# E-PARISH

## PROJECT REPORT

Submitted to the University of Kerala in partial fulfilment of the Degree of Bachelor of Science in Computer Science


**By**

**Jinoy Varghese**

**(Reg.No:32018806023)**

**Devadathan R**

**(Reg.No:32018806016)**

**Arun Ayyappan**

**(Reg.No:32018806010)**

Under the supervision of

**Mrs. Rachana P**




**DEPARTMENT OF COMPUTER SCIENCE**

**MAR THOMA COLLEGE OF SCIENCE AND TECHNOLOGY**

(Affiliated to the University of Kerala)

Chadayamangalam, Kollam (Dist), Kerala-691534


2020

# DECLARATION

We do here by declare that this project entitled **E-PARISH** is a record of independent project work carried out by us under the supervision of the internal guide **Mrs. RACHANA P,** Associate Professor, Department of Computer Science, Mar Thoma College of Science & Technology, Ayur in the fulfilment of the award of B.Sc. Computer Science of the University of Kerala during the academic year of 2020. No part of this has previously formed the basis for the award of any Degree Diploma Associate Ship, Fellowship or other similar titles of this or any other University of Society.

**Jinoy Varghese** **(Reg.No:32018806023)**

**Devadathan R** **(Reg.No:32018806016)**

**Arun Ayyappan** **(Reg.No:32018806010)**

**Place: Ayur**

**Date:**

**Mrs. Rachana P**

Associate Professor

**Department of Computer Science**

Mar Thoma College of Science and Technology,
Chadayamangalam(P.O),Ayur
Kollam-691534

## Certificate

This is to certify that the project entitled E-PARISH is an authentic report of the project done by **Jinoy Varghese(Reg.No:32018806023),Devadathan R(Reg.No:32018806016), Arun Ayyappan (Reg No:32018806010)** in partial fulfilment of the requirement for the award of the Degree in B.Sc Computer Science of the University of Kerala during the academic year of 2019-2020 under my supervision and guidance.

Place : Ayur

Date :

**Mrs. Rachana P**

(Internal guide)

**MAR THOMA COLLEGE OF SCIENCE AND TECHNOLOGY**

**CHADAYAMANGALAM(P.O), AYUR**

**DEPARTMENT OF COMPUTER SCIENCE**

**Certificate**

**Certified that this report titled……**

**is a bonafide record of the project work done by**

**Sri/Kum…….** **Reg. No…….**

**Sri/Kum…….** **Reg. No…….**

**Sri/Kum…….** **Reg. No…….**

**Under our supervision and guidance, towards partial fulfilment of the requirement for the award of the Degree of B.Sc (Computer Science) of the University of Kerala.**

Internal Guide:

(Dated signature) HOD

Date of Examination…….

## Examiners:

# ACKNOWLEDGEMENT

We would like to thank the almighty **God**, for showering all his heavenly blessings upon us for the successful completion of our project work **E-PARISH.**

We are thankful to our Principal **Prof. Dr.Joseph Mathai**, Mar Thoma College of Science and Technology, Ayur for providing us all facilities during our course of study.

We are thankful to **Mrs. Shanu Thomas**, Head of the Department of Computer Science, Mar Thoma College of Science and Technology ,Ayur for the valuable guidance and support throughout the project work.

We express our affectionate respect and heartful thanks to internal guide **Mrs. Rachana P**, Associate Professor, Department of Computer Science, Mar Thoma College of Science and Technology, Ayur who had guided us with a great deal of patience.

Last but not least, we express our deepest gratitude to our beloved parents, classmates and other teachers for their constant encouragement for the completion of this project.

# ABSTRACT

The E-Parish is a web based application which can manage the entire functionalities of a CHURCH. This E-Parish has all the record that were related with a church such as the death, birth, accounting, family details, chat, programs held etc. This website has 4 modules: admin, registered user, employee and guest user. The admin can only create the employees and the employees can create the registered users(family). The guest user can directly view the website without any authentication. The registered user can update their personal details. The employee can only manage the account section and also he can update civil registration in the absence of admin. The admin can update the programs held and the programs that are going to held. There is a chat box which can be accessed by anyone except the guest user so that everyone can express their ideas and view.

**Modules:**

- **Administrator**
- **Employee**
- **Registered User**
- **Guest User**

# CONTENTS

# 1.INTRODUCTION

## 1.1 ABOUT THE PROJECT

The E-Parish is a web based application which can manage the entire functionalities of a CHURCH. This E-Parish has all the record that were related with a church such as the death, birth, accounting, family details, chat, programs held etc. This website has 4 modules: admin, registered user, employee and guest user. The admin can only create the employees and the employees can create the registered users(family). The guest user can directly view the website without any authentication. The registered user can update their personal details only. The employee have only the rights to manage the account section and also he can update civil registration in the absence of admin. The admin can update the programs held and the programs that are going to held. There is a chat box which can be accessed by anyone except the guest user so that everyone can express their ideas and views.

## 1.2 MODULE DESCRIPTION

The modules included in this system are:

### 1.Administrator

Administrator is the overall controller of the system. He can performs,

- Handle the Employee
- Upload programs and event details
- View complaints and feedbacks
- View accounts and reports
- Upload Baptism, Marriage, Death and Image
- Edit Profile
- Chat
- Approve and reject the applied certificate

### 2.Employee

Employee can register into the system. After the registration, the employee can login by using their username and password. They can performs,

- Update Profile
- Update Baptism, Marriage, Death in the absence of Vicar
- Handle the Registered user
- View Baptism, Marriage, Death, Image, accounts, reports and programs
- Chat
- Update accounts and reports

**3.Registered User**

Registered user can register into the system. After the registration, the registered user can login by their username and password and performs,

- View Baptism, Marriage, Death, Events, Accounts, Images
- Check and update personal details
- Chat
- Posting complaints and feedback
- Apply for Baptism, Marriage, Death certificates
- Download or print the approved certificates

**4.Guest User**

- View Vicar details, Images, Programs, Events
- Donation
- Complaints and feedback posting

## Advantages

- Members can apply for certificate by online
- Get church notifications
- Communicate between church members
- Less time consuming
- Preview of account book

# 2.SYSTEM STUDY AND ANALYSIS

## 2.1 INTRODUCTION

During our project we went through the different system development life cycle. First of all we started with system study which helped us to understand the scope of the system. During this phase we were able to understand the limitations of the existing system and it also helped us in realizing the requirements from the client's perspective. The web application E-Parish will provide to bring out the different activities in Church under one click. Many similar websites exists.

## 2.2 EXISTING SYSTEM

In the existing system, all process done manually and the copy of certificates are not provided through online. The security level provided by the current system is very low.

## 2.3 DRAWBACKS OF EXISTING SYSTEM

- Certificate not provide through online.
- No communication with members.
- Accounts and reports not published
- No online donation

## 2.4 PROPOSED SYSTEM

E-PARISH is a web application developed using PHP and MariaDB Server. This Web application is for Online Church which provides facilities like donation to church, interact with members etc. This software package also allow to know different programs in church.

## 2.5 ADVANTAGES OF PROPOSED SYSTEM

- User friendly and interactive.
- Better Service.
- Minimum time needed for the various processing.
- Security of data.
- Greater efficiency
- Quick Access
- Fast Response

## 2.6 FEASIBILITY STUDY

The main aim of feasibility study is to determine whether it would be functionally and technically to develop the product. The feasibility study involves the analysis of the collection of relevant information relating to the product such as the different data item which would be the input to the system, The processing required to be carried out on these data, the output data required to be produced by the system, as well as various constraints on the behaviour of the system. A feasibility study is a rest of the system proposal according to its working, impact on the organization, ability to meet users and effective use of resources. The objective of feasibility study is acquiring the sense of scope of the system.

The development of a computer based system is more likely to be projects that are feasible. Three essential factors are involved in the feasibility analysis are:

- Technical feasibility
- Economic feasibility
- Operational feasibility

### Technical feasibility

Technical feasibility focuses on the possibility of doing the project with the current equipment, existing software technologies available personnel. The assessment of this feasibility must be based on the study of system requirements in terms of input, output, programs, procedures, and staff having identified an outline of system. It also includes the possibility of expanding the system developed and technical guarantees of accuracy, reliability, case of access and also the data security,The technical needs were taken into consideration to see if the project is feasible with the current resources available.

### Economic feasibility

The cost to conduct a full system implementation is very low. The proposed system can be developed with the existing resources. There is no need for any additional hardware or software requirements for both developing and implementation. The cost of other resources needed for development is minimum. The technologies to be used to develop the system were so choosen to minimize cost. Most modification can be done easily with less effort. Since the system is developed as apart of project work, there is no manual cost of spending for the proposed system. Hence the proposed system is feasible.

**Operational feasibility**

One of the main problems faced during development of a new system is getting acceptance from user. Even if a system is technically and economically feasible but the users of the system are resistant to use it then there is no use. In this stage the following issues are considered.

- Is the proposed system is user friendly?
- Is there sufficient support for the project from the management and users?
- Will the proposed system cause harm?
- Will it proposed poorer result in any area?
- Will loss of control result in any area?

The proposed system is so effective, user friendly and functionally reliable that the users will find that the new system reduced their effort. Since the users are very much involved in planning and development of the project. The result produced is accurate and optimized. The proposed system will have good control on all parts of the Church and it will take care of current activities.

## 2. 7 DATA FLOW DIAGRAM (DFD)

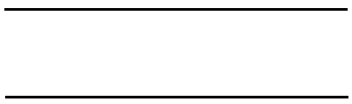
DFD are the most commonly used way of documenting the process of flow and required system. As their name suggests, they are a pictorial way of showing flow of data into, around the system. DFD was introduced by Demacro, Gane and Sarson. Data Flow Diagrams are constructed with four major components. They are :

**Data Flow Diagram Symbols**

**1.Entities**

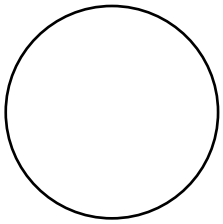
External entities represent the sources of data that enter the system or recipients of data that leave the system.

**2.Data Store**



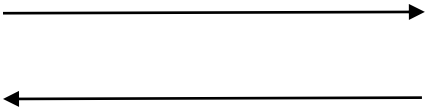
Data Store is represented by using two parallel lines. It represents a logical files.

**3.Process**



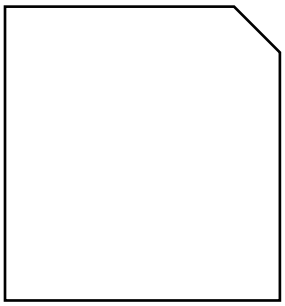
Process represent active in which data is manipulated by being stored or retrieved and transformed in some way. A circle represents it.

**4.Data Flows**



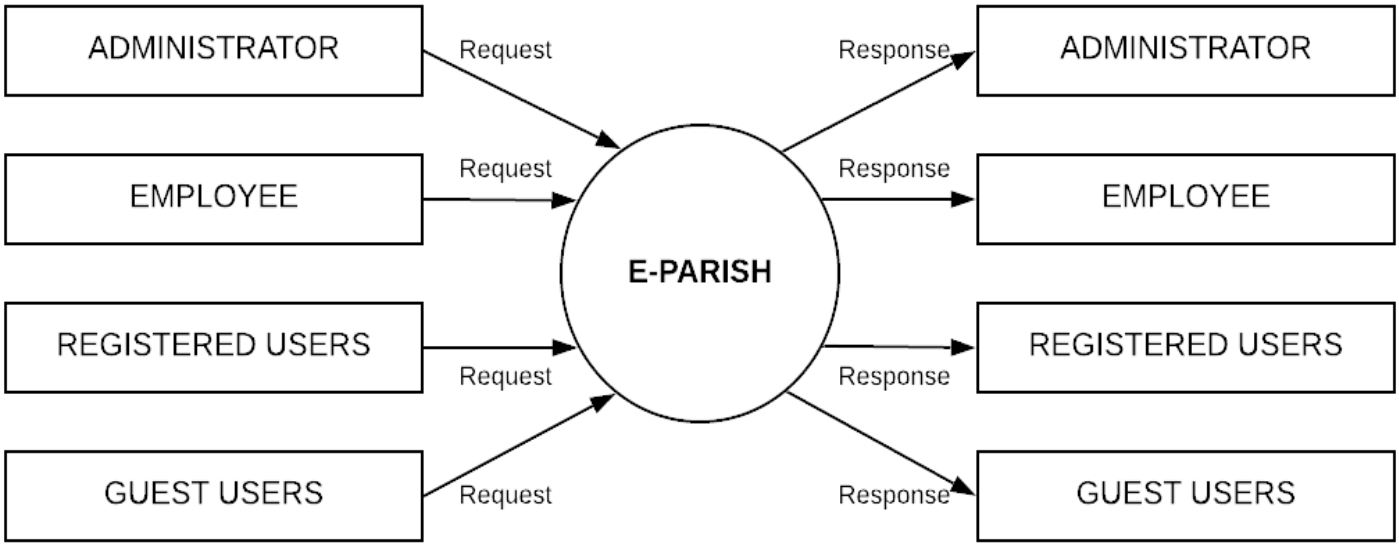
Data flow show the flow of information from its source to its destination's line represents data flow, with arrowheads showing the direction of flow.

**5.Output**

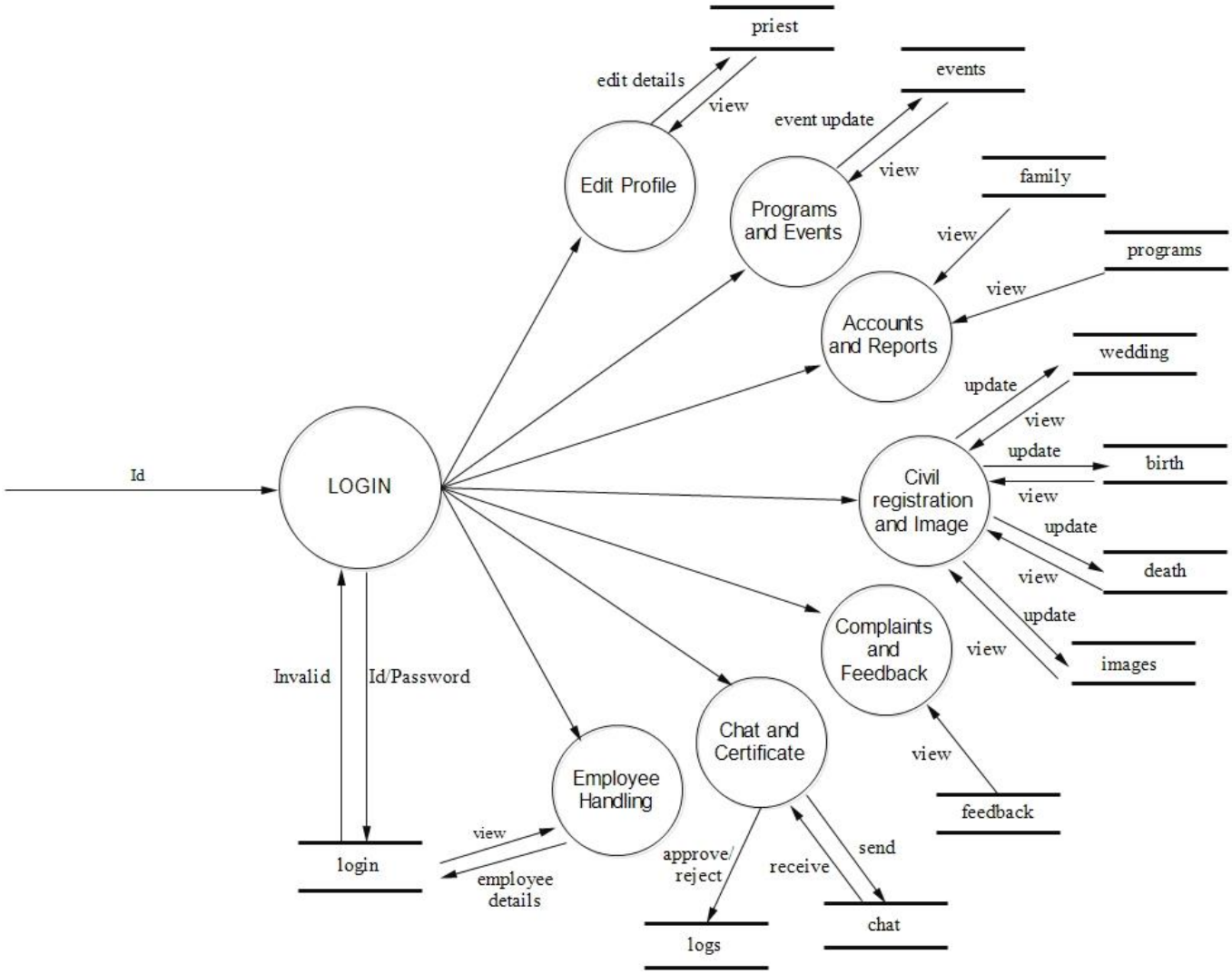


The output shows the output of data that enter the system.

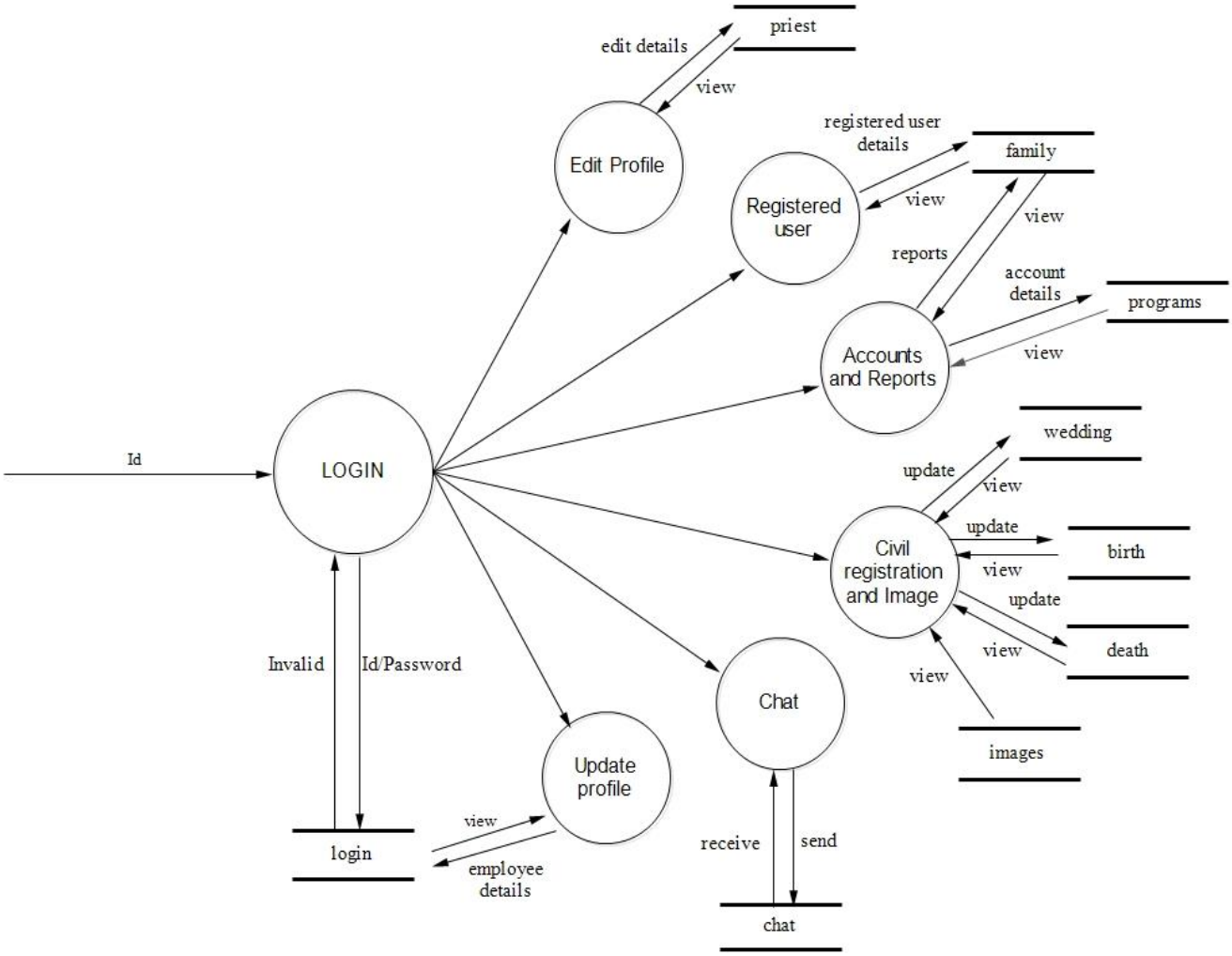
*Level zero DFD : E-PARISH*



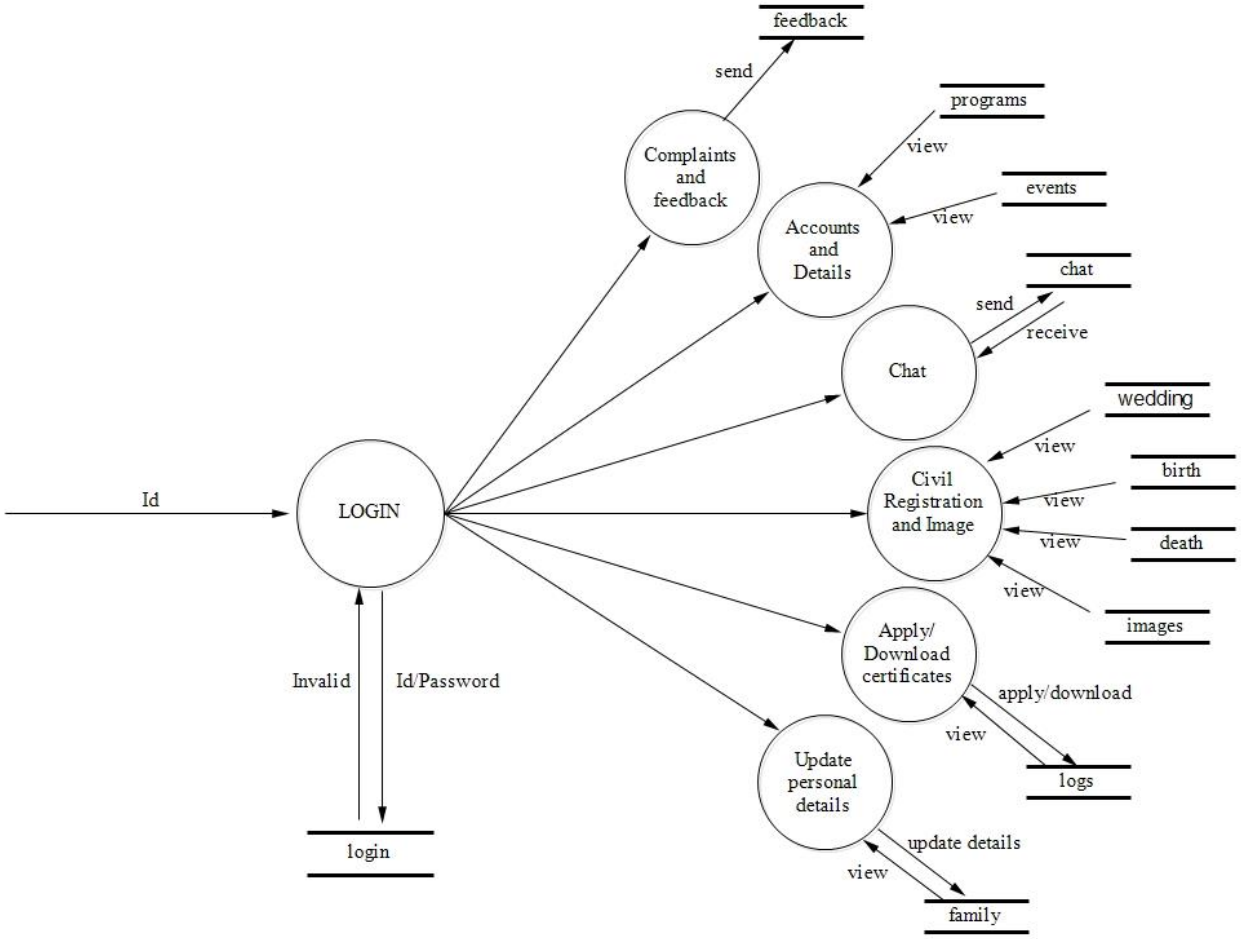
*Level 1 DFD of : Administrator*



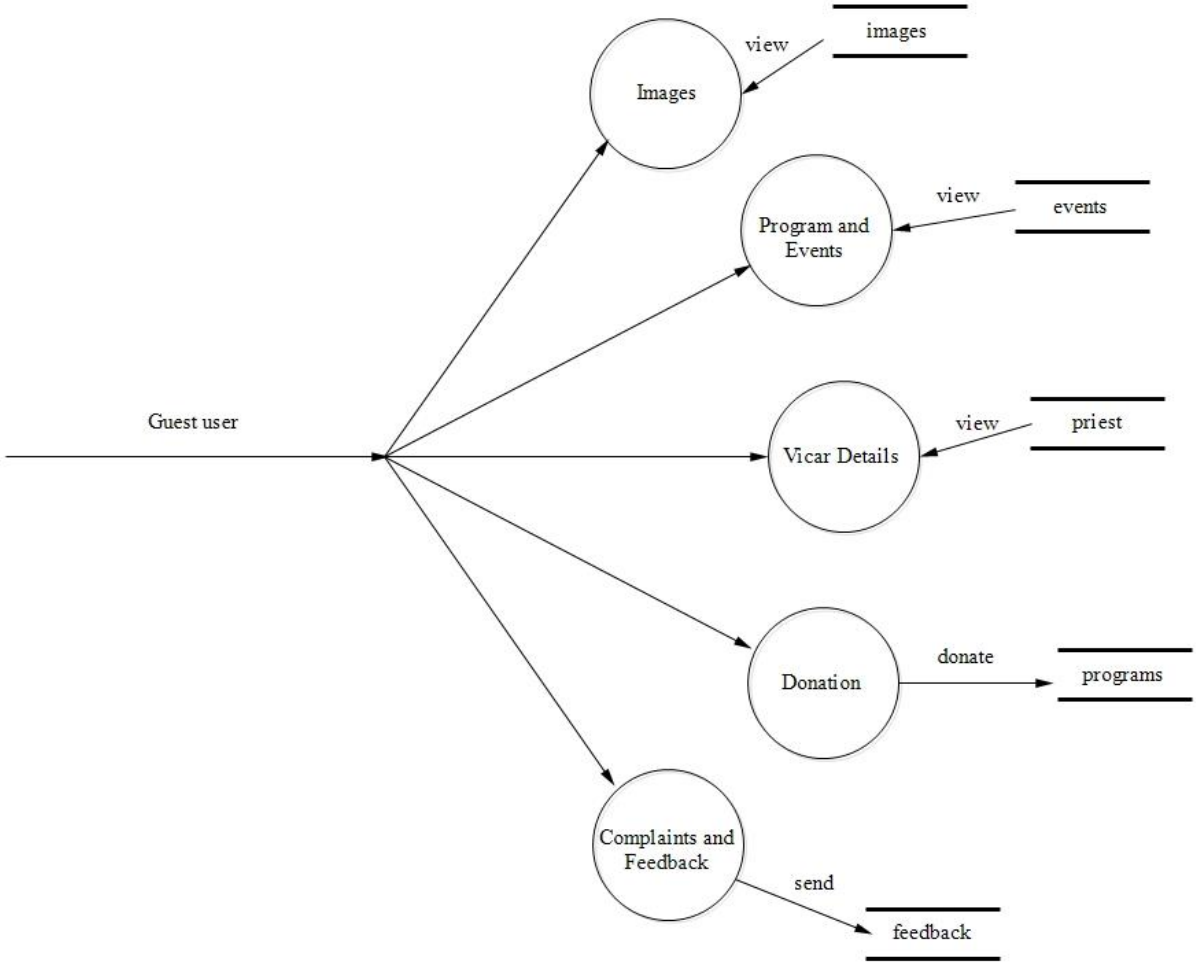
*Level 1 DFD of : Employee*



*Level 1 DFD of : Registered user*



*Level 1 DFD of : Guest user:*



# 3.SYSTEM DESIGN

## INTRODUCTION

System design is the first step in the development phase for many engineered product or system. It may be defined as the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permits its physical realization. This Phase is the first step in moving from the problem domain to the solution domain. It is an iterative Process through which requirements are transmitted into a "blueprint" for constructing the software initially; the blue depicts a holistic view of software. That is design is represented at a high level of abstraction, functional and behavioral requirements.

System design develops the architectural detail required to build a system or product. The system design process encompasses the following activities.

- Partition the analysis model into subsystem.
- Identify the concurrency that is dictated by the problem.
- Allocate the subsystems to processors and tasks.
- Develop a design for the user interface.
- Choose a basic strategy for implementing data management.
- Identify global resources and control mechanisms required to access them.
- Design an appropriate control mechanism for the system including task management

System design provides an understanding of the procedural details, necessary for implementing the system recommended in the feasibility study. Basically it is all about the creation of the new system. This is critical phase since it decides the quality of the system. It has a major impact on the testing and implementation phase.

System design is the most creative and challenging phase of the system life cycle. The term design describes the final system and the process by which is to be developed. During the system design phase the designers must design how to produce an efficient and effective system. There are two levels of system design: Logical design and physical design.

In the logical design, the designer produces a specification of the major features of the system which meets the objectives. The delivered product of logical design includes current requirements of the system components: Input design, Output design and Database design.

Physical design takes this logical design blueprint and produces the program specifications. Design specifications instruct programmers about what the system should do. Structured design is data flow based methodology that partitions a program into a hierarchy of modules organized top-down manner with details.

### 3.1 INPUT DESIGN

The input design is the process of converting the user-oriented inputs in to the computer based format. The goal of designing input data is to make automation as easy and free from errors as possible. The input design requirements such as user friendliness, consistent format and interactive dialogues for giving the right message and help for the user at right time are also considered for the development of the project.

Inaccurate input data is the most common cause of error in processing data. Errors entered by the data entry operators can be controlled by the input design. The arrangement of messages as well as placement of data, headings and titles on display screens or source document is also a part of input design. The design of input also includes specifying the means by which end user and system operators direct the system what action to take. The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps that are necessary to put transaction data into a usable form for processing data entry.

In this project, all the necessary text boxes are validated. The input forms are designed in Visual Studio Code. If any non-empty fields are not filled, it will display an error message and will wait until user types the necessary and correct input. Initially, to access the services of this software, the user has to log on with a login name and password which are validated. Once logged on, he can access the various services, navigate to different profiles.

## 3.2 OUTPUT DESIGN

Output generally refers to the results and information that are generated by the system. When designing output, system analyst must accomplish the following.

- Determine what information to present.
- Decide whether to display, print the information and select the output medium
- Arrange the presentation of information in an acceptable format.
- Decide how to distribute the output to intended recipients.

The output design is specified on layout forms, sheets that describe the location characteristics, and formats of the column heading and pagination. In this project, output forms are designed in PHP. Each form has a heading or caption which specifies what services is been given to the users for making the software user-friendly. All information are stored in the database and when anyone logs on and request for a service, the corresponding page is fetched from the server after validation and is rendered.

## 3. 3 DATA BASE DESIGN

A relational database is a collection of data items organized as a set of formally described tables from which data can be accessed or reassembled in many different ways without having to recognize database tables. The RDB was invented by E.F Codd at IBM in 1970.

An RDBMS is a program that lets you create, update, and administer a relational database. Most commercial RDBMS use the SQL to access the data base, although SQL was invented after the development of relational model and is not necessary for its use. A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored data 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 cleanly 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 is transferred into a design for 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 DB design runs parallel with the system design. The organization of data in the database is aimed to achieve two major objectives. They are:

- Data independence
- Data integrity

The data base design is made up of three levels

1) Conceptual level (High Level)

2) Physical level (Low Level)

3) View level(Representation level)

### 3. 3. 1 DATA NORMALIZATION

Normalization is the process of efficiently organizing data in a database. Two goals or normalization are: eliminate redundant data and ensure data dependencies make sense. Both these goals reduce the amount of space a database consumes and ensure that data is logically stored. The database commonly has developed a series of guidelines for ensuring that databases are normalized. These are referred to as normal forms and are numbered from one through five (1NF to 5NF).

**Data in First Normal Form**

- A relation is in first normal form if and only if all its attributes are based upon a single domain

**Data in Second Normal Form**

- A relation r(R) is in second normal form if and only if the following two conditions are not simultaneously

1) r (R) is in First Normal Form
2) No non prime attribute is partially dependent on any key

**Data in Third Normal Form**

- A relation is in Third normal form if and only if the following conditions arc satisfied simultaneously.

1) Is already in second normal form
2) No non-prime attribute is transitively dependent on the key

**Advantages of Normalization**

- Helps in reduction in the complexity of maintaining data integrity by removing the redundant data.
- Reduces inconsistency of data.
- Eliminate repeating fields.
- Create a row for each occurrence of a repeated field.
- Allows exploitation of column functions.

## 3.2 TABLE DESIGN

***1.login***

**Description:** Used to store username and password of users

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| Num | int | 6 | Primary key | User num |
| Id | varchar | 20 | Not null | user id |
| Password | varchar | 32 | Not null | user password |
| Type | varchar | 20 | Not null | user type |

***2.priest***

**Description:** Used to store Vicar details

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| Username | varchar | 20 | Not null | user's name |
| jd | date | Date | Null | join date |
| cbefore | varchar | 30 | Null | previous church |
| dob | date | Date | Null | date of birth |
| image | varchar | 200 | Null | user's image |
| experience | int | 6 | Null | user's experience |
| tfd | varchar | 100 | Null | message from user |
| flag | int | 11 | Null | flag |

### *3.family*

**Description:** Used to store family member's details

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| num | int | 6 | Not Null | family num |
| family | varchar | 30 | Null | family name |
| members | int | 6 | Null | No: of family member |
| membername | varchar | 30 | Null | family member name |
| phno | bigint | 20 | Null | phone number |
| owner | varchar | 20 | Null | family owner name |
| tax | int | 6 | Null | tax |
| due | int | 6 | Null | due |
| donation | int | 6 | Null | donation |
| extra | int | 6 | Null | extra |
| user | varchar | 20 | Null | id |
| total | int | 6 | Null | total |
| date | date | date | Null | date |
| christmas | int | 6 | Null | amount |
| diocese | int | 6 | Null | amount |
| convention | int | 6 | Null | amount |
| treatment | int | 6 | Null | amount |
| gospel | int | 6 | Null | amount |

***4.events***

**Description:** Used to store the event details

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| num | int | 6 | Not null | num |
| name | varchar | 20 | Null | name |
| value | varchar | 100 | Null | value |

***5.birth***

**Description:** Used to store the Baptism details

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| num | int | 6 | Not null | num |
| bname | varchar | 20 | Null | baby name |
| date | date | date | Null | date |
| so | varchar | 20 | Null | father's name |
| id | id | 6 | Null | id |

### *6.wedding*

**Description:** Used to store marriage details

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| bride | varchar | 20 | Null | bride name |
| date | date | date | Null | marriage date |
| groom | varchar | 20 | Null | groom name |
| id | int | 6 | Not null | id |
| so | varchar | 20 | Null | father's name |
| wid | int | 6 | Null | wedding id |

### *7.death*

**Description:** Used to store death details

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| num | int | 6 | Not null | num |
| dname | varchar | 20 | Null | name of person |
| date | date | date | Null | death date |
| so | varchar | 20 | Null | father's name |
| id | int | 6 | Null | id |
| age | int | 11 | Null | age |

### *8.chat*

**Description:** Used to store the messages sent

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| num | int | 6 | Not null | num |
| chat | varchar | 400 | Null | message |
| name | varchar | 20 | Null | name |

### *9.images*

**Description:** Used to store the images

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| id | Int | 6 | Not Null | id |
| image | varchar | 100 | Null | uploaded image |

### *10.programs*

**Description:** Used to store the account and report details

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| name | varchar | 20 | Null | name |
| credit | int | 6 | Null | credits |
| debit | int | 6 | Null | debit |
| total | int | 6 | Null | total |
| num | int | 6 | Not null | num |

### *11.logs*

**Description:** Used to store the applied certificate details

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| num | Int | 6 | Not null | num |
| des | Varchar | 20 | Null | Category of certificate |
| status | Varchar | 20 | Null | certificate status |
| ldate | Date | Date | Null | applied date |
| id | int | 20 | Null | id |

### *12.feedback*

**Description:** Used to Store the feedback send in the website

| Column | Datatype | Size | Constraints | Description |
|---|---|---|---|---|
| num | int | 6 | Not null | num |
| name | varchar | 20 | Null | name of sender |
| phno | bigint | 20 | Null | phone number |
| email | varchar | 30 | Null | email |
| msg | varchar | 200 | Null | message |

## 3.3 LANGUAGE OVERVIEW

### PHP

Is a server-side scripting language designed for web development but also used as a general- purpose programming language. Originally created by Rasmus Lerdorf 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 Pre-processor, 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 an 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 graphical applications. PHP is free software released under the PHP License, which is 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.

The PHP interpreter only executes PHP code within its delimiters. Anything outside its delimiters is not processed by PHP. The most common delimiters are <?php to open and"> to close PHP sections. <script language="php"> and </script> delimiters are also available, as are the shortened forms <? or <?= and ?> as well as ASP-style short forms <% or <%= and %>. WW1e short delimiters are used, they make script files less portable as support for them can be disabled in the PHP configuration, and so they are discouraged. The purpose of all these delimiters is to separate PHP code from non-PHP code, including HTML.

PHP, which stands for “Hypertext Pre-Processor”, is a server-side; HTML embedded scripting language used to create dynamic Web pages. Much of its syntax is borrowed from C. Java and Perl with some unique features thrown in. The goal of the language is to allow Web developers to write dynamically generated pages quickly.

### Advantages of PHP

- Cost is low
- PHP is an open source software

- PHP is easy to learn
- PHP is embedded within HTML

The HTML-embedding of PHP has many helpful consequences

- PHP can quickly be added to code produced by WYSIWYG editors.
- PHP lends itself to a division of labour between designers and scripters.
- Every line of HTML does not need to be rewritten in a programming language.
- PHP can reduce labour costs and increase efficiency due to its shallow learning curve and ease of use.
- PHP has Cross-platform compatibility
- PHP is not tag-based
- PHP is much faster for almost every use than CGI scripts.
- PHP makes it easy to communicate with other programs and protocols.
- PHP is fast becoming one of the most popular choices for so-called two-tier development.
- PHP is developed and supported in a collaborative fashion by a worldwide community of users.

**Hyper Text Transfer Protocol (HTTP)**

HTTP is the protocol "spoken" by web servers. Client programs that can speak I-ITT P. known as browsers, are used by the people on the Internet to connect to HTTP servers. The servers provide access to distributed hyper linked documents, applications and databases. HTTP is a stateless. object oriented application level protocol that has been in the existence since the early days of the WWW. NSCA HTTP is a HTTP/1.0 compliant web server and is credited with being one of the first HTTP servers available. It supports multiple schemes of authentication.

**Html - The Frame Work For Webpages**

Hypertext Mark-up Language (HTML) is the text mark-up language on the World Wide Web. The mark-up commands applied to the web based content tell the browser software the structure of document and, when appropriate, how we want the content to be displayed. It has a well-defined syntax and HTML documents have a formal structure. With the introduction of scripting languages such as Javascript, the concept of dynamic HTML (DHTML) is becoming more and more popular and is used to create highly interactive web pages. When browser reads a

document that has html markup in it, it determines how to render it on screen by considering the html elements embedded within the document.

**jQuery**

jQuery is a library of javascript Functions. jQuery is a lightweight "write less, do more" Javascript library. The jQuery library is stored as a single javascript file, containing all the jQuery methods.

The jQuery library contains the following features

- HTML element selections
- HTML element manipulation
- CSS manipulation
- HTML event functions
- Javascript Effects and animations
- HTML DOM traversal and modification
- Utilities.

**CSS**

- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- Styles were added to HTML 5. 0 to solve a problem
- External Style Sheets can save a lot of work External Style Sheets are stored in CSS
- All browsers support CSS today.

**VISUAL STUDIO CODE**

Visual Studio Code is a source code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js and C++. Instead of a project system it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language, contrary to Microsoft Visual Studio which

uses the proprietary solution file and project-specific project files. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many of Visual Studio Code features are not exposed through menus or the user interface, but can be accessed via the command palette. Visual Studio Code can be extended via extensions.

**DATABASE**

A database is a separate application that stores a collection of data. Each database has one or more distinct AP is for creating, accessing, managing, searching, and replicating the data it holds. Other kinds of data stores can be used, such as files on the file system or large hash tables in memory.

A database system must provide following features

- A variety of user interfaces.
- Physical data independence.
- Logical data independence.
- Query optimization.
- Data integrity.
- Concurrency control.
- Backup and recovery.
- Security and authorization

**MariaDB**

MariaDB is a community-developed, commercially supported fork of the MySQL relational database management system (RDBMS), intended to remain free and open-source software under the GNU General Public License. Development is led by some of the original developers of MySQL, who forked it due to concerns over its acquisition by Oracle Corporation in 2009. Information is stored in "Tables" which can be thought of as the equivalent of Excel spreadsheets. A single database can contain many tables at once and store thousands of individual records.

The features of MariaDB server are:

- MariaDB is licenced under GPL, LGPL, or BSD.
- MariaDB includes a wide selection of storage engines, including high-performance storage engines, for working with other RDBMS data sources.
- MariaDB uses a standard and popular querying language.
- MariaDB runs on a number of operating systems and supports a wide variety of programming languages.
- MariaDB offers support for PHP, one of the most popular web development languages.
- MariaDB offers Galera cluster technology.
- MariaDB also offers many operations and commands unavailable in MySQL, and eliminates/replaces features impacting performance negatively.

## 3.4 SELECTION OF TOOLS (S/W,H/W REQUIREMENT)

### Minimum Hardware Requirements

| | | |
|---|---|---|
| Processor | : | Celeron |
| RAM | : | 1 GB |
| Hard Disk | : | 80GB |
| CD-ROM | : | 200 MB |
| Keyboard | : | Standard 101/102 key |
| Mouse | : | Optical mouse |
| Monitor | : | Plug and Play monitor |
| Printer | : | Ink jet |

### Software Specification

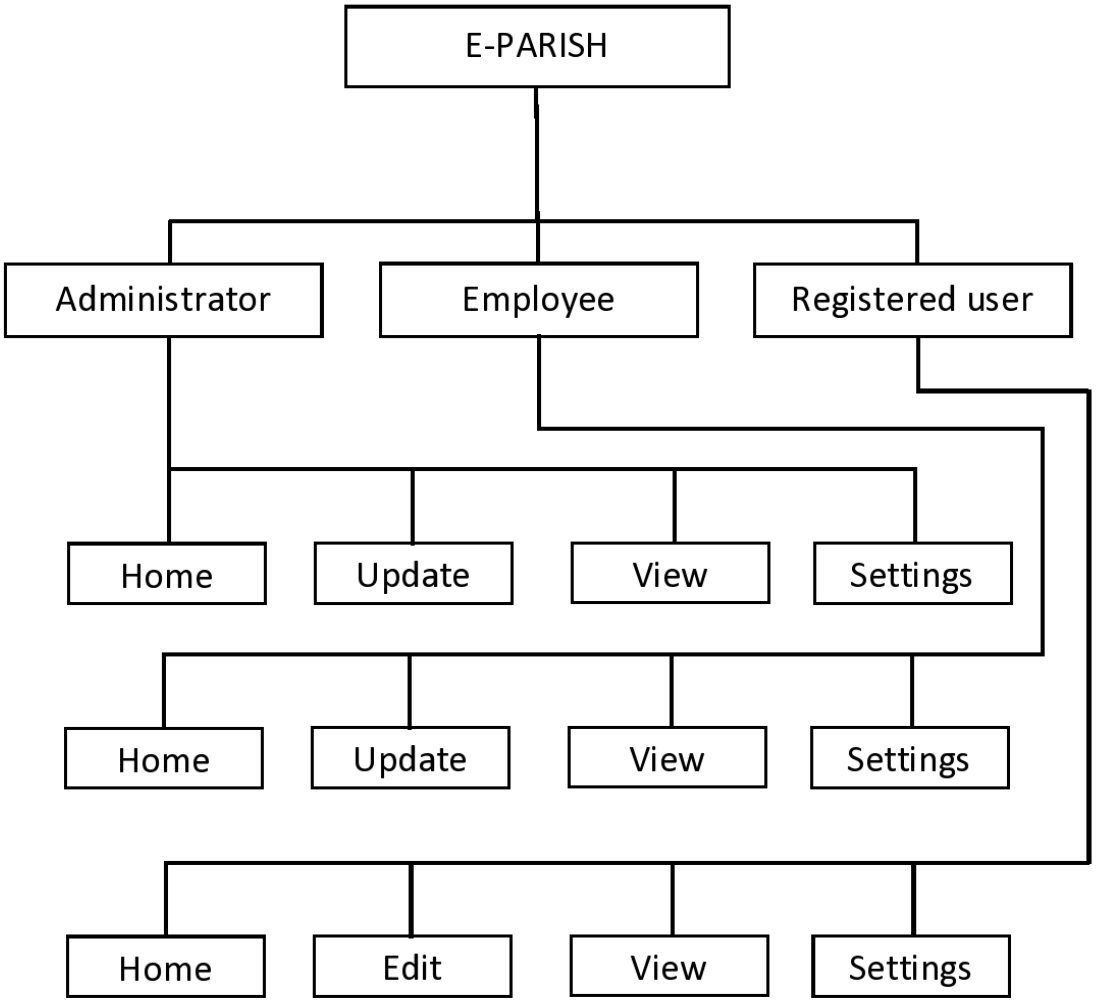
| | | |
|---|---|---|
| Operating System | : | Windows 7 & Above |
| Front End | : | PHP |
| Back End | : | MariaDB |
| Browser | : | Any browser with HTML 5 support |

### Hosting Details

| | | |
|---|---|---|
| Host | : | infinityfree.net |
| Mail | : | migadu.com |
| Storage | : | Unlimited |
| Bandwidth | : | Unlimited |
| Storage Device | : | SSD(Solid State Drive) |

| | | |
|---|---|---|
| Protection | : | Cloudflare |
| Domain | : | eparish.ml |
| Domain Name Provider | : | freenom.com |
| TLS Version | : | 1.3 |
| IP Address | : | 185.27.134.217 |

### 3.5 MENU TREE


E-PARISH
Administrator
Employee
Registered user
Home
Update
View
Settings
Home
Update
View
Settings
Home
Edit
View
Settings

# 4.SYSTEM TESTING AND IMPLEMENTATION

## 4.1 SYSTEM TESTING

Testing is the process of executing a program with the intent of finding any errors. A good test of course has the high probability of finding a yet undiscovered error. A successful testing is the one that uncovers a yet undiscovered error. A test is vital to the success of the system. System that makes logical assumptions that if all parts of this system are correct, then foal will be successfully achieved. The candidate system is subjected to variety of tests online like responsiveness, its value, stress and security. System testing can be broadly classified into:

- Black box testing
- White box testing
- Unit testing
- Integration testing
- Validation testing

**Black Box Testing**

When computer software is considered b1ack box testing alludes to tests that are conducted at the software interface. A black box test examines some fundamental aspects of a system with little regard for the internal logical structure of the software. Black box testing attempts to find errors in the following categories :

- Incorrect or missing function
- Interface errors
- Performance errors
- Errors in data structures or external database access
- Initialization and termination errors

In our application, we use a number of functions to perform operations. Using the black box testing we made sure that all functions are executing correctly by giving the required result.

**White Box Testing**

It is a testing method that uses control structure of procedural design to derive testing. Knowing the internal working of a product tests can be conducted to ensure that the internal operations performs according to specification and all internal components have been adequately

exercised. White box testing of software is predicated on close examination of conditions and/or loops tests logical paths through the software. Using this testing method, the software engineer can do tests that :

Guarantee that all independent paths within a module have been exercised at least Once.

- Exercise all logical decisions on their true and false values.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure their validity.

Here all logical structures are tested in their true and false conditions. We also made sure that all loops are performing well at their boundaries. For the checking appropriate data inputs are given and they are processed correctly. Individual functions are tested separately for each of the above conditions.

**Unit Testing**

This is the first level of testing. Here different functions used in the software development are split into different modules and tested to see whether they satisfy our needs. Code produced during the coding phase of the software development process and the internal logic of the module is tested here. After coding each function was tested individually. The logical errors found were corrected.

**Integration Testing**

This is systematic technique for constructing the structure while conducting tests to uncover errors with interfacing. Here the different functions of software are combined into sub system, which are again tested. The various unit tested functions of the software were integrated and rigorous integration testing was conducted to make the application free of any interface errors that may occur. In this phase various functions are combined. Once the individual functions were tested, we tested the control hierarchy in a top down integration manner.

**Validation Testing**

It provides the final assurance that the software meets all functional, behavioural and performance requirements. Then software changed for the better performance. When the application was made free of all logical and interface errors, validation testing was conducted by

inputting dummy data to ensure that the software developed satisfied all the requirements of the user. This includes providing various valid and invalid inputs.

System tests carried out to validate dully developed system with a view assuring that it meets its requirements. There are essentially three kinds of system testing:

**1. Alpha Testing**

It refers to the system testing that is carried out by the test team within the organization.

**2. Beta Testing**

Beta testing is the system testing performed by a selected group of friendly customers.

**3.Acceptance Testing**

Acceptance testing is the system testing performed by the customer to determine whether or not to accept the delivery of the system. The application is tested to ensure the requirements. Different sets of input data are entered to validate the system. In all cases the system produces the reasonable output.

## 4.2 SYSTEM IMPLEMENTATION

Implementation is the process of converting a new or revised system design into operation. It is the key stage in achieving a successful new system because, usually it reveals a lot of up heal. It must therefore be carefully planned and controlled. Apart from planning the two major tasks of preparing for implementation are education and training of users and testing of the system. Education of users should really take place much earlier in the project, Training has to be given to the web masters regarding the new system. Implementation is the stage of project where the theoretical design is turned into working system or it is the key stage in achieving a successful new system. Therefore it must be carefully planned and controlled. It can also be considered to be the most crucial stage in achieving a successful new system and in giving the user confidence that the new system and in giving the user confidence that the new system will work and be effective.

Implementation is the final and important phase. It is the phase where theoretical design is turned into working system, which works for the user in the most effective manner. It involves careful planning, investigation of the present system and the constraints involved, user training, system testing and successful running of developed proposed system. The implementation process begins with preparing a plan for the implementation of the system. According to this plan the

activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system, The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of a system using various kinds of data. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

### 4.3 FUTURE ENHANCEMENT

Enhancement means adding, modifying or developing the code to support the changes in the specification. It is the process of adding new capabilities such as report, new interface without other systems and new features such as better screen or report layout. Every module in the system is being developed carefully stich that the future enhancements do not affect the basic performance of the system. In future we can add any links or services to the System very easily. Moreover, due to limited time allotted for the project, there are features, which I couldn't implement. Thus the system offers the scope of future enhancement. As this software is reliable to use, any modification in accordance with the necessity of the user can be done for the future use. Any additional feature can be implemented very easily. So what we call this software also a user friendly. Some of the future developments that can be incorporated in this software are

- In current system tender verification process is semi computerized we can implement it total computerized.
- This application is implementing all over the world.

# 5. SYSTEM MAINTENANCE

## 5. SYSTEM MAINTENANCE

It is possible to produce systems of any size which do not need to be changed. Over the lifetime of a system, its original requirements will be modified to reflect the changing user.

After implementation, maintenance is the important process. Usually once the system is implemented, the software developers and customer would sign a contract. According to the time mentioned in the contract all errors and requirements would be charged. During the contract period we would frequently visit the site where the system is implemented and check the system performance such as response time and also how it works at peak hours. If any problem is found, it is corrected.

The four types of maintenance activities are listed below.

### Corrective Maintenance

This is concerned with fixing reported errors in the software. Coding errors are relatively cheap to correct; design errors are more expensive as they may involve the rewriting of several program components. Requirements errors are the most expensive to repair because of the extensive system redesign which may be necessary.

### Adaptive Maintenance

Adaptive maintenance means changing the software to some new environment such as different hardware platform or for use with different operating system. The software functionality does not radically change.

### Perfective Maintenance

This involves implementing a functional or non-functional system requirement. These are generated by software consumers as their organization or business changes.

### Preventive Maintenance

This occurs when software is changed to improve future maintainability or reliability or to provide a better basis for future enhancement. In this project, all the above maintenance were implemented.

# 6.SECURITY MECHANISMS

# 6. SECURITY MECHANISMS

This project provides some security features. We can implement application security in the developing system. After registering details, they have to authenticate by providing the appropriate username and password. Securities are provided in this project that the data remains confidential. We can implement security through username and passwords. Username and password facility is implemented to avoid unauthorized access of information. For providing more security they have to change their passwords with their wishes. And it is more secure to change password periodically and it must be kept confidentially. A combination of alphabets, numbers and special characters make a password strong.

# 7.UPGRADABILITY POSSIBILITIES

# 7. UPGRADABILITY POSSIBILITIES

The technology is changing day to day. The efficiency of developed system can be improved by modifications. The quality of this online site can be improved by keeping wise list of reports and other documents effectively. So it is easy to add or remove modules. Software development in PHP is very flexible and all application was tested with live data and has proved respond successful. So it is quite and helps in smooth migration from manual system to computerized system.

Our project "E-PARISH" is a project that contain only four modules such as administrator, employee, registered user, and guest user. It is easy to develop and upgrade. It is compatible with any future developments.

# 8.CONCLUSION

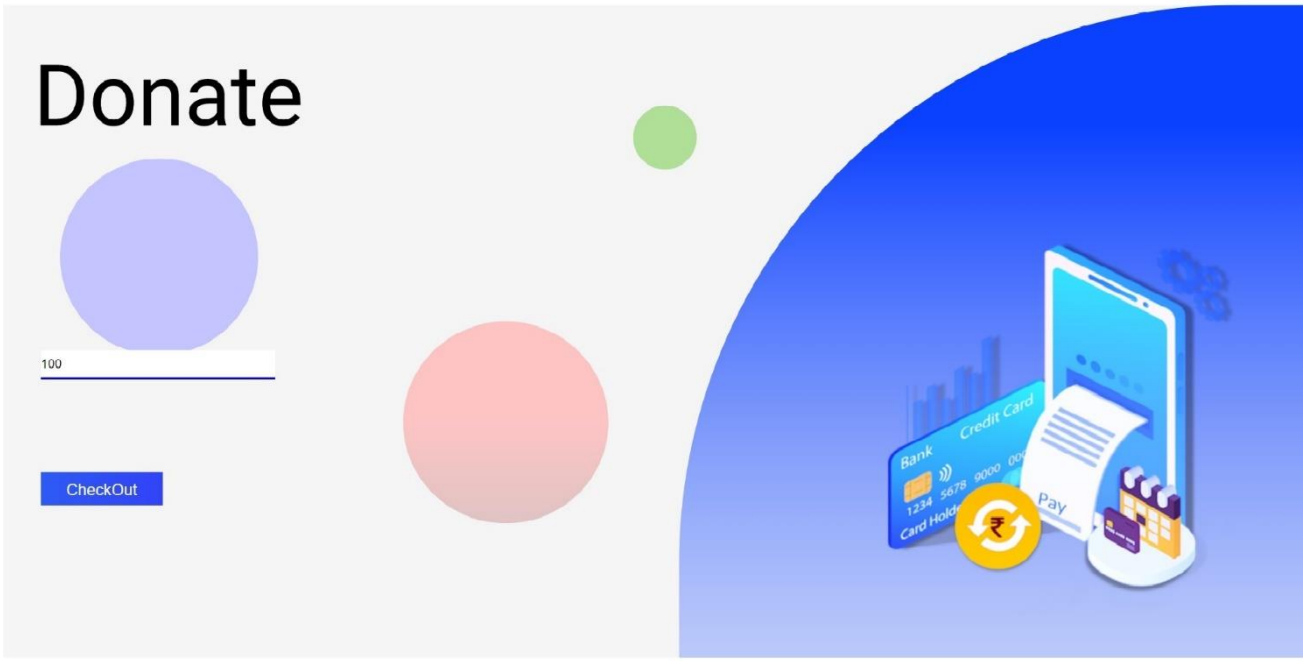
# 8. CONCLUSION

The new system has overcome most of the limitations of the existing system and works according to the design specification given. The developed systems dispense the problem and meet the needs of by providing reliable and comprehensive information. All the requirements projected by the user have been met by the system. The newly developed system consumes less processing time and all the details are updated and processed immediately. since the screen provides online help messages and is, very user-friendly, any user will get familiarized with its usage. Modules are designed to be highly flexible so that any failure occur that can be easily solve without facing many problems.

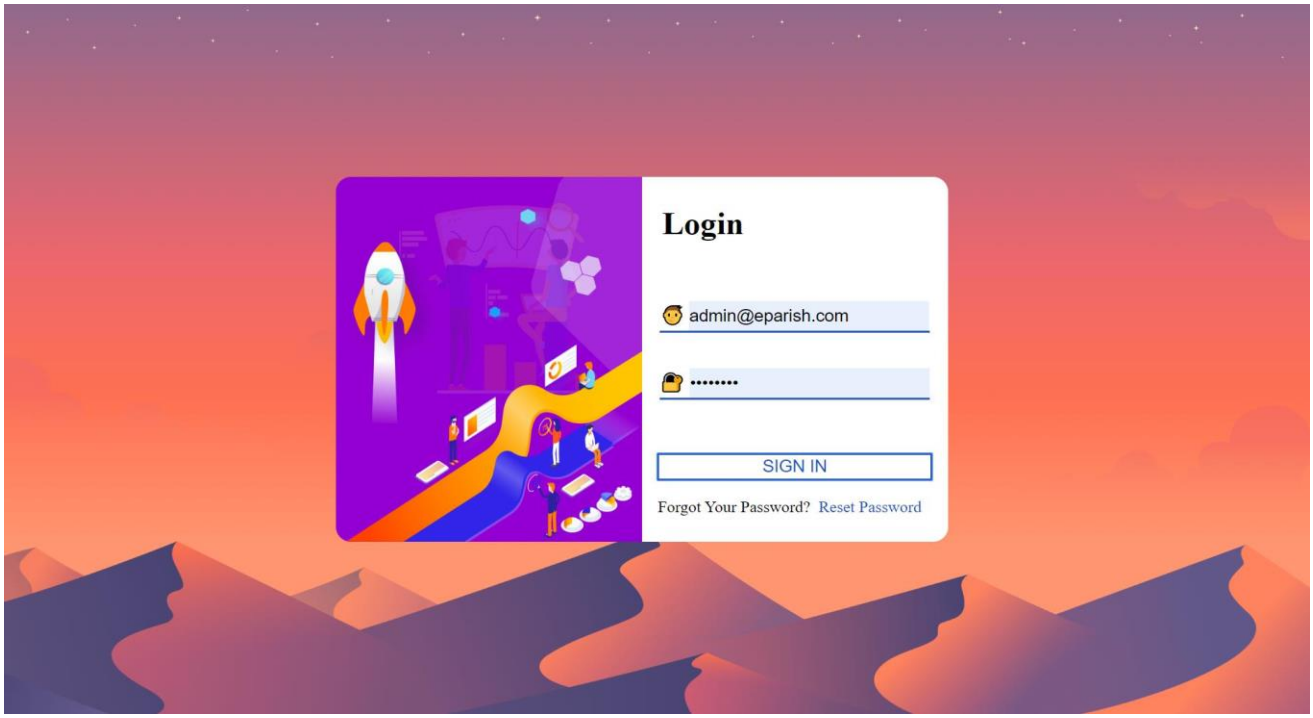
# 9.APPENDIX

# 9.1 APPENDIX A-FORM LAYOUTS

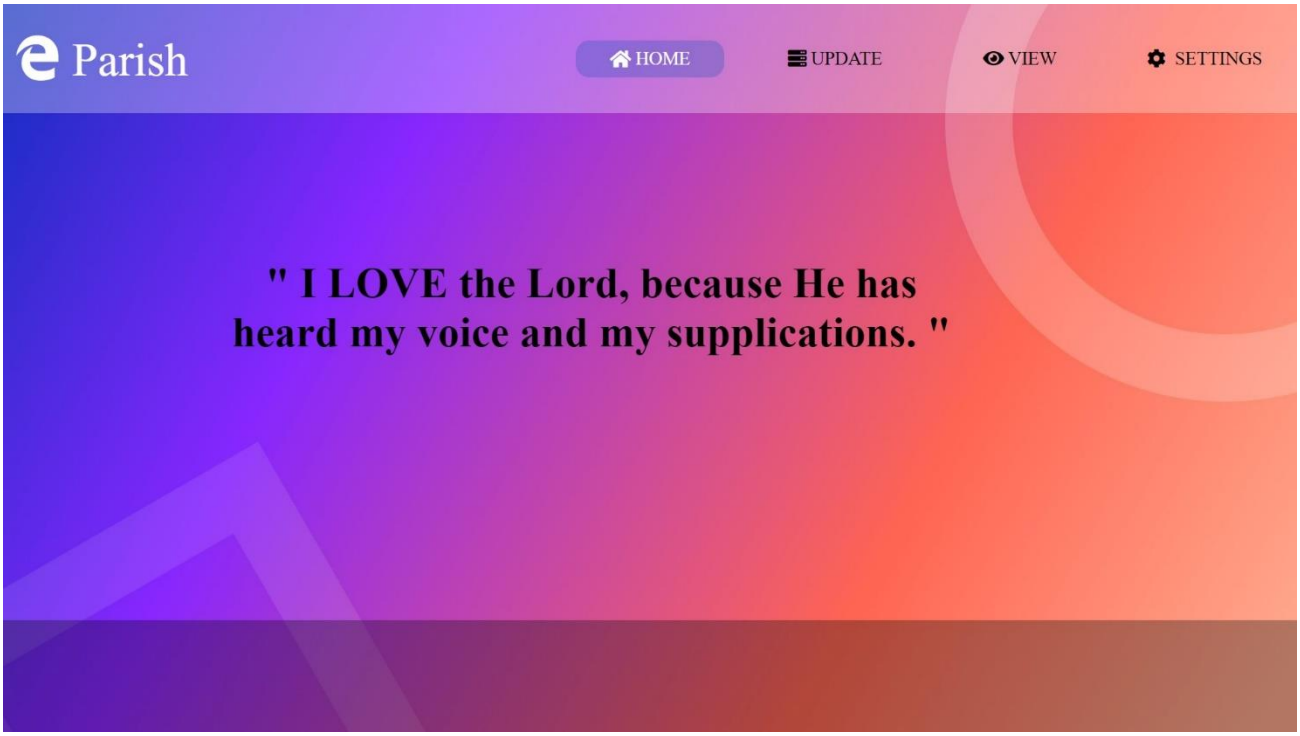

*1.Home page*



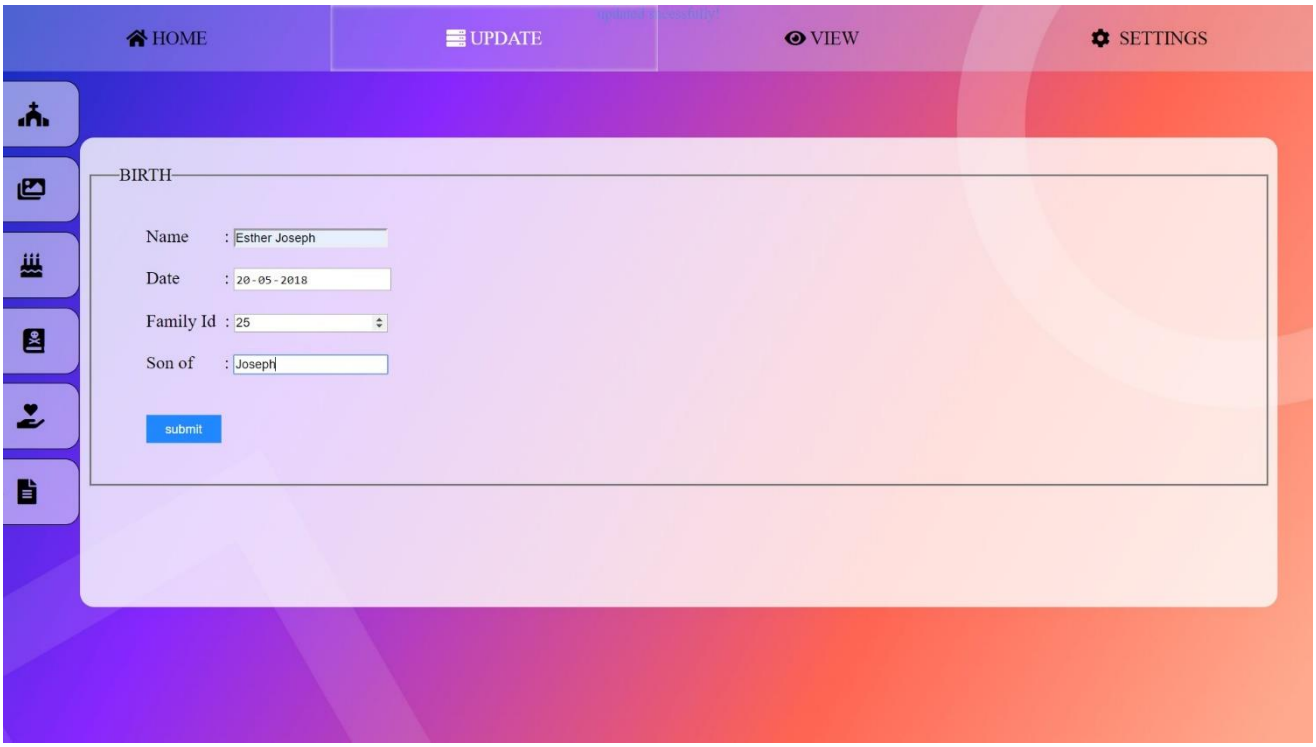

*2.Donation*



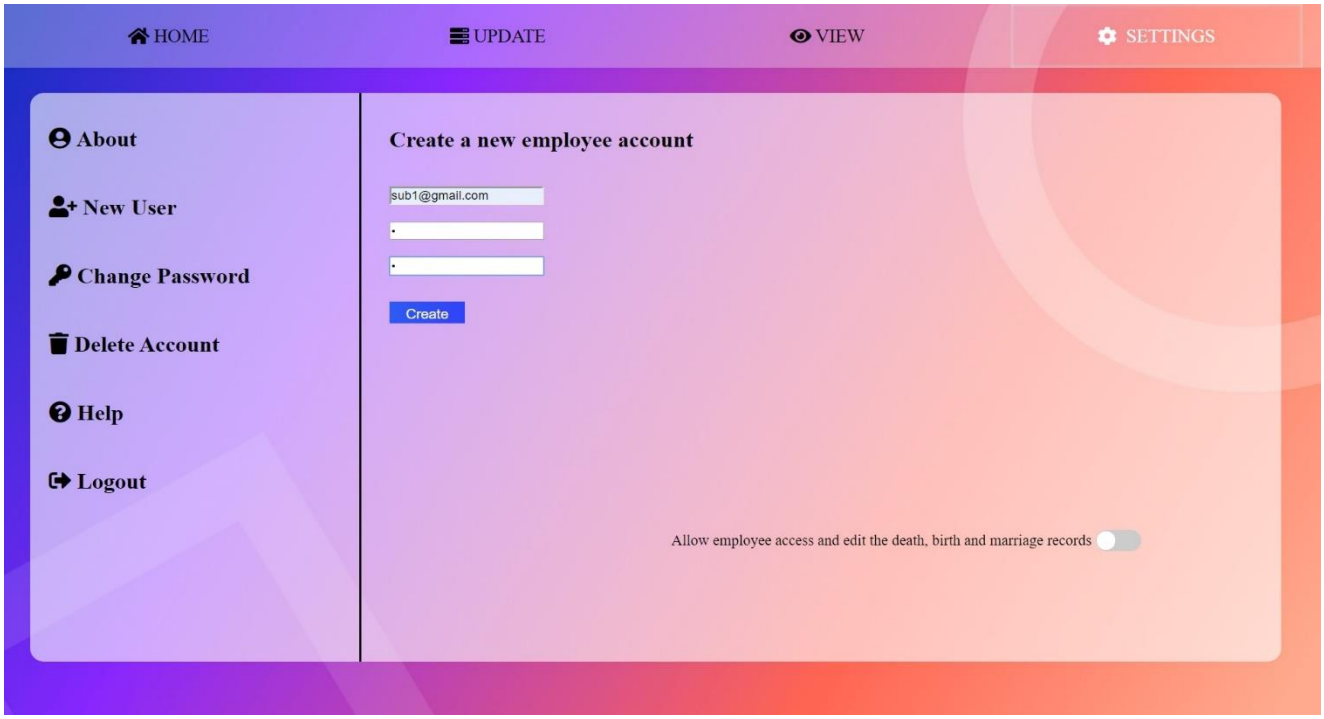

*3.Login Page*



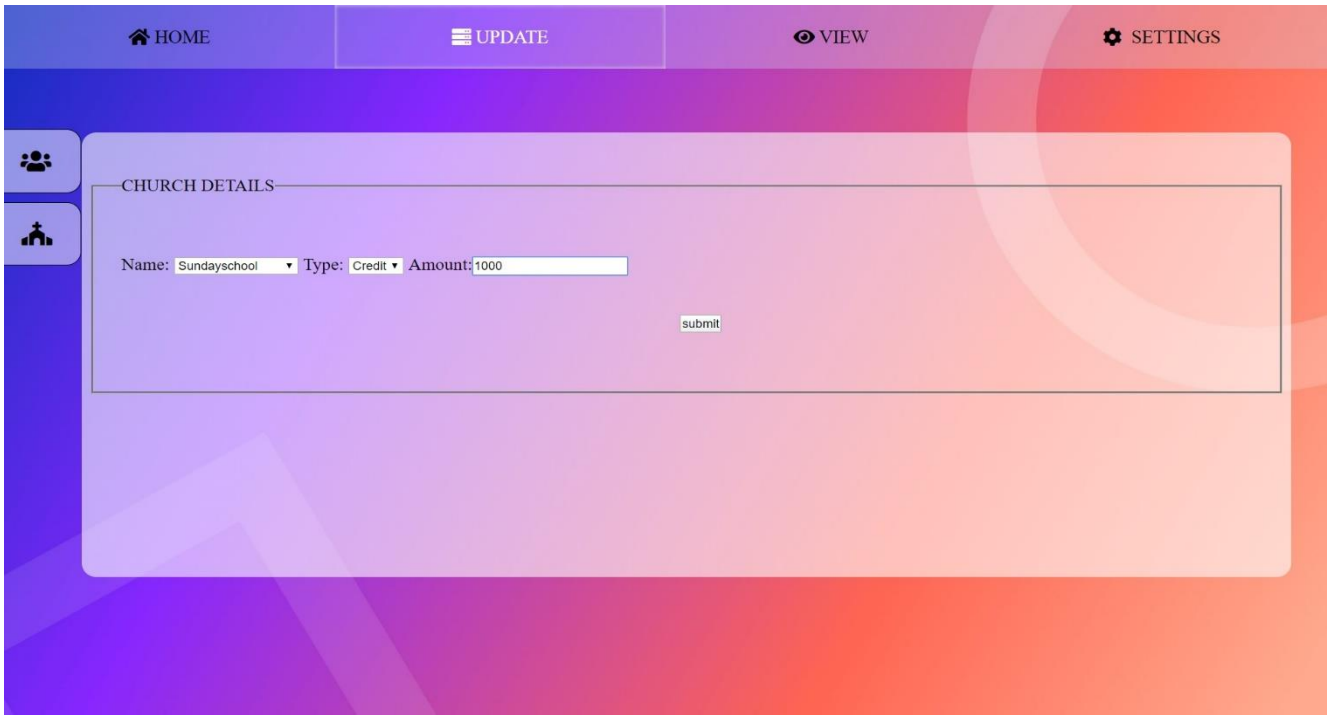

*4.Home Page*



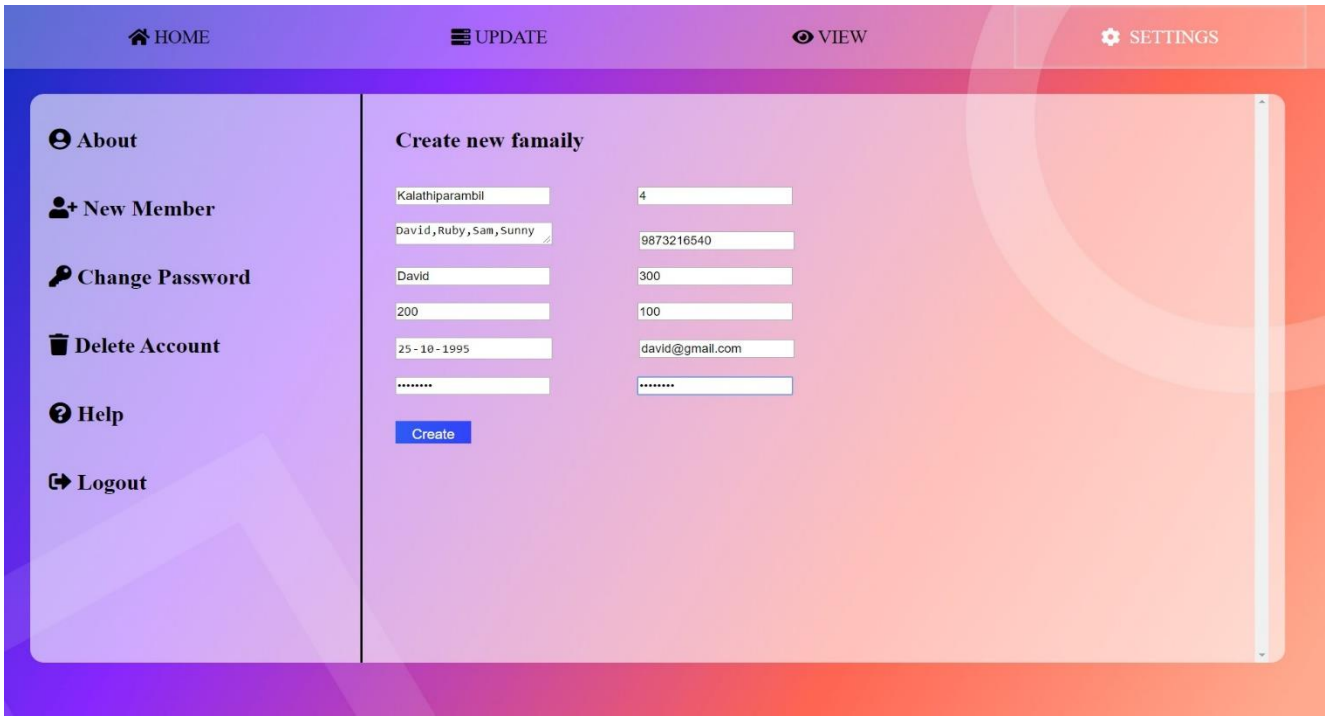

*5.Civil Registration*



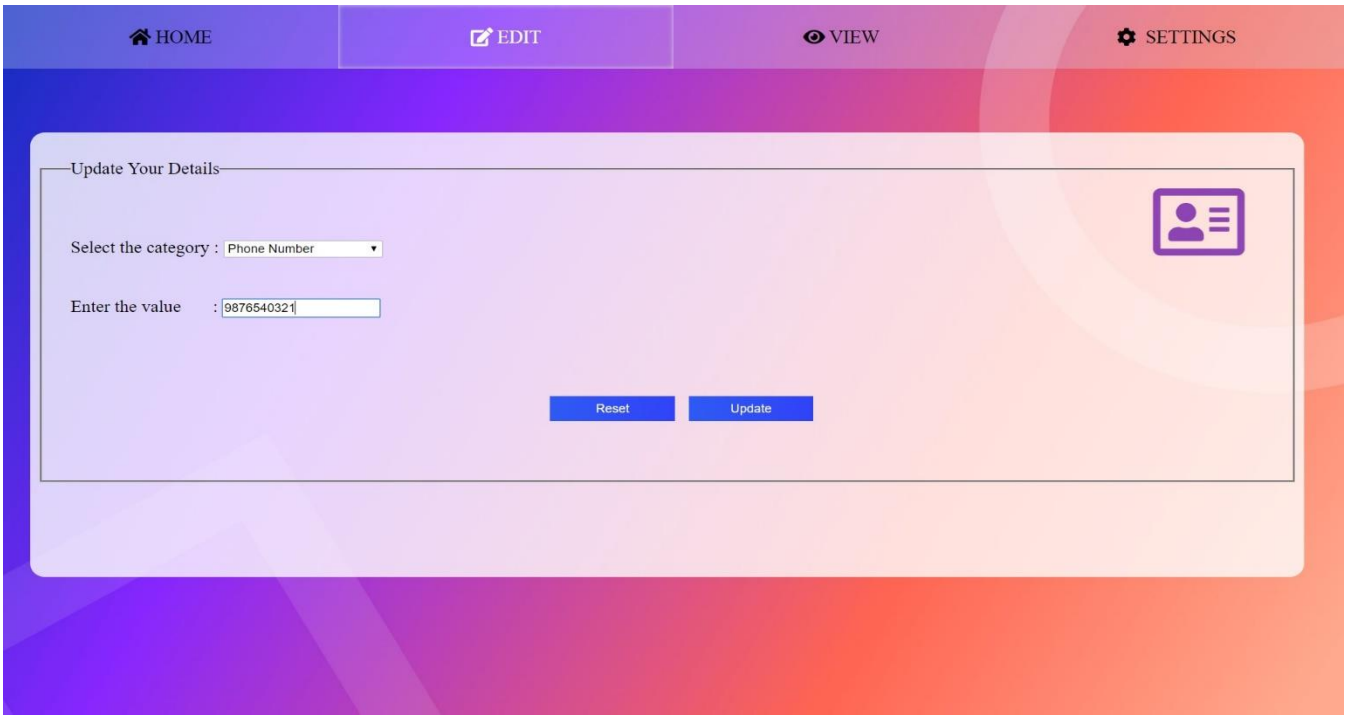

*6. Add Employee*



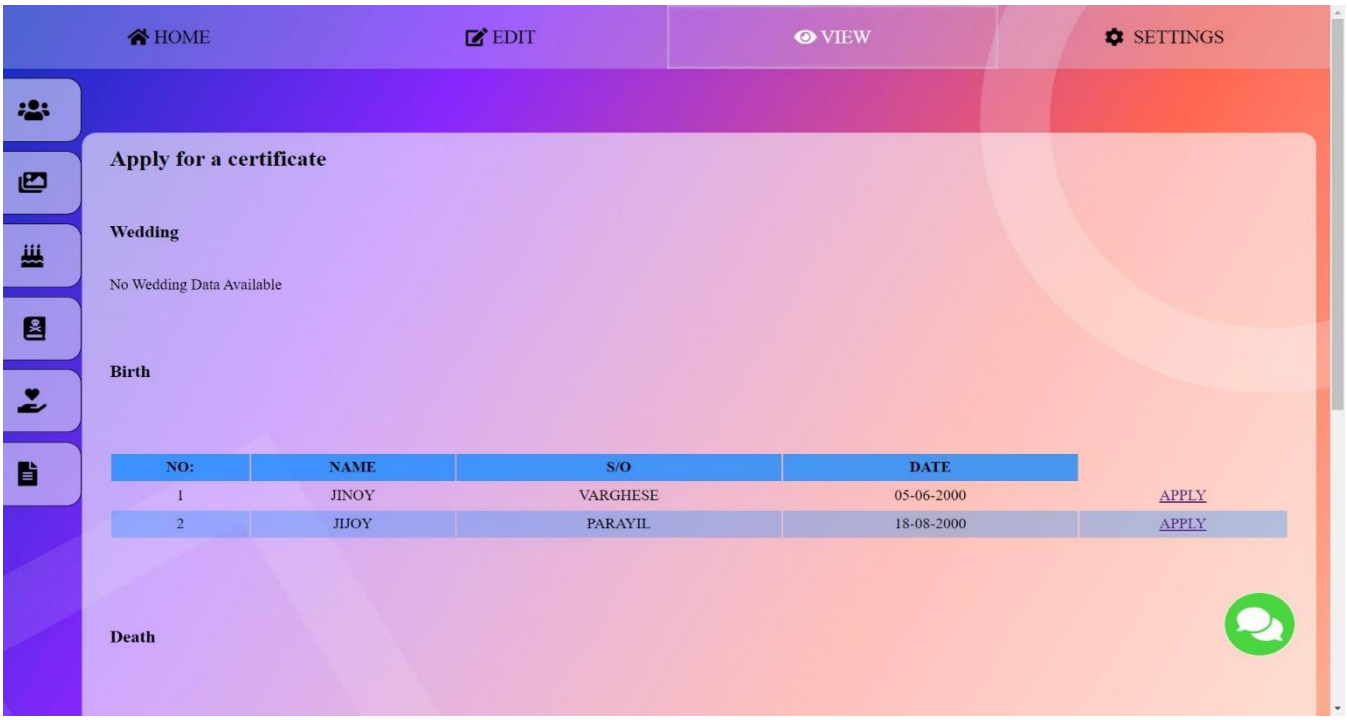

*7.Add Accounts Details*




*8.Add Family*




*9.Edit Family*




*10.Apply Certificate*

# 9.2 APPENDIX B-SYSTEM CODING

**eparish.php**

```
<?php
include "config.php";
if($_SERVER["REQUEST_METHOD"]=="POST"&&isset($_POST['send']))
{
$name=$_POST['name'];
$contact=$_POST['contact'];
$email=$_POST['email'];
$msg=$_POST['msg'];
if(empty($name))
{
echo "<center>Enter your name</center>";
}
elseif(empty($contact))
{
echo "<center>Enter your contact</center>";
}
elseif(empty($email))
{
echo "<center>Enter your email</center>";
}
elseif(empty($msg))
{
echo "<center>Enter your message</center>";
}
else
{
$sql="INSERT INTO feedback(name,phno,email,msg)values('$name','$contact','$email','$msg')";
if($conn->query($sql))
{
echo "<div class=update><center><font >Feedback submitted
sucessfully!</font></center></div>";
}
else
{
echo "<div class=update><center><font >Feedback submition failed!</font></center></div>";
}
}
}
?>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link href="fontawesome-free-5.11.2-web/css/all.min.css" rel=stylesheet>
<script src=jquery.txt></script>
<link rel="shortcut icon" type="image/png" href="images/eedge.png"/>
<link href="https://unpkg.com/aos@2.3.0/dist/aos.css" rel="stylesheet">
```

```
<title>eParish</title>
<style>
*
{
margin:0;
padding: 0;
}
a
{
text-decoration: none;
}
.left-msg
{
width:50vw;
float:left;
}
.right-msg
{
width:50vw;
float:right;
height:440px;
}
.vicar h1,.footer h1
{
margin-top:-100px;
margin-left:50px;
}
.vicar h1::after,.footer h1::after
{
content:'';
background: -webkit-linear-gradient(144deg, rgba(255,0,255,1) 0%, rgba(237,0,255,1) 27%,
rgba(189,1,255,1) 65%, rgba(154,3,255,1) 100%);
display:block;
height:3px;
width:130px;
margin:10px auto 50px;
position:absolute;
left:50px;
}
.vicar ul
{
margin-top:30px;
margin-left:120px;
list-style-type: none;
}
.vicar ul li
{
line-height: 40px;
```

```
font-size:15px;
position: relative;
}
.vicar ul li::after
{
content:'';
height:8px;
width:8px;
background: -webkit-linear-gradient(144deg, rgba(255,0,255,1) 0%, rgba(237,0,255,1) 27%, rgba(189,1,255,1) 65%, rgba(154,3,255,1) 100%);
transform: rotate(45deg);
position: absolute;
top:17px;
left:-25px;
}
.footer .input1
{
width:300px;
height:30px;
padding:8px;
margin-top:30px;
margin-left:120px;
border: 1px solid black;
}
.footer .input2
{
width:300px;
height:100px;
padding:8px;
margin-top:30px;
margin-left:120px;
border: 1px solid rgb(0, 0, 0);
}
::placeholder
{
font-size: 13px;;
}
.footer-button
{
border:none;
background: -webkit-linear-gradient(144deg, rgba(255,0,255,1) 0%, rgba(237,0,255,1) 27%, rgba(189,1,255,1) 65%, rgba(154,3,255,1) 100%);
padding:5px;
color:white;
margin-top:30px;
margin-left:120px;
border:1px  solid blueviolet;
}
```

```
.footer-button:hover
{
background:rgba(189,1,255,.04);
color:rgb(121, 10, 224);
}
.footer
{
padding-bottom: 60px;
}
.location,.phno,.email
{
padding:8px;
margin-left:32px;
margin-top: 0px;
font-size: 19px;
}
.fa-map-marker-alt,.fa-envelope,.fa-phone
{
margin-left:10px;
margin-top: 10px;
float:left;
font-size: 20px;
}
.imgbox
{
width:450px;
position:absolute;
height:265px;
background-color: blueviolet;
margin-top: -265px;
margin-left: 800px;
}
.vicarimg
{
width:100%;
height: 100%;
}
.footer
{
margin-top: 310px;
height:500px;
}
.footer h1
{
text-align: center;
margin-left: -40px;
}
.footer h1::after
```

```
{
left: 44%;
}
::-webkit-scrollbar
{
display: none;
}
.sticky
{
width:100%;
height:10vh;
position: absolute;
background-color: #0303039d;
z-index: 1;
position: absolute;
display:flex;
align-items: center;
}
.fab
{
font-size:40px;
color:white;
}
.logo
{
margin-left:12px;
}
.img
{
width:100%;
height:89vh;
background:url("back1.jpg");
background-size:cover;
background-attachment: fixed;
display:flex;
justify-content: center;
align-items: center;
font-size: 70px;
font-family:cursive;
padding:20px;
color:white;
}
.login
{
position: absolute;
border: 2px solid white;
font-size: 20px;
color:white;
```

```
border-radius: 15px;
width:80px;
text-align: center;
margin-left: 92%;
text-decoration:none;
}
.menu1
{
color:white;
width:65px;
text-decoration:none;
font-size: 23px;
position: absolute;
text-align: center;
margin-left: 85%;
transition: .1s ease-in-out;
height:23px;
}
.menu1:hover
{
border-bottom: 2px solid white;
}
.b1,.b2,.b3,.b4
{
width:250px;
height:350px;
text-align:center;
color: black;
float:left;
cursor:pointer;
padding-left:15px;
padding-right:15px;
}
.b1:hover,.b2:hover,.b3:hover,.b4:hover
{
background: -webkit-linear-gradient(144deg, rgba(255,0,255,1) 0%, rgba(237,0,255,1) 27%,
rgba(189,1,255,1) 65%, rgba(154,3,255,1) 100%);
color:white !important;
box-shadow:-5px 5px 10px 0 rgba(0,0,0,.4);
transition:.3s;
}
.b1:hover i,.b2:hover i,.b3:hover i,.b4:hover i
{
color:white;
border: 3px solid white;
}
.body
{
```

```
display:flex;
justify-content: space-around;
align-items: center;
margin-top: 200px;
margin-bottom: 200px;
}
.fa-calendar-alt,.fa-church,.fa-images,.fa-hand-holding-usd
{
font-size: 40px;
margin-top: 20px;
border: 3px solid rgb(193, 53, 235);
padding:15px;
border-radius: 100%;
text-align: center;
width:40px;
height:40px;
color:rgb(175, 1, 255);
}
.fa-church
{
font-size:35px;
text-align:center;
}
hr
{
height:2px;
background-color: black;
margin-top: 20px;
margin-bottom: 20px;
border:0px;
}
.clouds
{
position:absolute;
top:0;
left:0;
width:100%;
height:100%;
}
.clouds img
{
position:absolute;
bottom:0;
max-width:100%;
animation:cloudanimate calc(10s * var(--i)) linear infinite;
}
@keyframes cloudanimate
{
```

```
0%
{
transform:translateX(-100%);
}
100%
{
transform:translateX(100%);
}
}
</style>
</head>
<body>
<div class=sticky>
<div class=logo>
<i class="fab fa-edge"> Parish</i>
</div>
 <a href="http://localhost/project/loginform" class=login>
login <i class="fa fa-sign-in-alt" style="font-size: medium;"></i>
</a>
</div>
<div class="img">
<div>Welcome to eParish</div>
<div class=clouds>
<img src="icon/cloud1.png" style="--i:1;">
<img src="icon/cloud2.png" style="--i:2;">
<img src="icon/cloud3.png" style="--i:3;">
<img src="icon/cloud4.png" style="--i:4;">
<img src="icon/cloud5.png" style="--i:5;">
</div>
</div>
<div class="body">
<a href=events>
<div class="b1" data-aos="flip-right"><h1><i class="far fa-calendar-alt"></i></h1><font size=6
>Events</font><br><br>
Do not be anxious about anything, but in every situation, by prayer and petition, with
thanksgiving, present your requests to God. And the peace of God, which transcends all
understanding, will guard your hearts and your minds in Christ Jesus.Be a part of every events.
</div>
</a>
<a href=history>
<div class="b2" data-aos="flip-right"><h1><i class="fas fa-church"></i></h1><font
size=6>History</font><br><br>
History can be seen as the introduction of great men and heroes who established their charisma,
wisdom and power to make an impact on the whole world.
People will always remember what men have done to the society and how far they have taken
civilization.
</div>
</a>
```

```
<a href=image>
<div class="b3" data-aos="flip-left"><h1><i class="far fa-images"></i></h1><font
size=6>Gallery</font><br><br>
Memories without a picture are like misprints in a book. You might think their are too much
cintent but when you open it their is nothing but unconnected moments which you regret that you
had the time to relive it. A good picture depicts a good memory.
</div>
</a>
<a href=pay/TxnTest>
<div class="b4" data-aos="flip-left"><h1><i class="fas fa-hand-holding-usd"></h1></i><font
size=6>Donation</font><br><br>
Whoever sows sparingly will also reap sparingly, and whoever sows generously will also reap
generously. Each of you should give what you have decided in your heart to give, not reluctantly
or under compulsion, for God loves a cheerful giver. Donate each other.
</div>
</a>
</div>
<br><br><br><br><br>
<div class=vicar>
<?php
$sql="select * from priest";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
?>
<h1  data-aos="fade-right">About Vicar</h1>
<ul>
<li data-aos="fade-right">I am <?php echo $row["username"]; ?></li>
<li data-aos="fade-right">Joined on <?php echo date('d-m-Y',strtotime($row["jd"])); ?></li>
<li data-aos="fade-right">Last served at <?php echo $row["cbefore"]; ?></li>
<li data-aos="fade-right">Wish me on <?php echo date('d-m-Y',strtotime($row["dob"])); ?></li>
<li data-aos="fade-right">Had experience about <?php echo $row["experience"]; ?> years</li>
</ul>
<div class="imgbox">
<?php
$sql = mysqli_query($conn, "SELECT * FROM priest");
$row = mysqli_fetch_array($sql);
echo "<img src='images/".$row['image']."' class=vicarimg>";
?>
</div>
</div>
<div class=footer data-aos="fade-up">
<h1 style="margin-top: 50px;">Get In Touch</h1>
<br><br><br><br>
<div class="left-msg">
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post"
data-aos="slide-up">
<input type=text name=name placeholder="Your Name" class=input1 required>
<br>
```

```
<input type=number name=contact placeholder="Phone Number" class=input1 required>
<br>
<input type=email name=email placeholder="Email ID" class=input1 required>
<br>
<textarea placeholder="Your Message" name=msg class=input2 required></textarea>
<br>
<button type="submit" name="send" class="footer-button">Send Message</button>
</form>
<br>
<br>
<br>
</div>
<div class="right-msg" data-aos="slide-up">
<br>
<i class="fas fa-map-marker-alt"></i>
<div class="location"> Mathra Kokkad Road</div>
<br>
<i class="fas fa-phone" style="transform:rotatey(180deg);"></i>
<div class="phno"> +91 9207224063</div>
<br>
<i class="fas fa-envelope"></i>
<div class="email"> admin@eparish.ml</div>
<br>
<div class="follow">
<a href=#><i class="fab fa-facebook-square" style="color:rgb(39, 93, 209);font-size:
29px;margin-right: 18px;margin-left: 11px;"></i> </a>
<a href=#><i class="fab fa-youtube"  style="color:rgb(255, 4, 4);font-size: 29px;margin-right:
18px;"></i> </a>
<a href=#><i class="fab fa-twitter"  style="color:rgb(0, 140, 255);font-size: 29px;margin-right:
18px;"></i> </a>
<a href=#><i class="fab fa-instagram"  style="color:rgb(240, 15, 101);font-size: 29px;margin-
right: 18px;"></i> </a>
<a href=#><i class="fab fa-whatsapp-square"  style="color:rgb(41, 240, 15);font-size:
29px;margin-right: 18px;"></i> </a>
</div>
</div>
</div>
<script src="https://code.jquery.com/jquery-1.11.3.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/highlight.js/8.6/highlight.min.js"></script>
<script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>
<script>
AOS.init({
easing: 'ease-out-back',
duration: 2000
});
</script>
<script>
hljs.initHighlightingOnLoad();
```

```
$('.hero__scroll').on('click', function(e) {
$('html, body').animate({
scrollTop: $(window).height()
}, 1200);
});
</script>
</body>
</html>
```

**welcome.php**

```
<?php
include('config.php');
include "session.php";
if($_SESSION['type']!="admin")
{
header("location:loginform.php");
}
?>
<html>
<head><title>eParish</title>
<link href="fontawesome-free-5.11.2-web/css/all.min.css" rel=stylesheet>
<link rel="shortcut icon" type="image/png" href="images/eedge.png"/>
<style>
*
{
padding:0;
margin:0;
}
::-webkit-scrollbar
{
display:none;
}
body
{
background:url("29.jpg");
background-size:cover;
background-attachment:fixed
}
.stickify
{
width:100%;
height:15vh;
background-color:rgba(255,255,255,.3);
COLOR:black;
display:flex;
justify-content:SPACE-BETWEEN;
align-items:center;
```

```
}
.stickify a div
{
FLOAT:LEFT;
font-size:20px;
background-color:rgba(255,255,255,0);
width:100%;
height:100%;
justify-content:center;
align-items:center;
display:flex;
}
.stickify a:hover
{
COLOR:white;
background-color:rgba(0,0,0,.6);
height:40px;
border-radius:15px;
}
.stickify .none:hover
{
COLOR:white;
background-color:rgba(34,83,189,0);
height:40px;
border-radius:15px;
}
a
{
margin-left:20px;
margin-right:20px;
height:40%;
background-color:rgba(255,255,255,0);
text-decoration:none;
width:30%;
align-items:center;
display:flex;
color:black;
}
.stickify a:nth-child(4)
{
background-color:rgba(34,83,189,.3);
color:white;
height:40px;
border-radius:15px;
}
@keyframes animate
{
```

```
0%
{
transform:rotate(0deg);
}
100%
{
transform:rotate(360deg);
}
}
a:nth-child(7):hover i
{
animation:animate 2s linear infinite;
}
.fab
{
font-size:50px;
color:white;
}
.logo
{
margin-left:12px;
}
.odometer
{
font-size:100px;
margin:200px 0;
text-align: center;
width: 100%;
-webkit-animation-duration: 3s;
animation-duration: 3s;
}
.bible
{
width:auto
height:100px;
float:left;
}
.song
{
width:auto;
height:100px;
float:right;

}
.speech
{
width:auto;
height:100px;
```

```
float:left;
}
.mess
{
width:auto;
height:100px;
float:right;
}
.sub1
{
display:block;
border:1px solid white;
border-radius:1px;
margin-top:120px;
margin-left:45px;
width:55vw;
background-color:#fffe;
height:100.8px;
animation:animate1 2s 1;
}
.sub2
{
display:block;
border:1px solid white;
border-radius:1px;
margin-top:85px;
margin-right:40px;
width:55vw;
background-color:#fffe;
height:100.7px;
float:right;
animation:animate2 2s 1;
}
.sub3
{
display:block;
border:1px solid white;
border-radius:1px;
margin-top:85px;
margin-left:45px;
width:55vw;
background-color:#fffe;
height:100.7px;
float:left;
animation:animate3 2s 1;
}
.sub4
{
```

```
display:block;
border:1px solid white;
border-radius:1px;
margin-top:85px;
margin-right:40px;
width:55vw;
background-color:#fffe;
height:100.7px;
float:right;
animation:animate4 2s 1;
}
.main
{
margin-bottom:50px;
background-color:rgba(0,0,0,.3);
height:900px;
}
@keyframes animate1
{
0%
{
margin-left:-900px;
opacity:0;
}
100%
{
margin-left:45px;
opacity:1;
}
}
@keyframes animate2
{
0%
{
margin-right:-900px;
opacity:0;
}
100%
{
margin-right:40px;
opacity:1;
}
}
@keyframes animate3
{
0%
{
margin-left:-900px;
```

```
opacity:0;
margin-top:400px;
}
100%
{
margin-left:45px;
opacity:1;
}
}
@keyframes animate4
{
0%
{
margin-right:-900px;
opacity:0;
}
100%
{
margin-right:40px;
opacity:1;
}
}
@keyframes animate11
{
0%
{
opacity:0;
}
100%
{
opacity:1;
}
}
</style>
</head>
<body>
<div class=head>
<!--navigation-->
<div class=stickify>
<div class=logo><i class="fab fa-edge"> Parish</i></div>
<a href=# class=none><div ></div></a>
<a href=# class=none><div ></div></a>
<a href=welcome.php><div class=home><i class="fas fa-home"></i> HOME</div></a>
<a href=insert.php><div class=insert><i class="fas fa-server"></i> UPDATE</div></a>
<a href=view.php><div class=view><i class="fas fa-eye"></i> VIEW</div></a>
<a href=settings.php><div class=logout><i class="fas fa-cog"></i>  
SETTINGS</div></a>
<!--/navigation-->
```

```
</div>
</div>
<br><br><br>
<h1 style='margin-top:100px;width:780px;margin-left:245px;font-size:46px;animation:animate11
3s 1'><center>"
<?php
$ssql="select tfd from priest";
$sresult=$conn->query($ssql);
$srow=$sresult->fetch_assoc();
echo $srow["tfd"];
?>
 "</center></h1>
<br><br><br><br><br><br><br><br><br>
<br><br><br><br><br>
<font size=5 style=float:right;color:white;margin-top:7px;>  Members  </font><font
class="number-animation" style=float:right;color:white; size=6>
<?php
$sql="SELECT COUNT(num) FROM family";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo $row['COUNT(num)'];
?>
</font>
<div class=main>
<br>
<div class="sub1">
<img src=bible.jpg class=bible>
<?php
$sql="SELECT * FROM events where name='bible1'";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo "<b><div style='margin-left:20vw;font-size:1.4vw'><br>Chapter 1 : ".$row['value']."<br>";
$sql="SELECT * FROM events where name='bible2'";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo "<br>Chapter 2 : ".$row['value']."</div></b>";
?>
</div>
<div class="sub2">
<img src=song.jpg class=song>
<?php
$sql="SELECT * FROM events where name='song1'";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo "<b><div style='margin-left:10vw;font-size:1.4vw'><br>Song 1 :
".$row['value']."           
      ";
$sql="SELECT * FROM events where name='song2'";
```

```
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo "Song 2 : ".$row['value']."<br>";
$sql="SELECT * FROM events where name='song3'";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo "<br>Song 3 :
".$row['value']."           
      ";
$sql="SELECT * FROM events where name='song4'";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo " Song 4 : ".$row['value']."</div></b>";
?>
</div>
<div class="sub3">
<img src=speech.jfif class=speech>
<?php
$sql="SELECT * FROM events where name='speech'";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo "<b><div style='margin-left:23vw;font-size:1.7vw;margin-top:2.5vh;'><br> Speech :
".$row['value']."</div></b>";
?>
</div>
<div class="sub4">
<img src=mess.jpg class=mess>
<?php
$sql="SELECT * FROM events where name='msg'";
$result=$conn->query($sql);
$row=$result->fetch_assoc();
echo "<b><div style='font-size:1.7vw;display:flex;justify-content:center;align-
items:center;margin-left:10px;margin-right:185px;'><br>  ".$row['value']."</div></b>";
?>
</div>
<br>
</div>
<!--number animation-->
<div class="odometer animated fadeIn" id="odometer"></div>
<script>
var odometer=new Odometer({
el:$('.odometer')[0],
value:1234,
theme: 'minimal'
duration:3000
});
odometer.render();
$($'odometer').text(5445);
```

```
</script>
<!--number animation-->
</body>
</html>
```

**insert2.php**

```
<?php
include "session.php";
if($_SESSION['type']!="user")
{
header("location:loginform.php");
}
if($_SERVER["REQUEST_METHOD"]=="POST"&&isset($_POST['subf']))
{
$user=$_SESSION["username"];
$password=$_SESSION["password"];
$item=$_POST['num2'];
$value=$_POST['num3'];
if(empty($item))
{
echo "<center><font color=white>ENTER THE CATEGORY !</font></center>";
}
elseif(empty($value))
{
echo "<center><font color=white>ENTER THE VALUE !</font></center>";
}
else
{
$sql="update family set $item='$value' where user='$user' ";
if($conn->query($sql))
{
echo "<center><font color=white><div class=update>updated
succefully</div></font></center>";
}
else
{
echo "<center><font color=white>updation failed</font></center>";
}
}
}
?>
<html>
<head><title>eParish</title>
<link rel="shortcut icon" type="image/png" href="images/eedge.png"/>
<link href="fontawesome-free-5.11.2-web/css/all.min.css" rel=stylesheet>
<script src=jquery.txt></script>
<style>
```

```
*
{
padding:0;
margin:0;
}

body
{
background:url("29.jpg");
background-size:cover;
}
.stickify
{
width:100%;
height:9vh;
background-color:rgba(212,223,247,.3);
COLOR:black;
display:flex;
justify-content:SPACE-BETWEEN;
align-items:center;
position:sticky;
top:0;
}
.stickify a div
{
FLOAT:LEFT;
font-size:20px;
background-color:rgba(255,255,255,0);
width:100%;
height:100%;
justify-content:center;
align-items:center;
display:flex;
}
.stickify a:hover
{
COLOR:white;
background-color:rgba(0,0,0,.6);
}
a
{
height:100%;
background-color:rgba(255,255,255,0);
text-decoration:none;
width:30%;
align-items:center;
display:flex;
color:black;
```

```
}
a:nth-child(2)
{
box-shadow:inset 0 0 5px #ddd;
color:white;
}
.box
{
margin-top:9vh;
height:60vh;
width:100%;
}
.insbox1
{
background-color:rgba(255,255,255,.8);
height:60vh;
width:93.6vw;
border-radius:15px;
margin:3.6vh;
padding:10px;
}
.insbox2
{
background-color:rgba(255,255,255,.8);
height:20vh;
width:95vw;
border-radius:15px;
margin:3.5vh;
display:none;
}
@keyframes animate
{
0%
{
transform:rotate(0deg);
}
100%
{
transform:rotate(360deg);
}
}
a:nth-child(4):hover i
{
animation:animate 2s linear infinite;
}
.update
{
animation:update1 10s 1;
```

```
height:0px;
color:#265fd9;
transition:2s ease-in-out;
}
@keyframes update1
{
0%
{
height:auto;
color:white;
}
99%
{
height:auto;
color:white;
}
100%
{
height:0px;
color:#265fd9;
}
}
input[type=submit],input[type=reset]
{
width:130px;
height:28px;
color:white;
background: linear-gradient(90deg, rgba(0,18,36,1) 0%, rgba(47,91,245,1) 0%, rgba(49,67,249,1)
100%);
cursor:pointer;
border:0;
animation:anima 2s 1;
}
input[type=reset]
{
animation:anim 2s 1;
}
.fa-address-card
{
position:absolute;
font-size:85px;
margin-left:83%;
margin-top:2%;
opacity:.7;
animation:animat 10s linear infinite;
}
@keyframes animat
{
```

```
0%
{
color:darkviolet;
}
10%
{
color:green;
}
20%
{
color:blue;
}
30%
{
color:red;
}
40%
{
color:yellow;
}
50%
{
color:orange;
}
60%
{
color:indigo;
}
70%
{
color:black;
}
80%
{
color:brown;
}
90%
{
color:grey;
}
100
{
color:pink;
}
}
@keyframes anima
{
0%
```

```
{
margin-left:-1000px;
}
100%
{
margin-left:0px;
}
}
</style>
</head>
<body>
<!--Navigation bar-->
<div class=stickify>
<a href=welcome2><div class=home><i class="fas fa-home"></i> HOME</div></a>
<a href=insert2><div class=insert><i class="fas fa-edit"></i> EDIT</div></a>
<a href=view2><div class=view><i class="fas fa-eye"></i> VIEW</div></a>
<a href=settings2><div class=logout><i class="fas fa-cog"></i>   SETTINGS</div></a>
</div>
<!--/Navigation bar-->
<div class=box>
<div class=insbox1>
<br>
<i class="far fa-address-card"></i>
<form method=post action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
id=frm1>
<fieldset style="font-size:19px;padding-left:30px;">
<legend> Update Your Details </legend>
<br><br><br>
 Select the category : <select name=num2 required style=width:166px;>
<option value="tax">Subscription</option>
<option value="members">Members</option>
<option value="membername">Member Name</option>
<option value="owner">Owner</option>
<option value="phno">Phone Number</option>
</select>
<br><br><br>
Enter the value       : <input type=text name=num3
required>
<br><br><br><br><br>
<center>
<input type=reset value=Reset>  
<input type=submit value=Update name=subf>
</center>
<br><br><br>
</fieldset>
</form>
</div>
<br><br>
```

```
<div class=insbox2> </div>
<br>
<br>
</div>
<br>
<br>
</body>
</html>
```

# 10.REFERENCES

**a.Books Reference**


- Rajib Mall, “Fundamentals of Software Engineering”, Fourth Edition
- Ramon A. Mata-toledo and Pauline K. Cushman, “Fundamentals of Database”, Schaum’s ouline series
- Thomas Powell, “HTML: The Complete Reference ”,Second Edition
- Julie C Meloni, “PHP, MySQL & Apache”, Fifth Edition


**b.Websites**

- https://www.geeksforgeeks.org/
- https://www.w3schools.com/
- https://www.cspsyco.blogspot.com/
- https://www.php.net/
- https://www.stackoverflow.com/
- https://www.tutorialspoint.com/