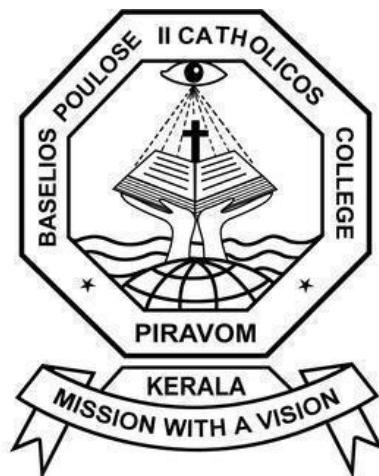


**BASELIOS POULOSE II CATHOLICOS COLLEGE
BASELIOS MOUNT, PIRAVOM**

**Re-accredited with ‘A’ Grade by
NAAC (Affiliated to Mahatma Gandhi University)**

**DEPARTMENT OF COMPUTER
APPLICATIONS**



2022-23

**Project
Report**

on

ESRA

[Ecumenical Site for Religious Administration]

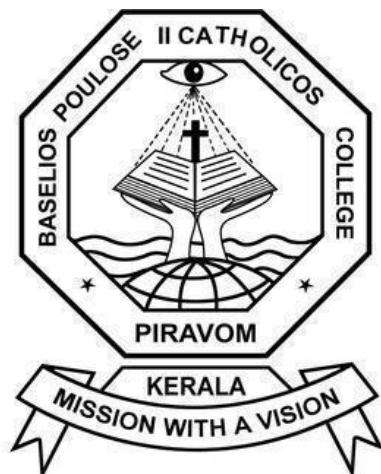
**BASELIOS POULOSE II CATHOLICOS COLLEGE
BASELIOS MOUNT, PIRAVOM**

Re-accredited with 'A' Grade by

PIRAVOM

2022-23

DEPARTMENT OF COMPUTER APPLICATIONS



Project Report

On ESRA

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF COMPUTER APPLICATION

**Guided by: Dr Kurian M J
(Dept. of Computer application)**

**Submitted by:
Jaiden Bose**

**BASELIOS POULOSE II CATHOLICOS COLLEGE
BASELIOS MOUNT, PIRAVOM**

Re-accredited with ‘A’ Grade by

PIRAVOM

2022-23

DEPARTMENT OF COMPUTER APPLICATIONS



Certificate

This is to certify that the project entitled “ESRA” submitted in partial fulfillment for the award of the degree of BACHELOR OF COMPUTER APPLICATION is a bonafide report of the project done by Jaiden Bose (Reg no: 200021096658) during the year 2022-23.

Internal Guide:

Dr. Kurian MJ

Examiner: 1

College Seal

Head of the department:

Dr. Kurian MJ

Department Seal

DECLARATION

I hereby declare that the this project work entitled “ESRA “ is a record of original work done by me under the guidance of Dr. Kurian MJ , Professor, Department of Computer Applications and the work has not formed the basis for the award of any degree or diploma or similar title to any candidate of any university subject.

Internal Guide:

Dr. Kurian MJ

Signature of Student

ACKNOWLEDGEMENT

At the outset, I thank God Almighty for making the endeavor a success.

I express my gratitude to **Dr. Tiji Zacharia**, Principal, Baselios Poulose II Catholicos College, for providing me with adequate facilities, ways and means by which I was able to complete the project work. I express my sincere thanks to my internal guide **Dr.Kurian M J**, who guided me properly from the beginning to the end of my project and examined the draft of this project,suggestions and modifications .

With immense pleasure I take this opportunity to record our sincere thanks to the Head of the Department **Dr. Kurian M J**, Professor, Department of Computer Applications for his motivation throughout this project .

Last but not the least, I also express my gratitude to all other members of the faculty and well wishers who assisted me in various occasions during the project work.

Jaiden Bose

ABSTRACT

The Web application entitled “ ESRA” is an online platform for managing church and its associative activities. The system is specially designed for people who wish to access the services provided by the church online.

The system consists of three users : Admin, Priest, Family. The admin has the privilege control on the system. The admin can verify and edit the information regarding the family, priest and church related functions. The admin verifies all service bookings of the system. Family members are general users who can simply register with basic details and login to access the facilities provided by the church. They can book auditoriums, book services, apply for certificates and view directory. Priests are added by the admin. They could view their profiles and make changes and can also acknowledge the service requests .

TABLE OF CONTENTS

ACKNOWLEDGEMENT

ABSTRACT

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION	15
1.1. BACKGROUND AND MOTIVATION	15
1.2. THE PROPOSED SYSTEM	15
1.3. PROJECT SCOPE	16
1.3.1 LIMITATIONS OF EXISTING SYSTEM	16
1.3.2 ADVANTAGES OF PROPOSED SYSTEM	17
2. SYSTEM ANALYSIS	19
2.1. INTRODUCTION	19
2.2. STAKEHOLDERS OF THIS PROJECT	20
2.2.1. ADMINISTRATOR	20
2.2.2. PRIEST	20
2.2.3. FAMILY	21
2.3. SOFTWARE REQUIREMENT SPECIFICATIONS	22
2.3.1. ADMIN	22
2.3.2. PRIEST	22
2.3.3. FAMILY	22
2.4. FEASIBILITY STUDY	23
2.4.1 TECHNICAL FEASIBILITY	24
2.4.2 OPERATIONAL FEASIBILITY	24
2.4.3 ECONOMIC FEASIBILITY	25
2.5. SOFTWARE DEVELOPMENT LIFECYCLE MODEL	25
2.6. HARDWARE AND SOFTWARE REQUIREMENTS	27
2.6.1. SOFTWARE SPECIFICATIONS	27

2.6.1.1.	HTML	27
2.6.1.2.	CSS	28
2.6.1.3.	AJAX	28
2.6.1.4.	JQUERY	29
2.6.1.5.	BOOTSTRAP	29
2.6.1.6.	JAVASCRIPT	29
2.6.1.7.	PHP	30
2.6.1.8.	WAMP SERVER	30
2.6.1.9.	MYSQL	31
2.6.1.10.	WINDOWS 10	34
2.6.1.11.	MICROSOFT WORD	34
2.6.1.12.	SMART DRAW	35
2.6.2.	HARDWARE REQUIRE	36
3. SYSTEM DESIGN		37
3.1.	SYSTEM ARCHITECTURE	37
3.2.	MODULE DESIGN	38
3.3.	DATABASE DESIGN	38
3.3.1.	NORMALIZATION	39
3.3.2.	TABLE STRUCTURE	41
3.3.3.	DATA FLOW DIAGRAM	47
3.3.3.1.	INTRODUCTION TO DATA FLOW DIAGRAMS	47
3.3.3.2.	DATA FLOW DIAGRAM	50
3.4.	INTERFACE DESIGN	58
3.4.1.	USER INTERFACE SCREEN DESIGN	58
3.4.2.	OUTPUT DESIGN	61
4. IMPLEMENTATION		65
4.1.	CODING STANDARD	65
4.2.	SAMPLE CODE	66
5. TESTING		71
5.1.	TEST CASES	71
5.1.1.	UNIT TESTING	72

5.1.2. INTEGRATION TESTING	74
5.1.3. BLACK BOX TESTING	75
5.1.4. WHITE BOX TESTING	76
5.1.5. VALIDATION TESTING	77
5.1.6. USER ACCEPTANCE TESTING	78
5.2. TEST CASE DOCUMENTS	79
6. CONCLUSION	82
6.1. FUTURE ENHANCEMENTS	82
7. REFERENCES	83
8. APPENDIX	84
8.1 SCREENSHOTS	84

LIST OF TABLES

Sl. no.	Table name	Page number
1	tb_admin	42
2	tbl_certificatetype	42
3	tb_priest	42
4	tb_auditorium	42
5	tb_category	43
6	tb_galllery	43
7	tb_bookauditorium	43
8	tb_ceremony	44
9	tb_news	44
10	tb_place	44
11	tb_family	44
12	tb_district	44
13	tb_donation	45
14	tb_event	45
15	tb_certificate	45
16	tb_churchgallery	46
17	tb_service	46
18	tb_user	46
19	tb_unit	47
20	tb_servicetype	47

LIST OF FIGURES

Sl. no.	Figure name	Page number
1.	Context level DFD for ESRA	50
2	First level DFD for ESRA	51
3	Second level DFD for Admin Login	52
4	Second level DFD for Priest Login	52
5	Second level DFD for Family Login	53
6	Second level DFD for Priest Registration	53
7	Second level DFD for Family Registration	53
8	Second level DFD for Adding Place	54
9	Second level DFD for Adding Events	54
10	Second level DFD for managing Family	55
11	Second level DFD for booking Auditorium	55
12	Second level DFD for booking Service	55
13	Second level DFD for booking certificate	56
14	Second level DFD for Service Report	56
15	Second level DFD for Certificate Report	57
16	Second level DFD for Directory	57
17	Family Registration Form	60
18	User Registration Form	61
19	Table format output based on users	63
20	Table report f services booked	64
21	Unit Testing	73
22	Unit Testing Result	73
23	Integration Testing	75
24	Validation Testing	77
25	Already Exist Validation Testing	78

26	Already Exist Validation Testing Result	78
27	Homepage	84
27	Gallery	87
28	Login Page	88
29	Registration Page	88
30	Admin Home Page	89
31	Add District Page	89
32	Service Details Showing Page	90
33	Service Report	90
34	Directory Page	91
35	Family Page	91
36	Family Home Page	92
37	Auditorium Booking Page	92
38	Payment Page	93
39	Service Booking Page	94
40	User Registration	94
41	Priest Homepage	95
42	Priest Profile	96
43	Service Requests shown on Priest Account	96

1.INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

A church, church building or church house is a building used for Christian worship services and other Christian religious activities. Religion has always been a part in building the culture among humans and uniting them. As long as humans live, there will be different kinds of religious worship. During the time of covid pandemic, It became arduous for the worshippers to gather in church and do the offerings and all, also it became hard to keep in touch with the community and access the service provided by the church. There are also difficulties for the people who work abroad to keep in touch with the church community in their homeland. In such a situation, online access to the church community and services provided by the church will be a great benefit for the worshippers. A consolidated website for church management will aid both the administrator and members of the church.

1.2 THE PROPOSED SYSTEM

The proposed system ensures a solution for such problems mentioned above by making the administration and church services online and computerized. A website for each church will provide easier access for the believers in church related services and also it will aid in ordering the functioning of church in a systematic way. The users can register to the site by entering their details and after registering they can login using their unique user id and password. After login to the system, they can access the services provided by the church including applying for certificates provided by the church, booking the parish halls, requesting priest services and for providing offerings and donation. The system admin has the provision to login to the system by using a unique

username and password . The administrator of the system is the person who has the permission for adding priests, managing the requests for hall booking and certificate requests, adding the events and news related to church, adding the data necessary for the user to register etc. He/ she also has the provision to make necessary changes in the above details if necessary. The system administrator has the authoritative power to view the details of all the priests and users who have registered into the system. He/she can also make them inactive/active.

1.3 PROJECT SCOPE

1.3.1. Limitations of Existing System

- **Data storage:-** In manual systems, paper files require a huge amount of storage space and paper storage creates several problems like spoilage, deterioration by way of aging, humidity etc... Paper based systems are generally very bulky both to handle to store and office space are expensive.
- **Speed of processing:-** The speed of execution of data is slow in the existing system. Processing is slower where large volumes of data need to be dealt with. Slower processing means that some information that could be provided if computerized systems were used will not be provided at all, because there is no time.
- **Speed of retrieval of information:-** The speed of retrieval information is very slow in this system. Since all details are entered on registers, if we want to retrieve the information about an old customer, we want to go through all the past records until we find the right one.
- **Time and manpower consuming:-** A considerable amount of time is required for recording details into the system. Report generation of various areas is done manually using great amount of manpower and

time. Erroneous records may lead to misleading information, which is more likely in a manual system.

- **Accuracy:-** In the existing system the error rate is high and it is difficult to locate the errors and correct them. Calculations made on papers often lead to cash mismatch and inaccurate results.
- **Alternations:-** It is difficult to make corrections. If a manual document contains errors or needs updating it is often necessary to recreate the whole document from scratch, rather than just a new version with the relevant details changed.
- **Redundancy:-** If a customer gives different work at different times, each time the customer arrives, the administrator wants to store the personal details repeatedly with each work.
- **User friendliness:-** In the existing system, the degree of user friendliness is considerably low. This system involves readability of the records and maintenance of different details. The technique used in the system is more complicated and there is a lack of technical background towards the system.
- **Back up:-** Back up of data cannot be done easily since all data are in different registers and are written on paper.

1.3.2 Advantages Of Proposed System

Data entry screens are designed such that they are very user friendly and minimum typing is required from the user,

- Not much training required.
- System provides various information's report quickly and accurately in easily understandable formats.
- The new Web application is more user friendly.
- It aims at paperless work.
- Fast access information.
- Efficient traceability.
- Talking into the speed of computer access, large data in less time and facilities

provided by the access.

- Duplication of data will be avoided.
- Menu driven interface provides ease to use.
- Availability of previous data for future reference.

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

Software Engineering is the analysis, design, construction, verification and management of technical or social entities. To engineer software accurately, a software engineering process must be defined. System analysis is a detailed study of the various operations performed by the system and their relationship within and module of the system. It is a structured method for solving the problems related to the development of a new system. The detailed investigation of the present system is the focal point of system analysis. This phase involves the study of parent system and identification of system objectives. Information has to be collected from all people who are affected by or who use the system. During analysis, data are collected on the variable files, decision point and transactions handled by the present system. The main aim of system is to provide the efficient and user friendly automation. So the system analysis process should be performed with extreme precision, so that an accurate picture of existing system, its disadvantages and the requirements of the new system can be obtained.

System analysis involves gathering the necessary information and using the structured tool for analysis. This includes studying the existing system and its drawbacks, designing a new system and conducting cost benefit analysis. System analysis is a problem solving activity that requires intensive communication between the system users and system developers. The system is studied to the minute detail and analyzed. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through various phases of processing of inputs.

There are a number of different approaches to system analysis. When a computer based information system is developed, systems analysis (according to the Waterfall model) would constitute the following steps:

- The development of a feasibility study, involving determining whether a project is economically, technologically and operationally feasible.

- Conducting fact-finding measures, designed to ascertain the requirements of the system's end-users. These typically span interviews, questionnaires, or visual observations of the work of existing systems.

- Gauging how the end-users would operate the system (in terms of general experience in using computer hardware or software), what the system would be used for and so on.

Techniques such as interviews, questionnaires etc. can be used for the detailed study of these processes. The data collected by these sources must be scrutinized to arrive at a conclusion.

The conclusion is an understanding of how the system functions. This system is called the Existing System. The Existing system is then subjected to close observation and the 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 a proposal which is the Proposed System. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is then presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

2.1STAKE HOLDERS OF THIS PROJECT

2.2.1 Administrator

Administrator is the person who manages the software. He is the person who focuses on the data and reports of the software. He/she is the person who adds locations ,gallery images, auditorium information, events , news, ceremonies, priests and modules.

In the case of other stakeholders, they have only access to read or search the questions provided in the database except the privilege for priests to accept or reject service requests. But in the case of an administrator he is the top person, he can view and access all details.

2.2.2 Priests

Priests are added by the administrator. They can also view and update their own information. In addition, they can view the service requests made by the user and accept or reject the requests.

2.2.3 Family

They are the main users of this application. They will be able to view all the available services, look into their description and images. Customers are also needed to register by providing their basic information such as name, email address, phone number, address, etc., to access the website. They can also view and update their information. They can also access the services offered by the website in relation to church.

2.1 SOFTWARE REQUIREMENT SPECIFICATION

2.3.1 Admin

1. Admin can login to the system.
2. Admin should have the provision to add auditoriums, service types, certificate types, news, events, gallery images.
3. Admin also have the privilege to edit or update the above details.
4. Admin have the privilege to register priests.
5. Admin have the privilege for accepting or rejecting families.
6. Admin should have the provision to logout.

2.3.2 Priest

1. The system should have a provision to login the priest by entering priest Username and password.
2. After logging in, the priests have the permission to view and edit their profile.
3. The priests should have the permission to edit the password that is already given by the admin.
4. The priests should have the permission to view the service requests made by the user and to accept or reject them.

2.3.3 Family

1. Users can login to the system.
2. Users should have the privilege to access the different services provided by the church that are given in the system.
3. The system should have the provision to change passwords.
4. The system should have the provision to logout

Table 2.1 Sign of table

Sl. No.	Name & Designation	Date	Accepted (Yes/No)
1	Dr. Kurian M J Professor, Dept of Computer Application BPC College, Piravom		
2	Jaiden Bose Developer		

2.2 FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- To analyze whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.
- When our project guide as well as our client Resmi K R told us regarding the mini project and about Word to the Wise for getting the desired product developed, it came up with a rough idea about what all functions the software must perform and which all features are expected from the software.
- Referencing this information, we did studies and discussions about whether the desired system and its functionality are feasible to develop and the output of this phase is a feasibility study report that should contain adequate comments and recommendations.
- Various types of feasibility that we checked include technical feasibility, operational feasibility,

and economic feasibility.

2.4.1 Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specific user requirements. Technical feasibility also performs the following tasks.

- Analyses the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

From our perspective there are two languages PHP, HTML and database MySQL which are used to develop these web based applications. PHP is used in the front end and MySQL is used in the back end. The Word to the Wise is web based and thus can be accessed through any browsers. As we are using these latest technologies which are currently trending and used by a number of developers across the globe, we can say that our project is technically feasible.

2.4.2 Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyses whether users will adapt to a new software.

- Determines whether the organization is satisfied by the alternative solutions

proposed by the software development team.

We found that our project will be satisfactory for the client since we were discussing every detail about the software with the client at every step. The most important part of operational feasibility study is the input from the client. So the software is built completely according to the requirements of the client. We have used the current industry standards for the software. Hence we can say that this software is operationally feasible.

2.4.3 Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

It is estimated that our project is economically feasible as development cost is very minimal since the tools and technologies used are available online. It's a group student project so there are no personnel costs. Development time is well planned and will not affect other operations and activities of the individuals. Once the system has been developed, the companies purchasing the system will be provided with a manual for training purposes. There is no need to purchase new hardware since the existing computers can still be used to implement the new system.

2.3 SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

One of the basic notions of the software development process is SDLC models which stand for Software Development Life Cycle models. SDLC – is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full removal from the exploitation. Software development lifecycle (SDLC) is a framework that defines the steps involved in the development of software. It covers the detailed plan for building, deploying and maintaining the software. SDLC defines the complete cycle of development i.e. all the tasks involved in gathering a requirement for the maintenance of a Product.

Some of the common SDLC models are Waterfall Model, V-Shaped Model, Prototype Model, Spiral Model, Iterative Incremental Model, Big Bang Model, Agile Model. We used an Agile Model for our Project.

Agile Model

Agile Model is a combination of the Iterative and incremental model. This model focuses more on flexibility while developing a product rather than on the requirement. In the agile methodology after every development iteration, the client is able to see the result and understand if he is satisfied with it or he is not. Extreme programming is one of the practical uses of the agile model. The basis of this model consists of short meetings where we can review our project. In Agile, a product is broken into small incremental builds. It is not developed as a complete product in one go. At the end of each sprint, the project guide verifies the product and after his approval, it is finalized. Client feedback is taken for improvement and his suggestions and enhancement are worked on in the next sprint. Testing is done in each sprint to minimize the risk of any failures.

Advantages of Agile Model:

- It allows more flexibility to adapt to the changes.
- The new feature can be added easily.
- Customer satisfaction as the feedback and suggestions are taken at every stage.
- Risks are minimized thanks to the flexible change process

Disadvantages:

- Lack of documentation.
- If a customer is not clear about how exactly they want the product to be, then the project would fail.
- With all the corrections and changes there is possibility that the project will exceed expected time

2.4 HARDWARE AND SOFTWARE REQUIREMENTS

2.4.4 Software Specifications

This project is built upon the latest technology software.

Front End	:	HTML
Development tool	:	HTML, JavaScript, CSS, Bootstrap, Ajax, jQuery, PHP
Database	:	MySQL
Web server	:	Wamp server
Operating System	:	Windows 10

HTML

HTML is a computer language devised to allow Website creation. These Websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet

audience under the direction of the W3C, the organization charged with designing and maintaining the language.

HTML consists of a series of short codes typed into a text-file by the site author - these are the tags. The text is then saved as a HTML file, and viewed through a browser, like Internet Explorer. This browser reads the file and translates the text into a visible form, hopefully rendering the page as the author had intended. Writing your own HTML entails using tags correctly

to create your vision. You can use anything from a rudimentary text-editor to a powerful graphical editor to create HTML pages.

CSS

Stands for "Cascading Style Sheet." Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML. CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text. While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets.

Ajax

Ajax is a set of web development techniques using many web technologies on the client side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behavior of the existing page. By decoupling the data interchange layer from the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page.

In practice, modern implementations commonly utilize JSON instead of

XML. reload the entire page. In practice, modern implementations commonly utilize JSON instead of XML.

jQuery

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open source software using the permissive MIT License. As of May 2019, jQuery is used by 73% of the 10 million most popular websites. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having 3 to 4 times more usage than any other JavaScript library.

Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobilefirst front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of Web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other Web browsers.

PHP

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHPway back in 1994.

PHP is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

The key advantages of learning PHP are:

PHP is a recursive acronym for "PHP: Hypertext Preprocessor". PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.

It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server. PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time. PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time. PHP is forgiving: PHP language tries to be as forgiving as possible.

PHP Syntax is C-Like.

Wamp Server

Stands for "Windows, Apache, MySQL, and PHP." WAMP is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP).

It is often used for web development and internal testing, but may also be used to serve live websites. The most important part of the WAMP package is Apache (or "Apache HTTP Server") which is used to run the web server within Windows.

By running a local Apache web server on a Windows machine, a web developer can test web pages in a web browser without publishing them live on the Internet.

WAMP also includes MySQL and PHP, which are two of the most common technologies used for creating dynamic websites. MySQL is a high-speed database, while PHP is a scripting language that can be used to access data from the database. By installing these two components locally, a developer can build and test a dynamic website before publishing it to a public web server.

While Apache, MySQL, and PHP are open source components that can be installed individually, they are usually installed together. One popular package is called "WampServer," which provides a user-friendly way to install and configure the "AMP" components on Windows.

MySQL

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout its history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in. MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's. MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by OracleCorporation.

MySQL is a database management system. 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, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

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

The SQL part of —MySQL|| stands for —Structured Query Language||. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQLstandard has been evolving

since 1986 and several versions exist. In this manual, —SQL-92| refers to the standard released in 1992,

—SQL:1999| refers to the standard released in 1999, 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 any time. MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.

The MySQL software uses the GPL (GNU General Public License),<http://www.fsf.org/licenses/>, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL needing to embed MySQL code into a commercial application, you can buy a commercially licensed version from us.

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

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

A large amount of contributed MySQL software is available. MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

Windows 10

Operating System is defined as a program that manages the computer hardware. An operating system can be viewed as a scheduler, where it has resources for which it has charge. Resources include CPU, memory, I/O device and disk space. In another view, the operating system is a new machine. The third view is that the operating system is a multiplexer which allows sharing of resources, provides protection from interference and provides a level of cooperation between users. This project is developed using Windows 10 as the operating system and supports its latest versions. Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and to retail on July 29, 2015. One of Windows 10's most notable features is support for universal apps. Windows 10 also introduced the Microsoft Edge web browser, a virtual desktop system, a window and desktop management feature called Task View, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX12. Windows 10 received mostly positive reviews upon its original release in July 2015. Critics praised Microsoft's decision to provide a desktop-oriented interfacing line with previous versions of Windows, contrasting the tablet-oriented approach of 8, although Windows 10's touch-oriented user interface mode was criticized for containing regressions upon the touch-oriented interface of Windows 8. Critics also praised the improvements to Windows 10's bundled software over Windows 8.1, Xbox Live integration, as well as the functionality and capabilities of the Cortana personal assistant and the replacement of Internet Explorer with Microsoft Edge. However, media outlets have been critical of changes to operating system behaviors, including mandatory update installation, privacy concerns over data collection performed by the OS for Microsoft and its partners and the adware-like tactics used to promote the operating system on its release.

Microsoft Word

Microsoft Word (or simply Word) is a word processor developed by Microsoft. It was first released on October 25, 1983 under the name *Multi-Tool Word* for Xenix systems. Subsequent versions were later written for several other platforms including IBM PCs running DOS (1983), Apple Macintosh running the Classic Mac OS (1985), AT&T Unix PC (1985), Atari ST (1988), OS/2 (1989), Microsoft Windows (1989), SCO Unix (1994), and macOS (formerly OS X; 2001).

Commercial versions of Word are licensed as a standalone product or as a component of Microsoft Office, Windows RT or the discontinued Microsoft Works suite. Unlike most MS-DOS programs at the time, Microsoft Word was designed to be used with a mouse. Advertisements depicted the Microsoft Mouse, and described Word as a WYSIWYG, windowed word processor with the ability to undo and display bold, italic, and underlined text, although it could not render fonts. It was not initially popular, since its user interface was different from the leading word processor at the time, WordStar. However, Microsoft steadily improved the product, releasing versions 2.0 through 5.0 over the next six years. In 1985, Microsoft ported Word to the classic Mac OS (known as Macintosh System Software at the time). This was made easier by Word for DOS having been designed for use with high-resolution displays and laser printers, even though none were yet available to the general public. Following the precedents of LisaWrite and MacWrite, Word for Mac OS added true WYSIWYG features. It fulfilled a need for a word processor that was more capable than MacWrite. After its release, Word for Mac OS's sales were higher than its MS-DOS counterpart for at least four years.

SmartDraw

SmartDraw is a diagram tool used to make flowcharts, organization charts, mind maps, project charts, and other business visuals. SmartDraw has two versions: an online edition and a downloadable edition for Windows desktop.

SmartDraw integrates with Microsoft Office products including Word, PowerPoint, and Excel and G Suite applications like Google Docs and Google Sheets. SmartDraw has apps for Atlassian's Confluence, Jira, and Trello. SmartDraw is compatible with Google Drive, Dropbox, Box, and OneDrive.

Since 1994, the mission of SmartDraw Software has been to expand the ways in which people communicate so that we can clearly understand each other, make informed decisions, and work together to improve our businesses and the world. We accomplish this by creating software and services that make it possible for people to capture and present information as visuals, while being a pleasure to use. In 2019, we took this to the next level by launching VisualScript, which makes it easy to visualize data in relational formats like trees, flows, and timelines, automatically, without any human input. VisualScript is a relationship visualization platform that empowers organizations to visualize data across siloed ecosystems and gain critical insights in real-time. Today, SmartDraw Software is

one of the most sophisticated digital marketing organizations in the world with over 90,000 unique visitors to our website each business day and in excess of 3,000,000 installations of our apps each year. SmartDraw is used by more than half of the Fortune 500 and by over 250,000 public and private enterprises of all sizes around the world. Privately held, SmartDraw Software is headquartered in San Diego, California.

Hardware Requirements

The selection of hardware configuring is a very task related to the software development, particularly inefficient RAM may adversely affect the speed and corresponding on the efficiency of the entire system. The processor should be powerful to handle all the operations

The hard disk should be sufficient to solve the database and the application.

Minimum hardware requirement :

CPU : Pentium IV Processor

Memory : 256 MB Above

Cache : 512 KB

Above Hard disk : 20 GB

Above Monitor : Any

Keyboard : Any

Mouse : Any

3.SYSTEM DESIGN

3.1SYSTEM ARCHITECTURE

A system architecture or system's architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system,

System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

The system architecture can best be thought of as a set of representations of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people. System architecture is primarily concerned with the internal interfaces among the system's components or subsystems, and the interface between the system and its external environment, especially the user.

The structural design reduces complexity, facilitates change and results in easier implementation by encouraging parallel development of different parts of the system. The procedural design transforms structural elements of program architecture into a procedural description of software components. The architectural design considers architecture as the most important functional requirement. The system is based on the three-tier architecture.

The first level is the user interface (presentation logic), which displays controls, receives and validates user input. The second level is the business layer (business logic) where the application specific logic takes place. The third level is the data layer where the application information is stored in files or databases. It contains logic about retrieving and updating data. The important feature about the three-tier design is that the information only travels from one

level to adjacent level.

3.1 MODULE DESIGN

1. LOGIN

This module allows the user to login to the website using his/her unique user id and password so that he/she can access the services provided in the website.

2. REGISTRATION

This module helps the admin, priests and families to register to our website. He/she can use the registration form for registering his/her account. While registering the faculty needs to provide some data for example email id, password etc.

3. MANAGE DATA

This module allows the admin of the site to add and edit data in the website. This includes adding details like district, places, certificate types, service types, events, news and also managing the families, priests.

4. BOOKING

This module allows the user to book auditoriums ,apply for certificates and to access services provided by the church.

5. REPORT

This module allows the admin of the site to view the details of the site through various reports. This includes table reports. All these reports provide information about the services provided by the website and all the users, So that the administrator can easily manage the website.

3.2 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific

objectives are considered:

- Ease of learning and use
- Controlled redundancy
- Data independence
- More information at low cost
- Accuracy and integrity
- Recovery from failure
- Privacy and security
- Performance

A database is an integrated collection of data and provides centralized access to the data.

Usually the centralised data managing the software is called RDBMS. The main significant difference between RDBMS and other DBMS is the separation of data as seen by the program and data has in direct access to stores device. This is the difference between logical and physical data.

3.3.1 Normalisation

Designing a database is complete task and the normalization theory is a useful aid in the design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form. There will be need for most databases to grow by adding new attributes and new relations. The data will be used in new ways. Tuples will be added and deleted. Information stored may undergo updating also. New association may also be added. In such situations the performance of a database is entirely depend upon its design.

A bad database design may lead to certain undesirable things like:

- Repetition of information
- Inability to represent certain information
- Loss of information

To minimize these anomalies, Normalisation may be used. If the database is in a normalized form, the data can be growing without, in most cases, forcing the rewriting application programs. This is important because of the excessive and growing cost of maintaining an organization's application programs and its data from the disrupting effects of database

growth.

As the quality of application programs increases, the cost of maintaining the without normalization will rise to prohibitive levels. A normalized database can also encompass many related activities of an organization thereby minimizing the need for rewriting the applications of programs. Thus, normalization helps one attain a good database design and there by ensures continued efficiency of database.

Normalization theory is built around the concept of normal forms. A relation is said to be in normal form if it satisfies a certain specified set of constraints. For example, a relation is said to be in first normal form (1NF) if it satisfies the constraint that it contains atomic values only. Thus every normalized relation is in 1NF. Numerous normal forms have been defined. Codd defined the first three normal forms.

All normalized relations are in 1NF, some 1NF relations are also in 2NF and some 2NF relations are also in 3NF. 2NF relations are more desirable than 1NF and 3NF are more desirable than 2NF. That is, the database designer should prefer 3NF than 1NF or 2NF. Normalization procedure states that a relation that is in some given normal form can be converted into a set of relations in a more desirable form. We can define this procedure as the successive reduction of a given collection of relations to some more desirable form. This procedure is reversible. That is, it is always possible to take the output from the procedure and convert them back into input. In this process, no information is lost. So it is also called “no loss decomposition”.

First Normal Form

A relation is in first normal form (1NF) if and all its attributes are based on single domain. The objective of normalizing a table is to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

Second Normal Form

A table is said to be second Normal Form (2NF), when it is in 1NF and every attribute in record is functionally dependent upon the whole key, and not just a part of the key.

Third Normal Form

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

3.3.2 Table Structure

Table is a collection of complete details about a particular subject. These data are saved in rows and Columns. The data of each Row are different units. Hence, rows are called RECORDS and Columns of each row are called FIELDS.

Data is stored in tables, which is available in the backend the items and data, which are entered in the input, form id directly stored in this table using linking of database. We can link more than one table to input forms. We can collect the details from the different tables to display on the output.

There are mainly 20 tables in the project. They are,

1. tb_admin
2. tbl_certificatetype
3. tb_priest
4. tb_auditorium
5. tb_category
6. tb_galllery
7. tb_bookauditorium
8. tb_ceremony
9. tb_news
10. tb_place
11. tb_family
12. tb_district
13. tb_donation
14. tb_event
15. tb_certificate
16. tb_churchgallery
17. tb_service
18. tb_user
19. tb_unit
20. tb_servicetype

1. Table: tb_admin

Description: This table is used to store the details of the administrator.

Slno	Field Name	Data type	Size	Constraints
1	admin_id	INT		Primary key
2	admin_name	VARCHAR	30	
3	admin_address	VARCHAR	100	
4	unit_id	INT		Foreign Key
5	admin_mail	VARCHAR	50	
6	admin_pass	VARCHAR	20	
7	admin_gender	VARCHAR	20	
8	admin_dob	VARCHAR	20	
9	admin_contact	VARCHAR	20	
10	admin_image	VARCHAR	100	

2. Table: tbl_certificatetype

Description: This table is used to store certificate types.

Slno	Field Name	Data type	Size	Constraints
1	certificatetype_id	INT		Primary key
2	certificatetype_name	VARCHAR	100	
3	certificater_name	VARCHAR	100	

3. Table: tb_priest

Description: This table is used to store details of the priest.

Slno	Field Name	Data type	Size	Constraints
1	priest_id	INT		Primary key
2	priest_name	VARCHAR	40	
3	priest_address	VARCHAR	100	
4	priest_email	VARCHAR	50	
5	priest_pass	VARCHAR	50	
6	priest_number	VARCHAR	20	
7	place_id	INT	11	Foreign Key
8	priest_dob	VARCHAR	20	
9	priest_image	VARCHAR	100	

4. Table: tb_auditorium

Description: This table is used to store the details of auditorium.

Slno	Field Name	Data type	Size	Constraints
1	auditorium_id	INT		Primary key
2	auditorium_name	VARCHAR	100	
3	auditorium_seating	VARCHAR	100	
4	auditorium_rate	VARCHAR	100	
5	auditorium_image	VARCHAR	100	

5.Table: tb_category

Description: This table is used to store various categories.

Slno	Field Name	Data type	Size	Constraints
1	category_id	INT		Primary key
2	category_name	VARCHAR	20	

6.Table: tb_gallery

Description: This table is used to store images.

Slno	Field Name	Data type	Size	Constraints
1	gallery_id	INT		Primary key
2	gallery_photo	VARCHAR	100	
3	gallery_caption	VARCHAR	100	
4	auditorium_id	INT		Foreign key

7.Table: tb_bookauditorium

Description: This table is used to store the details of auditorium booking.

Slno	Field Name	Data type	Size	Constraints
1	book_id	INT		Primary key
2	auditorium_id	INT		Foreign key
3	book_date	DATE		
4	booked_date	DATE		
5	fam_id	INT		Foreign key
6	book_details	VARCHAR	100	
7	booking_name	VARCHAR	100	

8.Table: tb_ceremony

Description: This table is used to store ceremony details.

Slno	Field Name	Data type	Size	Constraints
1	ceremony_id	INT		Primary key
2	ceremony_image	VARCHAR	100	
3	ceremony_name	VARCHAR	100	
4	ceremony_data	VARCHAR	100	
5	ceremony_date	VARCHAR	100	

9.Table: tb_news

Description: This table is used to store news about church.

Slno	Field Name	Data type	Size	Constraints
1	news_id	INT		Primary key
2	news_date	VARCHAR	100	
3	news_content	VARCHAR	100	
4	news_image	VARCHAR	100	

10.Table: tb_place

Description: This table is used to add places.

Slno	Field Name	Data type	Size	Constraints
1	place_id	INT		Primary key
2	district_id	INT		Foreign key
3	place_name	VARCHAR	100	

11.Table: tb_family

Description: This table is used to store family details.

Slno	Field Name	Data type	Size	Constraints
1	fam_id	INT		Primary key
2	fam_name	VARCHAR	50	
3	fam_headname	VARCHAR	30	
4	fam_member	VARCHAR	20	
5	fam_address	VARCHAR	100	
6	fam_mail	VARCHAR	50	
7	fam_pass	VARCHAR	50	
8	unit_id	INT		Foreign key
9	fam_gender	VARCHAR	20	
10	fam_dob	VARCHAR	20	
11	fam_contact	VARCHAR	20	
12	fam_image	VARCHAR	50	
13	fam_status	INT		

14	fam_occupation	VARCHAR	100	
----	----------------	---------	-----	--

12.Table: tb_district

Description: This table is used to store different districts.

Slno	Field Name	Data type	Size	Constraints
1	district_id	INT		Primary key
2	district_name	VARCHAR	40	

13.Table: tb_donation

Description: This table is used to store donations received.

Slno	Field Name	Data type	Size	Constraints
1	donation_id	INT		Primary key
2	donation_date	VARCHAR	100	
3	donation_amount	INT		
4	category_id	INT		Foreign key
5	fam_id	INT		Foreign key
6	donation_status	INT		
7	donation_name	VARCHAR	11	

14.Table: tb_event

Description: This table is used to store the events.

Slno	Field Name	Data type	Size	Constraints
1	event_id	INT		Primary key
2	event_name	VARCHAR	100	
3	event_date	VARCHAR	100	
4	event_location	VARCHAR	100	

15.Table: tb_certificate

Description: This table is used to store details of certificate requests

Slno	Field Name	Data type	Size	Constraints
1	certificate_id	INT		Primary key
2	certificate_name	VARCHAR	60	
3	certificate_requirements	VARCHAR	500	
4	fam_id	INT		Foreign key
5	certificatetype_id	INT		
6	booking_status	INT		

16.Table: tb_churchgallery

Description: This table is used to store photos of church.

Slno	Field Name	Data type	Size	Constraints
1	g_id	INT		Primary key
2	g_photo	VARCHAR	100	

17.Table: tb_service

Description: This table is used to store details of services booked.

Slno	Field Name	Data type	Size	Constraints
1	service_id	INT		Primary key
2	service_date	VARCHAR	100	
3	service_location	VARCHAR	100	
4	service_for_date	VARCHAR	100	
5	fam_id	INT		Foreign key
6	servicetype_id	INT		Foreign key
7	service_for_time	VARCHAR	100	
8	priest_id	INT		Foreign key
9	service_bookmark	VARCHAR	100	
10	service_comments	VARCHAR	100	

18.Table: tb_user

Description: This table is used to store user details(family members).

Slno	Field Name	Data type	Size	Constraints
1	user_id	INT		Primary key
2	user_name	VARCHAR	40	
3	user_address	VARCHAR	100	
4	unit_id	INT		Foreign key
5	user_mail	VARCHAR	100	
6	user_pass	VARCHAR	50	

7	user_gender	VARCHAR	20	
8	user_dob	VARCHAR	20	
9	user_contact	VARCHAR	25	
10	user_image	VARCHAR	100	
11	fam_id	INT		Foreign key

19.Table: tb_unit

Description: This table is used to add kudumba units.

Slno	Field Name	Data type	Size	Constraints
1	unit_id	INT		Primary key
2	unit_name	VARCHAR	60	
3	place_id	INT		Foreign key

20.Table: tb_servicetype

Description: This table is used to add service types.

Slno	Field Name	Data type	Size	Constraints
1	servicetype_id	INT		Primary key
2	servicetype_name	VARCHAR	100	

3.3.3 Data Flow Diagram

3.3.3.1 Introduction To Data Flow Diagram

Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles. Entities supplying data

are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labeled with a descriptive name. Process names are further identified with a number.

The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the input (source), outputs (destination), database (files) and procedures (data flow), all in a format that meet the user's requirements.

The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow.

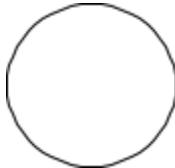
This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. It

is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

Rules for constructing a Data Flow Diagram

1. Arrows should not cross each other
2. Squares, circles and files must bear names.
3. Decomposed data flow squares and circles can have same time
4. Choose meaningful names for data flow
5. Draw all data flows around the outside of the diagram

Basic Data Flow Diagram Symbols

	A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow.
	Circles stands for process that converts data in to information. A process represents transformation where incoming data flows are changed into outgoing data flows.

	<p>A data store is a repository of data that is to be stored for use by one or more processes. It may be as simple as a buffer or queue or as sophisticated as a relational database. They should have clear names. If a process merely uses the content of a store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store, then a double-headed arrow is used.</p>
	<p>A source or sink is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the context of the data flow model.</p>

3.3.3.2 Data Flow Diagram

Each component in a DFD is labelled with a descriptive name. Process names are further identified with numbers. Context level DFD is drawn first. Then the process is decomposed into several elementary levels and is represented in the order of importance. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization.

A DFD methodology is quite effective; especially when the required design.

Zeroth Level DFD for ESRA

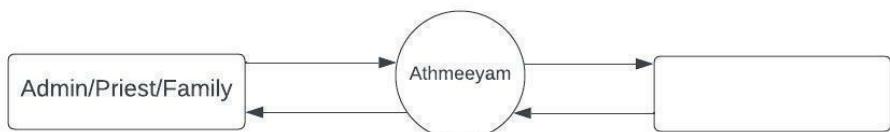


Fig 3.1 Context level DFD for ESRA.

First Level DFD for ESRA

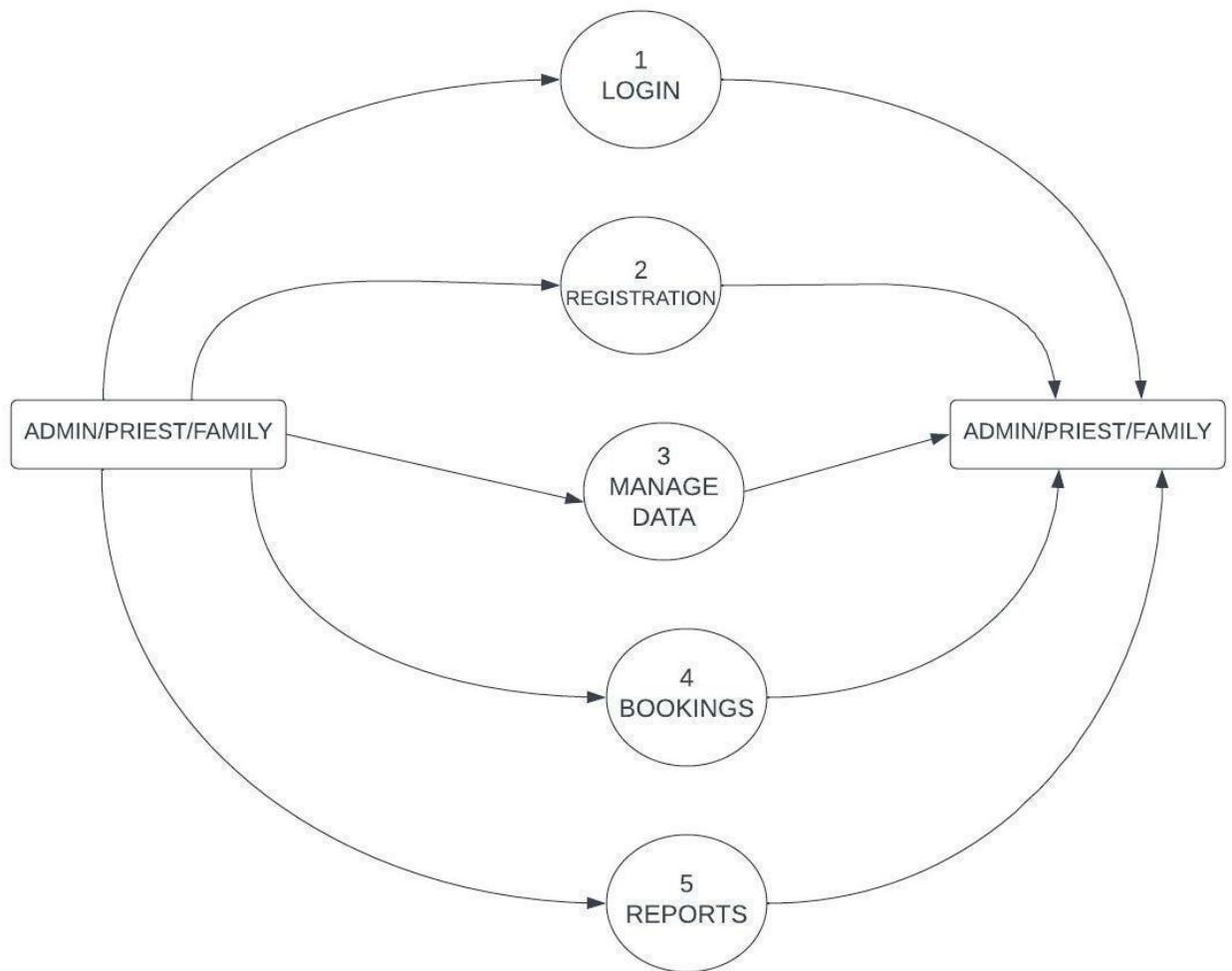


Fig 3.2 First level DFD for ESRA.

Second Level DFD for ESRA

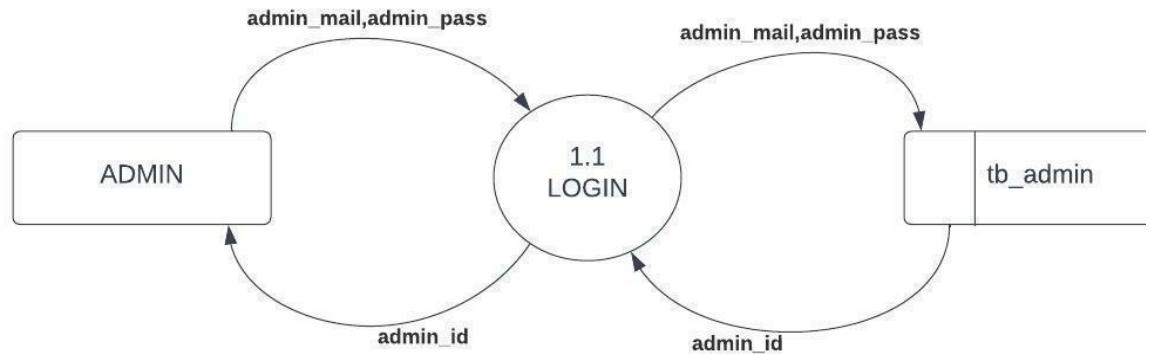


Fig 3.3 Second level DFD for Priest login

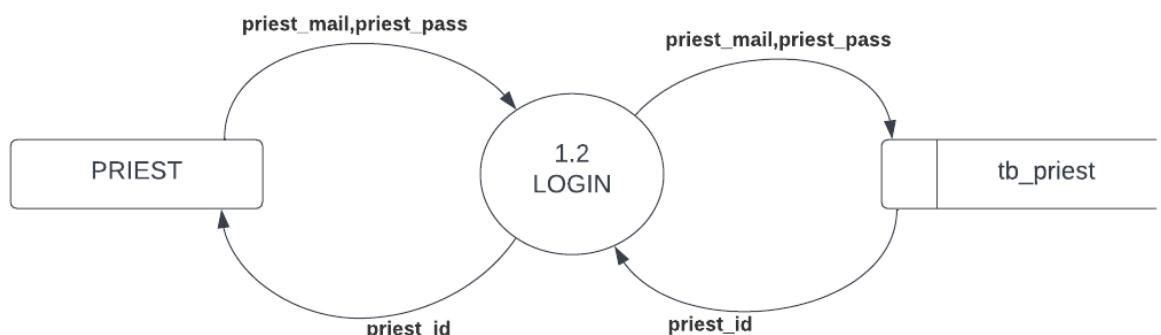


Fig 3.4 Second level DFD for Family login

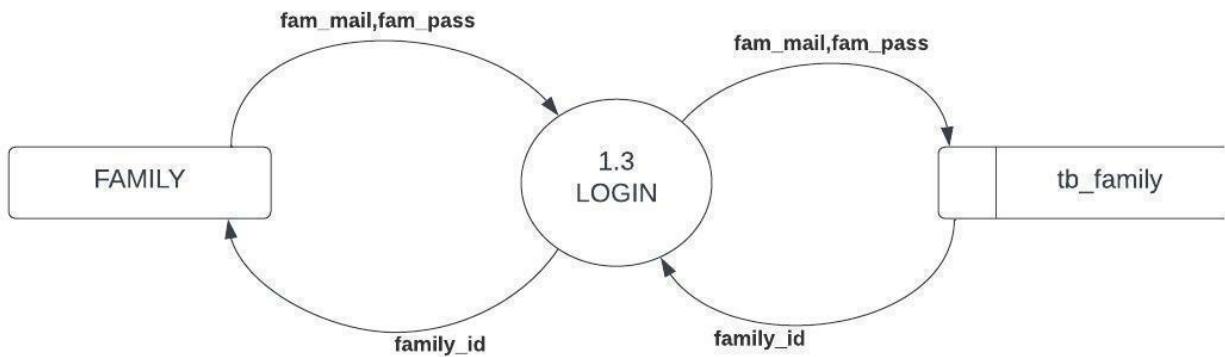


Fig 3.5 Second level DFD for Priest Registration

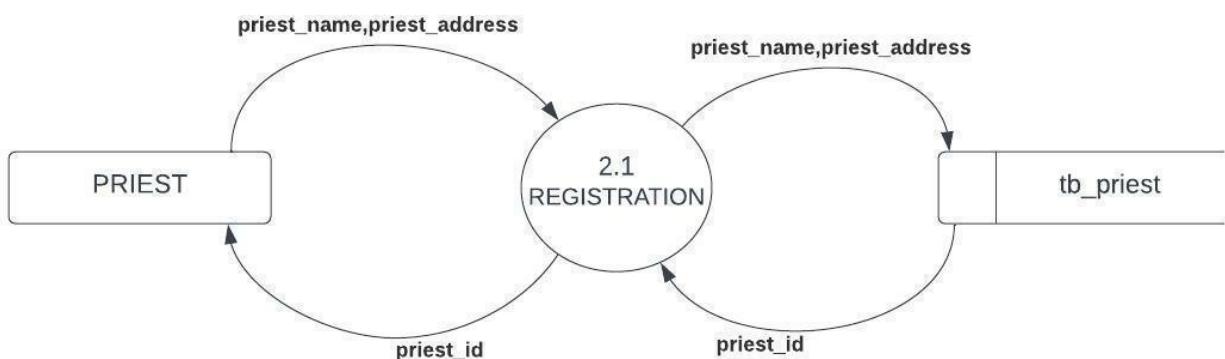


Fig 3.6 Second level DFD for Family Registration

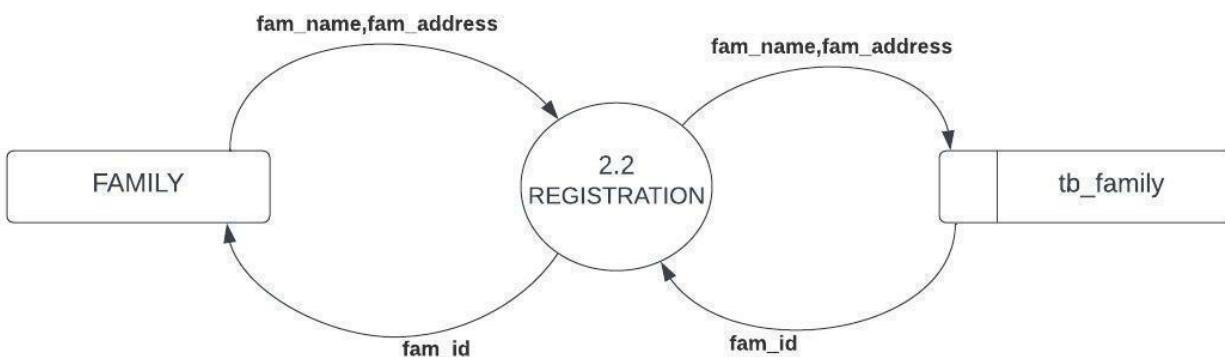


Fig 3.7 Second level DFD for Adding Place

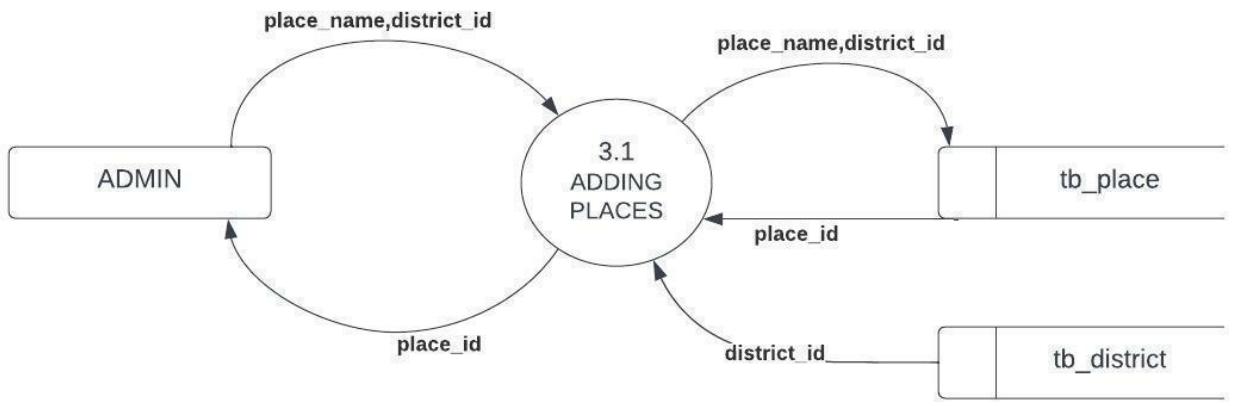


Fig 3.8 Second level DFD for Adding Events.

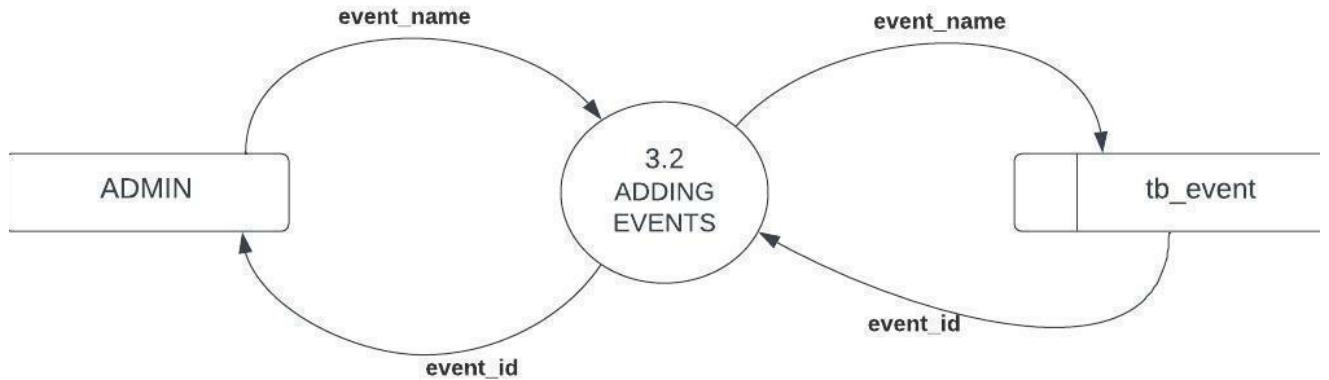


Fig 3.9 Second level DFD for Managing Family

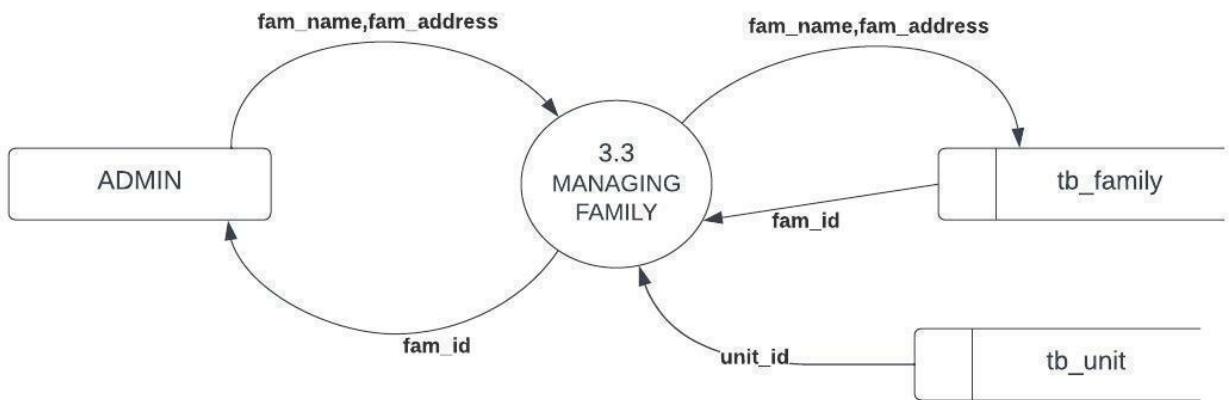


Fig 3.10 Second level DFD for booking Auditorium

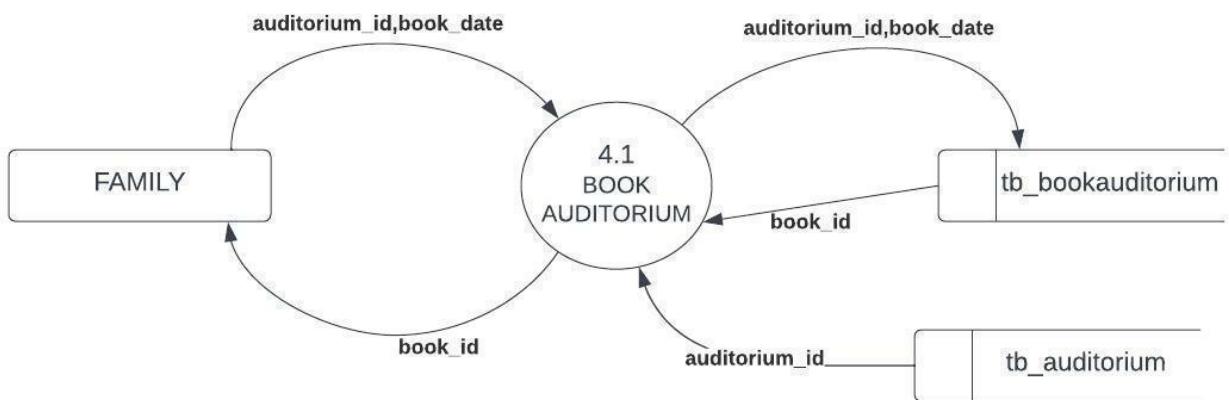


Fig 3.11 Second level DFD for booking Service.

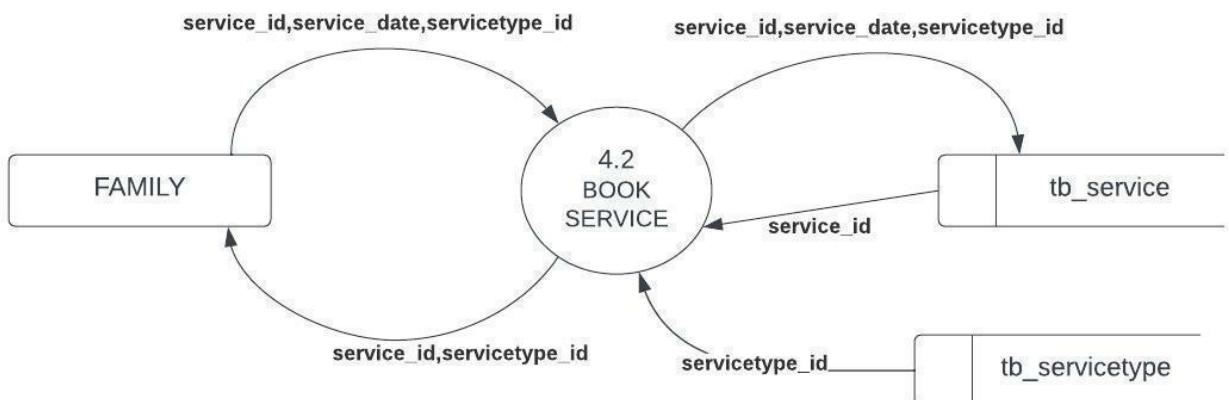


Fig 3.12 Second level DFD for booking Certificate.

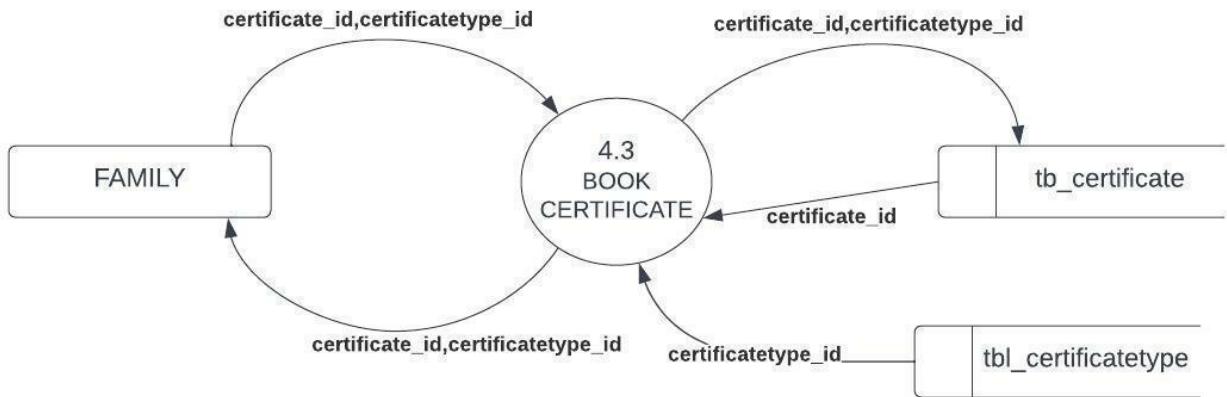


Fig 3.13 Second level DFD for Service Report..

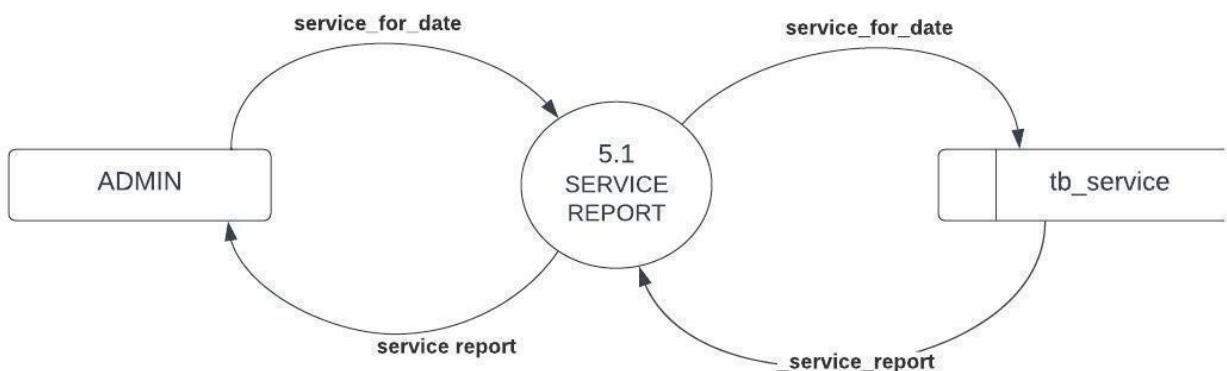


Fig 3.14 Second level DFD for Certificate Report..

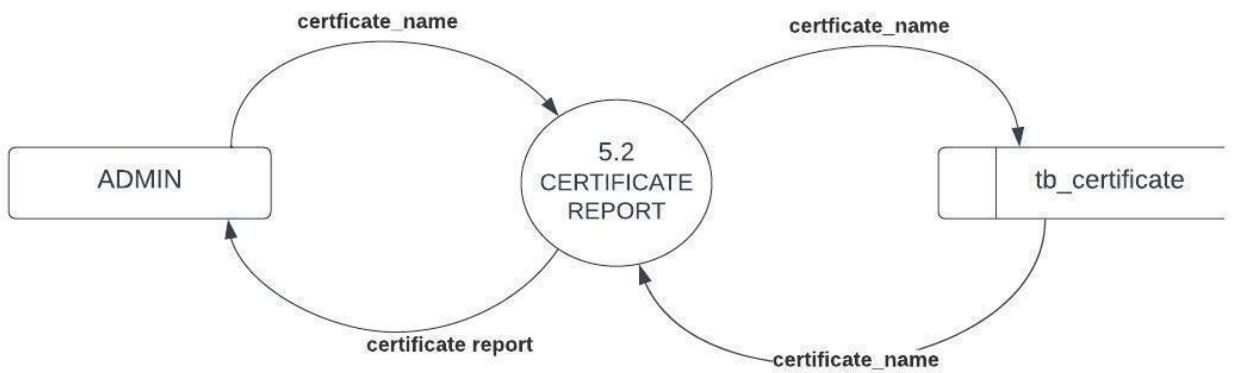
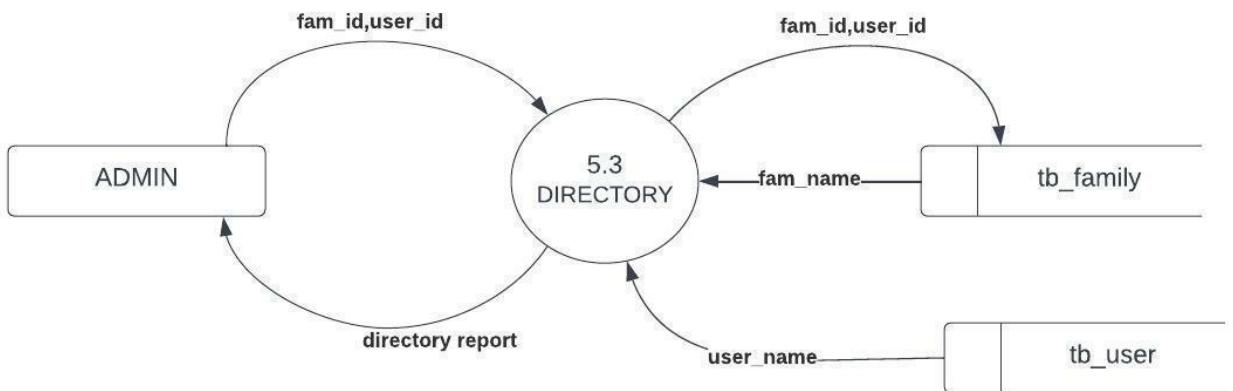


Fig 3.14 Second level DFD for Directory.



3.2 INTERFACE DESIGN

These modules can apply to hardware, software or the interface between a user and a machine. An example of a user interface could include a GUI, a control panel for a nuclear power plant, or even the cockpit of an aircraft. In systems engineering, all the inputs and outputs of a system, subsystem, and its components are listed in an interface control document often as part of the requirements of the engineering project. The development of a user interface is a unique field.

3.4.1 User Interface Screen Design

The user interface design is very important for any application. The interface design describes how the software communicates within itself, to system that interpreted with it and with humans who use it. The input design is the process of converting the user-oriented inputs into the computer based format. The data is fed into the system using simple inactive forms. The forms have been supplied with messages so that the user can enter data without facing any difficulty. They data is validated wherever it requires in the project. This ensures that only the correct data have been incorporated into system. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right messages and help for the user at right are also considered for developme

for this project.

Input Design is a part of the overall design. The input methods can be broadly classified into batch and online. Internal controls must be established for monitoring the number of inputs and for ensuring that the data are valid. The basic steps involved in input design are:

- Review input requirements.
- Decide how the input data flow will be implemented.
- Decide the source document.
- Prototype on line input screens.
- Design the input screens.

The quality of the system input determines the quality of the system output. Input specifications describe the manner in which data enter the system for processing. Input design features can ensure the reliability of the system and produce results from accurate data. The input design also determines whether the user can interact efficiently with the system.

These are the two sample input forms

The screenshot shows a web browser window with the title "Family Registration". The URL in the address bar is "localhost://jaiden/Project/Guest/familyRegistration.php". The page header includes the text "ST. MARY'S JACOBITE SYRIAN CONGREGATION PIRAVOM". Below the header are navigation links: "HOMEPAGE", "GALLERY", "SIGN UP", and "LOGIN". The main content area is titled "*****FAMILY REGISTRATION*****". It contains a form with five input fields:

- Family Name: [input field]
- Family Head Name: [input field]
- Occupation: [input field]
- No of Members: [input field]
- Address: [input field]

Placeselect.....
Kudumba Unitselect.....
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Others
Date of Birth	dd-mm-yyyy <input type="text"/>
Contact No	<input type="text"/>
Photo	<input type="file"/> Choose File No file chosen
<input type="button" value="CANCEL"/> <input type="button" value="SIGN IN"/>	

Family Registration Form

This input form is for the registration of a new family. It contains text boxes for inputting the name, email id, place, address and phone number of the family head, and a password box for giving the password. After clicking the sign in button the family details will be saved. And with this username and password he/she can login to the system.



ST. MARY'S JACOBITE SYRIAN CONGREGATION PIRAVOM

[HOMEPAGE](#) [USER](#) [DIRECTORY](#) [CERTIFICATE](#) [AUDITORIUM](#) [DONATION](#) [SERVICE](#) [VIEW AUDITORIUM BOOKING](#)

[VIEW CERTIFICATE](#) [LOGOUT](#)

*****USER REGISTRATION*****

Name	<input type="text"/>
Address	<input type="text"/>

localhost/Jaiden/Project/Family/ApplyCertificate.php
Show all
X

Date of Birth	dd-mm-yyyy	<input type="button" value=""/>
Contact No		
Photo	<input type="file"/> Choose File	No file chosen
<input type="button" value="ADD USER"/>		

Sl no	Name	Address	Email	Gender	Kudumba Unit	DoB	Contact	
1	Justin Bose	Edakkuzhiyil (H) Pazhoor P.O Piravom	justin@gmail.com	male	Unit 1	1997-08-04	8848551633	<input type="button" value="DELETE"/>
2	Beena Bose	Edakkuzhiyil (H) Pazhoor P.O Piravom	beena@gmail.com	female	Unit 1	1988-05-09	9544921516	<input type="button" value="DELETE"/>
3	Mariyamma Chacko	Edakkuzhiyil House Pazhoor P.O Piravom	mariyamma@gmail.com	female	Unit 1	1946-11-15	9446684472	<input type="button" value="DELETE"/>

User Registration Form

This input form is used to add family members of a family to our website. Family Head can add family members to this website. It contains text boxes for inputting name, email id, place, occupation and phone number of the member, and a password box for giving the password. This form allows to select gender through radio buttons. After filling all these input boxes, he should click the add user button to add the subject details to this website.

3.4.2 Output Design

A quality output is one, which meets the requirements of end user and presents the information clearly. In any system result of processing are communicated to the user and to the other system through outputs. In the output design it is determined how the information is to be displayed for immediate need.

It is the most important and direct source information is to the user. Efficient and intelligent output design improves the system's relationships with the user and helps in decision - making. The objective of the output design is to convey the information of all the past activities, current status and to emphasize important events. The output generally refers to the results and information that is generated from the system. Outputs from computers are required primarily to communicate the results of processing to the users.

Output also provides a means of storage by copying the results for later reference in consultation. There is a chance that some of the end users will not actually operate the input data or information through workstations, but will see the output from the system.

Two phases of the output design are:

1. Output Definition
2. Output Specification

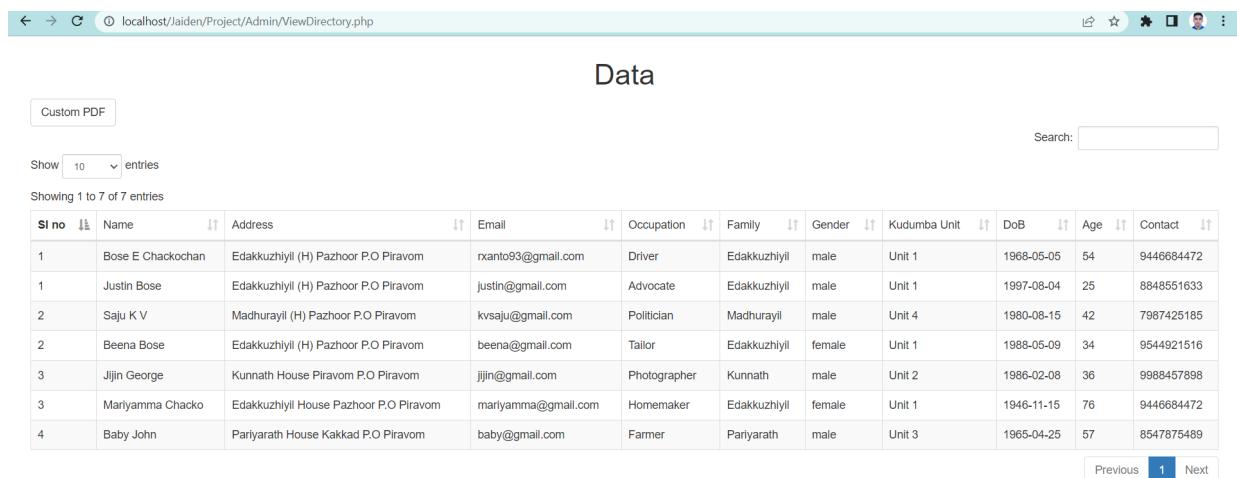
Output Definition takes into account the type of output contents, its frequency and its volume, the appropriate output media is determined for output. Once the media is chosen, the detail specification of output documents are carried out. The nature of output required from the proposed system is determined during logical design stage. It takes the outline of the output from the logical design and produces output as specified during the logical design phase.

In a project, when designing the output, the system analyst must accomplish the following:

- Determine the information to present.
- Decide whether to display, print, speak the information and select the output medium.
- Arrange the information in acceptable format.
- Decide how to distribute the output to the intended receipt.

Thus by following the above specifications, a high quality output can be generated.

In our projects outputs are generated in table format based on data.



The screenshot shows a web browser window with the URL localhost/Jaiden/Project/Admin/ViewDirectory.php. The page title is "Data". At the top, there are buttons for "Custom PDF" and a search bar labeled "Search: []". Below the header, there is a table with the following columns: Sl no, Name, Address, Email, Occupation, Family, Gender, Kudumba Unit, DoB, Age, Contact. The table contains 7 entries. At the bottom right of the table, there are navigation buttons for "Previous", "1", and "Next".

Sl no	Name	Address	Email	Occupation	Family	Gender	Kudumba Unit	DoB	Age	Contact
1	Bose E Chackochan	Edakkuzhiyil (H) Pazhoor P.O Piravom	rxanto93@gmail.com	Driver	Edakkuzhiyil	male	Unit 1	1968-05-05	54	9446684472
1	Justin Bose	Edakkuzhiyil (H) Pazhoor P.O Piravom	justin@gmail.com	Advocate	Edakkuzhiyil	male	Unit 1	1997-08-04	25	8848551633
2	Saju K V	Madhurayil (H) Pazhoor P.O Piravom	kvsaju@gmail.com	Politician	Madhurayil	male	Unit 4	1980-08-15	42	7987425185
2	Beena Bose	Edakkuzhiyil (H) Pazhoor P.O Piravom	beena@gmail.com	Tailor	Edakkuzhiyil	female	Unit 1	1988-05-09	34	9544921516
3	Jijin George	Kunnath House Piravom P.O Piravom	jijin@gmail.com	Photographer	Kunnath	male	Unit 2	1986-02-08	36	9988457898
3	Mariyamma Chacko	Edakkuzhiyil House Pazhoor P.O Piravom	mariyamma@gmail.com	Homemaker	Edakkuzhiyil	female	Unit 1	1946-11-15	76	9446684472
4	Baby John	Pariyarathe House Kakkaad P.O Piravom	baby@gmail.com	Farmer	Pariyarathe	male	Unit 3	1965-04-25	57	8547875489



Table format output based on users

SERVICE REPORT						
<input type="button" value="Custom PDF"/> <input type="text" value="Search:"/> <input type="button" value="Show all"/>						
Showing 1 to 1 of 1 entries						
Sl no	Service Type	Service Date	Service Time	Done For	Family	Done By(Priest Name)
1	House Warming	2022-11-24	19:24	Bose E Chackochan	Edakkuzhiyil	Varghese Panachiyl

Previous Next

Ruchiyyidam Docu....docx	Ruchiyyidam Docu....pdf	Show all	x
Type here to search			

Table report of services booked

4. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. The implementation stage is a systems project in its own right. It includes careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the changeover, training of the staff in the changeover procedure and evaluation of changeover method.

4.1 CODING STANDARDS

Writing an efficient software code requires a thorough knowledge of programming. This knowledge can be implemented by following a coding style which comprises several guidelines that help in writing the software code efficiently and with minimum errors. These guidelines, known as coding guidelines, are used to implement individual programming language constructs, comments, formatting, and so on. These guidelines, if followed, help in preventing errors, controlling the complexity of the program, and increasing the readability and understandability of the program.

A set of comprehensive coding guidelines encompasses all aspects of code development. To ensure that all developers work in a harmonized manner (the source code should reflect a harmonized style as a single developer had written the entire code in one session), the developers should be aware of the coding guidelines before starting a software project. Moreover, coding guidelines should state how to deal with the existing code when the software incorporates it or when maintenance is performed.

Since there are numerous programming languages for writing software codes, each having different features and capabilities, coding style guidelines differ from one language to another. However, there are some basic guidelines which are followed in all programming languages. These include naming conventions, commenting conventions, and formatting conventions.

1. **File header comments** are useful in providing information related to a file as a whole and comprise identification information such as date of creation, Dame

of

the creator, and a brief description of the software code.

2. **Trailing comments** are used to provide explanation of a single line of code. These comments are used to clarify the complex code. These also specify the function of the abbreviated variable names that are not clear. In some languages, trailing comments are used with the help of a double slash (//).
 3. **Indentation:** This refers to one or more spaces left at the beginning of statements in the program. Indentation is useful in making the code easily readable. However, the spaces used for indentation should be followed in the entire program.
 4. **Implementing coding guidelines:** If coding guidelines are used in a proper manner, errors can be detected at the time of writing the software code. Such detection in early stages helps in increasing the performance of the software as well as reducing the additional and unplanned costs of correcting and removing errors. Moreover, if a well-defined coding guideline is applied, the program yields a software system that is easy to comprehend and maintain.

4.2 SAMPLE CODE

- Adding district

```
<?php

ob_start();

include('..../Assets/Connection/Connection.php');

include('Head.php');

$did = "";

$dname = "";

if(isset($_POST['btn_save'])) {

    $district = $_POST['txt_district'];

    $query = "SELECT * FROM districts WHERE district_name = '$district'";

    $result = mysqli_query($conn, $query);

    if(mysqli_num_rows($result) > 0) {
        echo "District already exists";
    } else {
        $query = "INSERT INTO districts (district_name) VALUES ('$district')";

        $result = mysqli_query($conn, $query);

        if($result) {
            echo "District added successfully";
        } else {
            echo "Error adding district";
        }
    }
}
```

```

$ins = "insert into tb_district(district_name)
values('".$district."')";

if($conn->query($ins))
{
    header("Location:District.php");
}

}

if(isset($_GET['id']))
{
    $del = "delete from tb_district where district_id =
'".$_GET['id']."'";

    if($conn->query($del))
    {
        header("Location:District.php");
    }
}

?>

<body>

    <section class="main_content dashboard_part">
        <div class="main_content_iner ">
            <div class="container-fluid p-0">
                <div class="row justify-content-center">

```

```

<div class="col-12">
    <div class="QA_section">
        <!--Form-->
        <div class="white_box_tittle
list_header">
            <div class="col-lg-12">
                <div class="white_box mb_30">
                    <div class="box_header ">
                        <div
class="main-title">
                            <h3 class="mb-0">
                                Table District</h3>
                            </div>
                        </div>
                    <form method="post">
                        <div
class="form-group">
                            <label
for="txt_district">District Name</label>
                            <input
required="" type="text" class="form-control" id="txt_district"
name="txt_district" autocomplete="off" required>
                        </div>
                        <div
class="form-group" align="center">
                            <input type="submit" class="btn-dark" style="width:100px;
border-radius: 10px 5px " name="btn_save" value="Save">
                        </div>
                    </form>
                </div>
            </div>
        <div class="QA_table mb_30">

```


5.

TESTING

Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language. These conventions usually cover file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, etc. These are guidelines for software structural quality. Software programmers are highly recommended to follow these guidelines to help improve the readability of their source code and make software maintenance easier.

5.1 TEST CASES

The objective of system testing is to ensure that all individual programs are working as expected, that the programs link together to meet the requirements specified and to ensure that the computer system and the associated clerical and other procedures work together. The initial phase of system testing is the responsibility of the analyst who determines what conditions are to be tested, generates test data, produced a schedule of expected results, runs the tests and compares the computer produced results with the expected results with the expected results. The analyst may also be involved in procedures testing. When the analyst is satisfied that the system is working properly, he hands it over to the users for testing. The importance of system testing by the user must be stressed. Ultimately it is the user must verify the system and give the go-ahead.

During testing, the system is used experimentally to ensure that the software does not fail, i.e., that it will run according to its specifications and in the way users expect it to. Special test data is input for processing (test plan) and the results are examined to locate unexpected results. A limited number of users may also be allowed to use the system so analysts can see whether they try to use it in unexpected ways. It is preferably to find these surprises before the organization implements the system and depends on it. In many organizations, testing is performed by person other than those who write the original programs. Using persons who do not know how certain parts were designed or programmed ensures more complete and unbiased testing and more reliable software.

Parallel running is often regarded as the final phase of system testing. Since the parallel operation of two systems is very demanding in terms of user resources it should be embarked on only if the user is satisfied with the results of testing -- it should not be started if problems are known to exist. Testing is the major quality control measure during software development. Its basic function is to detect errors in the software. Thus the goal of testing is to uncover requirement design and coding errors in the program.

Testing is the process of correcting a program with the intent of finding an error. Different types of testing are,

1. Unit Testing
2. Integrated Testing
3. Black Box Testing
4. White Box Testing
5. Validation Testing
6. User Acceptance Testing

5.1.1 Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. In this testing we test each module individual and integrated the overall system. Unit testing focuses verification efforts on the smaller unit of software design in the module. This is also known as module testing. The modules of the system are tested separately. The testing is carried out during programming stage itself. In this testing step each module is found to work satisfactorily as regard to the expected output from the module. There are some validation checks for verifying the data input given by the user which both the formal and validity of the entered. It is very easy to find and debug the system.

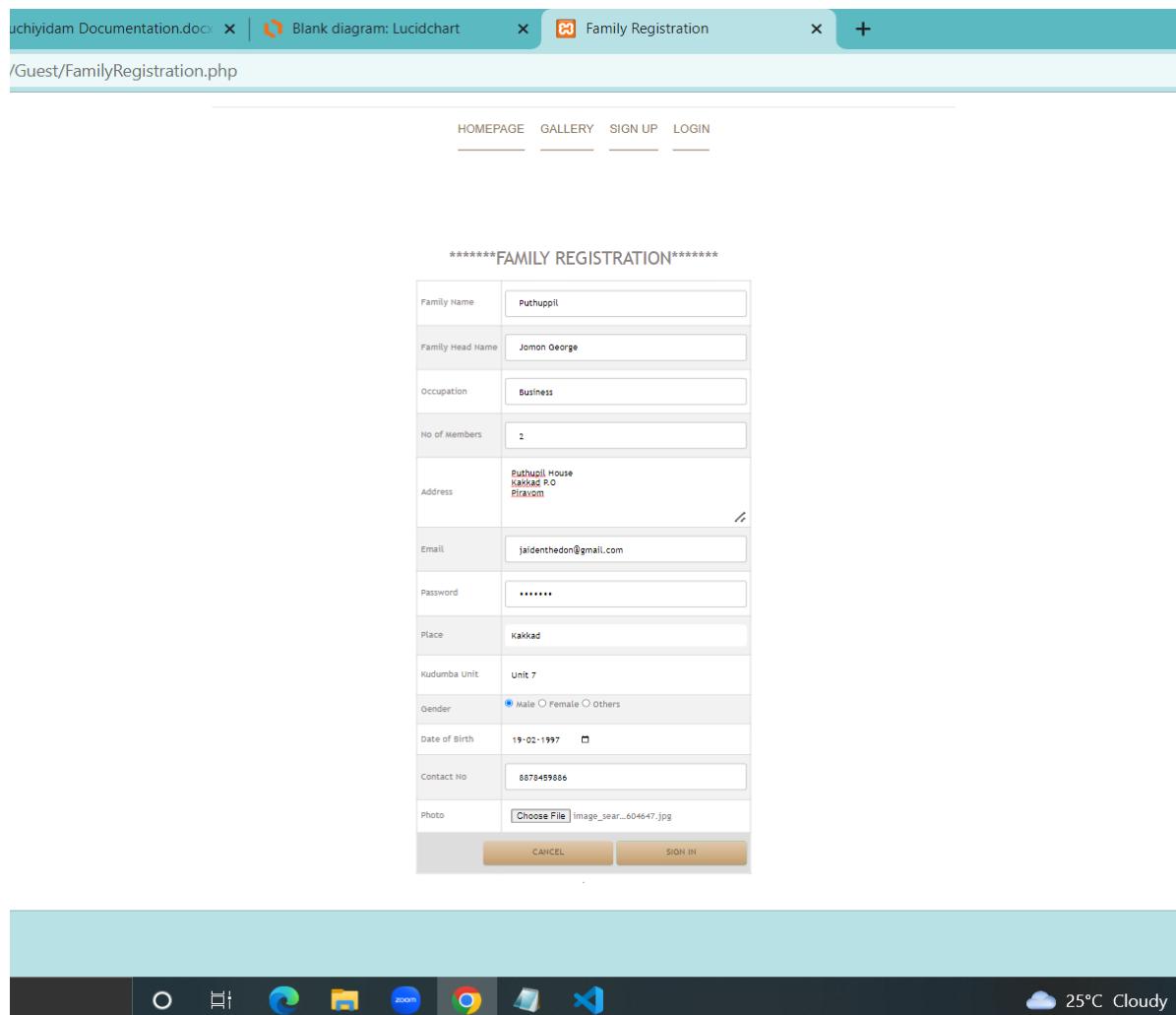


Fig 5.1 Unit Testing

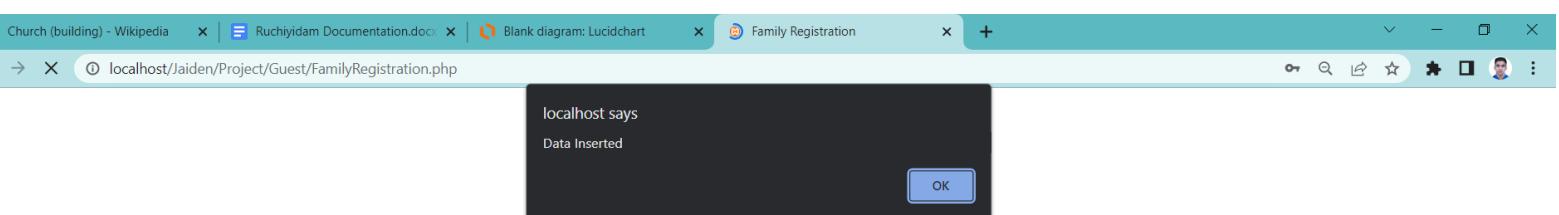


Fig 5.1 Unit Testing Result

I have continued Unit Testing from the starting of the coding phase itself. Whenever I completed one small sub module, some amount of testing was done based on the requirements to see if the functionality is aligned to the gathered requirements.

5.1.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practices since it allows interface is uses to be located more quickly and fixed. Data can be lost across an interface; one module can have an adverse effect on the other sub functions when combined by, may not produce the desired major functions. Integrated testing is the systematic testing for constructing the uncover errors within the interface. This testing was done with sample data. The developed system has run successful for this sample data. The need for integrated test is to find the overall system performance.

Integration testing is a logical extension of unit testing. In its simplest form, two units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. Integration testing identifies problems that occur when units are combined. By using a test plan that requires you to test each unit and ensure the viability of each before combining units, you know that any errors discovered when combining units are likely related to the interface between units. This method reduces the number of possibilities to a far simpler level of analysis. Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

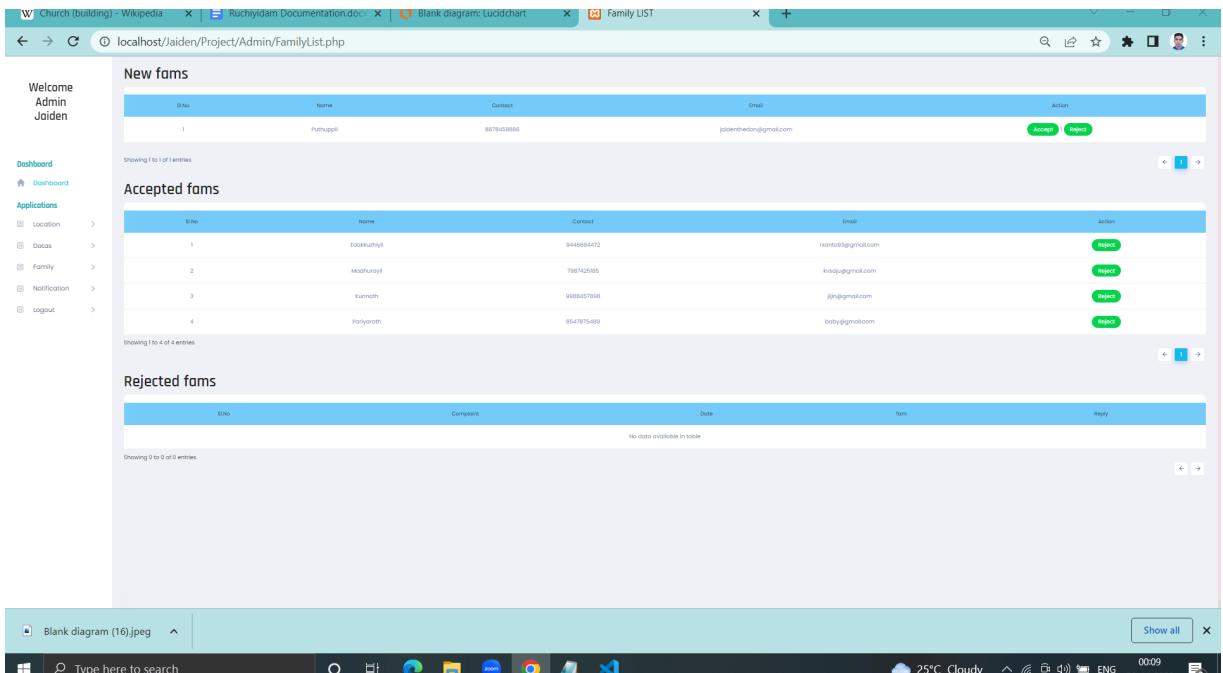


Fig 5.3 Integration Testing

I have performed integration testing whenever I have combined two modules together. When two modules are combined I have checked whether the functionality works correctly or not through integration testing.

5.1.3 Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well. In black box testing the structure of the program is not considered. Test cases are decided solely on the basis of the requirements or the specification of the program or module, and the internals of the module or program are not considered for selection of the test cases.

In the Black Box testing tester only knows the input that can be given to the system and what output the system should give. In other words, the basis of deciding test cases in functional testing is requirements or specifications of the system or module. This form of testing is also called functional or behavioral testing. One advantage of the black box technique is that no programming knowledge is required. Whatever biases the programmers may have had, the tester

likely has a different set and may

emphasise different areas of functionality. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight." Because they do not examine the source code, there are situations when a tester writes many test cases to check something that could have been tested by only one test case, or leaves some parts of the program untested.

5.1.4 White Box Testing

White-box testing (also known as clear box testing, glass box testing, and transparent box testing and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements. White Box testing is concerned with testing the implementation of the program. The intent of this testing is not to exercise all the different input or output conditions but to exercise the different programming structures and data structures used in the program.

White-box test design techniques include:

- Control flow testing
- Data flow testing
- Branch testing
- Path testing
- Statement coverage
- Decision coverage

5.1.5 Validation Testing

At the culmination of Black Box testing, software is completely assembled as a package, interface errors have been uncovered and corrected and final series of software tests, Validation tests begins. Validation testing can be defined many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably accepted by the customer. After validation test has been conducted one of the two possible conditions exists.

1. The function or performance characteristics conform to specification and are accepted.
2. A derivation from specification uncovered and a deficiency list is created.

The screenshot shows a web browser window with the title "ST. MARY'S JACOBITE SYRIAN CONGREGATION PIRAVOM". The page header includes links for HOMEPAGE, GALLERY, SIGN UP, and LOGIN. Below the header, the main content area is titled "*****FAMILY REGISTRATION*****". The form consists of several input fields:

Family Name	<input type="text"/> (highlighted in orange)
Family Head Name	<input type="text"/> (highlighted in orange) - Error message: Please fill out this field.
Occupation	<input type="text"/>
No of Members	<input type="text"/>
Address	<input type="text"/>

At the bottom of the browser window, there is a taskbar with icons for File, Search, and various system status indicators like battery level and network connection.

Fig 5.4 Validation Testing

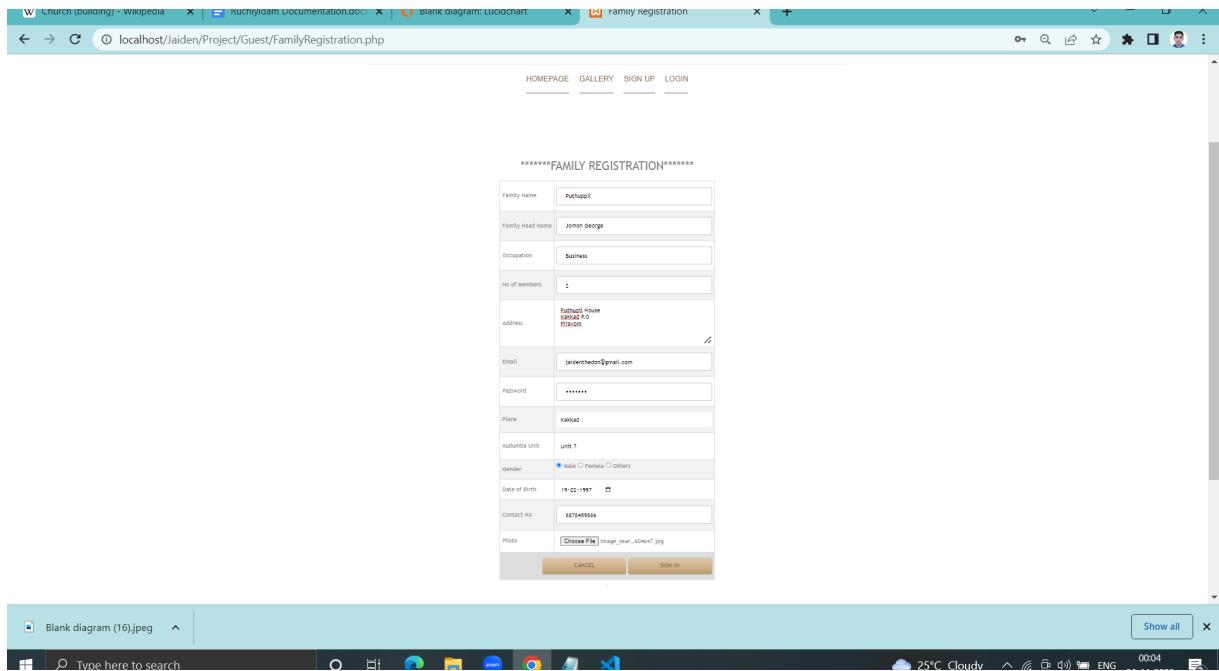


Fig 5.5 Already Exists Validation Testing



Fig 5.6 Already Exists Validation Testing Result

5.1.4 User Acceptance Testing

Acceptance Testing is a level of the software testing process where a system is tested for acceptability. User Acceptance testing is the software testing process where system tested for acceptability & validates the end to end business flow. Such type of testing executed by client in separate environment & confirms whether system meets the requirements as per requirement specification or not.

UAT is performed after System Testing is done and all or most of the major defects have been fixed. This testing is to be conducted in the final stage of Software Development Life Cycle(SDLC) prior to system being delivered to a live environment. UAT users or end users are concentrating on end to end scenarios & typically involves running a suite of tests on the completed system.

User Acceptance testing also known as Customer Acceptance testing (CAT), if the

system is being built or developed by an external supplier. The CAT or UAT are the final confirmation from the client before the system is ready for production. The business customers are the primary owners of these UAT tests. These tests are created by business customers and articulated in business domain languages. So ideally it is collaboration between business customers, business analysts, testers and developers. It consists of test suites which involve multiple test cases & each test case contains input data (if required) as well as the expected output. The result of test case is either a pass or fail.

5.2 TEST CASE DOCUMENTS

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application. A sample of test case document format is given below.

Table 5.1 Test Case

TC No.	Test Steps	Expected Result	Actual Result	Status	Comment
1	Run application and navigate to login screen	Login screen is displayed. A field for entering username, a field for entering password and a button to submit should be present	Login screen has been displayed, fields for entering email address and password together with a log in button is available.	Pass	
2	Enter an invalid username and invalid password and press the button	A message should be displayed stating that user name and password are invalid	A message has been displayed stating that user name and password are invalid	Pass	

3	Enter a valid username and password and press the button	User must successfully login to the webpages.	A message has been displayed stating that the login successful and navigate into home page	Pass	Using admin as the username and admin as the password
---	--	---	--	------	---

4	Enter a valid username and leave password and press the button	A message should be displayed stating that please enter the user name and password	A message has been displayed stating that please enter the user name and password	Pass	
5	Leave username and password and press the button	A message should be displayed stating that please enter the user name and password	A message has been displayed stating that please enter the user name and password	Pass	
6	Leave username and enter a valid password and press the button	A message should be displayed stating that please enter the user name and password	A message has been displayed stating that please enter the user name and password	Pass	

6. CONCLUSION

The project was successfully completed with in the time span allotted. All the modules are tested separately and put together to form the main system. Finally, the modules are tested with real data and it worked successfully. Thus the system has fulfilled the entire objective defined.

This project will help the user to reduce man power and consume less time access the services provided by the church without going to the church. Our goal of developing this “ESRA” has come to a good result without many defects.

The main motive for developing this system is for the welfare and easier access of the society to church by giving all time access to church.

6.1 FUTURE ENHANCEMENTS

The system has been designed in such a way that it can be modified with very little effort when such needs arise in the future. New features can be added with slight modifications of software which make it easy to expand the scope of this project. Though the system is working on various assumptions, it can be modified easily to any kind of requirements.

Even though we have tried our best to present the information effectively and efficiently, there can be further enhancement in the application. We have taken care of all the critical aspects, which were needed to be taken care of. Because of fast changes in the world of programming this system will gradually get outdated and less effective. For the time being it's possible to overcome problems by amendments and minor modifications to acknowledge the need of fundamental design. Though the new system provides a base for improving the efficiency of operations, there are a lot of future enhancements that can be added to this project. Keeping this in view, a provision has been made in the system to facilitate easy modification and updating in the future. Any modification will not affect the normal working of the system. It can also be converted into a mobile application

7. REFERENCES

- Web Programming using PHP, Dr. Jeeva Jose, Kalyani Publications.
- www.stackoverflow.com
- www.w3schools.com
- www.gihub.com
- Pamkaj Jalote-An Integrated approach to Software Engineering
,Second Edition, Narosa Publishing Company

8. APPENDIX

8.1 SCREENSHOTS

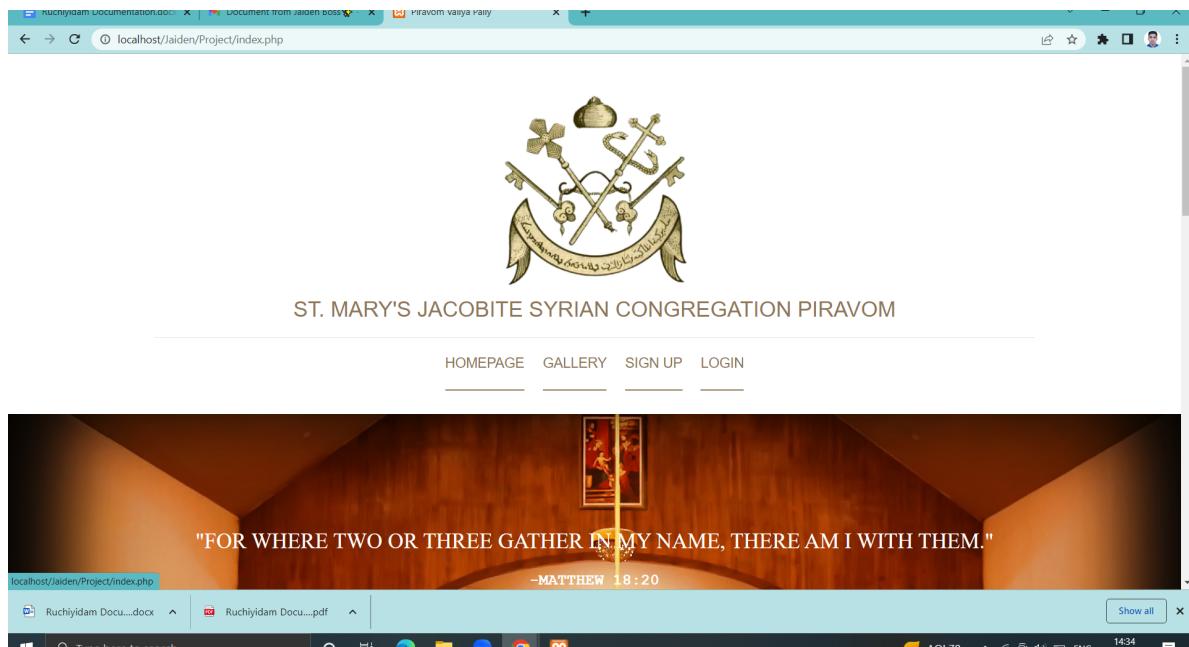


Fig 8.1 Homepage

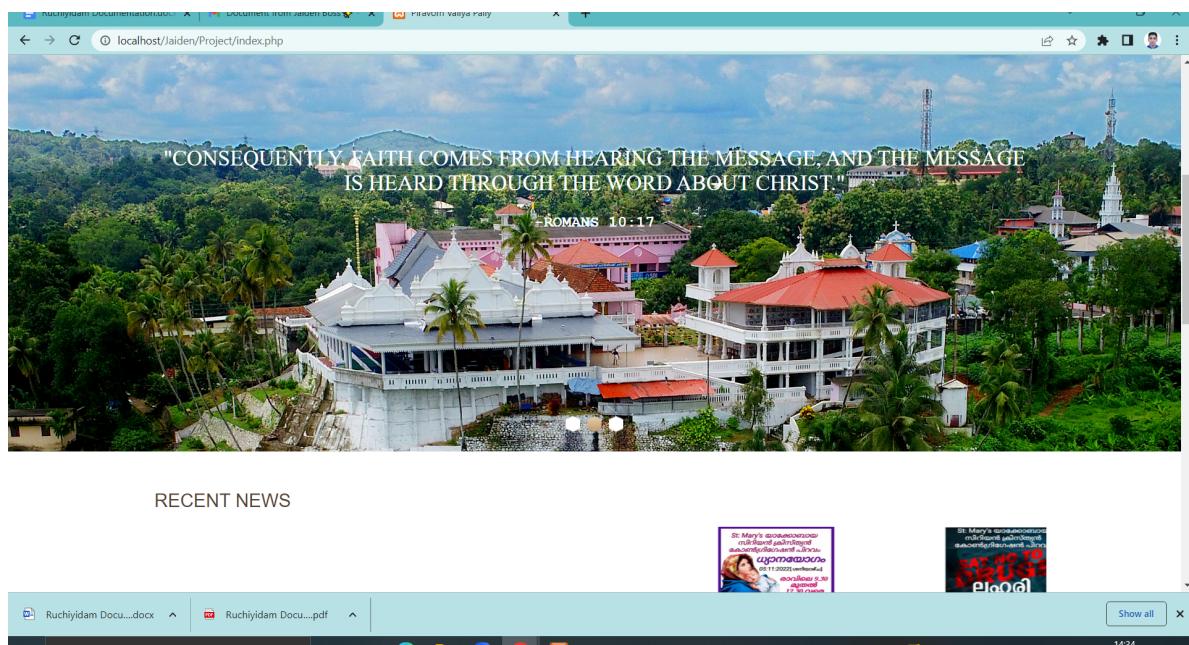


Fig 8.2 Homepage



RECENT NEWS



DHYANA YOGAM

2022-11-05

UPCOMING EVENTS

LOSOK AMBO (AWARENESS CLASS AGAINST DRUGS CONDUCTED BY SRI AJAYANADH(DYSP PUTHENCRUZ))

2022-11-13

post office junction
piravom

DHYANA YOGAM

2022-11-12

parish hall

LATEST CEREMONY

GEORGE WEDS SERENE



marriage
2022-11-13

JOBIN WEDS NEENA



Fig 8.3 Homepage

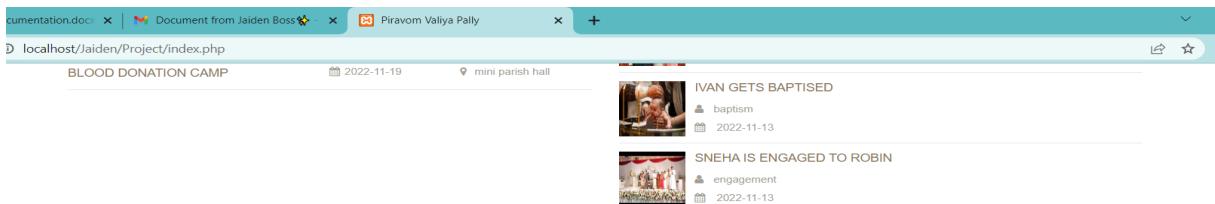


Fig 8.3b Homepage

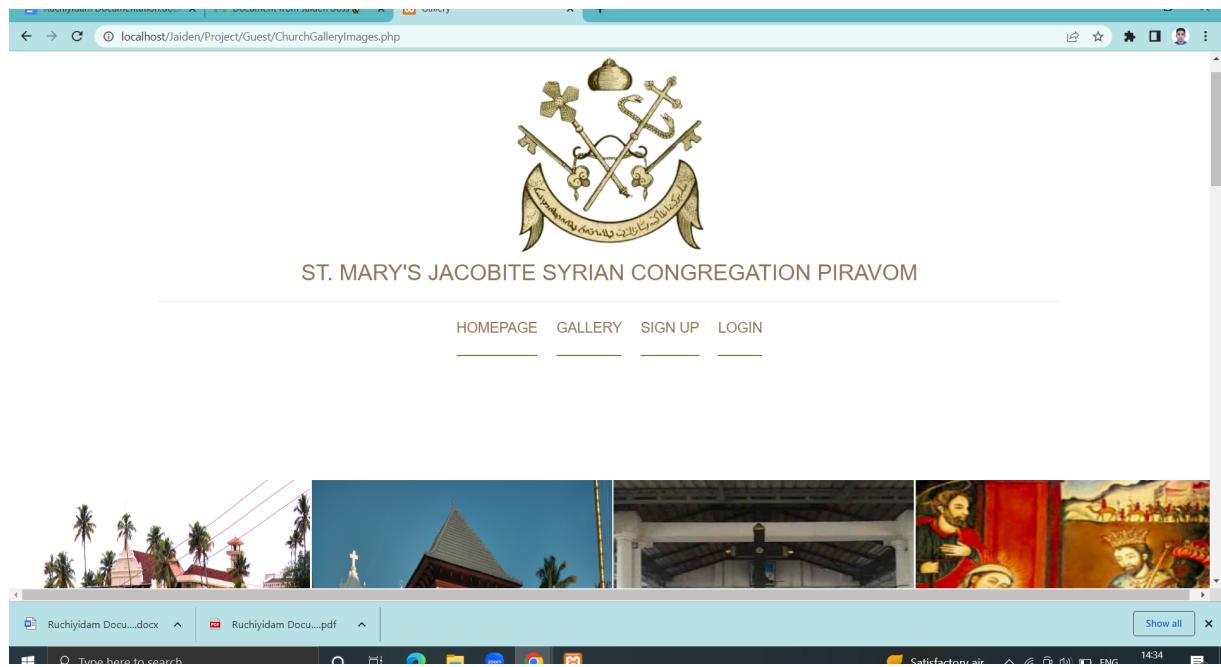


Fig 8.4 Gallery

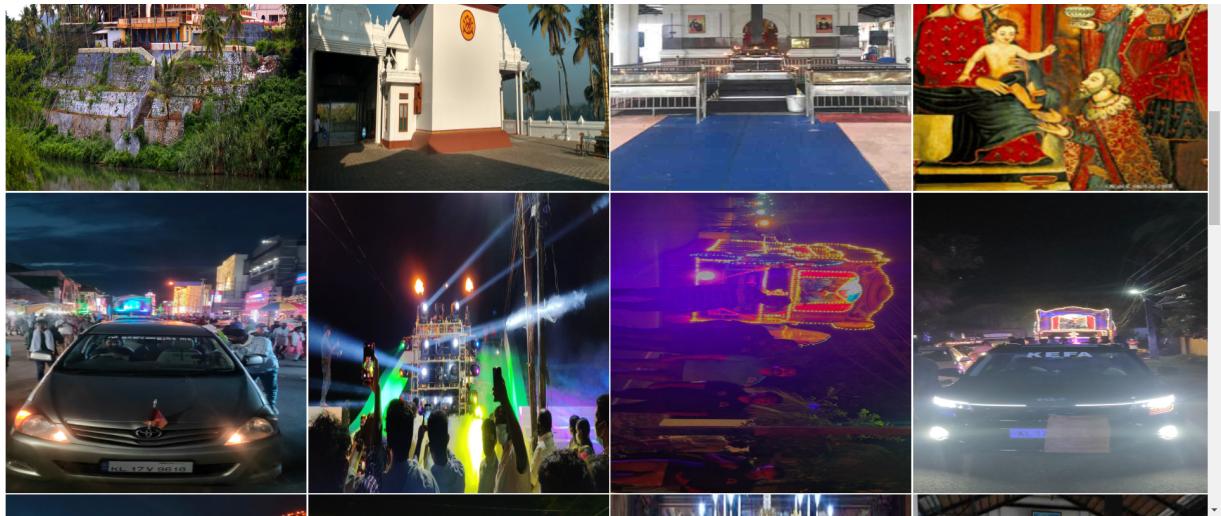


Fig 8.5 Gallery

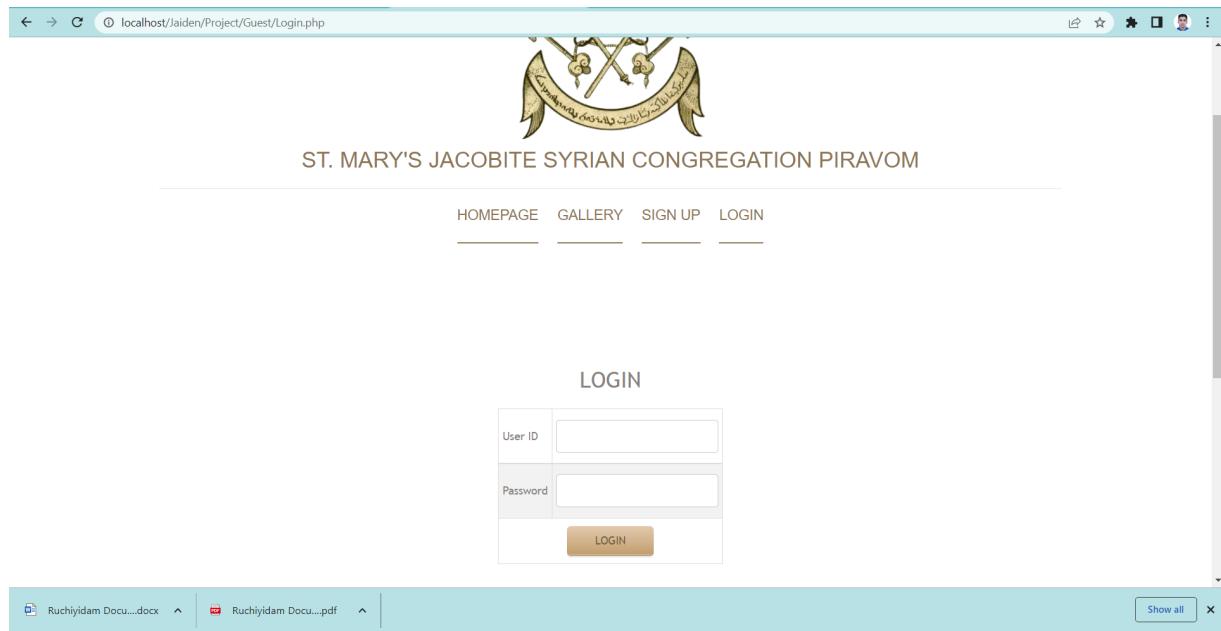


Fig 8.6 Login Page

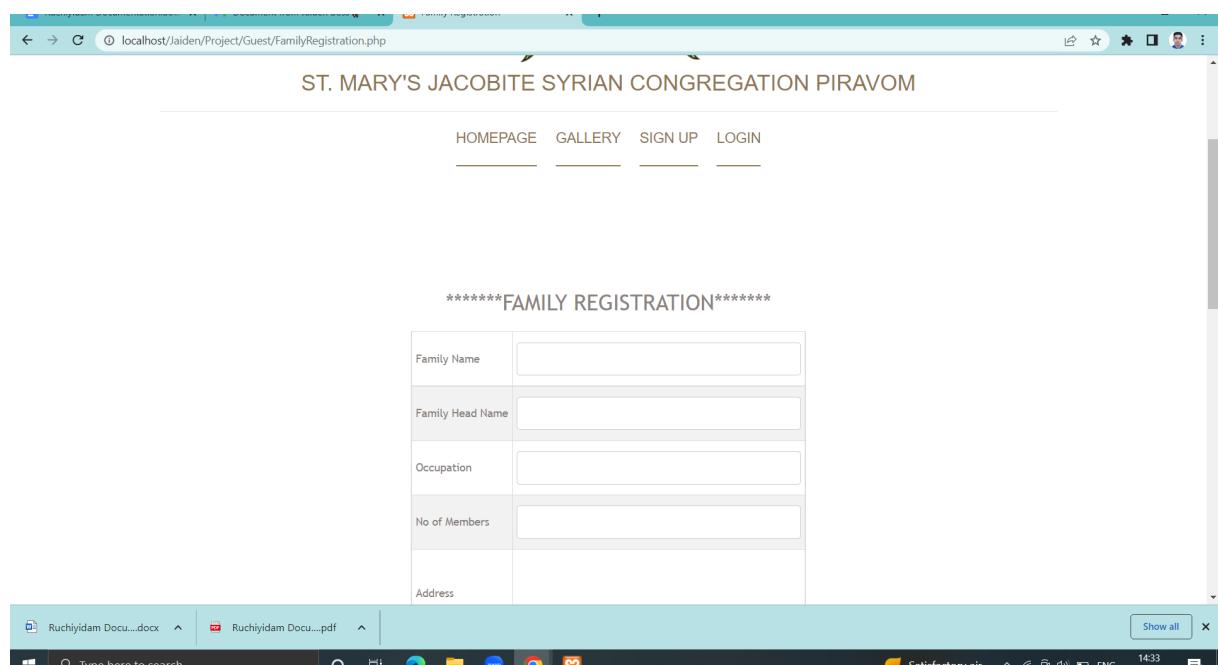


Fig 8.7 Registration Page

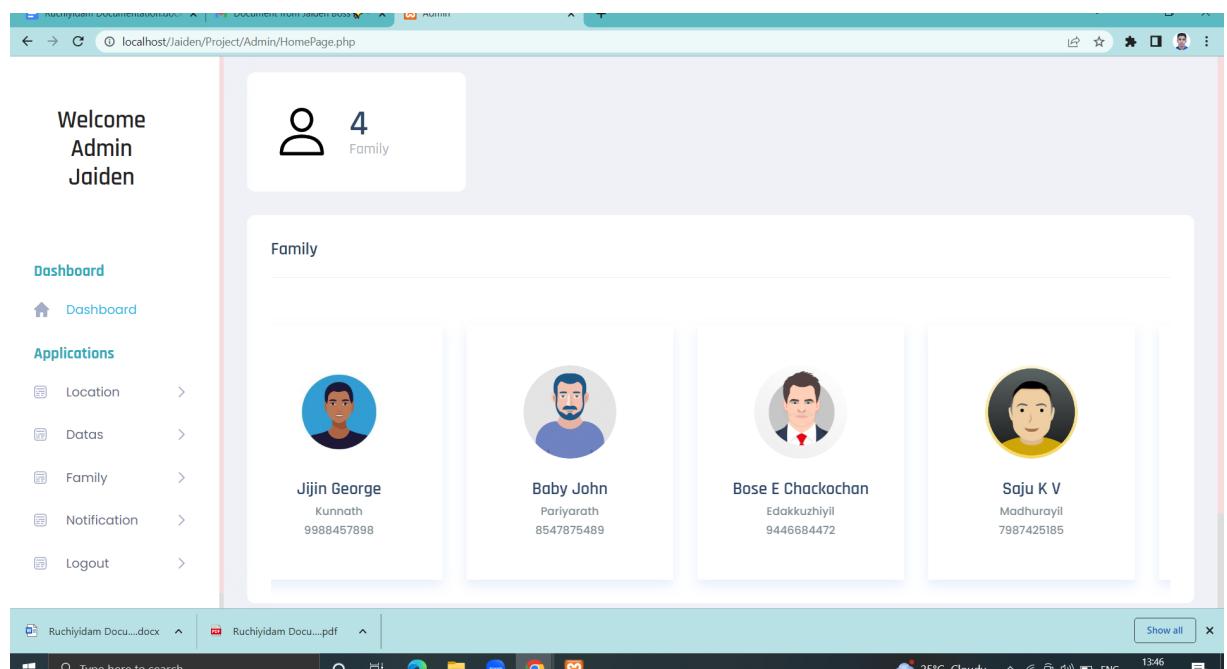


Fig 8.8 Admin Homepage

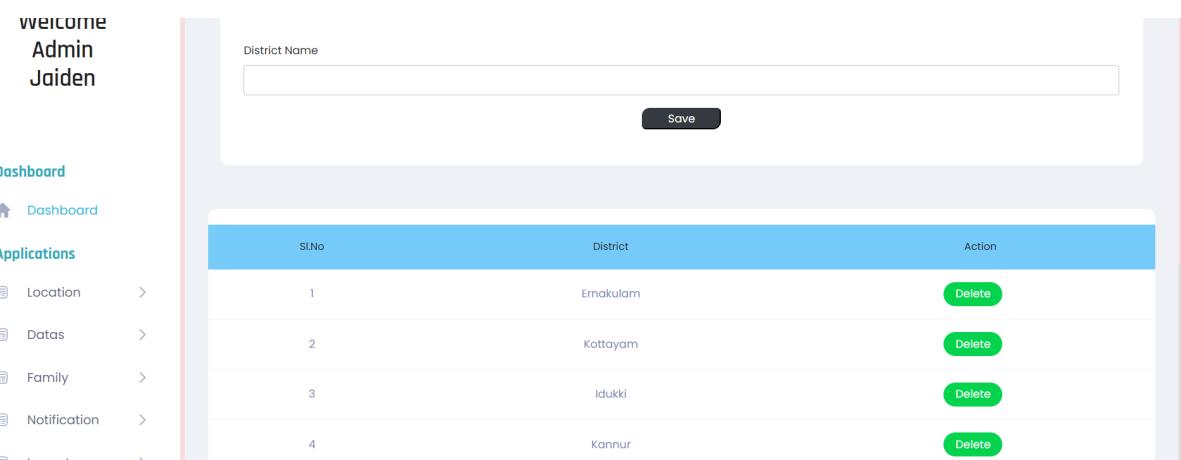


Fig 8.9 Add District Page

SI no	Service Name	Service Date	Service Time	Service Location	Comments	Requested By	Contact No.	Status
1	House Warming	2022-11-24	19:24	Mamalakavala	Near AGCM old age home, opposite to Vijaya Hotel	Bose E Chackochan	9446684472	Accepted By Varghese Panachiyil
2	Gospel	2022-11-27	20:30	Kallumary	At Kallumary St George Chappel	Gospel service is requested for the Kudumbanait meeting.	Bose E Chackochan	9446684472
3	House Warming	2022-11-17	16:33	jgeghje	geuljuv	yufif	Bose E Chackochan	9446684472

Fig 8.10 Service Details Showing Page

SERVICE REPORT						
<input type="button" value="Custom PDF"/> <input type="text" value="Search:"/> <input type="button" value="Show all"/>						
<input type="button" value="Show 10 entries"/>						
Showing 1 to 1 of 1 entries						
SI no	Service Type	Service Date	Service Time	Done For	Family	Done By(Priest Name)
1	House Warming	2022-11-24	19:24	Bose E Chackochan	Edakkuzhiyil	Varghese Panachiyil

Fig 8.11 Service Report

Data

[Custom PDF](#)

Show 10 entries

Showing 1 to 7 of 7 entries

Sl no	Name	Address	Email	Occupation	Family	Gender	Kudumba Unit	DoB	Age	Contact
1	Bose E Chackochan	Edakkuzhiyil (H) Pazhoor P.O Piravom	rxanto93@gmail.com	Driver	Edakkuzhiyil	male	Unit 1	1968-05-05	54	9446684472
1	Justin Bose	Edakkuzhiyil (H) Pazhoor P.O Piravom	justin@gmail.com	Advocate	Edakkuzhiyil	male	Unit 1	1997-08-04	25	8848551633
2	Saju K V	Madhurayil (H) Pazhoor P.O Piravom	kvsaju@gmail.com	Politician	Madhurayil	male	Unit 4	1980-08-15	42	7987425185
2	Beena Bose	Edakkuzhiyil (H) Pazhoor P.O Piravom	beena@gmail.com	Tailor	Edakkuzhiyil	female	Unit 1	1988-05-09	34	9544921516
3	Jijin George	Kunnath House Piravom P.O Piravom	jijin@gmail.com	Photographer	Kunnath	male	Unit 2	1986-02-08	36	9988457898
3	Mariyamma Chacko	Edakkuzhiyil House Pazhoor P.O Piravom	mariyamma@gmail.com	Homemaker	Edakkuzhiyil	female	Unit 1	1946-11-15	76	9446684472
4	Baby John	Pariyarath House Kakkad P.O Piravom	baby@gmail.com	Farmer	Pariyarath	male	Unit 3	1965-04-25	57	8547875489

Previous 1 Next

Fig 8.12 Directory Page

Welcome
Admin
Jaiden

[Dashboard](#)

[Dashboard](#)

[Applications](#)

- [Location](#)
- [Datas](#)
- [Family](#)
- [Notification](#)
- [Logout](#)

Accepted fams

Sl.No	Name	Contact	Email	Action
1	Edakkuzhiyil	9446684472	rxanto93@gmail.com	Reject
2	Madhurayil	7987425185	kvsaju@gmail.com	Reject
3	Kunnath	9988457898	jijin@gmail.com	Reject
4	Pariyarath	8547875489	baby@gmail.com	Reject

Fig 8.13 Family page

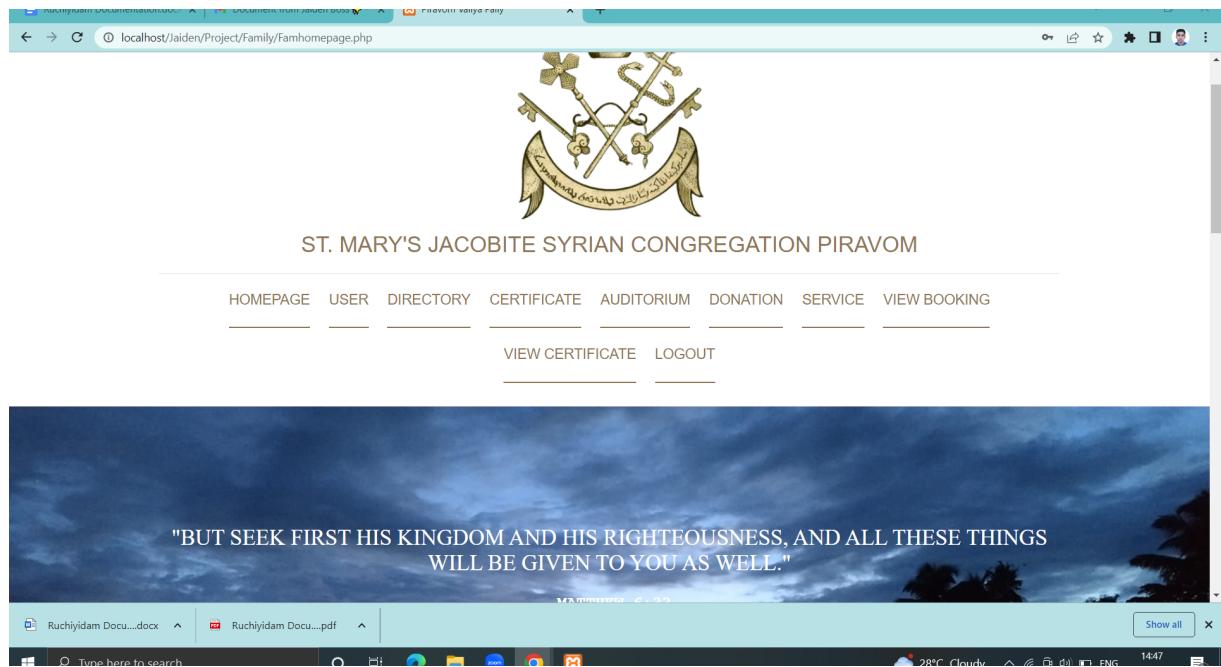


Fig 8.14 Family Homepage

The screenshot shows a web browser window with the URL localhost/laiden/Project/Family/Auditoriumbooking.php. At the top is a decorative header featuring a golden emblem with three keys and a banner. Below it is the text "ST. MARY'S JACOBITE SYRIAN CONGREGATION PIRAVOM". A horizontal menu bar follows, with items: HOMEPAGE, USER, DIRECTORY, CERTIFICATE, AUDITORIUM, DONATION, SERVICE, VIEW AUDITORIUM BOOKING, VIEW CERTIFICATE, and LOGOUT. Below the menu is a table with the following data:

Sl no	Auditorium Name	Seating Capacity	Rate	Image	Action
1	Parish Hall	1000	25000		Book/ Gallery
2	Mini Parish Hall	500	15000		Book/ Gallery

Fig 8.15 Auditorium booking page

The screenshot shows a web browser window with the URL localhost/laiden/Project/Family/Booking.php?did=1. The page has a header with links: HOME PAGE, USER, DIRECTORY, CERTIFICATE, AUDITORIUM, DONATION, SERVICE, and VIEW AUDITORIUM BOOKING. Below the header is a navigation bar with links: VIEW CERTIFICATE and LOGOUT. The main content area contains a form for booking an auditorium. The form includes fields for 'Function' (text input), 'Date(Booking To)' (date input with placeholder 'dd-mm-yyyy'), and 'Comments' (text area). At the bottom of the form are two buttons: 'BOOK' and 'CANCEL'. The background of the page shows a desktop environment with other windows open.

Fig 8.16 Auditorium booking page.

The screenshot shows a web browser window with the URL localhost/laiden/Project/Family/Payment.php. The page title is 'Payment Gateway'. The form is divided into sections: 'Account' (First Name, Last Name, Email Address), 'Date of Birth' (DD, MM, YYYY), and 'Payment Details' (Credit Card, Debit Card, Phone Number, Card Number, CVV, Expiry Date). A large green 'PAY NOW' button is at the bottom. The background shows a desktop environment with other windows open.

Fig 8.17 Payment Page

The screenshot shows a web browser window with three tabs open: 'Ruchiyidam Documentation.docx', 'Document from Jaiden Boss.pdf', and 'Book Service'. The 'Book Service' tab displays a form for booking a service. The form fields include:

Service Type	----Select----
Date	dd-mm-yyyy <input type="text"/>
Time	--:-- <input type="text"/>
Location	<input type="text"/>
Bookmark	<input type="checkbox"/>
Comments	<input type="text"/>

A large orange 'BOOK' button is at the bottom right of the form.

Below the form is a table showing a booking record:

Sl No	Service Name	Service Date	Service Time	Location	Bookmark	Comments	priest	Action
1	House Warming	2022-11-24	19:24	Mamalakkavala	<input checked="" type="checkbox"/>	Near AGCM old age home, opposite to Vilaya	Accepted by Varghese	Cancel

The browser's taskbar at the bottom shows other open documents: 'Ruchiyidam Docu...docx' and 'Ruchiyidam Docu...pdf'. The system tray indicates it's 28°C, Cloudy, and the time is 14:48.

Fig 8.18 Service Booking Page.

The screenshot shows a web browser window with three tabs open: 'Ruchiyidam Documentation.docx', 'Ruchiyidam Documentation.pdf', and 'localhost/Jaiden/Project/Family/User.php'. The 'User.php' tab displays the 'ST. MARY'S JACOBITE SYRIAN CONGREGATION PIRAVOM' website. The header menu includes links for HOMEPAGE, USER, DIRECTORY, CERTIFICATE, AUDITORIUM, DONATION, SERVICE, and VIEW AUDITORIUM BOOKING. Below the menu is a link for VIEW CERTIFICATE and a LOGOUT link.

The main content area is titled '*****USER REGISTRATION*****' and contains a form with four input fields:

Name	<input type="text"/>
Address	<input type="text"/>
Email	<input type="text"/>
Password	<input type="text"/>

The browser's taskbar at the bottom shows other open documents: 'Ruchiyidam Docu...docx' and 'Ruchiyidam Docu...pdf'. The system tray indicates it's 28°C, Cloudy, and the time is 14:52.

Fig 8.19 User Registration

Sl no	Name	Address	Email	Gender	Kudumba Unit	DoB	Contact	
1	Justin Bose	Edakkuzhiyil (H) Pazhoor P.O Piravom	justin@gmail.com	male	Unit 1	1997-08-04	8848551633	DELETE
2	Beena Bose	Edakkuzhiyil (H) Pazhoor P.O Piravom	beena@gmail.com	female	Unit 1	1988-05-09	9544921516	DELETE
3	Mariyamma Chacko	Edakkuzhiyil House Pazhoor P.O Piravom	mariyamma@gmail.com	female	Unit 1	1946-11-15	9446684472	DELETE

Fig 8.20 User Registration

"FOR WHERE TWO OR THREE GATHER IN MY NAME, THERE AM I WITH THEM."
-MATTHEW 18:20

Fig 8.21 Priest Homepage

Fig 8.22 Priest Profile

Fig 8.23 Service requests shown on priest account.