**E-PUMP**

**PROJECT REPORT**

***Submitted in partial fulfillment of the requirements for the award of BCA (Bachelor of Computer Application) Degree University of Kerala***

**SUBMITTED**

**By**

**ASWADEV G (Reg No:33218815007)**

**JIJO J S(RegNo:33218815015)**

**SANISH BABU (Reg No:33218815025)**

**Under the Guidance of**

**Mrs. Renjini J**

******

**DEPARTMENT OF COMPUTER APPLICATION**

**SREE NARAYANA COLLEGE OF TECHNOLOGY**

**VADAKKEVILA, KOLLAM-10**

**University of Kerala**

**Thiruvananthapuram**

**2021**

**SREE NARAYANA COLLEGE OF TECHNOLOGY**

**VADAKKEVILA, KOLLAM-10**

**University Of Kerala**

**CERTIFICATE**

Certified that this report entitled **“E-PUMP”**is a bonafide record of project work done by Smt. **JIJO J S, SANISH BABU, ASWADEV G** under our supervision and guidance, towards partial fulfillment of the requirements for the award of the Degree of**Bachelor of Computer Applicationof the University of Kerala.**

External Examiners Project Guide Head of the Department

Mrs.Renjini J Mrs. Salini S Nair

**DECLARATION**

We hereby declare that the project work entitled **“E-PUMP”** submitted in the Kerala University in the partial fulfillment of the requirements for the award of Degree inBCA (Bachelor of Computer Application) is a record of original work under the guidance of **Mrs. Renjini J,** Asst. Professor in the Computer Application Department, SreeNarayana College OfTechnology during our period of study in SreeNarayana College Of Technology,Vadakkevila.

We also declare that this project has notbeen submitted to any other University or Institute for the award of any degree, diploma, associateship, fellowship and other similar title or recognition.

**Signature**

**JIJO J S (Reg No:33218815015)**

**SANISH BABU (Reg No:33218815025)**

**ASWADEV G (Reg No:33218815007)**

**Countersigned**

**Head of the Department of Computer Application.**

**Sree Narayana College of Technology,**

**Vadakkevila, Kollam**

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning about the people who made it possible, whose constant guidance and encouragement has crowned the efforts with success. We would like to thank almighty god all blessings he has bestowed upon us to complete our work.

We take immense pleasure in expressingour heartfelt indepthness to our **PrincipalDr.AnithaSankar C** as to conduct and successfully complete this project. With heartful thanks,I would like to take up this opportunity to wish my internalguide**Mrs. RenjiniJ,Department of Computer Applications, Sree Narayana College of Technology** for the endless support,encouragement and suggestions in various stages of the development of the project. We also wish to extend our sincere and hearty respect and thanks **to Mrs.Salini S Nair, Head of the Department of Computer Applications, SreeNarayana College of Technology.**

We also wish to extend our sincere and hearty respects andthanks to **Mrs. Cindiya G Nathan,Mrs. Deepa Rajendran , Mrs. AswathyVR,Mr.Kannan S** who helped us to make the project a success.

Last, but not least we would like to thank our parents and friends for the support and help they provided to complete our project.

.

With gratitude,

**ASWADEV G**

**JIJO J S**

**SANISH BABU**

**ABSTRACT**

Data mining is the hot topic of the day. All organizations are moving towards implementation of IOT into their business for effective development. The project titled “E-Pump” is aimed at automating a petrol pump by providing support to the customer outside the organization. It is awebbased application, is to manage the booking and delivery of petrol and diesel to the customer in a particular location. It mainly focuses on helping people who get stuck in a place when no petrol pumps are nearby you can so on have the comfort of buying them on one click,without going to the fuel pump. And also the system provide the job opportunity in each petrol pumps. The database includes information about registered users & registered pump. The project has three levels of access, admin level, pump level and registered user level. These are the three modules used in the system. In admin module, admin can update the current fuel price, view & approve registered pump and orders, view feedback ect. In pump module, users are provided the option to book fuel from a remote location which will be reached by Google cloud services. Registered pump can view orders, add vacancies, approve applications, list pump and stock details, keep pump records. And the registered user module have provision to order fuel, view their previous orders, and apply for new job.

This project is developed using PHP as server scripting language and Mysql as database.Php source is widely used. Open scripting language and Mysql as backend.

**TABLE OF CONTENTS**

**Sl.No. Topic Page No.**

1. **INTRODUCTION** 1

2. **ORGANIZATIONAL PROFILE** 2

3. **SYSTEM ANALYSIS** ­­­­­­ 4

3.1 Existing System 4

3.2 Proposed system 5

3.3 Feasibility Analysis 5

3.3.1 Technical Feasibility 6

3.3.2 Operational Feasibility 6

3.3.3. Behavioral Feasibility 6

3.3.4 Economic Feasibility 7

3.4 System Specification 7

3.4.1 Software Specification 7

3.4.2 Hardware Specification 7

3.5 Software Requirements 8

3.5.1 Overview of PHP 8

3.5.2 HTML 9

3.5.3 WAMP 8

3.5.4 MYSQL 9

3.5.5 Apache Web Server 11

3.6 Data Flow Diagram(DFD) 10

3.7 Structure Chart 17

3.8 Menu Tree 20

3.9 System Flow Chart 21

4**. SYSTEM DESIGN**  27

4.1 Module Description 28

4.2 Input design 29

4.2.1 Input Objectives 29

4.3Output design 30

4.3.1 Output Objectives 30

4.4 Database design 31

4.4.1 Normalization 31

4.4.1.1 First Normal Form(1NF) 32

4.4.1.2 Second Normal Form(2NF) 32

4.4.1.3 Third Normal Form(3NF) 32

4.5 TABLE DESIGN 33

5.**SYSTEM CODING** 36

5.1 Program List 36

5.2 Source Code 36

6.**SYSTEM TESTING** 53

6.1 Unit Testing 54

6.2 Integration Testing 55

6.3 System Testing 55

7.**SYSTEM IMPLEMENTATION** 57

8.**SYSTEM MAINTENANCE**  58

8.1 Types of System Maintenance 58

8.1.1 Corrective Maintenance 58

8.1.2 Adaptive Maintenance 58

8.1.3 Perfective Maintenance 59

9.**CONCLUSION** 60

10.**FUTURE ENHANCEMENT** 61

**APPENDIX A: GANTT CHART** 62

**APPENDIX B : MEETING MINUTES** 63

**APPENDIX C: SCREEN LAYOUTS& REPORTS** 65

**APPENDIX D : TEST CASES** 69

11.**BIBLIOGRAPHY** 70

**CHAPTER-1**

**INTRODUCTION**

In current days mainly in our state the fuel pumps are operated manually. These fuel pumps cause time consuming and require more man power. In the current system have so many difficulties. Such that, there is no proper method for the booking and delivery of petrol and diesel to the customer in a particular location for helping people who get stuck in a place when no petrol pumps are nearby. It is very difficult to locate a nearest petrol pump using the current system. We can only see the nearest fuel stations from google map. To place petrol pumps in distant area is very costly to provide excellent facility to the customers. All these problems are sorted out by the use of IOT E-Pump, which requires less time to operate and it is effective and can be use anywhere from kollam district.

**CHAPTER-2**

**ORGANIZATIONAL PROFILE**

Technology Centre for Software and Research (TCSR) is a dynamic group of highly competent IT professionals providing cost effective and cutting-edge IT solutions with integration of Data Mining/Research tools and Applications. Their focus is

greater return for their client’s technology investment. They are a dynamic company where teamwork and common goals are the main driving force. Expert and timely delivery of customized solutions is what they can do best.

Technology Centre for Software and Research (TCSR) is a professional web design company, offering first-class website development services, Internet Marketing and Promotion, E Commerce and Content Management solutions to clients. Their services include new domain name search and registration, site design and development (including photography, graphics, and writing), and ongoing maintenance of your sites.

* Custom Design & Layout
* Custom HTML Programming
* Custom Graphic Design (GUI)
* Professional Digital Photography
* Library of Over thousands of Professional Web Templates
* Available JavaScript, Java, Perl, ASP, .Net, VC++, PHP, VBprogramming
* Flash & Shockwave Programming Technology
* Online Forms & E-mail
* Search Engine Optimization (SEO)
* Custom Website Performance Statistics
* Business Analytics Company’s Mission Statement

Technology Centre for Software and Research (TCSR) powering is innovative, cost-effective and quality solutions in Web applications, Intelligent Systems and Software Applications.

**CHAPTER-3**

**SYSTEM ANALYSIS**

This project is aimed at automating a fuel pump by providing support to the customer outside the organization. Data Mining is the hot topic of the day. All organizations are moving towards implementation of IOT devices into their business for effective development. IOT based E-Pump is a web based application, is to manage the booking and delivery of petrol and diesel to the customer in a particular location. It mainly focuses on helping people who get stuck in a place when no fuel stations are nearby. You can soon have the comfort of buying them on one click, without going to the fuel pump. In our system admin can check which employee sold how much petrol and what amount he submitted to account at the end of the day.

**3.1 EXISTING SYSTEM**

Today we are waiting in long queues at fuel stations, refuel while you are at home or work. It is very time consuming. And much more rush to fuel station to fetch fuel. The main aim of the system is to manage the booking and delivery of petrol and diesel to the customer in a particular location by the oil marketing companies. In existing system almost all petrol pumps have a controlling unit to perform the tasks like managing collect the money etc. petrol or diesel is available to the customer when they reached the pump. In present system all report work is done manually at the petrol pump. Admin user have to check each petrol pump machine to calculate the sold petrol, but in our proposed system user can check sold petrol from his monitor by clicking on some report button.

**3.2 PROPOSED SYSTEM**

IOT BASED E-PUMP is a webbased application, is to manage the booking and delivery of petrol and diesel to the customer in a particular location. It mainly focuses on helping people who get stuck in a place when no petrol pumps are nearby you can soon have the comfort of buying them on one click, without going to the fuel pump. As the work at petrol pump is done manually so it becomes hard for the supervisor to maintain a daily record. So we have developed this system to computerize all the reporting work of petrol pump. Our project gives the supervisor to maintain a daily record in an easiest way. Admin can add new pump and can also approve customer to the system. User can also check the petrol diesel price online. Admin can also find contact details of every customer from the system.He can find detail like his address , phone number , native address, current location of the customer, request from customers etc. Admin can update many information in the system that’s why his account is protected with user name and password. Only authentic user can login to admin account

**3.3FEASIBILITY ANALYSIS**

The main objective of the feasibility study is to treat the Technical, Operational, Logical and Economic feasibility of developing the computerized system. All systems are feasible, given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at System study phase itself. The feasibility study to be conducted for this project involves.

* Technical Feasibility
* Operational Feasibility
* Economic Feasibility

**3.3.1 Technical Feasibility**

The technical feasibility assessment is focused on gaining and understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

* + - The software and hardware requirements for the development of this project are already available.
    - The available equipment’s have the capacity to hold the data

required to use the new system.

* + - This system can be upgraded after the development.
    - The proposed system provides adequate responses to the users inquiry.

**3.3.2 Operational Feasibility**

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives.

* + During the development there is sufficient support for the management

from the users.

* + This system will be used and worked properly if it is being developed andimplemented.
  + There is no resistance from the user that will undermine the possible application benefits.
  + The well-planned design would ensure the optimal utilization of the computer resources

**3.3.3 Economic Feasibility**

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide .It includes quantification and identification of all the benefits expected. This assessment typically involves a cost benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with cost.

9The proposed system is economically feasible because of the followings

* In economic feasibility , the main purpose is to check the system is cost effective or not
* This system does not require any additional hardware or software.
* Financial benefit is equal to the cost

**3.4 SYSTEM SPECIFICATION**

**3.4.1 Software Specification**

Operating System : Windows 10

Server Side Programming :PHP

Markup Language :HTML

Database : MySQL

Designing tool: Adobe Macromedia Dream Weaver

Browser : Internet Explorer/ Mozilla Firefox

**3.4.2 Hardware Specification**

Processor : Core I3 processor

RAM : 4.00 GB

Hard disk : 500 GB

Keyboard : QWERTY Keyboard

Mouse : Optical Mouse

Monitor : LCD Monitor

**3.5SOFTWAREREQUIREMENTS**

**3.5.1Overview of PHP (PHP-5.0):**

PHPis a widely-used Open Source general-purpose scripting language that is especially suited for Web applications and can be embedded into HTML. It is a free software*.* Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly. One of the most important factors driving PHP’s popularity over the last couple of years has been its support for a variety of databases, including MySQL, Oracle, and Microsoft Access. One of the most powerful combinations in the open source arena today is the PHP/ MySQL combination. Like PHP,MySQL has open-source roots: it is a fast and reliable database management system that is rapidly acquiring a worldwide user base. By using PHP and MySQL together, users can benefit from the cost savings that accompany community-driven software, and also leverage off the immense number of freely available PHP/MySQL applications to reduce development time.

Typically, PHP code is embedded inside a regular HTML document, and is recognized and executed by the web server when the document is requested through a browser. Because PHP is a full-featured programming language, you can code all manner of complex thin gummies into your web pages using this technique; the server will execute your code and return the output to the browser in the format you specify. Because PHP code is executed on the server and not on the client, developers don’t have to worry about browser-specific quirks that could cause the code to break (as commonly happens with JavaScript); PHP code works independently of the user’s web browser.

**Advantages of PHP**

1. Vs. Pure Servlets - It is more convenient to write (and to modify) regular HTML than to have a zillion print statements that generate the HTML
2. Vs. Server-Side - Includes (SSI).SSI is a widely support technology for including externally defined pieces into a static web page.
3. Vs. JavaScript - JavaScript can generate HTML dynamically on the client. This is the useful capability, but only handles situations where the dynamic information is based on the client’s environment. With the exception of cookies, HTTP and form submission data is not available to JavaScript. And, since it runs on the client, JavaScript can’t access server-side resources like databases, catalogs, pricing information, and the like.
4. Vs. Static HTML - Regular HTML, of course, cannot contain dynamic information. PHP is so easy and convenient that it is quite feasible to augment HTML pages that only benefit marginally by the insertion of small amounts of dynamic data. Previously, the cost of using dynamic data would preclude its use in all but the most valuable instances.

**Syntax and Semantics**

PHP only parses code within its delimiters. Anything outside its delimiters is sent directly to the output and is not parsed by PHP. The most common delimiters are **<?php** and **?>**, which known as open and close delimiters respectively in the PHP language. **<script language="php">**and **</script>** delimiters are also available. Short tags can be used to start PHP code, <? or <?= (which is used to echo back a string or variable) and the tag to end PHP code, **?>**.

**3.5.2 HTML (Hyper Text Markup Language)**

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

**Advantages:**

1. A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
2. HTML is platform independent.
3. HTML tags are not case-sensitive.

**3.6 DATA FLOW DIAGRAM (DFD)**

Data Flow Diagram was first developed by Larry Constantine asa way of expressing system requirements in a graphical form; this led to modular design. A Data Flow Diagram, also known as “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. A DFD consist of a series of bubbles joined by lines.

Data Flow Diagram is useful in understanding a system and can be effectively used for partitioning during analysis. A Data Flow Diagram models a system by external entries from which data flows to a process which transforms the data and create output data which goes other process as input. The main merit of data Flow Diagram is that can provide an overview of what data a system would process , what information of data are done, what files are used and where the result flow. The graphical representation of the system makes it good communication tool between the user and the analyst.

It is difficult to understand the business through verbal description alone. Here Data Flow Diagrams helps in illustrating the essential component of a process and the way they interact. The symbols used in the Data Flow Diagram has been explained below:

**3.6.1. Data Flow Notations**

**External Entities**

This represents any outside agency, which interact with the system. It represents the source or destination of data for the system under the consideration.

Rectangle

**Process/Function**

A process represents activities in which data is manipulated by being stored or retrieved or transferred in some way. The process transforms the input data into output data.

Circle

**Data flows**

The data flow portrays in an interface among different components in a DFD. It represents flow of data between two processes or between a process and an external entity or between a process and data store.

Arrow

**Data Stores/Database**

A data store represents a logical life. Data store is represented by open ended rectangle. Each data store is connected to a process by means of a data flow symbol.

Open Rectangle

A circle is used to represent a process. A rectangle is used to represent source and destination of data. These are called external entities. The entities that supplying data are known as source and those that consume data are called destinations. An open rectangle is used to represent a data store and arrows represents data flows. Also the arrows shows the direction of data flows.

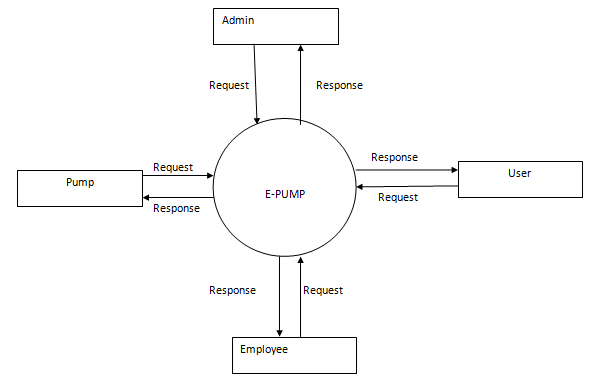
A few guidelines to aid the derivation of a DFD:

* The level 0 DFD should depict the software/system as a single bubble.
* Primary input and output should be noted.
* Refinement should begin by isolating candidate processes, data objects and the stores to be represented in the next level.
* All arrows and bubbles should be labeled with a meaningful name.
* Information flow continuity must be maintained from level to level and.
* One bubble at a time should be refined.

The level 0 DFD is expanded to level 1 DFD. The process represented in the level 1 DFD can be further refined to level 2 DFD. Information flow continuity is maintained between levels. The refinement continues until each bubble performs a simple function. That is, until the process represented by the bubble performs a function that would be easily implemented as a program. The DFD serves two purposes:

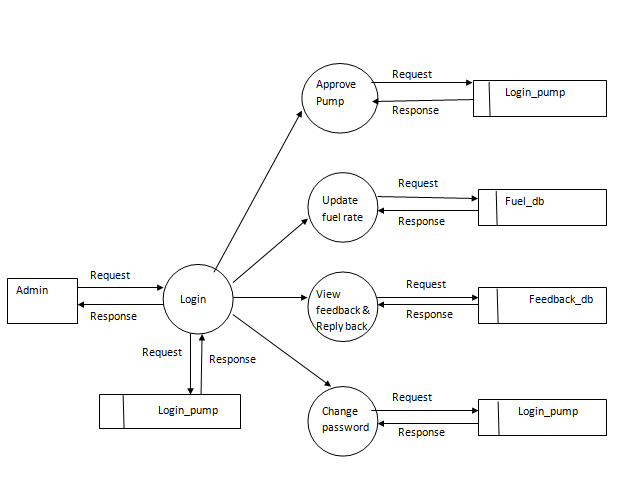
To provide an indication of how data is transformed as they move through the system,and to depict the functions (and sub functions) that transforms the data flow. DFD provide additional information that is used during the analysis of the information domain and serves as a basis for the modeling of functions.

Level 0 DFD

****

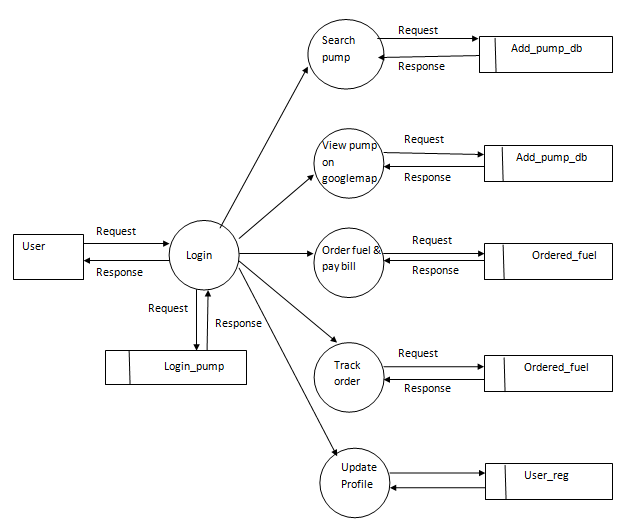
**Figure 3.6.1: Level 0 DFD**

Level 1 DFD – ADMIN



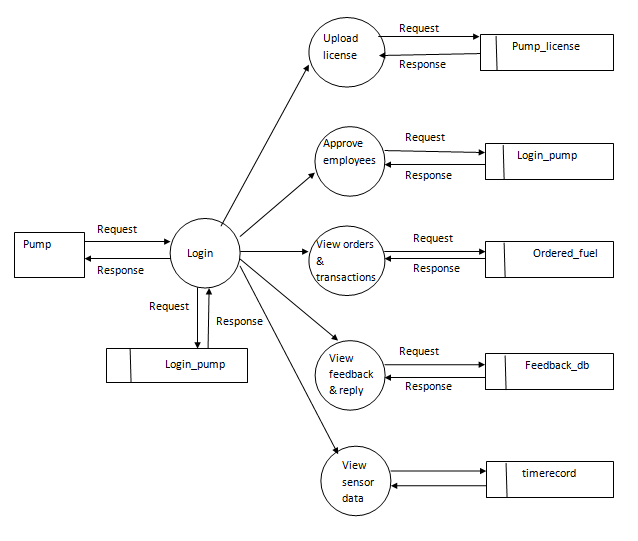
**Figure 3.6.2:Data Flow Diagram for Admin**

Level 2 DFD – User



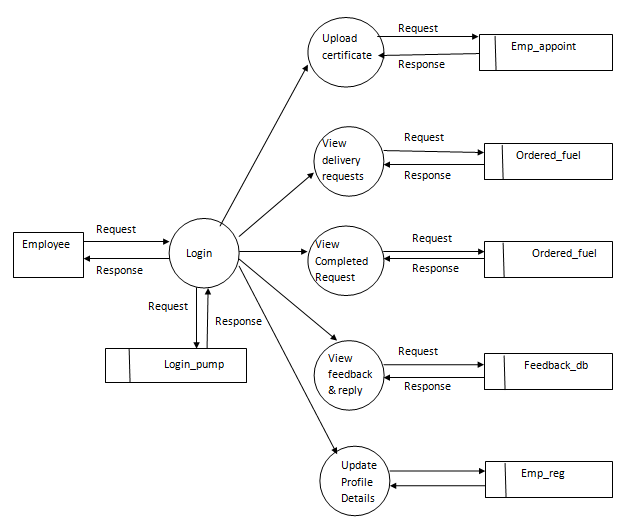
**Figure 3.6.3:Data Flow Diagram for User**

Level 3 DFD –Pump



**Figure 3.6.3:Data Flow Diagram for Pump**

Level 4 DFD –Employee



**Figure 3.6.5:Data Flow Diagram for Employee**

**3.7 STRUCTURE CHART**

Structure chart is a design tool that pictorially shows the relation between processing module in computer software. Describes the hierarchy of components and the data are transmitted between them. Includes analysis of input-to-output transformations and analysis of transaction. Structure charts show the relation of processing modules in computer software. Structure charts are developed prior to the writing of program code. They identify the data passes existing between individual modules that interact with one another.

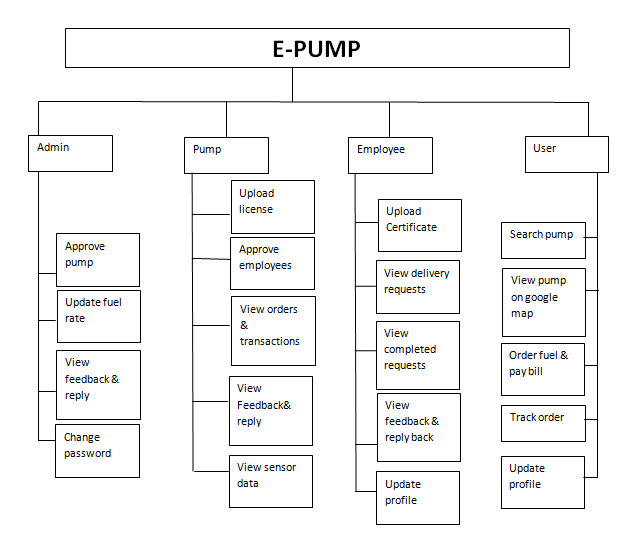
The basic building blocks using the structure are:

* A rectangle box represents a module (level). It is notated with the same name of the module.

An arrow connecting two modules implies that during program execution, control is passed from one module to another in the direction of the connecting arrow.

* Data flow arrows are small arrows appearing along the side module invocation arrows. It is associated with the corresponding data same.

A structure chart (SC) in software engineering and organizational theory is a chart, which shows the breakdown of the configuration system to the lowest manageable levels. This chart is used in structured programming to arrange the program modules in a tree structure. Each module is represented by a box, which contains the modules name. The tree structure visualizes the relationships between the modules. A structure chart is a topdown modular design tool constructed of squares representing the different modules in the system, and the lines that connect them. The lines represents the connection and or ownership between activities and sub activities as they are used in organization chart.

****

**Figure 3.7.1 Structure Chart**

**3.8 MENU TREE**

Menu tree is also helpful for representing the simplified version of the system. Which is in the form of a tree structure. Here firstly the entire system taken as the main part, and them each modules coming in that system can be represented as the branches and their functions of each module can be represented as same as leaf in the tree. So menu tree is the hierarchical representation of entire system, so it is very helpful for identifying the system easily.

**Symbols used in Menu Tree:**

A rectangle represents Menus

Rectangle

A line represents the Connection

Line

View orders & transactions

Pump

Employee

User

E-PUMP

Admin

Login

Home 

About

Registration

Approval pump

View feedback & reply back

Update fuel rate

Change password

Search pump

View pump on google map

Order fuel & pay bill

Update profile

Track Order

Update profile details

View feedback & reply back

View completed Orders

View delivery requests

Upload certificate

View sensor data

View feedback & reply back

Upload license

Approve employees

**Figure 3.8.1: Menu Tree**

**3.9SYSTEM FLOW CHART**

Flow chart that can be used for representing the flow of data in the system . Flow chart that can be created before the coding is started. Because flow chart is very helpful to developer to identify the main functions coming in the system, and flow of data in the system. By analyzing the flow chart of a system we can develop that system very easily. Flow charts are used n designing and documenting complex processes. Like other types of diagram, the help to visualize what is going on an thereby help the viewer to understand a process , and perhaps also find flaws, bottlenecks, and other less- obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are:

* A processing step, usually called activity, and denoted as rectangular box
* A decision usually denoted as a diamond

A flowchart is described as “cross-functional” when the page is divided into different swim lanes describing the control of different organizational units. A symbol appearing in a particular “ lane” is within the control of that organizational unit. This technique allows the author to locate the responsibility for performing an action or making a decision correctly, showing the responsibility of each organizational unit for different parts of a single process.

**Symbols used in the Flowchart**

**: Start or Stop**

Rounded Rectangle

**: Processing**

Rectangle

**: Input or Output**

Parallelogram

**: Condition**

Diamond

**: Connector**

Circle

**: Direction (Logic Flow)**

Flow Line

START

Admin

FAILED

Login

SUCCESS

View feedback & Replay back

Approve pump

Change password

Update fuel price

STOP

**Figure 3.9.1:Flowchart for Admin**

START

User

FAILED

Login

SUCCESS

Order fuel &Pay bill

Update profile

Track order

View Pump on google map

Search Pump

STOP

**Figure 3.9.2:Flowchart for User**

START

Pump

FAILED

Login

SUCCESS

View sensor data

Approve employee

View feedback & replay

View order & transaction

Upload License

STOP

**Figure 3.9.4:Flowchart for Pump**

START

Employee

FAILED

Login

SUCCESS

View delivery request

Upload Certificate

Update profile details

View feedback & replay

View completed request

STOP

**Figure 3.9.4: Flowchart for Employee**

**CHAPTER-4**

SUCCESS

FAILED

Track orders

view fertilizer inormation

Purchase,add feedback

Lgin

**SYSTEM DESIGN**

System design translates the system requirements into ways of operationalizing them. The design is a solution, a “how to” approach, compared to analysis, a “what is “orientation. The emphasis is on translating performance specifications into design specifications.

System design covers the following:

* Reviews the current physical system.
* Prepares output specifications.
* Prepares input specifications.
* Prepares edit, security, and control specifications.
* Specifies the implementation plan.
* Prepares a logical design walk through of the information flow, output, input, control and implementation plan.

**4.1 MODULE DESCRIPTION**

The System is divided into four modules. Each module specifies the

functional requirement of the system.

* **Admin**
* **Approve Pump**-Approve/reject pump
* **Update Fuel Rate-**Update fuel rate
* **View Feedback & Reply back**-View feedback given by users and reply back.
* **Change Password**-Change password
* **Pump**
* **Upload License-**Upload license of the pump.
* **Approve Employees-**Approve employees.
* **View Orders& Transactions-**View orders and transactions in a day.
* **View Feedback & Reply Back-**View feedback given by users and reply back.
* **View Sensor Data**-Sensor data
* **Employee**
* **Update Certificate-** Upload certificate.
* **View Delivery Requests–**View new delivery requests.
* **View Completed Requests-**View completed orders.
* **View Feedback & Reply Back-**View feedback given by users and reply back.
* **Update Profile -**Update profile details like password,name,etc.
* **User**
* **Search Pump-** Search pumps by district or by fuel.
* **View Pump on Google Map–**View pump on googlemap.
* **Order Fuel & Pay Bill-**Order fuel & Pay the bill.
* **Track Orders-**Track order status.
* **Update Profile-**Update profile details like password,name,etc.

**4.2 INPUT DESIGN**

Input design is the method by which valid data are accepted from the user. This part of the designing requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify these errors. Inaccurate input data are the most common cause of errors in data processing. Input design consists of the following processes:-

* Designing graphical user entry screen is easy to use.
* Designing procedures and functions to valid the data as per business rules.
* Designing functions needed to store data into a usable form for processing.
* Designing the common integrated functions that can be used by all other users when needed.

**4.2.1 Input Objectives**

Controlling Amount of Input: Wherever user input is required, giving possible input values as default in that area reduces the amount of user keystrokes. Thus the user can pass on to next data without much typing. This makes the data entry much fast and error free. When the user has the format of input to be given, it will be very easy for the user to give input in the same format.

Avoiding Delay: A processing delay resulting from data entry operations is called a bottleneck. Such bottlenecks are made obsolete in this project by breaking up the amount of data to be entered in each form into different smaller and simpler forms.

Avoiding Errors in Data: The rate at which errors occur depends on the quantity of the data. As told in the above objective these errors are reduced by making the number of data to be entered in each form is reduced.

Avoiding Extra Steps: To fulfill any operation the user have no need to do complex steps, instead any operation can be done with simple easy to use steps.

**4.3OUTPUT DESIGN**

* Output design is one of the most important features of the information system. When the output is not of good quality, the users will be averse to use the newly designed system and may not use the system. There are many types of outputs, all of which can be either highly useful or can be critical to the users, depending on the manner and degree to which they are used.
* Outputs from computer system are required primarily to communicate the results of processing to users. They are also used to provide a permanent hard copy of the results for later consultation.

**4.3.1 Output Objectives**

The output from an information system should accomplish one or more of the following objectives:

* Convey information about past activities, current status, or projections of the future.
* Signal important events, opportunities, problems or warnings
* Trigger an action
* Confirm an action

**4.4DATABASE DESIGN**

The database design is a logical development in the methods used by the computers to access and manipulate data stored in the various parts of the computer systems. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. The main objectives of databases are data integrity and data independence.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. The database serves as the repository of data, so a well-designed database can lead to a better program structure and reduce procedural complexity. In a database environment, common data are available and used by several users

Database Management System (DBMS) allow the data to be protected and organized separately from other resources like hardware, software, and programs. DBMS is a software package, which contains components that are not found other data management packages. The significant of DBMS is the separation of data as seen by the programs and data as stored on the direct access storage devices. That is the difference between the logical and physical data.

The main objectives covered in database design are:

* Controlled redundancy

Data independence

* Accuracy and integrity
* Privacy and security
* Performance

**4.4.1.Normalization**

Normalization is the process of decomposing the attributes in an application, which results in a set of tables with very simple structure. The purpose of normalization is to make tables as simple as possible. Normalization is carried out in this system for the following reasons

* To structure the data so that there is no repetition of data, this helps in saving space.
* To permit simple retrieval of data in response to query and report request
* To simplify the maintenance of the data through updates, insertions and deletions.
* To reduce the need to restructure or reorganize data which new application requirements arise

Primary key is assigned for this purpose. The primary key fields in almost all the tables help to ease the search and improve efficiency. The proposed system is using the second normal form as it is found most suitable. In second normal form each row must contain associated field that describes an attribute of the entry that the table describes.

Here are the most commonly used normal forms:

* First normal form(1NF)
* Second normal form(2NF)
* Third normal form(3NF)

**4.4.1.1 First normal form (1NF)**

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

**4.4.1.2 Second normal form (2NF)**

A table is said to be in 2NF if both the following conditions hold:

* Table is in 1NF (First normal form)
* No non-prime attribute is dependent on the proper subset of any candidate key of table.

An attribute that is not part of any candidate key is known as non-prime attribute.

**4.4.1.3 Third Normal form (3NF)**

A table design is said to be in 3NF if both the following conditions hold:

* Table must be in 2NF
* [Transitive functional dependency](https://beginnersbook.com/2015/04/transitive-dependency-in-dbms/) of non-prime attribute on any super key should be removed.

An attribute that is not part of any [candidate key](https://beginnersbook.com/2015/04/candidate-key-in-dbms/) is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:

* X is a [super key](https://beginnersbook.com/2015/04/super-key-in-dbms/) of table
* Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as prime attribute.

The database tables used in this project are given below:

**Tablename** : Login

**Description** :Logindetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Fieldname | Datatype | Constrains | Description |
| 1 | Lid | Int(11) | Primarykey | Loginid |
| 2 | Username | Text | Notnull | Username |
| 3 | Password | Text | Notnull | Password |
| 4 | Type | Text | Notnull | Usertype |
| 5 | Status | Int(3) | Notnull | Status |

**Tablename** :user\_reg

**Description** :userregistration table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Fieldname | Datatype | Constrains | Description |
| 1 | Uid | Int(11) | Primarykey | Userid |
| 2 | Uname | Text | Notnull | Username |
| 3 | Lid | Int(11) | Foriegnkey | Loginid |
| 4 | Uhousenane | Text | Notnull | userhousename |
| 5 | Ustreet | Text | Notnull | Userstreet |
| 6 | Udistrict | Text | Notnull | Userdistrict |
| 7 | Ustate | Text | Notnull | Userstate |
| 8 | Upin | Int | Notnull | Userpincode |
| 9 | Uphone | bigint(12) | Notnull | userphonenumber |

**Tablename** :employee

**Description** :employee registration details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Fieldname | Datatype | Constrains | Description |
| 1 | Eid | Int(11) | Primarykey | Employeeid |
| 2 | Ename | Text | Notnull | employeename |
| 3 | Lid | Int(11) | Foriegnkey | Loginid |
| 4 | Ehousenane | Text | Notnull | employeeaddress |
| 5 | Street | Text | Notnull | employeeGender |
| 6 | Edistrict | Text | Notnull | Employee Date of birth |
| 7 | Estate | Text | Notnull | employeeQualification |
| 8 | Epin | Bigint(12) | Notnull | employeePhoneno |

**Tablename** :pump\_manager\_reg

**Description** :pump manager registration details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Fieldname | Datatype | Constrains | Description |
| 1 | Manid | Int(11) | Primarykey | Mangerid |
| 2 | Man\_name | Text | Notnull | mangeranme |
| 3 | Pump\_name | Int(11) | Notnull | Pumpname |
| 4 | CompanyName | Text | Notnull | companyname |
| 5 | Lno | Int(11) | Notnull | licensenumber |
| 6 | Manphone | Bigint(12) | Notnull | mangephonenumber |
| 7 | Lid | Int(11) | Notnull | Loginid |

**Tablename** :fuel\_db

**Description** :fueldetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno |  |  |  |  |
| 1 | Fuel\_id | Int(11) | Primary key | Fuelid |
| 2 | Fname | Text | notnull | Fuelname |
| 3 | Rate/ltr | floar | notnull | Fuelrate/ltr |

**Tablename** :fuel\_at\_pump

**Description** :fueldetail at pump

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno |  |  |  |  |
| 1 | Pump\_id | Int(11) | Primarykey | Pumped |
| 2 | Manager\_id | Int(11) | Foriegnkey | Managerid |
| 3 | Fname | Text | Notnull | Fuelname |
| 4 | Ltr | float | Notnull | Liter |

**Tablename** :pump\_license

**Description** :licensedtails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno |  |  |  |  |
| 1 | lic\_id | Int(11) | Primarykey | Pumped |
| 2 | Lic\_ | Int(11) | Notnull | Licensenumber |
| 3 | Man\_id | Int(11) | Foriegnkey | Managerid |
| 4 | Franchise name | Text | Notnull | Franchise\_name |
| 5 | Licimageid | text | notnull | License image |

**Tablename** :ordered\_fuel

**Description** :ordered\_fueldetail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Slno | Fieldname | Datatype | Constrains | Description |
| 1 | Order\_id | Int(11) | Primarykey | Ordered |
| 2 | User\_id | Int(11) | Foriegnkey | Userid |
| 3 | Pump\_id | Int(11) | Foriegnkey | Pumpeid |
| 4 | Emp\_id | Int(11) | Foriegnkey | Employeeid |
| 5 | Fuel\_name | Text | Notnull | Fuelname |
| 6 | Pincode | Int(11) | Notnull | Pincode |
| 7 | Date | Date | Notnull | Date |
| 8 | Advance amnt | Float | Notnull | Advance amount |
| 9 | Total | Float | Notnull | Total amount |
| 10 | Qty | Float | Notnull | Quantity |
| 11 | Stat\_id | Int(11) | Notnull | Deliverystatus |

**CHAPTER-5**

**SYSTEM CODING**

**5.1Program List**

|  |  |
| --- | --- |
| **Program Name** | **Description** |
| orders.php | Order fuel |
| homepage.php | Employee Registration Page |
| homepage.php | User Registration Page |
| viewprofile.php | View User profile |
| logout.php | Logout |

**5.1.1Source Code**

Orders.php

**<?php**

**$nm=$\_GET["nm"];**

**$x=0;**

**$total\_amount=0;**

**$con=mysql\_connect("localhost","root","");**

**mysql\_select\_db("pump");**

**if ($nm=='')**

**{**

**# code...**

**}**

**else**

**{**

**session\_start();**

**$mid=$\_SESSION["id"];**

**//echo $mid;**

**$a=mysql\_query("select pump\_id from add\_pump\_db where man\_id='$mid'");**

**// echo $a;**

**$original\_date = $nm; //input:: 2019-03-31**

**// Creating timestamp from given date**

**$timestamp = strtotime($original\_date);**

**// Creating new date format from that timestamp**

**$new\_date = date("Y-m-d", $timestamp);**

**// Outputs: 31-03-2019**

**$r=mysql\_query($s);**

**$m=0;**

**echo '<html><body><table style="width:100%" class="table"><tr>';**

**while ($m <mysql\_num\_fields($r))**

**{**

**$meta = mysql\_fetch\_field($r, $m);**

**echo '<th>' .ucfirst($meta->name) . '</th>';**

**$m = $m + 1;**

**}**

**//echo '<th>Remove</th>';**

**while ($b = mysql\_fetch\_row($a))**

**{**

**$o=current($b);**

**//$b['pump\_id'];**

**$r=mysql\_query($s);**

**$m = 0;**

**while ($row = mysql\_fetch\_row($r))**

**{**

**echo '<tr>';**

**$count = count($row);**

**$y = 0;**

**//$idval='1';**

**while ($y < $count)**

**{**

**$c\_row = current($row);**

**//if($y==0)**

**// $idval=$id\_row;**

**if ($y==5)**

**{**

**echo '<td>'.$nm.'</td>';**

**}**

**else if ($y==6)**

**{**

**$ad\_amount=$c\_row;**

**echo '<td>' . $c\_row . '</td>';**

**}**

**else if ($y==7)**

**{**

**$tot\_amount=$c\_row;**

**echo '<td>' . $c\_row . '</td>';**

**}**

**else if ($y==9)**

**{**

**if ($c\_row==0)**

**{**

**echo '<td>Not Delivered</td>';**

**}**

**else**

**{**

**echo '<td>Delivered</td>';**

**}**

**}**

**else if($y==10)**

**{**

**if ($c\_row==0)**

**{**

**echo '<td>Full amount not payed</td>';**

**}**

**else if ($c\_row==1)**

**{**

**$total\_amount=$total\_amount+$ad\_amount;**

**echo '<td>Advance amount Payed</td>';**

**}**

**else**

**{**

**$total\_amount=$total\_amount+$tot\_amount;**

**echo '<td>payment complete</td>';**

**}**

**}**

**else**

**{**

**$x=$x+1;**

**echo '<td>' . $c\_row . '</td>';**

**}**

**next($row);**

**$y = $y + 1;**

**}**

**$m = $m + 1;**

**echo '</tr>';**

**}**

**}**

**echo '</table>';**

**if ($x==0)**

**{**

**echo "<center><p style='color:yellow'>No Orders on the Records</p></center>";**

**}**

**else**

**{**

**//<h4>Total litres</h4>**

**echo "<table style='color:white'><td><h4>Total Amount :</h4></td><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td><td><h3>".$total\_amount."</td></h3></table>";**

**}**

**echo "</body></html>";**

**}**

**?>**

**5.1.2Source Code**

**homepage.php**

<div class="modal fade" id="modal4" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content modal-popup">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h2 class="modal-title" style="color: #0004ff">Employee Registration</h2>

</div>

<?php

$aeid=autoid("emp\_reg","emp\_id");

?>

<form action="controller\_epump.php" method="post">

<center>

<table>

<th>

<center>

<h3 style="color: white">Login ID :: </h3>

</center>

</th>

<th>

<center>

<input name="empid" type="text" class="idform-control" style="width: 280px" id="empid" disabled=disabled placeholder="User ID" value=<?php echo $aeid; ?>>

</center>

<input type=hidden name="empid" value=<?php echo $aeid ?>>

</th>

</table>

<input name="empname" type="text" class="form-control" id="empname" placeholder="Employee Name" required="required" maxlength="20"

pattern="[a-zA-Z]{3,20}" title="3 to 25 character required">

<input name="pass" type="Password" class="form-control" id="pass" placeholder="Password" required="required" pattern="{8,20}"

title="Should contain atleast 8 characters">

<input name="repass" type="Password" class="form-control" id="repass" placeholder="Repeat Password" required="required">

<table>

<tr>

<td>

<center>

<h3 style="color: white">GENDER </h3>

</center>

</td>

<td>

<center>

<select name="gen" class="idform-control" style="width: 320px">

<option value="male" style="color: black">Male</option>

<option value="female" style="color: black">Female</option>

<option value="other" style="color: black">Other</option>

</select>

</center>

</td>

</tr>

<tr>

<td>

<center>

<h3 style="color: white">DOB </h3>

</center>

</td>

<td>

<center>

<input name="dob" type="Date" class="idform-control" style="width: 320px" id="dob" placeholder="Date of Birth" required="required">

</center>

</td>

</tr>

</table>

<input name="address" type="textarea" class="form-control" id="address" placeholder="Address" required="required">

<table>

<tr>

<td>

<center>

<h3 style="color: white">QUALIFICATION</h3>

</center>

</td>

<td>

<center>

<select name="qual" class="idform-control" style="width: 250px">

<option value="10" style="color: black">10th Level</option>

<option value="12" style="color: black">12th Level</option>

<option value="Graduate" style="color: black">UG or Above</option>

</select>

</center>

</td>

</tr>

</table>

<input name="email" type="Email" class="form-control" id="email" placeholder="Email ID" required="required">

<input name="phno" type="text" class="form-control" id="phno" placeholder="Phone number" required="required" pattern="[0-9]{10}"

title="Phone number should contain 10 numbers" maxlength="10">

<input name="empsignup" type="submit" class="form-control" id="empsignup" value="Sign up">

</center>

</form>

</div>

</div>

</div>

**5.1.3Source Code**

**adminhome.php**

<div class="modal fade" id="modal3" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content modal-popup">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<h2 class="modal-title" style="color: #0004ff">User Registration</h2>

</div>

<?php

$auid=autoid("user\_reg","us\_id");

?>

<form action="controller\_epump.php" method="post">

<center>

<table>

<th>

<center>

<h3 style="color: white">Login ID :: </h3>

</center>

</th>

<th>

<center>

<input name="usid" type="text" class="idform-control" id="usid" disabled=disabled placeholder="User ID" value=<?php echo $auid; ?>>

</center>

<input type=hidden name="usid" value=<?php echo $auid ?>>

</th>

</table>

<input name="usname" type="text" class="form-control" id="usname" placeholder="User Name" required="required" maxlength="20"

pattern="[a-zA-Z]{3,20}" title="3 to 25 character required">

<input name="pass" type="Password" class="form-control" id="pass" placeholder="Password" required="required" pattern="{8,20}"

title="Should contain atleast 8 characters">

<input name="repass" type="Password" class="form-control" id="repass" placeholder="Repeat Password" required="required">

<input name="phno" type="text" class="form-control" id="phno" placeholder="Phone number" required="required" pattern="[0-9]{10}"

title="Phone number should contain 10 numbers" maxlength="10">

<input name="ussignup" type="submit" class="form-control" id="ussignup" value="Sign up">

</center>

</form>

</div>

</div>

</div>

**5.1.4 View User Profile**

**Viewprofile.php**

<?php

//session\_start();

$id=$\_SESSION['id'];

$type=$\_SESSION['type'];

if($type=='admin')

{

$sql="select \* from login\_pump where id='$id'";

$result = mysql\_query($sql);

$i = 0;

echo '<html><body><table border=3 align=center><tr>';

while ($i<mysql\_num\_fields($result))

{

$meta = mysql\_fetch\_field($result, $i);

echo '<th>' .ucfirst($meta->name) . '</th>';

$i = $i + 1;

}

$i = 0;

while ($row = mysql\_fetch\_row($result))

{

echo '<tr>';

$count = count($row);

$y = 0;

$idval='1';

while ($y < $count)

{

$c\_row = current($row);

if($y==0)

$idval=$c\_row;

echo '<td>' . $c\_row . '</td>';

next($row);

$y = $y + 1;

}

// echo "<td><a href='admin\_home.php?aid=$idval'>CLICK</a></td>";

//echo "<td><img width=100 height=100 src=userdocs/".$idval."\_1.jpg></a></td>";

//echo '<td><a href=rating.php><imgsrc="images/star.jpg"></a></td>';

echo '</tr>';

$i = $i + 1;

}

echo '</table></body></html>';

//mysql\_free\_result($result);

//echo "<script>location.href='admin\_home.php';</script>";

}

else if($type=='user')

{

$sq="update user\_reg set pass='$newpass' AND repass='$newpass' where usid='$id'";

mysql\_query($sq);

}

else if($type=='pump manager')

{

$sq="update pump\_man\_reg set pass='$newpass' AND repass='$newpass' where man\_id='$id'";

mysql\_query($sq);

}

else if($type=='employee')

{

$sq="update emp\_reg set pass='$newpass' AND repass='$newpass' where emp\_id='$id'";

mysql\_query($sq);

}

return 0;

?>

**5.1.5Source Code**

**logout.php**

<?php

//---------------------------------------------------------- LOG OUT-----------------------------------------------------------------------

session\_start();

session\_unset();

session\_destroy();

echo "<script>alert('Successfully Loged out');</script>";

echo "<script>location.href='../../homepage.php';</script>";

?>

**CHAPTER-6**

**SYSTEM TESTING**

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system.

In the code testing the logic of the developed system is tested. For this every module of the program iys executed to find an error. To perform specification test, the examination of the specifications stating what the program should do and how it should perform under various conditions.

Unit testing focuses first on the modules in the proposed system to locate errors. This enables to detect errors in the coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided. In unit testing step each module has to be checked separately.

System testing does not test the software as a whole, but rather than integration of each module in the system. The primary concern is the compatibility of individual modules. One has to find areas where modules have been designed with different specifications of data lengths, type and data element name.

Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software must be executed several times in order to find out the errors in the different modules of the system.

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out theerrors. The validation phase reveals the failures and the bugs in the developed system. It will be come to know about the practical difficulties the system faces when operated in the true environment. By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted to check whether the specifications stating the program are performing under various conditions.

The testing steps performed in “Online charity” are given below:

* Unit testing
* Integration testing
* System testing

**6.1 UNIT TESTING**

In unit testing different modules are tested against the specifications produced during the design of the modules. Unit testing is essential for the verification of the code produced during the coding phase, and hence the goal is to test the internal logic of the modules.

The testing is carried out during the programming itself. After designing and coding each form they are run to see whether there are any anomalies. Some of the various test cases used to test the system are as follows:

* Test cases are given for testing against requirements of the unit being tested.
* Test case for path or branch covering.
* Test case for data flow coverage.
* Testing with classes of bad data

In unit testing the program unit that make up the system are tested individually. Unit testing focuses first on the modules, independent of one another to locate errors. This enables to detect in coding and the logic with in the module alone. This testing is also used to ensure the integrity of data stored temporarily.

**6.2 INTEGRATION TESTING**

Integration testing is systematic technique for constructing the program structure, while at the same time conducting test to uncover errors associated with interfacing. That is the program is constructed and tested in small segments, which makes it easier to isolate and correct.

Invalid modules are invariably related to one another and interact in a total system. Each portion of the system is tested against the entire module with both testing and live data before the entire system is ready to be implemented. When the individual modules were found works satisfactory, the system integration test was carried out. Data was collected in such a way that all program paths could be covered. Using these data a complete test was made. All outputs were generated. Different users were allowed to work on the system to check its performance.

So here in “Online charity” integration testing contains admin module, User module, Guest user module and login

**6.3SYSTEM TESTING**

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system.

In the code testing the logic of the developed system is tested. For this every module of the program is executed to find an error. To perform specification test, the examination of the specifications stating what the program should do and how it should perform under various conditions.

Unit testing focuses first on the modules in the proposed system to locate errors. This enables to detect errors in the coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided. In unit testing step each module has to be checked separately.

System testing does not test the software as a whole, but rather than integration of each module in the system. The primary concern is the compatibility of individual modules. One has to find areas where modules have been designed with different specifications of data lengths, type and data element name.

Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software must be executed several times in order to find out the errors in the different modules of the system.

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out theerrors. The validation phase reveals the failures and the bugs in the developed system. It will be come to know about the practical difficulties the system faces when operated in the true environment. By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted to check whether the specifications stating the program are performing under various conditions.

The testing steps performed in “Online charity” are given below:

* Unit testing
* Integration testing
* Validation testing

**CHAPTER-7**

**SYSTEM IMPLEMENTATION**

Implementation is the process of bringing developed system into operational use and turning it over to the user. Implementation is the stage of the project where the theoretical design is turned into working system. In this stage the installation of the package in the real environment, to the satisfaction of the intended user and the operation of the system. Implementation includes all those activities that take place to convert from the old system to new one. Implementation is the phase, in which one has to be cautious, because the efforts undertaken during the project will be fruitful only if the software is properly implementation according to the plan made. The implementation phase is less creative than system design. The system implementation was carried out using five main aspects:

* Transition planning
* Training
* Security
* Protection
* Quality control

The term maintenance is used to describe the software engineering activities that occur following the delivery of the software product to the customer. The maintenance phase of the software lifecycle is the time period in which software performs useful work. We may define maintenance by describing four activities that are undertaken after a program is released for use.

During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or mare error is called corrective maintenance. Thus problem correction involves modification and revalidation of software to corrective maintenance.

**CHAPTER-8**

**SYSTEM MAINTANENCE**

Maintenance is the enigma of system development. It holds the software industry captive, trying up programming resources. Analysts and programmers spend far more time maintaining resources. As important as it is, many programmers and analysts are reluctant to perform or identify themselves with the maintenance effort. There is psychological, personality, and professional reasons for this.

Whereas the cost of hardware has steadily declined, the cost of producing programs has skyrocketed. That is software maintenance is expensive. One way to reduce maintenance costs is through maintenance management and software modification audits. Software modification consists of program rewrites system level updates, and re-audits of low-ranking programs to verify and correct the soft spots. The outcome should be more reliable software, a reduced maintenance backlog, and higher satisfaction and morale among the maintenance staff.

After the installation phase is completed and the user staff is adjusted to the changes created by the candidate system evaluation and maintenance. Like any system, there is an aging process that requires periodic maintenance of hardware & software for customer evaluation, tasks required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage.

**8.1 TYPES OF SYSTEM MAINTENANCE**

**8.1.1 Corrective Maintenance**

Corrective maintenance of a system becomes necessary to rectify the bugs while the system is in use.

**8.1.2 Adaptive Maintenance**

A system might need maintenance when the customers need the product to run on new platforms, on new operating systems, or when they need the product to be interfaced with new hardware and software.

**8.1.3 Perfective Maintenance**

A system needs maintenance to support the new features that users want it to support,to change different functionalities of the system according of customer demands, or to enhance the performance of the system.

**CHAPTER-9**

**CONCLUSION**

In current days mainly in our state the fuel pumps are operated manually. These fuel pumps cause time consuming and require more man power. In the current system have so many difficulties. Such that, there is no proper method for the booking and delivery of petrol and diesel to the customer in a particular location for helping people who get stuck in a place when no petrol pumps are nearby. It is very difficult to locate a nearest petrol pump using the current system. We can only see the nearest fuel stations from google map. To place petrol pumps in distant area is very costly to provide excellent facility to the customers. All these problems are sorted out by the use of IOT E-Pump, which requires less time to operate and it is effective and can be use anywhere from kollam district.

**CHAPTER-10**

**FUTURE ENHANCEMENT**

The system is designed and developed in such a way that further expansion or modification can be made to permit the evaluation. The focus of the system is to inherit the requirement and update the system as per the needs. This project is so perfectly designed that it satisfies the requirements. The main future enhancements of the system is to order fuel from anywhere in the country**.**

**APPENDIXA:GANTT CHART**

Gantt chart shows the time relationship between ‘events’ of the production program has been regarded as revolutionary in management. Gantt chart recognizes the total program goals and it should be regarded as a series of inter-related supporting plan (or events), that people can comprehend and follow. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical “TODAY” line.

Following figure shows the project schedule Gantt chart of the project.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **GANTT CHART** | | | | | | | | | |
| E-PUMP | | | | | | | | | |
| **TASK** | **2021 January 5 TO 2021 April 9** | | | | | | | | |
| **5Jan- 18Jan** | **19Jan- 25Jan** | **26Jan- 5Feb** | **8Feb- 26Feb** | **1Mar- 12Mar** | **15Mar-19Mar** | **22Mar- 26Mar** | **29 Mar- 2Apr** | **5Apr- 9Apr** |
| **Requirement Gathering** |  |  |  |  |  |  |  |  |  |
| **Requirement Analysis** |  |  |  |  |  |  |  |  |  |
| **System Design** |  |  |  |  |  |  |  |  |  |
| **Coding and Unit testing** |  |  |  |  |  |  |  |  |  |
| **System Testing** |  |  |  |  |  |  |  |  |  |
| **Project Reporting and documentation** |  |  |  |  |  |  |  |  |  |

**Figure A.1 Gantt chart**

**APPENDIX B:MEETING MINUTES**

**B.1 Meeting Minute – 1**

Date : 05-01-2021

Time : 10.30 pm -11.30 pm

Location : TCSR PROJECTS

Present : JIJO J S,SANISH BABU,ASWADEV G

On this day we collect all the required data for the development of this system.So we select TCSRforprojectwork.Mr. Sajeev Jaladharangave us lots of details and suggestions for the successful development of the project.On the basis of those data we select modules and assign them to each of the members and also started system analysis on this day.

**B.2 Meeting Minute – 2**

Date : 08-02-2021

Time : 10.30 pm -11.30 pm

Location : TCSR PROJECTS

Present : JIJO J S,SANISH BABU,ASWADEV G

In this day we reached our project center and start to design the system.This system has more than 10tables for store and retrieve data,which are designed with the help of our project our giude.The forms are designed by analyzing some related sites and the tables designed.

**B.3 Meeting Minute – 3**

Date : 19-03-2021

Time : 10.30 pm -11.30 pm

Location : TCSR PROJECTS

Present : JIJO J S,SANISH BABU,ASWADEV G

On this day we start the system coding with the help of our guide.The coding is done in PHP;this phase is more complex to our project development.Various suggestions from our friends and teachers were attained here.The project module wise of project coding is done by each team members.

**B.4 Meeting Minute – 4**

Date : 06-04-2021

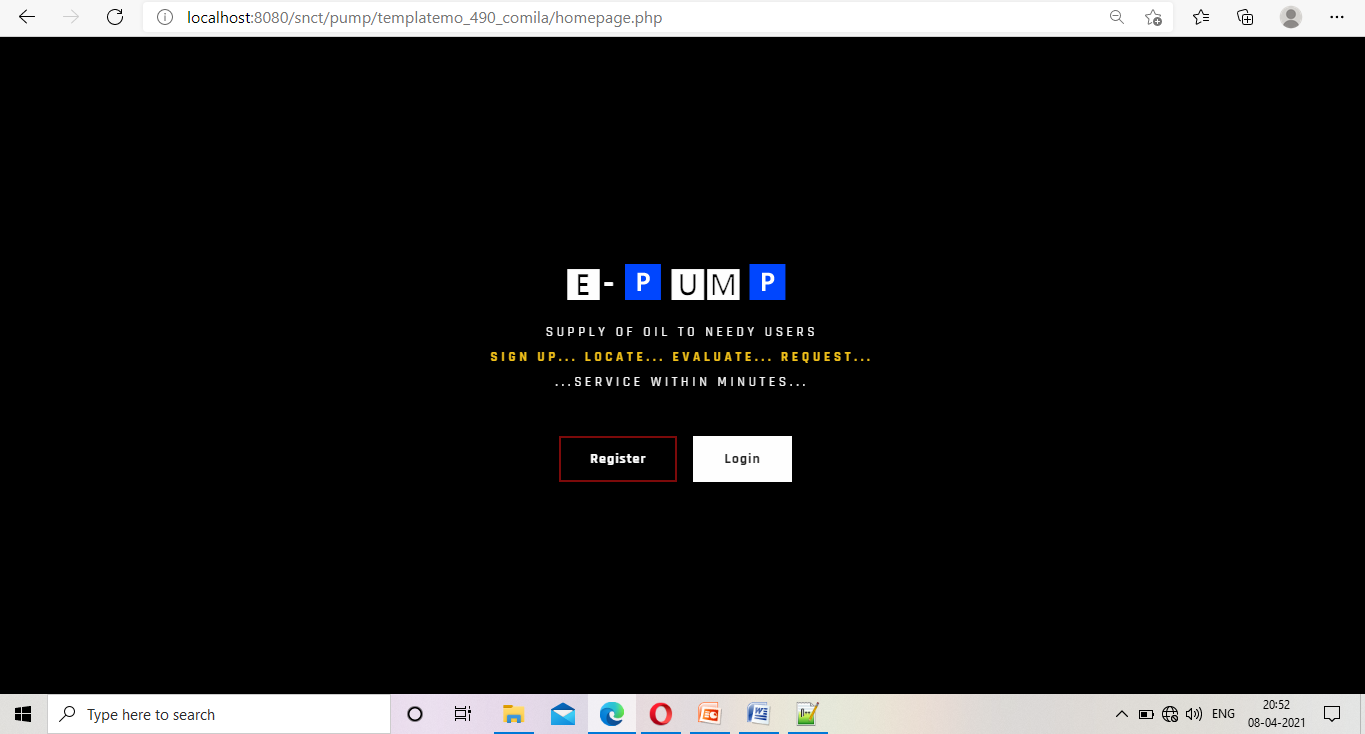
Time : 10.30 pm -11.30 pm

Location : TCSR PROJECTS

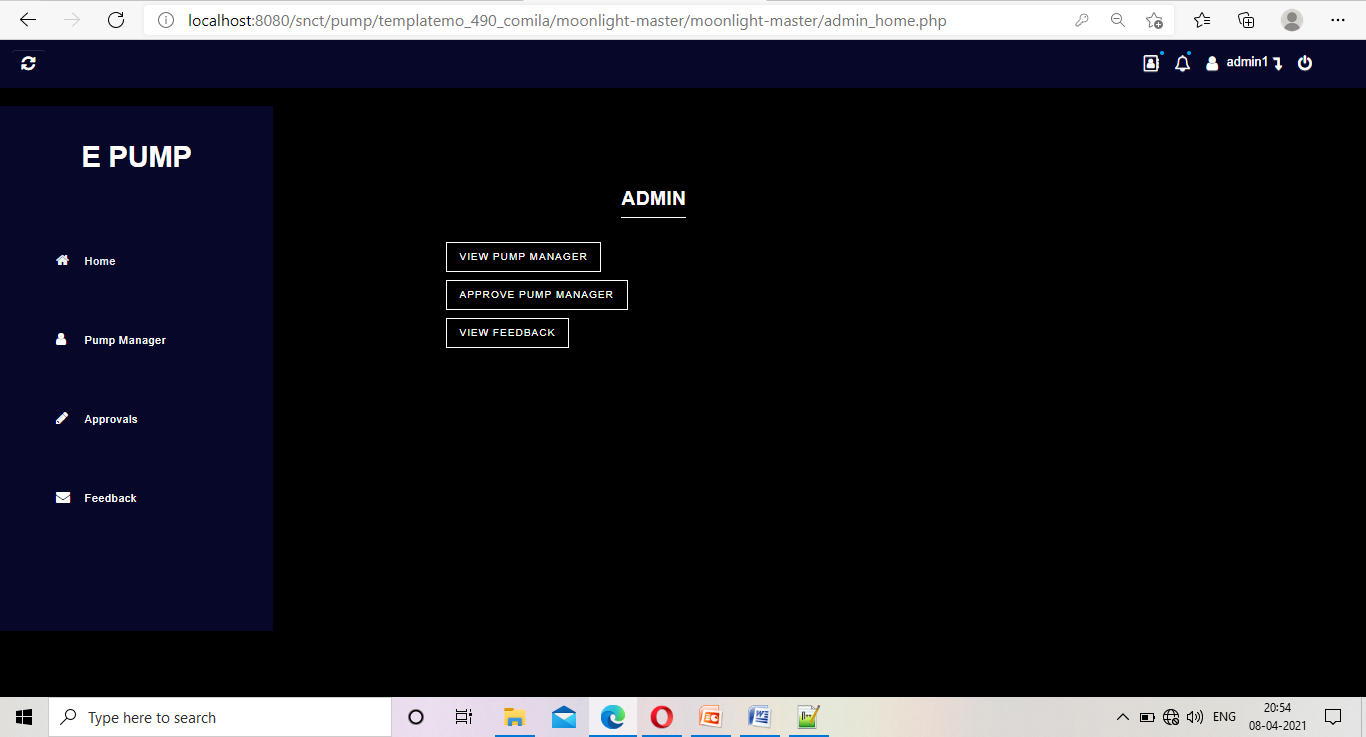
Present : JIJO J S,SANISH BABU,ASWADEV G

On this day we start testing each module of our project.Aftertesting,each module can be implemented.Also,maintenance of our project is done.Finally,we have checked the working of our project..

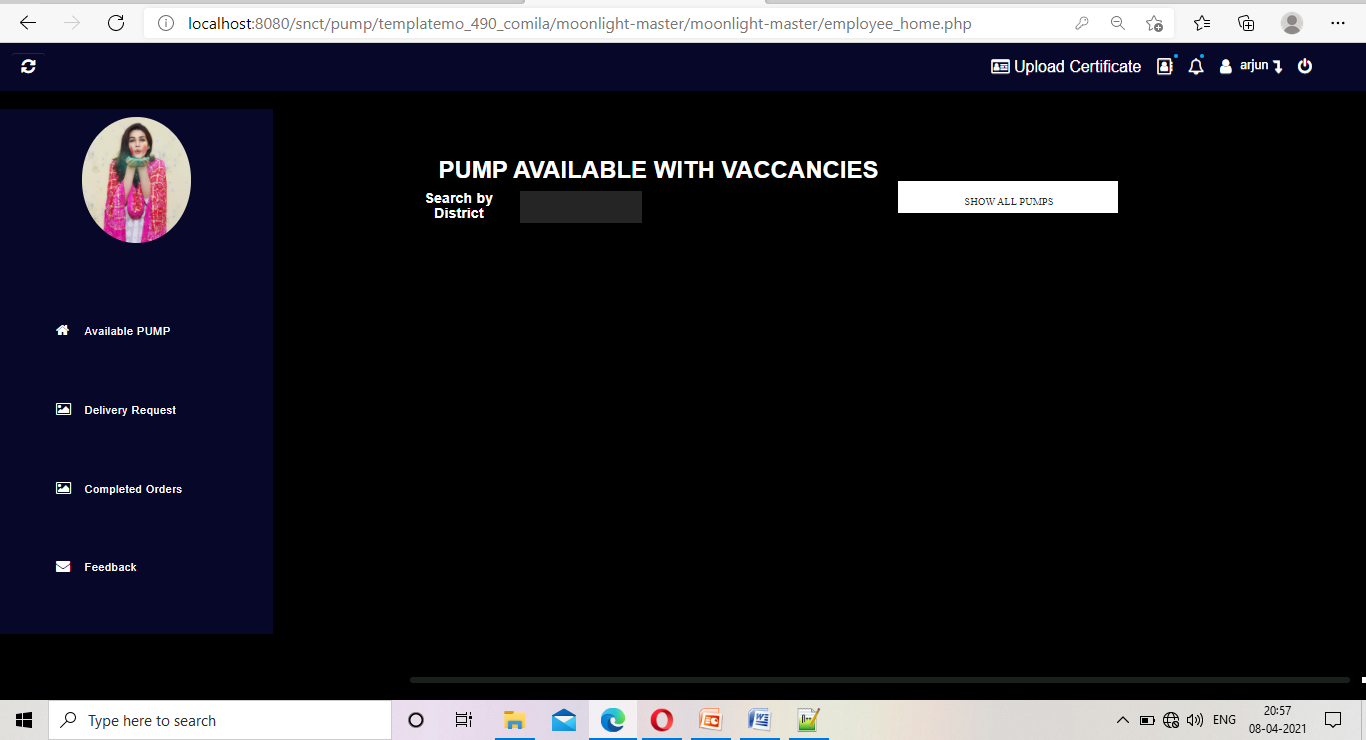
**APPENDIX A :SCREEN LAYOUTS AND REPORT**

****

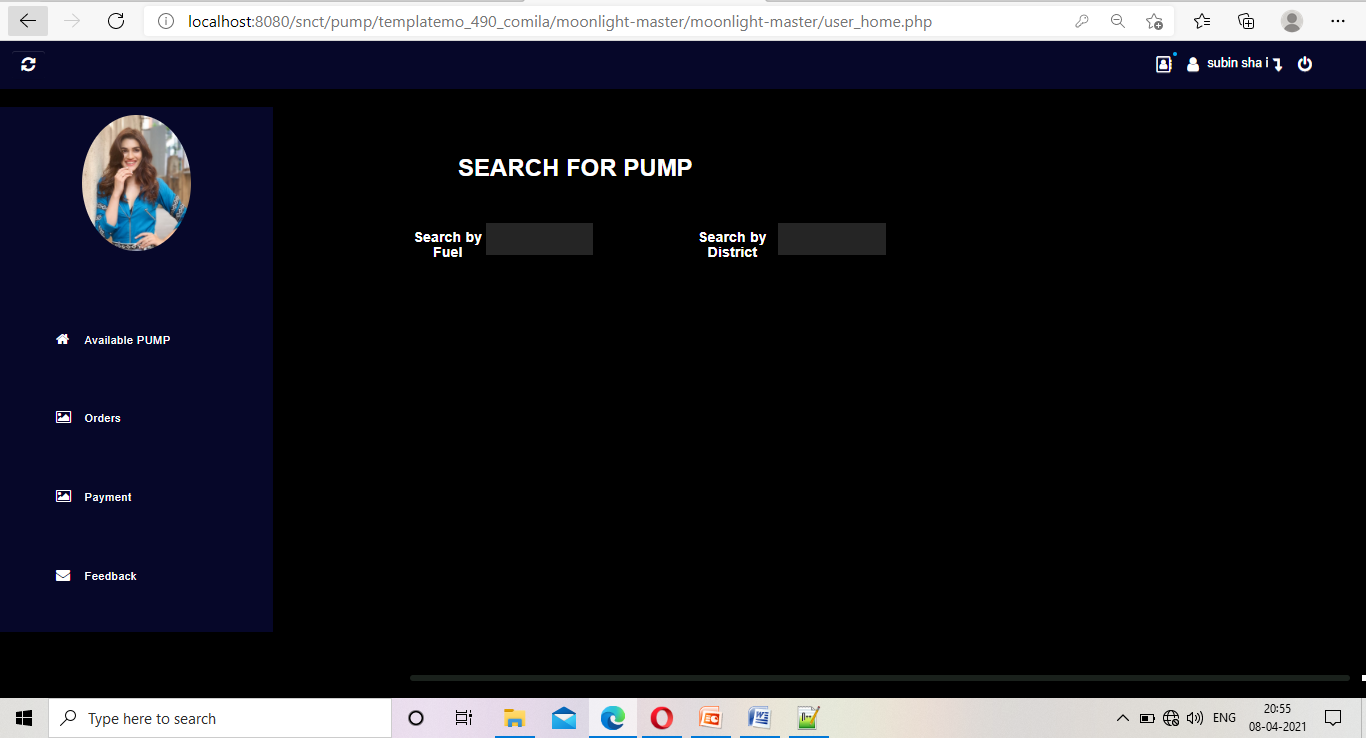
**Figure 1 : Home page**

****

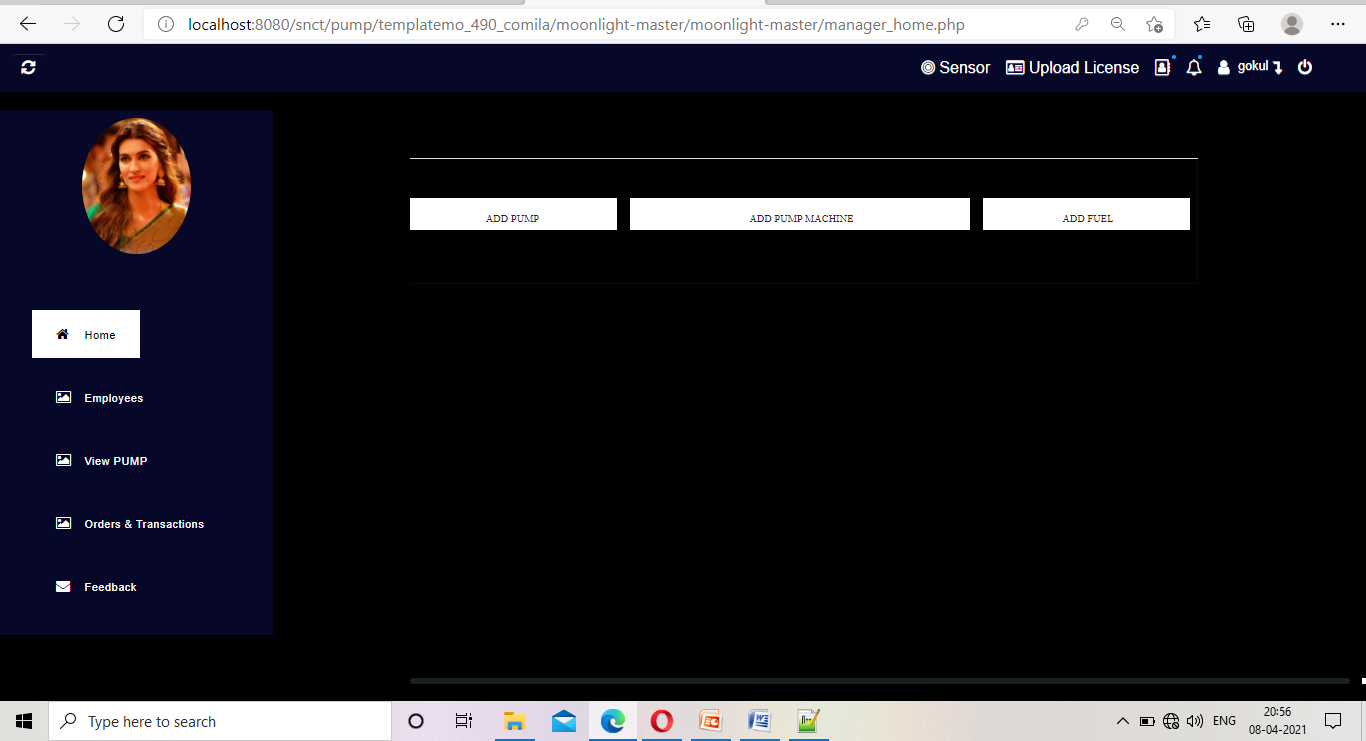
**Figure 2 :Admin home page**

****

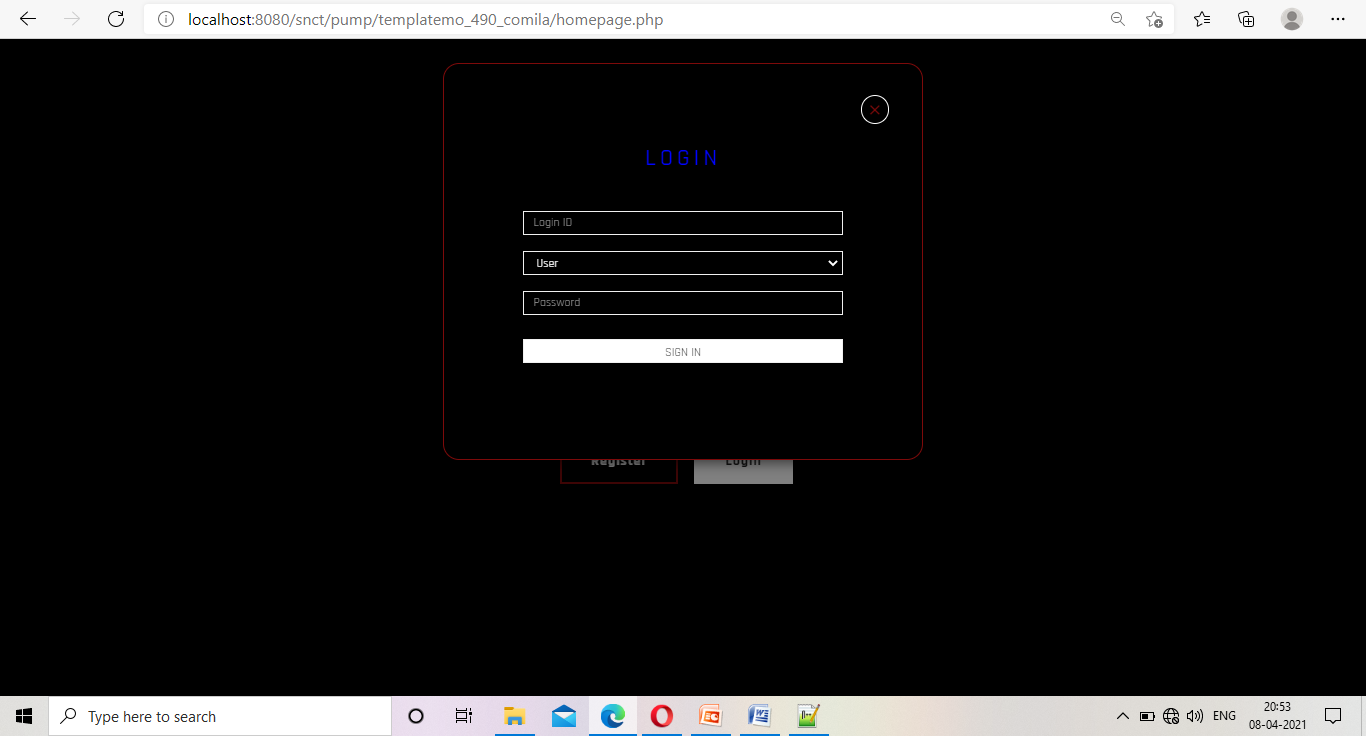
**Figure 3 :Employee home page**

****

**Figure 4 :User home Page**

****

**Figure C.5 :Pump home page**

****

**Figure C.6 :Login page**

**APPENDIX D: TEST CASES**

**Table D.1: TEST CASES FOR LOGIN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Test Procedure** | **Test Data** | **Expected**  **Result** | **Actual Result** |
| Farmer  Registration | To test whether the registration details of farmer  are valid | Enter Student  Details | Arunima  Arunima | Valid registration | Successfully registered |
| Login | To test whether the username, password of user are valid | Enter  username and password | admin@gmail.com  \*\*\*\*\*\*  admin@gmail.com  admin | Password not matching  Valid login | Incorrect password  Valid login |
| Login | To test whether the username, password of user are valid | Enter  username and password | anu@gmail.com  \*\*\*\*\*\*\*  anu@gmail.com  Anu@123 | Wrong  password  Valid login | Incorrect user name or password  Valid login |

**BIBLIOGRAPHY**

**Texts**

[1]. Rajib Mall, Fundamentals Of Software Engineering, Third Edition ,PHI Learning

Pvt. Ltd, Year 2009

[2]. Andrew S Tanenbaum , System Analysis, Prentice Hall India Publications, Year

2000

**Websites**

* http://www.phpissimple.blogspot.com
* http://www.w3schools.com/php/ [PHP]
* http://www.tutorialspoint.com/php/ [PHP]
* http://php.net/manual/en/tutorial.php [PHP]
* https://www.lynda.com/PHP-training-tutorials/282-0.html [PHP]
* http://code.tutsplus.com/tutorials/the-best-way-to-learn-php--net-22287 [PHP]
* http://www.tizag.com/mysqlTutorial/ [MySQL
* https://en.wikipedia.org/wiki/MySQL [MySQL]
* http://dev.mysql.com/doc/refman/5.7/en/ [MySQL]
* https://en.wikipedia.org/wiki/CodeIgniter [Codeigniter]
* https://www.codeigniter.com/ [Codeigniter]
* https://ellislab.com/codeigniter [Codeigniter
* https://ellislab.com/codeigniter/user-guide/ [Codeigniter]