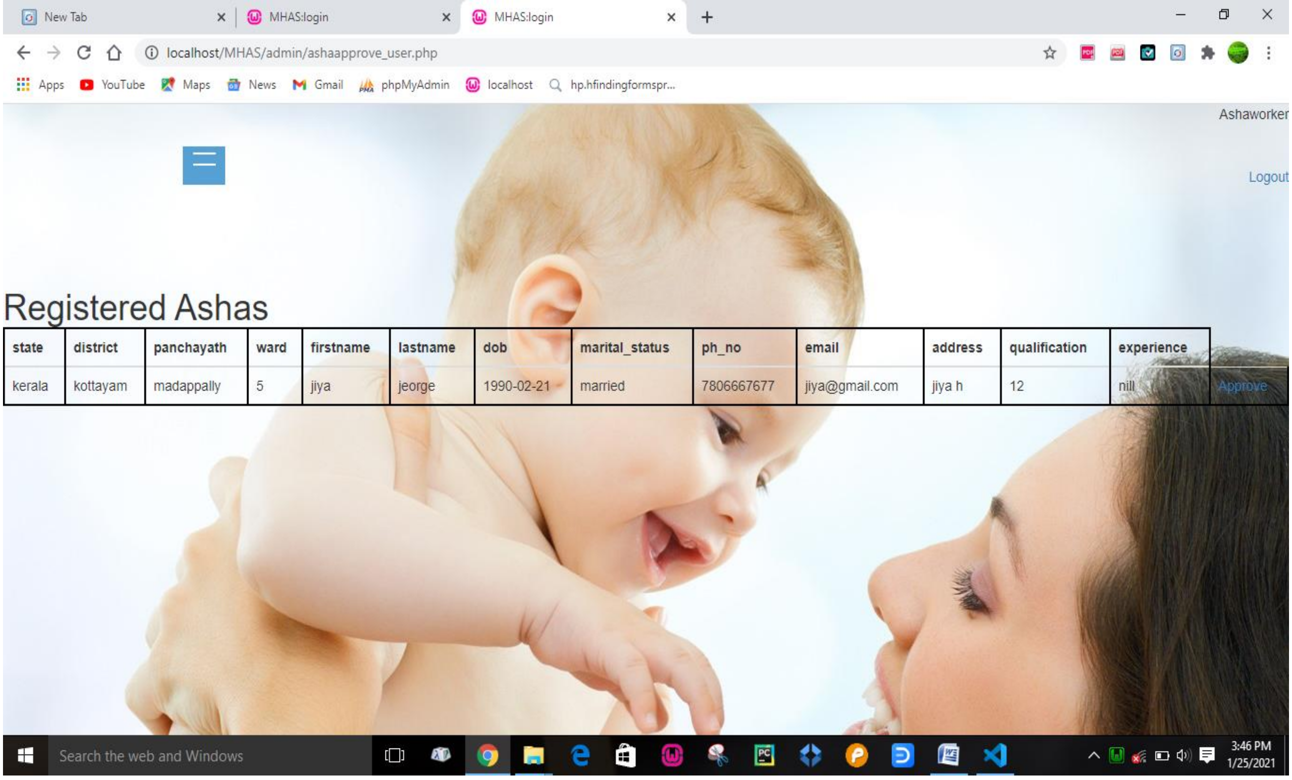


| state | district | panchayath | firstname | lastname | dob | marital_status | ph_no | email | address | qualification | experience | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| kerala | kottayam | vazhappally | miya | binu | 1975-06-01 | married | 8909890987 | miya2@gmail.com | miya house | 10 | 8 | delete |
| kerala | kottayam | vazhappally | mariya | john | 1984-01-01 | married | 9877897897 | mariya@gmail.com | mariya h | 12 | 3 | delete |
| kerala | kottayam | madappally | linta | mathew | 1984-01-20 | married | 8767656676 | linta3@gmail.com | linta house | 12 | 2 | delete |
| kerala | kottayam | madappally | meenu | madhav | 1988-11-29 | married | 6768866798 | meenu@gmail.com | meenu | 12 | nill | delete |

## ASHAWORKER PAGE



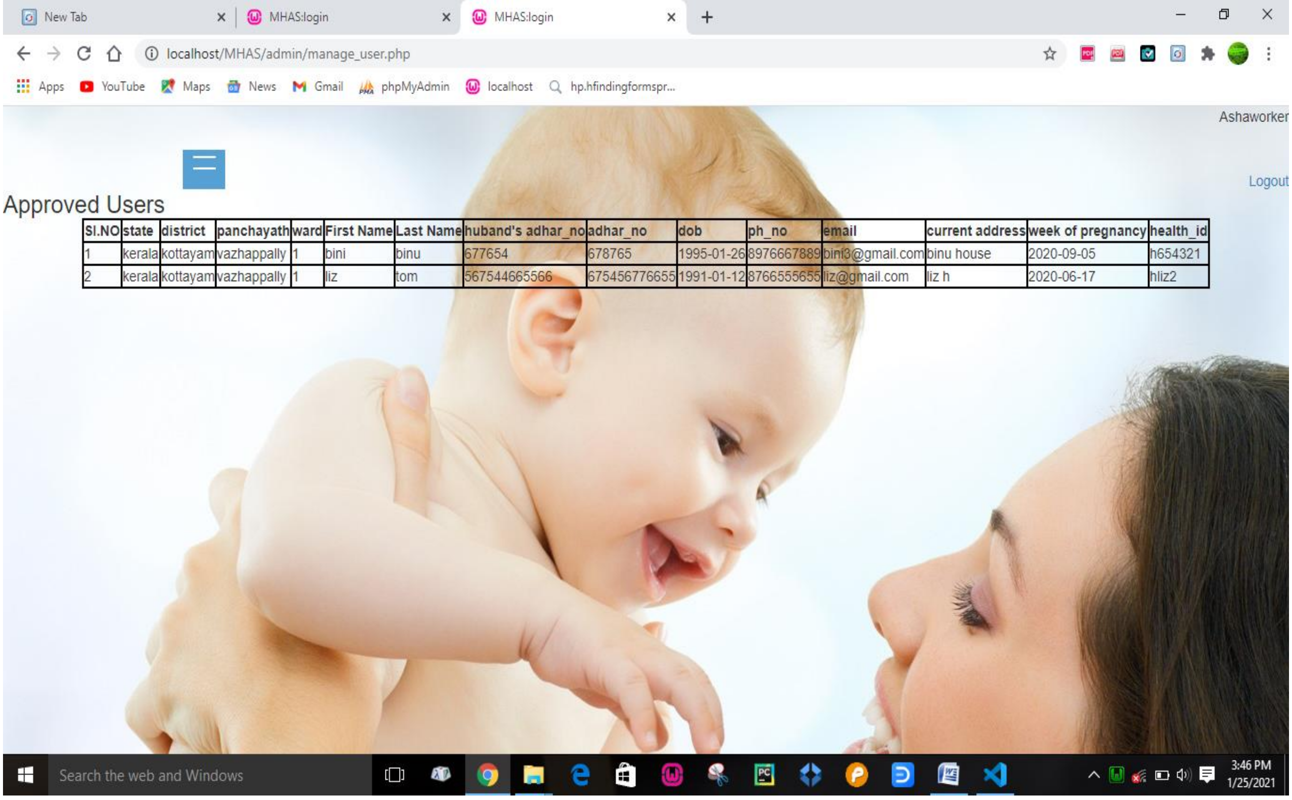
## APPROVE USER



Registered Ashas

| state | district | panchayath | ward | firstname | lastname | dob | marital_status | ph_no | email | address | qualification | experience | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| kerala | kottayam | madappally | 5 | jiya | jeorge | 1990-02-21 | married | 7806667677 | jiya@gmail.com | jiya h | 12 | nill | Approve |

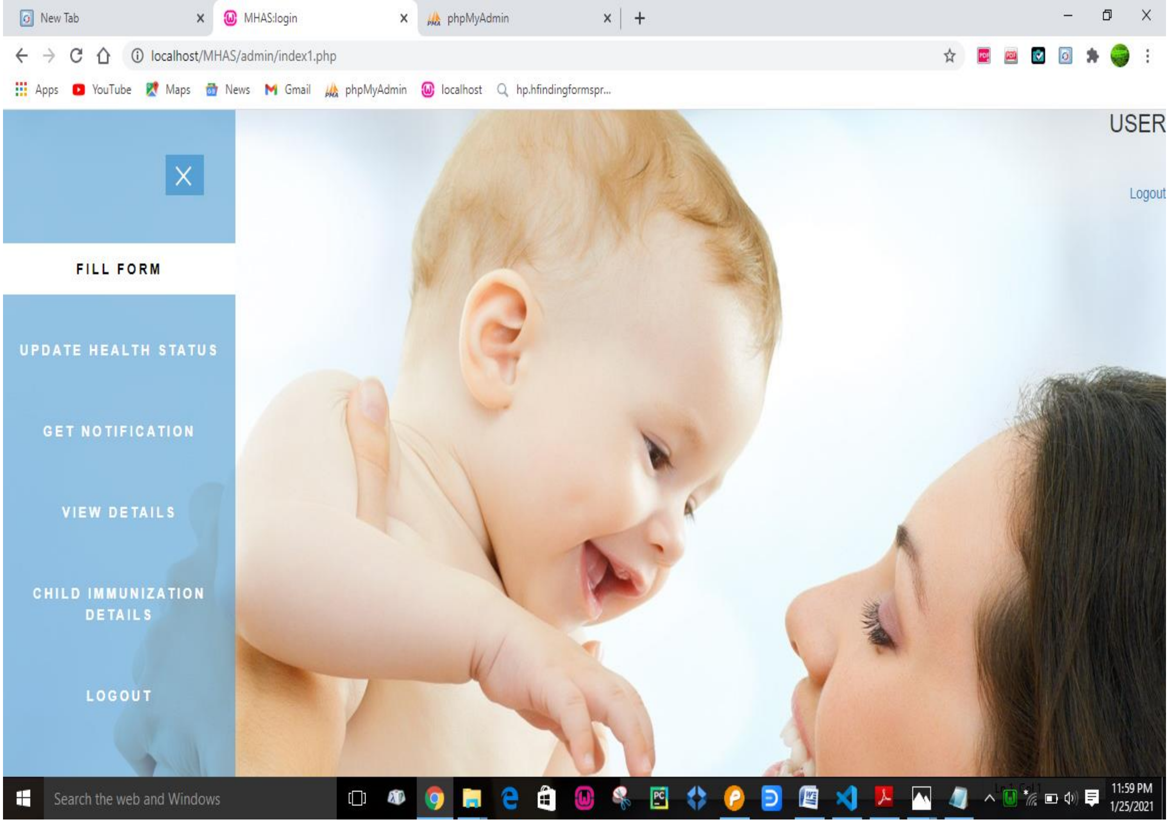
## VIEW USER



Approved Users

| Sl.NO | state | district | panchayath | ward | First Name | Last Name | huband's adhar_no | adhar_no | dob | ph_no | email | current address | week of pregnancy | health_id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | kerala | kottayam | vazhappally | 1 | bini | binu | 677654 | 678765 | 1995-01-26 | 8976667889 | bini3@gmail.com | binu house | 2020-09-05 | h654321 |
| 2 | kerala | kottayam | vazhappally | 1 | liz | tom | 567544665566 | 675456776655 | 1991-01-12 | 8766555655 | liz@gmail.com | liz h | 2020-06-17 | hliz2 |

## USER PAGE


New Tab
MHAS:login
phpMyAdmin
localhost/MHAS/admin/index1.php
Apps
YouTube
Maps
News
Gmail
phpMyAdmin
localhost
hp.hfindingformspr...
USER
Logout
FILL FORM
UPDATE HEALTH STATUS
GET NOTIFICATION
VIEW DETAILS
CHILD IMMUNIZATION DETAILS
LOGOUT
Search the web and Windows
11:59 PM
1/25/2021