WATER AUTHORITY MANAGEMENT SYSTEM

Project Report Submitted By

AMMU SARA MATHEW

Reg. No:AJC17MCA-I008

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**WATER AUTHORITY MANAGEMENT SYSTEM"** is the bonafide work of **AMMU SARA MATHEW (Reg.No:AJC17MCA-I008)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-22.

Ms. Paulin Paul
Internal Guide

Rev.Fr.Dr.Rubin Thottupurathu Jose
Coordinator

Rev.Fr.Dr.Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report **"WATER AUTHORITY MANAGEMENT SYSTEM"** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

Date: **AMMU SARA MATHEW**

KANJIRAPPALLY **Reg. No: AJC17MCA-I008**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr. Dr.Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinators **Rev.Fr. Dr. Rubin Thottupurathu Jose** for their valuable suggestions and for overwhelming concern and guidance from thebeginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Pauline Paul** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

AMMU SARA MATHEW

ABSTRACT

Due to the increasing use of water in day today life and pollution made it difficult for the availability of pure water. So it became an important issue to resolve this, then we introduced online water authority site were all the process and operations are monitored by the authorities. From requesting for a new connection the journey between the user and the water authority starts and it will continue till the disconnection. All updates and messages will be shared through the registered email and it will help to keep in touch with the user all the time. In this application three modules are used user, staff and admin. Admin can verify user details and handle all the process except payment as it is paid through payment gateway. Each customer have their own unique consumer numbers and they are identified by this id in the system. The staff updates the meter readings in the site and it will notify the user through the mail so they could pay their payments. This application ensures the quality and quantity is good and provide a user friendly communication.

.

CONTENT

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Water Authority Management System The purpose of this site is to assist in the management of water authority .It records and control the distribution of water in various parts based on the customer demands. Will provide quality water and waste water management services in an environmental friendly and sustainable manner. Will transform ourselves to a consumer friendly organization providing services at the doorstep. Will achieve this by being open and honest in our business dealings. Financially self sufficient valuing and developing our employees continuously improving our work practices. Preparation, execution, promotion, operation, maintenance and financing of these schemes for the supply of water and the disposal of the waste water to decrease the scarcity of water. Rendering all necessary services in regard to water supply and collection and disposal of waste water to the Government and on request to private institutions or individuals. Preparation of state plans for water supply and collection and disposal of waste water on the direction of the Government.

1.2 PROJECT SPECIFICATION

The proposed system is made to help the customers for an easy and convenient way of paying the payment and also helps customer's for virtually. We will also provide users to give feedbacks, they can view the user page, paymentdetails, complaint details etc.

The system includes 4 modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add or update labour categories, state and district details etc. Admin can view all the registered contractor and labour, can able to approve or reject users and also can able to view all registered customer details.

2. Customer/User Module

Customer can register and they can view the their data and do secure online payment. Customer can also find their meter reading, payment details can add feedbacks and complaints to them.

3. Staff Module

Staff can register and they can see all the projects and payments that are updated by the customers for payment and projects. So that, the Staff can study the plan details and requirements of the customer. Staff can make payment online. They can do construction management using this web site effectively.

4. Payment Module

Users can pay their payments online through the payment gateway, it could be based on their choice. They will also get the receipt as a pdf file.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

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

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

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

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Each customer can create their own profile .The proposed systemrectify the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can pay their bills by viewing the profile details, previous and current bill and water usage details.

2.3 DRAWBACKS OF EXISTING SYSTEM

- The area receiving water from a pipe under repair is without water until the work is done.
- Maintenance costs are higher.

2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth; on such consideration the system is proposed. In our proposed system there is admin who can view all the staffs, labours and customers. It allows customers do their transactions by using online payment method .Usersof this proposed system are admin, customer, staff and labour. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Our website is multifunctional which includes customer introduction, project tellus, all activity history details etc.

The master file that is the file which contains all the details of the data's which are kept for long time is admin master and customer master. The admin master contains all the details of all the transactions and projects in the system, on-going projects, rates and details of different categories of connection, daily progress report, payment options etc. The customer master contains personal details of the customer like customer id, customer name, profession, residential address, etc. This system is designed to help the customers to make the communication easy. The proposed system provides consistency of data and reduces the paper work. Also, the customer can easily select the type of connection they want based on the rating of each connection. This helps the customer to fulfill his/her needs with the help of reviews and feedback through online. This system helps the customer to get the work done smoothly, efficiently and in less time and it helps the admin to increase the work and income source. This system is made to help both the customer and the admin.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- **You are able to get creative and innovative project plans:-**

Our customers will get creative project plans with affordable price and they have the option to view previeous project for drought regions and can send feedbacks.

- **Better security: -**

For data to remain secure measures must be taken to prevent unauthorized access.The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality.

- **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility Study is an assessment of the practicality of a proposed project or system. The requirement engineering process start with a feasibility study. The input to the feasibility study is a set of preliminary business requirement, an outline description of the system and how the system is intended to support business processes. In feasibility study, you may consult information sources such as the managers of the departments where the system will be used, software engineers who are familiar with the type of system that is proposed, technology experts and end users of the system.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Technical Feasibility

Technical feasibility assesses the current resources and technology, which are required to accomplish user requirements in the software within the allocated time and budget. Technical feasibility performs 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.

This project requires basic skills on technologies like HTML and designing, PHP for backend development, MySQL for database management. Also require other skills in general for user management, payment and gateway, awareness about the structure of water authority and its functioning. Out development team is already equipped with these skills and they are very well expert in these sections, hence the project does no force to do more functionality and dynamic nature rather than what all normal projects do. Also, the client side has ensured to organise a team of experts from payment, metre reading and new connection departments. And also, we require a server machine that support HTML, PHP, MySQL and payment gateway functions to deploy the project, which is not a serious issue and is available. Since payment are carries out through this project, payment security and authentications features are also required in the deployment server.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

3.1.2 Behavioural Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and Installed. All behavioral aspects are considered carefully and conclude that the project is behavioral feasible.

3.1.3 Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. The cost incurred on the software development team,estimated cost of hardware and software, cost of performing feasibility study. Cost incurred on software development to produce long-term gains for an organization. Cost required to conduct full software investigation. Cost of hardware, software, development team, and training. The proposed project is economically feasible. As the hardware was installed from quite beginning, the cost on the project hardware is low.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

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

3.2.2 Software Specification

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

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP:HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed onmost web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.1 MySQL

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

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, youneed a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play acentral 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 SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL92" 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), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

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

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

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

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

4.2 UML DIAGRAM

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

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

UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

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

4.2.1 USE CASE DIAGRAM

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

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include itemordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined to their ro

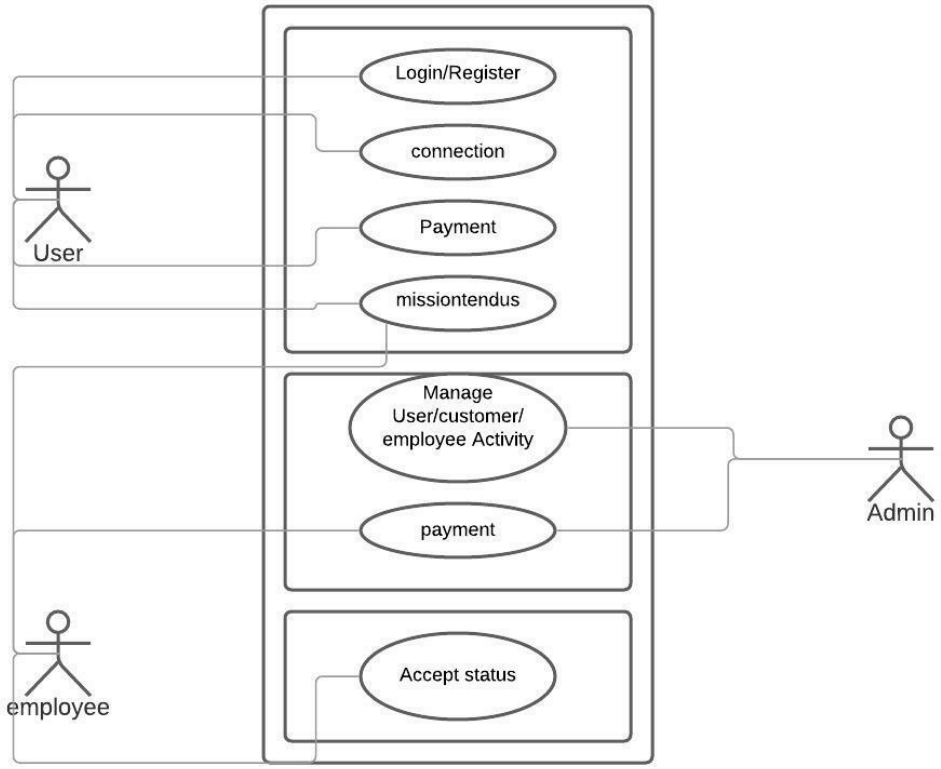


Fig 1 : Use case diagram for Water Authority Management System

4.2.2 OBJECT DIAGRAM

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

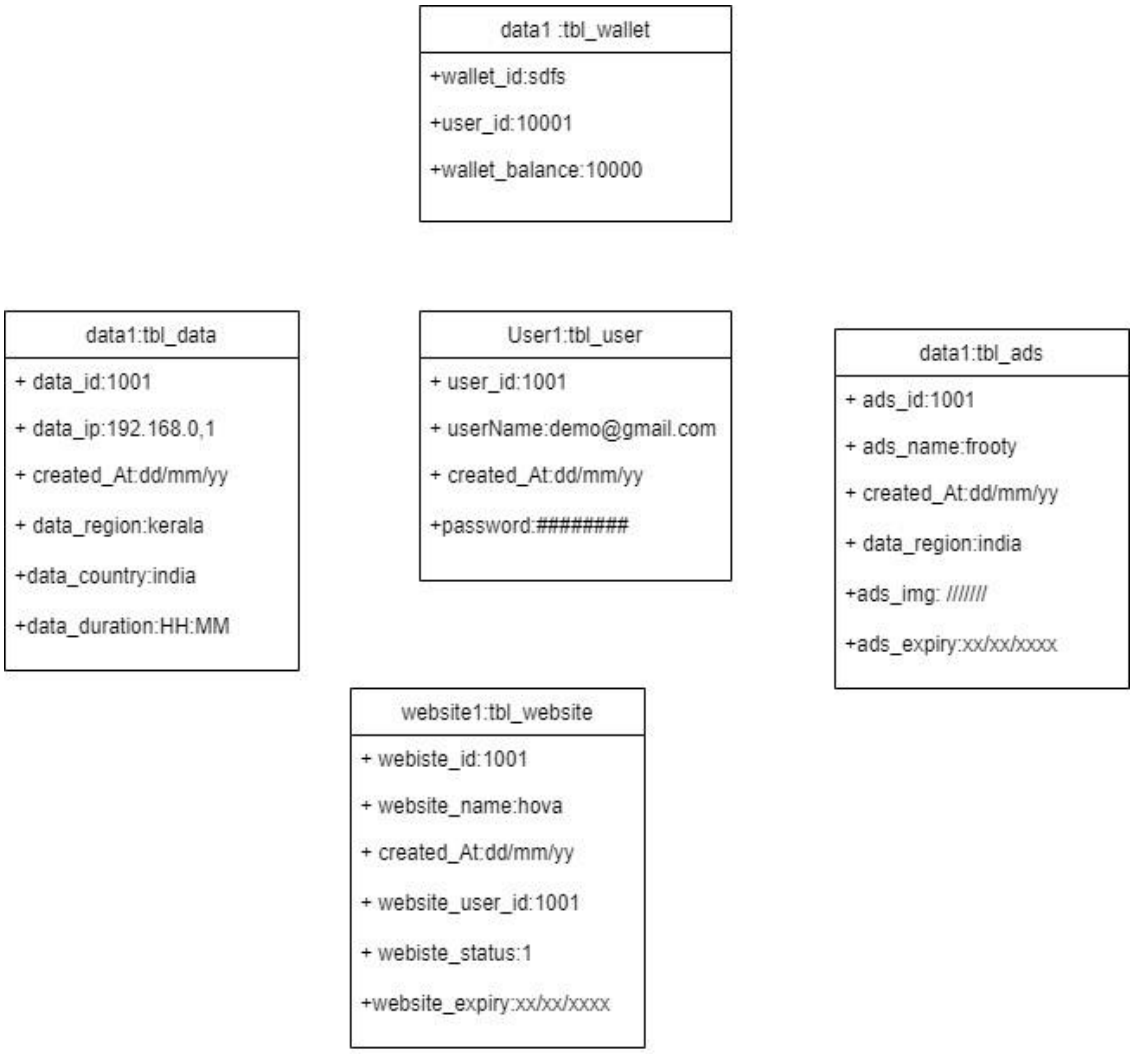


Fig 8: Object diagram for Proanalysis

4.2.3 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

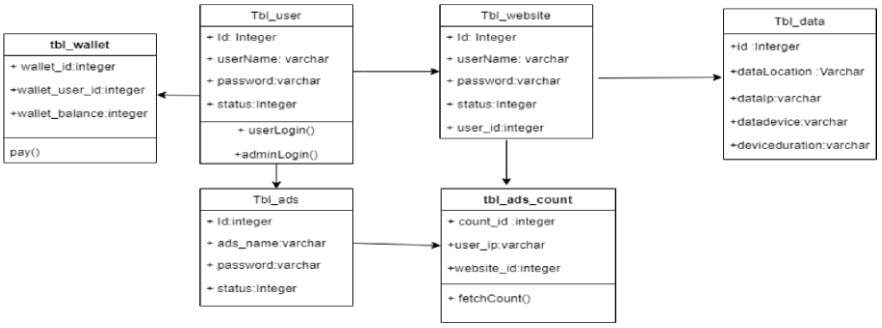


Fig 4: Class diagram for Proanalysis

4.2.4 SEQUENCE DIAGRAM

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

Sequence Diagram Notations –

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

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

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

Messages can be broadly classified into the following categories:

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

- Lost Message

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

Uses of sequence diagrams –

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

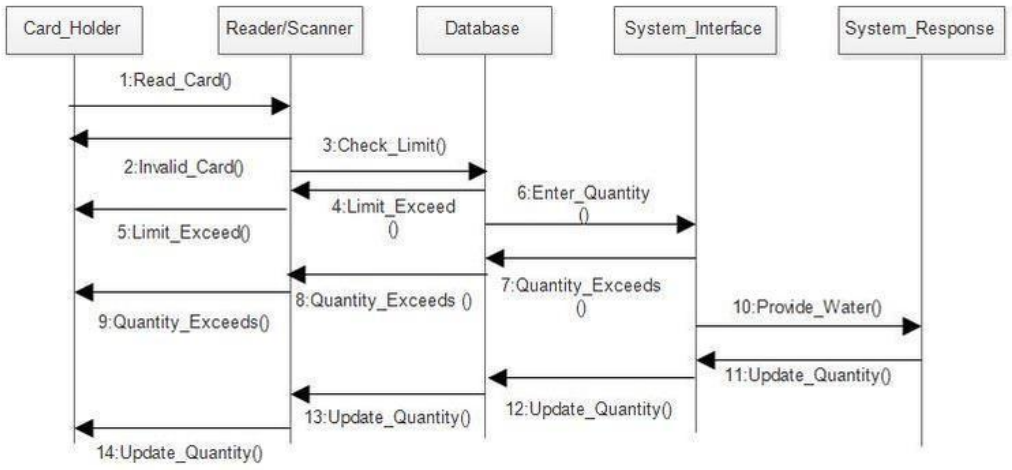


Fig 5: Sequence diagram for Water Authority Management System

4.2.5 STATE CHART DIAGRAM

A state machine of the states of objects defined by State Chart Diagram. The state machine can be called a machine that tells different states of an object and these states are controlled by internal or external events.

Basic of State Chart Diagram

1. It is one of the important UML diagrams used to model the spirited nature of a system.
2. The important aim of a diagram is to creation to termination from the states of objects.
3. These are also used for ahead and opposite engineering of a system. The main aim is to model the responsible system.
4. They describe different types of states of an object during its time of life and these states are updated by events and places where it used.
5. The main purposes of using State chart diagrams are follows,
    - Design the dynamic aspect of a system.
    - During its lifelong time defines the states of an object.
    - Model the time of a responsible system in states.
    - Tell a state product to design the states of an object.

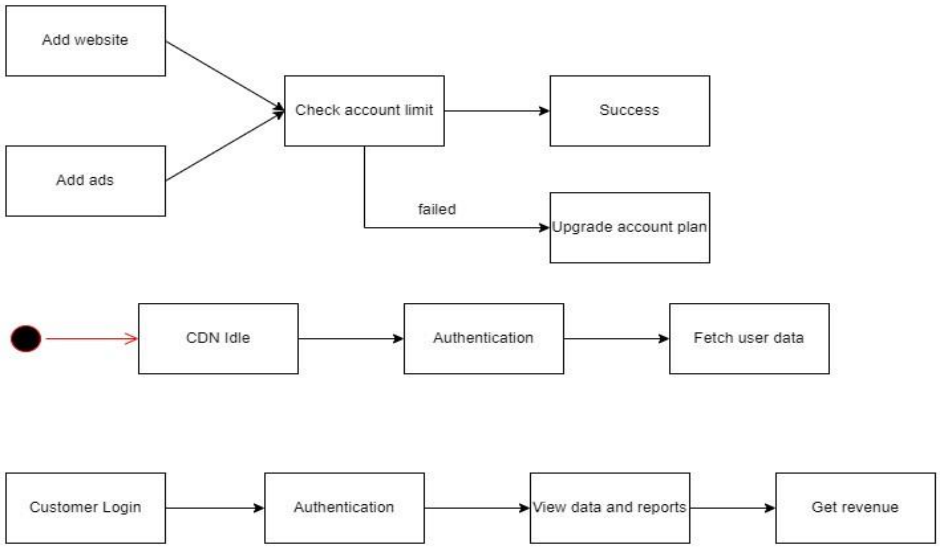


Fig 5: State chart diagram for Proanalysis

4.2.6 DEPLOYMENT DIAGRAM

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware. Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

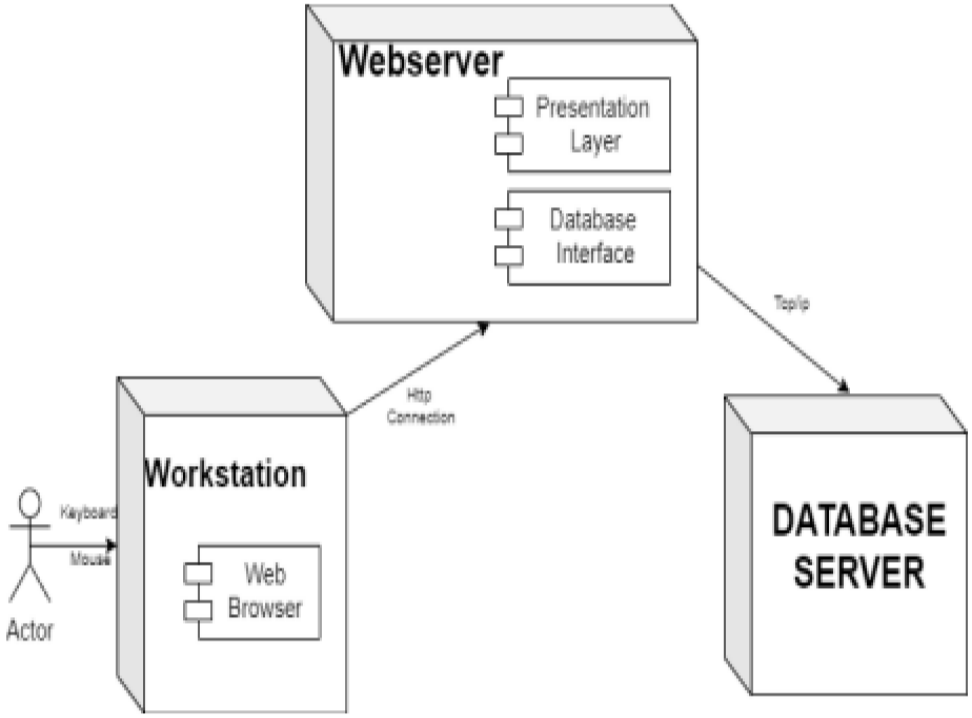
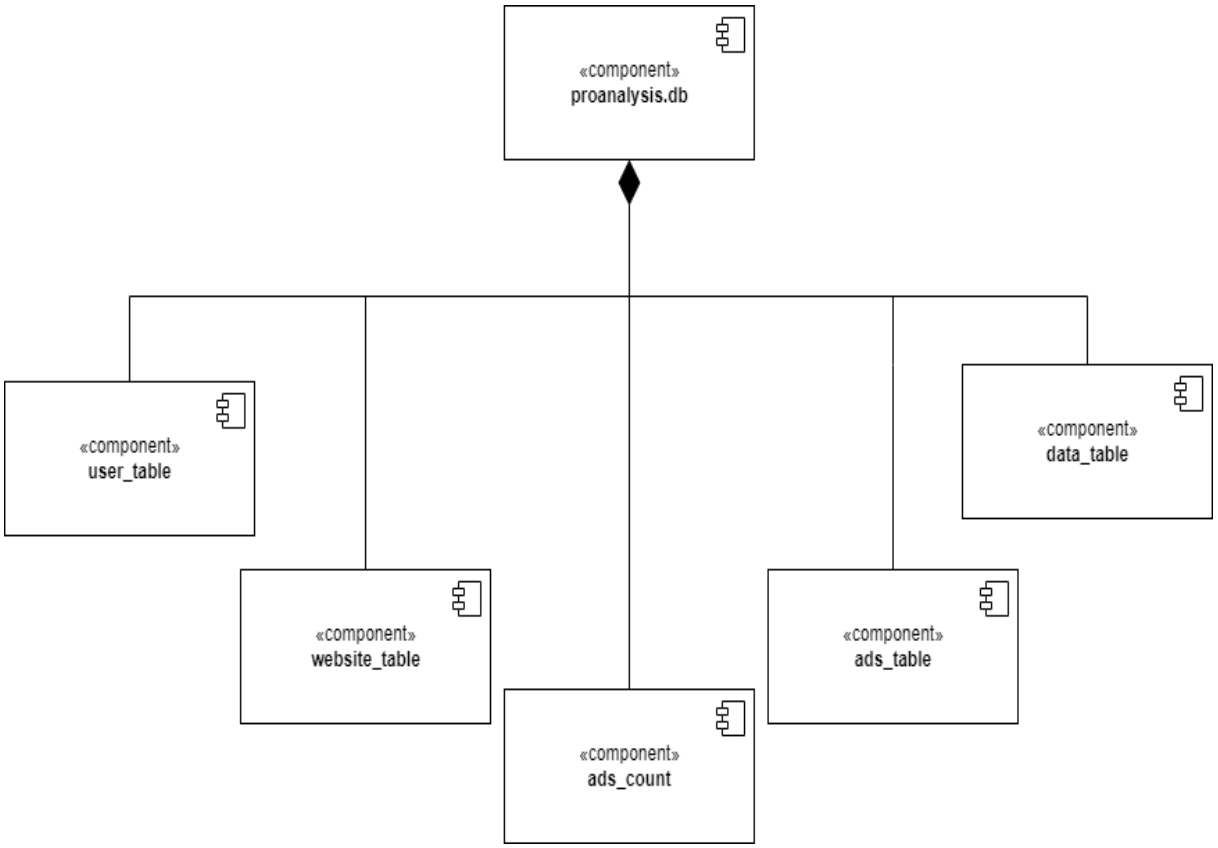


Fig 6: Deployment diagram for Proanalysis

4.2.7 COMPONENT DIAGRAM

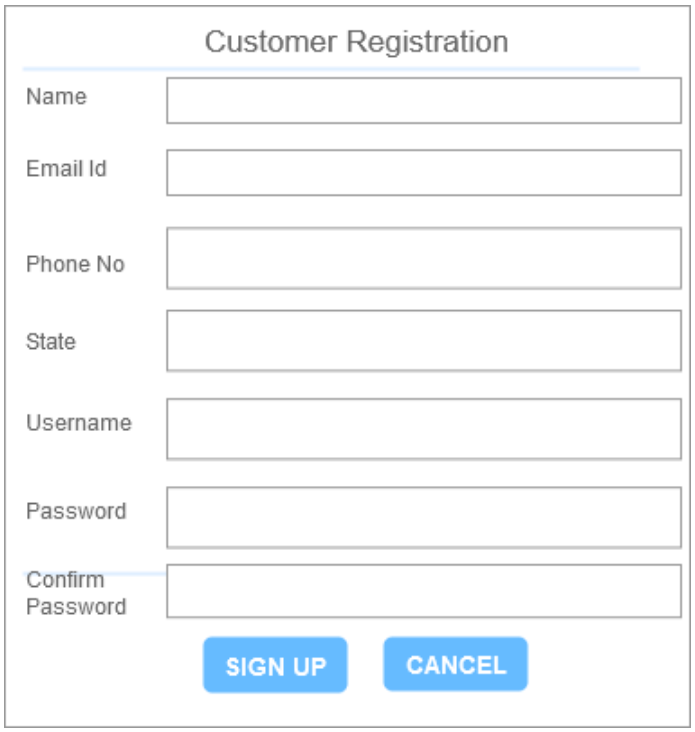
In Unified Modeling Language (UML), a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.



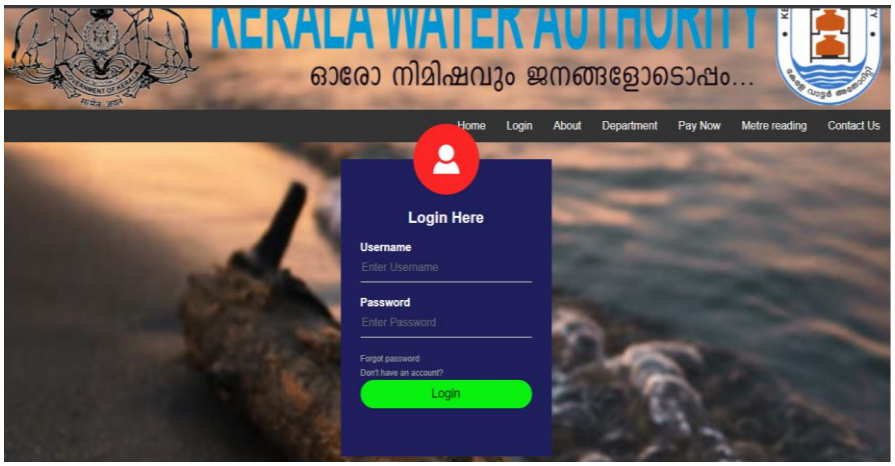
4.3 USER INTERFACE DESIGN USING FIGMA

4.3.1-INPUT DESIGN

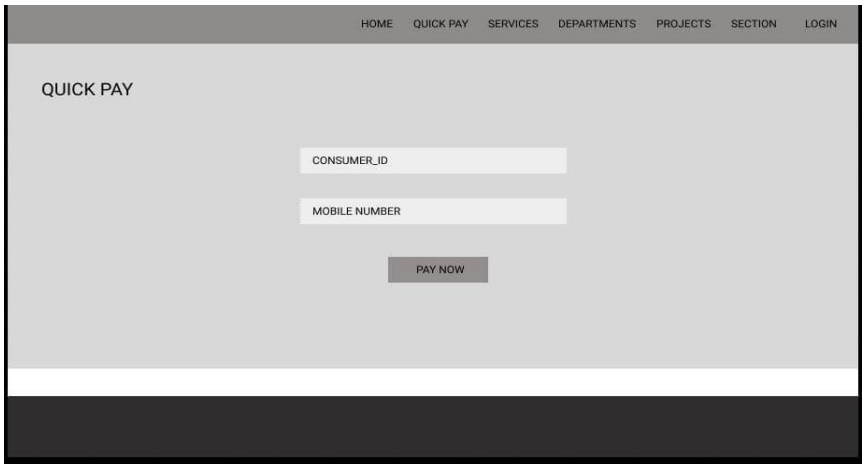
Form Name : Customer Registration



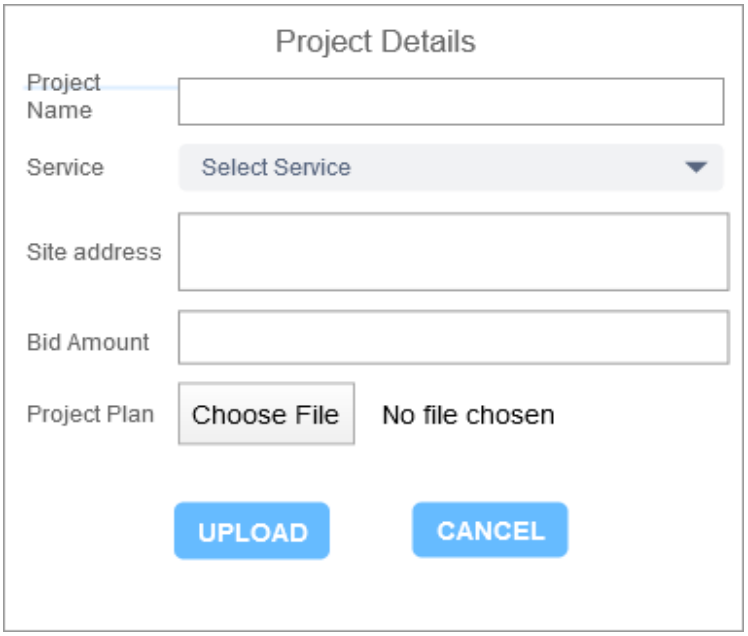
Form Name : User Login



Form Name : payment page



Form Name :Customer's file uploads



4.4 DATABASE DESIGN

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

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

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

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

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

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

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

4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

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

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

First Normal Form

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

Second Normal Form

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

Third Normal Form

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

TABLE DESIGN

Table No : 01

Table Name : tbl_login

Primary Key : login_id

Fieldname	Data Type	Size	Description
login_id	Int	10	Primary key of login table
Username	Varchar	20	unique username to login
Password	Varchar	20	password to login
user_type	Varchar	20	Type of users
Status	Varchar	20	Status of the user

Table No : 02

Table Name : tbl_laboursReg

Primary Key : lid

Fieldname	Data type	Size	Description
lid	Int	10	Primary key
Lab_name	Varchar	20	Labour name
phoneno	Varchar	20	Phone number of labour
email_id	Varchar	20	Email id of labour
dist_name	Int	10	Foreign key with reference from district table
state_name	Varchar	20	Foreign key with reference from state table
adharCardNo	Number	20	Adharcard number of labour
login_id	Int	10	Foreign key with reference from login table

Table No : 03

Table Name : tbl_customerReg

Primary Key : cust_id

Fieldname	Date type	Size	Description
cust_id	Int	10	Primary Key
cust_name	Varchar	20	Name of customer
Address	Varchar	50	Address of customer
postOffice	Varchar	20	Post office of customer
PIN Code	Number	20	Pin code number of customer
Phno	Number	20	Phone number of client
email_id	Varchar	20	Email id
dist_name	Int	10	Foreign key with reference from district table
state_name	Int	10	Foreign key with reference from state table
login_id	Int	10	Foreign key with reference from login table
Status	Varchar	20	Status of the customer

Table No : 04

Table Name : tbl_DailyProgressReport

Primary Key : report_id

Fieldname	Data type	Size	Description
report_id	Int	10	Primary key
title	Varchar	20	Title
activityDetails	Varchar	20	Activities performed today
date	Date	10	Today's date
status	Varchar	20	Status of the report

Table No : 05

Table Name : tbl_Cust_Bank_Accnt

Primary Key : payerid

Fieldname	Data type	Size	Description
payerid	Int	10	Primary key
accountName	Varchar	20	Account name of customer
accountNo	Number	20	Account number of customer
branchName	Varchar	20	Branch Name of customer bank account
IFSC Code	Varchar	20	IFSC Code of customer bank account
branchLocation	Varchar	20	Branch Location of customer bank account

Table No : 06

Table Name : tbl_complaint

Primary Key : comp_id

Field Name	Type	Size	Description
comp_id	Int	10	Primary key of complaint table
login_id	Int	10	Foreign key of login table by customer
to_login_id	Int	10	Foreign key of login table by contractor
complaint	Varchar	100	Brief summary of the issue
status	Int	10	Status of the complaint

Table No : 07

Table Name : tbl_feedback

Primary Key : fid

Fieldname	Data type	Size	Description
Fid	Int	10	Primary key
cust_id	Int	10	Foreign key with reference from customer table
Feedback	Varchar	100	Feedback

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

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

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

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

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

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

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

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

5.2 TEST PLAN

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

The levels of testing include:

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

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

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

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

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

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

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

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

5.2.4 Output Testing or User Acceptance Testing

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

- Input Screen Designs,
- Output Screen Designs,

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

5.2.5 Selenium testing

Selenium is one of the most widely used open-source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages. Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

Selenium supports a variety of programming languages through the use of drivers specific to each language. Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported program languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari. Selenium can be used to automate functional tests and can be integrated with automation test tools such as Maven**,** Jenkins**,** & Docker to achieve continuous testing. It can also be integrated with tools such as TestNG**, &** JUnit for managing test cases and generating reports.

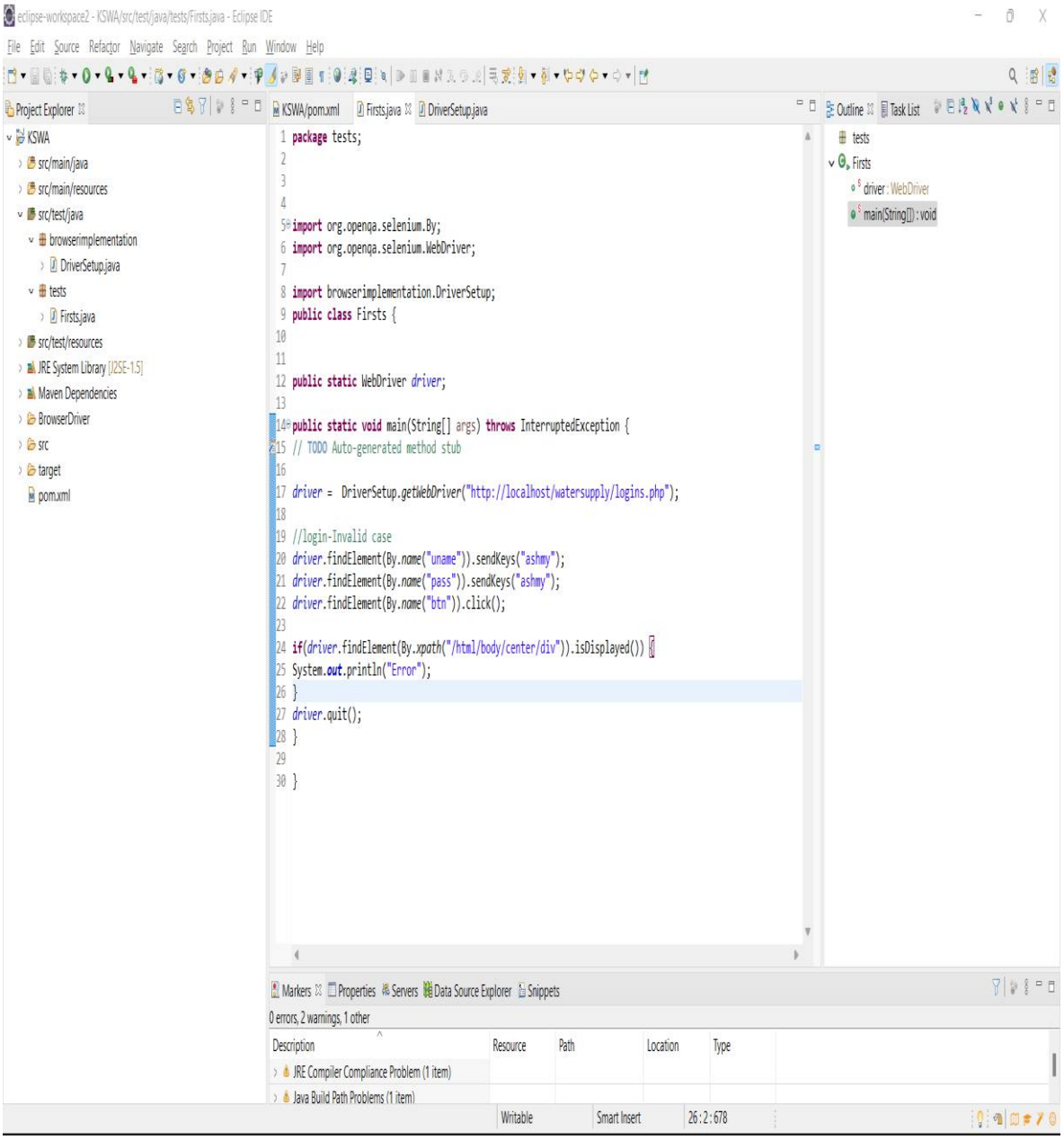
Test Case 1

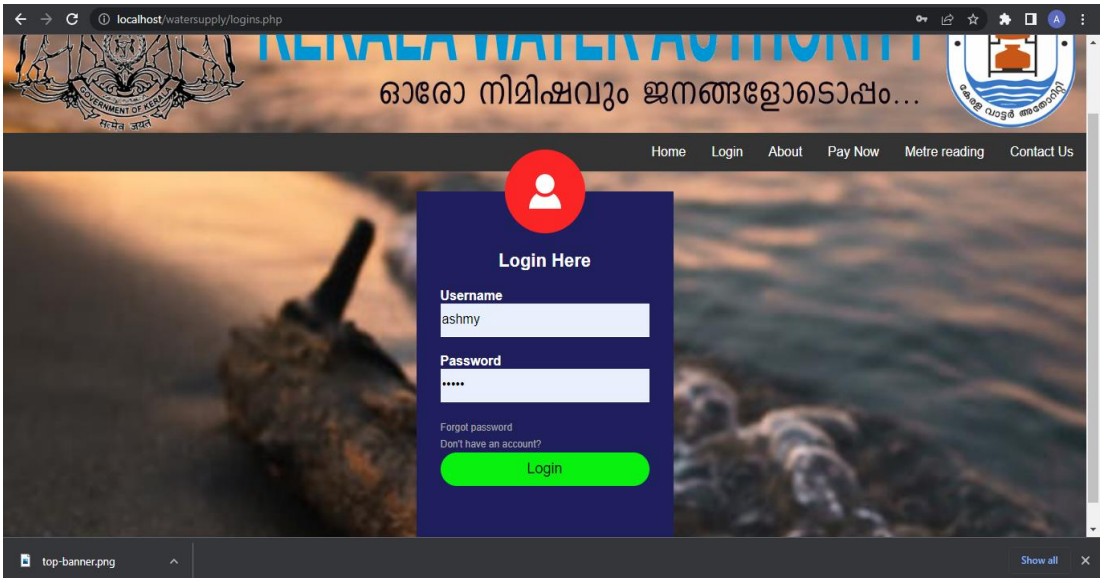
Project Name: Water Authority Management System					
Login Test Case					
Test Case ID: Fun_1			**Test Designed By:** AMMU SARA MATHEW		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 17-05-2022		
Module Name: Login Screen			**Test Executed By :** Ms. Paulin Paul		
Test Title : Verify login with mobile and password			**Test Execution Date:** 18-05-2022		
Description: Test the Login Page					
Pre-Condition : User has valid mobile number and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation toLogin Page		Login Page should be displaye d	Login page displayed	Pass
2	Provide Valid Mobile Number	Mobile Number: 9123456789	User should be able to Login	User Logged in and navigated to Home page of user side	Pass
3	Provide Valid Password	Password: ashmy			
4	Click on Sign In button				
5	Provide Invalid Mobile number or password	Email Id: 2345678902 Password: 1234	User should not be able to Login	Message for enter valid mobile number or password displayed	Pass
6	Provide Null Email Id or Password	Email Id: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Test Case 2

Project Name: Water Authority Management System					
Purchase History Test Case					
Test Case ID: Login			**Test Designed By:** AMMU SARA MATHEW		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 17-05-2022		
Module Name: Login Screen			**Test Executed By:** Ms. Paulin Paul		
Test Title: Verify login with username and password			**Test Execution Date:** 18-05-2022		
Description: Test the Purchase History Page					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigate to Login Page		Login Page should be displayed	Login Page displayed	Pass
2	Provide valid username	Username: ashmy	User should be able to login	User logged in and navigated to respective home page	Pass
3	Provide valid password	Password: ashmy@123			
4	Click on Login button				
5	Click on History in Orders		Purchase History must be displayed	Purchase History is displayed	Pass
5	Provide invalid username or password	Username: @ashmy Password: ashmy@1234	User should not be able to login	Message about Invalid Credentials is displayed	Fail
6	Provide NULL username or password	Username: (NULL) Password: (NULL)			
7	Click on Login button				
Post-Condition: User is validated with database and successfully logged in to their account. The account session details are logged in database.					

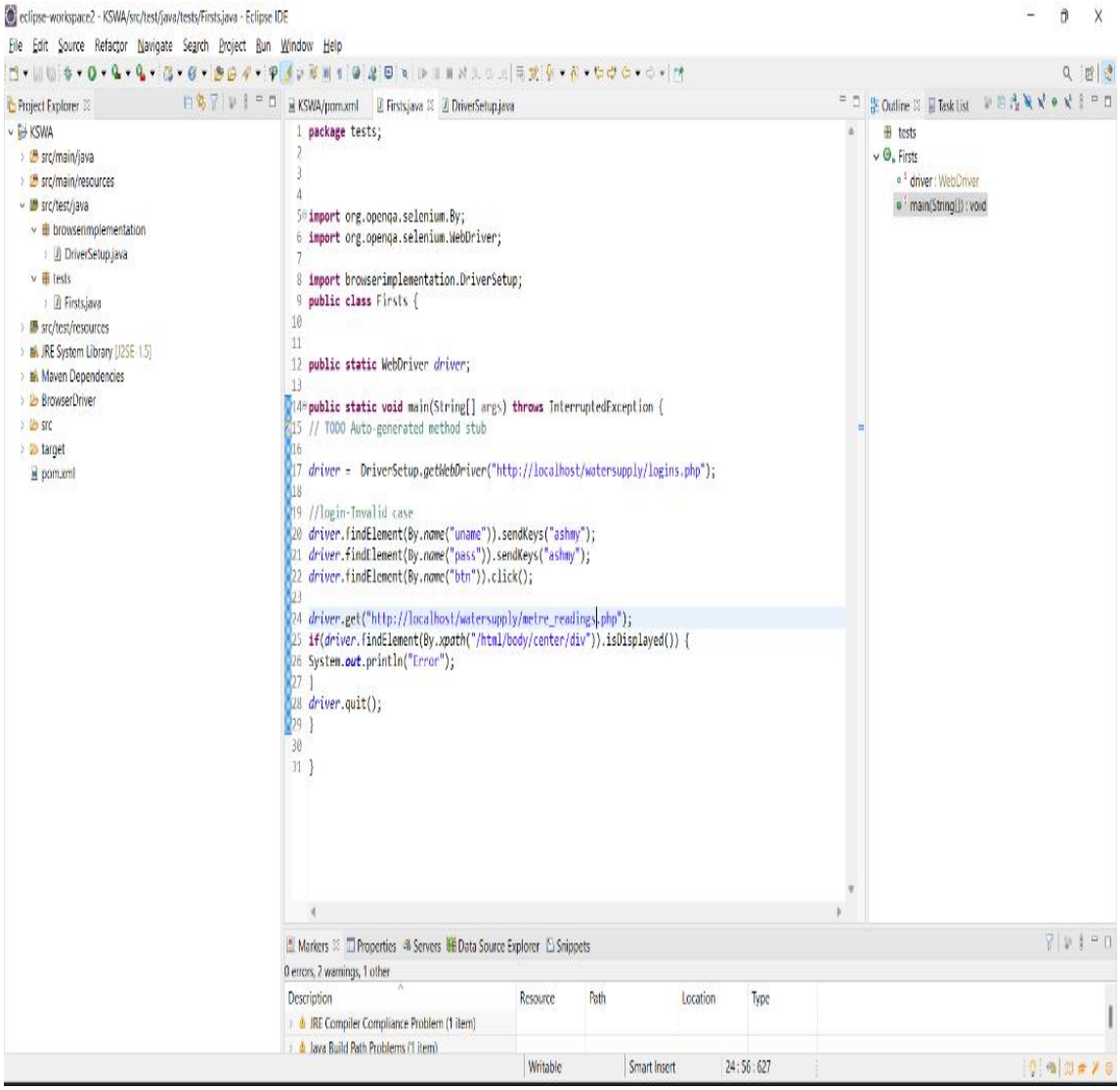
5.2.6: Login page testcase




localhost/watersupply/logins.php
KERALA WATER AUTHORITY
ഓരോ നിമിഷവും ജനങ്ങളോടൊപ്പം...
Home
Login
About
Pay Now
Metre reading
Contact Us
Login Here
Username
ashmy
Password
Forgot password
Don't have an account?
Login
top-banner.png
Show all


localhost/watersupply/KSWA%20home.php
Welcome ashmy
KERALA WATER AUTHORITY
ഓരോ നിമിഷവും ജനങ്ങളോടൊപ്പം...
Request
Profile
Home
Contact Us
About
Pay Now
Meter Related
Meter Readings
Logout
Pay
top-banner.png
Show all

5.2.7. Metre reading page testcase



CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

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

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

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

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to

ensure that the resistance does not build up, as one has to make sure that:

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

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

6.2.1 User Training

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

6.2.2 Training on the Application Software

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

6.2.3 System Maintenance

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

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current working system technology is not a bad one but still improving the system is always necessary in order to cope up with the change. The proposed system introduces facility for customer to upload projects by viewing profile of contractors. Provides lots of advantages like view profile of contractors, enabling unique customer id, enhanced user interface, payment options, add feedback, daily progress report option, complaint status and may more.

7.2 FUTURE SCOPE

Apart from all of this, the major future service is geographic information system (GIS), an important tool needed to supply managers with the information required to make decisions that will have a significant impact on the utilities demanded by their customers, It is to enable the water authority to prepare, update and manage the water distribution assets and related resources in a common repository.

.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.

WEBSITES:

- www.w3schools.com
- https://epay.kwa.kerala.gov.in/quickpay
- http://kwa.kerala.gov.in/ml/

CHAPTER 9

APPENDIX

9.1 Sample Code

Index.html

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

li {
  float: right;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

/* Change the link color to #111 (black) on hover */
li a:hover {
  background-color: #111;
}
* {box-sizing: border-box;}
body {font-family: Verdana, sans-serif;}
.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
```

```
  text-align: center;
  }

  /* Number text (1/3 etc) */
  .numbertext {
    color: #f2f2f2;
    font-size: 12px;
    padding: 8px 12px;
    position: absolute;
    top: 0;
  }

  /* The dots/bullets/indicators */
  .dot {
    height: 15px;
    width: 15px;
    margin: 0 2px;
    background-color: #bbb;
    border-radius: 50%;
    display: inline-block;
    transition: background-color 0.6s ease;
  }

  .active {
    background-color: #717171;
  }

  /* Fading animation */
  .fade {
    -webkit-animation-name: fade;
    -webkit-animation-duration: 1.5s;
    animation-name: fade;
    animation-duration: 1.5s;
  }

  @-webkit-keyframes fade {
    from {opacity: .4}
    to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}
</style>
<title>KSWA</title>
<link href="stylesheet/KSWA.css" rel="stylesheet" type="text/css">
<body>
```

```
<img src="images\Banner_Ed1.gif" style="width:100%">
<ul>
<li> <a href="ContactUs.php">Contact Us</a></li>
  <li><a href="meter related.php">Meter Related</a></li>
   <li><a href="metre_reading.php">Meter Readings</a></li>
 <li><a href="Paynow.php">Pay Now</a></li>
 <li><a href="about.php">About</a></li>
 <li><a href="logins.php">Login</a></li>
 <li><a href="index.php">Home</a></li>

</ul>

<div class="kswa">
<font color="red" size="32px" style="bold italic">

</font>
<font color="white" align="right-side">
</font>
  

</div>
<div class="slideshow-container">

<div class="mySlides fade">
<div class="numbertext">1 / 3</div>
<img src="images\cheruthoni-dam.jpg" style="width:100%">
<div class="text">Cheruthoni Reserviour,Idukki</div>
</div>

<div class="mySlides fade">
<div class="numbertext">2 / 3</div>
<img src="images\Pazhassi_Dam.jpg" style="width:100%">
<div class="text">pazhassi Reservior,Kannur</div>
</div>

<div class="mySlides fade">
<div class="numbertext">3 / 3</div>
<img src="images\KARAPUZHA12.jpg" style="width:100%">
<div class="text">Karapuzha Reserviour,Wayanad</div>
</div>


</div>
<br>

<div style="text-align:center">
<span class="dot"></span>
<span class="dot"></span>
<span class="dot"></span>
</div>

<script>
var slideIndex = 0;
```

```
 showSlides();
function showSlides() {
  var i;
  var slides = document.getElementsByClassName("mySlides");
  var dots = document.getElementsByClassName("dot");
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slideIndex++;
  if (slideIndex > slides.length) {slideIndex = 1}
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
</head>

</body>
</html>
```

Register.php

```
<?php

include "Connect.php";
if(isset($_POST["submit"]))
{

$sname=$_POST["name"];
$fname=$_POST["username"];
$lname=$_POST["name"];
$consumerno=$_POST["consumerno"];
$c=$_POST["address"];
$u=$_POST["locality"];
$p=$_POST["password"];


$q=mysqli_query($conn,"INSERT INTO `kswatable`(`name`, `username`, `email`, `consumerno`, `address`,
    `locality`, `password`, `status`) VALUES('$fname','$fname','$lname','$consumerno','$c','$u','$p','0')");
if($q)
{

echo "SUCCESS";
header("Location:logins.php");
  }else{
  echo mysqli_errno($conn);
  }
  }
  ?>
```

```
<!DOCTYPE html>
<html><style>
ul {
list-style-type: none;
margin: 0;
padding: 0;
overflow: hidden;
background-color: #333;
}

li {
float: right;
}

li a {
display: block;
color: white;
text-align: center;
padding: 14px 16px;
text-decoration: none;
}

/* Change the link color to #111 (black) on hover */
li a:hover {
background-color: rgba(1,0,0,0.43);
}
* {box-sizing: border-box;}
body {font-family: Verdana, sans-serif;}
.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
max-width: 1000px;
position: relative;
margin: auto;
}

/* Caption text */
.text {
color: #f2f2f2;
font-size: 15px;
padding: 8px 12px;
position: absolute;
bottom: 8px;

width: 100%;
text-align: center;
}
/* Number text (1/3 etc) */
.numbertext {

color: #f2f2f2;
```

```
font-size: 12px;
padding: 8px 12px;
position: absolute;
top: 0;
}

/* The dots/bullets/indicators */
.dot {
height: 15px;
width: 15px;
margin: 0 2px;
background-color: #bbb;
border-radius: 50%;
display: inline-block;
transition: background-color 0.6s ease;
}

.active {
background-color: black;
}

/* Fading animation */
.fade {
-webkit-animation-name: fade;
-webkit-animation-duration: 1.5s;
animation-name: fade;
animation-duration: 1.5s;
}

@-webkit-keyframes fade {
from {opacity: .4}
to {opacity: 1}
}

@keyframes fade {
from {opacity: .4}
to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
.text {font-size: 11px}
}

body {
color:white;

height: 100%;
/* Center and scale the image nicely */
background-position: center;
background-repeat: no-repeat;
background-size: cover;
font-family:Lucida Calligraphy;}
```

```
.welcome-button a {
font-size: 12px;



color:white;
background:rgba(1,0,0,0.45) ;
padding: 10px 20px;
text-transform: capitalize;
letter-spacing: 2px;
display: inline-block;
}
.welcome-button a:hover {
background: white;
color:red;
}

.form-w3ls {
background:rgba(1,0,0,0.43);/*--form purple --*/
padding: 40px 50px;
height : 20%;
width:30%;/*--purple form width --*/
margin:0 auto;
margin-top:25px;
float : center;
}

.header {

font-size: 20px;
color: black;
letter-spacing: 1px;
text-transform: uppercase;
}
div#signin-agile input[type="text"],div#signin-agile input[type="button"],div#signin-agile input[type="date"],div#signin-agile input[type="password"],div#signup-agile input[type="text"],div#signup-agile input[type="tel"],div#signup-agile input[type="email"],div#signup-agile input[type="password"]
{
padding: 10px;
width: 100%;
border: 1px solid rgba(30, 0, 41, 0);
color: white;
font-size: 16px;
background-color:rgba(1,0,0,0.44);/*--type space color --*/
margin-top: 1em;
margin-bottom: 2em;
font-family:Lucida Calligraphy;
}
div#signin-agile .sign-in, div#signin-agile .join-us
{
```

```
padding:10px;
background-color:rgba(1,0,0,0.43);
width:100%;



border:0;
margin-top: 1em;
margin-bottom: 2em;
color:white;
letter-spacing:1px;
font-family:arial;
font-size:18px;
text-transform:uppercase;
}
/*--//sing in button color and letter color--*/
div#signin-agile .sign-in:hover,div#signin-agile .join-us:hover {
background-color :black;
color:black;

}
div#signin-agile .button{
padding:10px;
background-color:rgba(1,0,0,0.43);
width:100%;
border:0;
color:;
letter-spacing:1px;

font-size:20px;
text-transform:uppercase;}
font-family:Lucida Calligraphy;
div#signin-agile .login100-form-btn:hover {
background-color :gray;
color:black;

}
</style>
<title>Registration_page</title>
<br><br><br><br>
<body>
<img src="images\Banner_Ed1.gif" style="width:75%">
<ul>
<li> <a href="ContactUs.php">Contact Us</a></li>
<li><a href="Meter Related.php">Meter Related</a></li>
<li><a href="Paynow.php">Pay Now</a></li>
<li><a href="about.php">About</a></li>
<li><a href="logins.php">Login</a></li>
<li><a href="KSWA home.php">Home</a></li>
<li><a href="metre reading.php">Meter Readings</a></li>
</ul>
```

```
<body background="images\1.jpg">
<title>Register</title>
<script type="text/javascript">
function chk()




{
var name=/^[a-zA-Z\s]+$/;
var ch=/^([a-z A-Z 0-9_\-\.])+\@([a-z A-Z 0-9_\-])+\.([a-z A-Z]{2,4}).$/;
var num=/^([0-9_\-])+$/;

if(document.getElementById("name").value.match(name))
{}
else
{



alert("Invaild Name");
document.getElementById("name").focus();
return false;
}
if(document.getElementById("pass1").value==document.getElementById("pass2").value)
{}
else
{
alert("Password Missmatch");
document.getElementById("pass2").focus();
return false;
}
}
</script>
</head>
<link href="stylesheet/style.css" rel="stylesheet" type="text/css">
<body><center>
<div id="content" style="margin-top:-5px;width:550px;">
<br>
<div id="Register" style="margin-top:-105px;">
<form method="post" action="Register.php">
<table align="center">
<tr>
<td>
<div id="Registerbox" align="center"><font size="5 px"><b>Register</b></font></div>
</td>
</tr>

<tr>
<td>
<input type="text" id="name" name="name" required placeholder="Enter your name"/>
<br><br>
</td></tr><tr><td>
```

```
<input type="text" id="username" name="username" required placeholder="Enter your username"/>
<br><br>
</td></tr><tr><td>
<input type="text" id="email" name="name" required placeholder="Enter your email"/>
<br><br></td></tr><tr><td>
<input type="text" id="consumerno" name="consumerno"  required placeholder="Enter your consumer no"/>
<br><br></tr><tr><td>
<input type="text" id="address" name="address" required placeholder="Enter your address"/>
<br><br></td></tr><tr><td>
<input type="text" id="locality" name="locality" required placeholder="Enter your locality"/>
<br><br></td></tr><tr><td>
<input type="password" id="pass1" name="password" required placeholder="Enter your Password"/><br><br>
</td></tr><tr>
<td style="text-align:center;">
<input type="submit" name="submit" value="Register" onclick="chk()"/><br><br><br>
</td></tr></table></form></div></div></div></center></body></html>
```

Paymentgateway.php

```
<?php
session_start();
require('stripe-php-master/config.php');
?>
<!DOCTYPE html>
<html>
  <title>Payment Gateway</title>
<head>
<style>
body {
  display: flex;
  justify-content: center;
  align-items: center;
  background: #242d60;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
  'Helvetica Neue', 'Ubuntu', sans-serif;
  height: 100vh;
  margin: 0;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
section {
  background: #ffffff;
  display: flex;
  flex-direction: column;
  width: 400px;
  height: 112px;
  border-radius: 6px;
  justify-content: space-between;
}
.product {
  display: flex;
```

```
  padding: 24px;
}
.description {
  display: flex;
  flex-direction: column;
  justify-content: center;
}
p {
  font-style: normal;
  font-weight: 500;
  font-size: 14px;
  line-height: 20px;
  letter-spacing: -0.154px;
  color: #242d60;
  height: 100%;
  width: 100%;
  padding: 0 20px;
  display: flex;
  align-items: center;
  justify-content: center;
  box-sizing: border-box;
}
h3,
h5 {
  font-style: normal;
  font-weight: 500;
  font-size: 14px;
  line-height: 20px;
  letter-spacing: -0.154px;
  color: #242d60;
  margin: 0;
}
h5 {
  opacity: 0.5;
}

button {
  height: 36px;
  background: #556cd6;
  color: white;
  width: 100%;
  font-size: 14px;
  border: 0;
  font-weight: 500;
  cursor: pointer;
  letter-spacing: 0.6;
  border-radius: 0 0 6px 6px;
  transition: all 0.2s ease;
  box-shadow: 0px 4px 5.5px 0px rgba(0, 0, 0, 0.07);
}
.cbutton
{
  height: 36px;
```

```
  background: #556cd6;
  color: white;
  width: 100%;
  font-size: 14px;
  margin-top:20px;
  border: 0;
  font-weight: 500;
  cursor: pointer;
  letter-spacing: 0.6;
  border-radius: 0 0 6px 6px;
  transition: all 0.2s ease;
  box-shadow: 0px 4px 5.5px 0px rgba(0, 0, 0, 0.07);
}
button:hover {
  opacity: 1;
}

.detailsection{
    display: block;
}
.anim{
    width:25%;
    margin: 0% 4%;
}
</style>

</head>
<div class="anim" id="anim"></div>
<section>
      <div class="product">
        <div class="description">
          <h3>The Amount to Be Pay By: <?php echo $_SESSION['n'];?></h3>
          <h5>₹<?php echo 1600;?></h5>
        </div>
      </div>
      <form action="order-history.php" method="POST">
      <button
          type="submit"
          value="Pay with Card"
          data-key="<?php echo $publishableKey?>"
          data-amount="<?php echo 1600  * 100;?>"
          data-currency="inr"
          data-name="KSWA"
          data-description="Water Supply"
        >Procced to Pay</button>
        <input type="button" id="delete" class="cbutton" name="cancelbutton" value="Cancel Payment">


        <script src="https://checkout.stripe.com/v2/checkout.js"></script>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.js"></script>
        <script src="js/delete.js"></script>
        <script>
```

```
    $(document).ready(function() {
      $(':submit').on('click', function(event) {
        event.preventDefault();

        var $button = $(this),
          $form = $button.parents('form');

        var opts = $.extend({}, $button.data(), {
          token: function(result) {
            $form.append($('<input>').attr({ type: 'hidden', name: 'stripeToken', value: result.id })).submit();
          }
        });

        StripeCheckout.open(opts);
      });
    });
    </script>
    <script src="./js/lottie.js"></script>
    <script src="js/app.js"></script>
</form>
  </section>

</body>
</html>
```

Metrereading.php

```
<?php
session_start();
$var=$_SESSION["n"];
?>

<!DOCTYPE html>
<html>
<head>
<title>metre reading</title>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

li {
  float: right;
}

li a {
  display: block;
  color: white;
  text-align: center;
```

```
  padding: 14px 16px;
  text-decoration: none;
}

/* Change the link color to #111 (black) on hover */
li a:hover {
  background-color: rgba(1,0,0,0.43);
}
* {box-sizing: border-box;}
body {font-family: Verdana, sans-serif;}
.mySlides {display: none;}
img {vertical-align: middle;}

/* Slideshow container */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}

/* Number text (1/3 etc) */
.numbertext {
  color: #f2f2f2;
  font-size: 12px;
  padding: 8px 12px;
  position: absolute;
  top: 0;
}

/* The dots/bullets/indicators */
.dot {
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active {
  background-color: black;
}
```

```
/* Fading animation */
.fade {
  -webkit-animation-name: fade;
  -webkit-animation-duration: 1.5s;
  animation-name: fade;
  animation-duration: 1.5s;
}

@-webkit-keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
  .text {font-size: 11px}
}body {
            color:white;

   height: 100%;

  /* Center and scale the image nicely */
 background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
  background-image:url("image/banner2.jpg");
    font-family:Lucida Calligraphy;
  }
  .welcome-button a {
    font-size: 12px;
    color:white;
    background:rgba(1,0,0,0.45) ;
    padding: 10px 20px;
    text-transform: capitalize;
    letter-spacing: 2px;
    display: inline-block;
}
.welcome-button a:hover {
  background: white;
  color:red;
}

   .form-w3ls {
  background:rgba(1,0,0,0.43);/*--form purple --*/
  padding: 40px 50px;
  height : 20%;
  width:30%;/*--purple form width --*/
  margin:0 auto;
```

```
  margin-top:25px;
  float : center;
}

   .header {

    font-size: 20px;
    color: black;
    letter-spacing: 1px;
    text-transform: uppercase;
   }
   div#signin-agile input[type="text"],div#signin-agile input[type="button"],div#signin-agile
input[type="date"],div#signin-agile input[type="password"],div#signup-agile input[type="text"],
div#signup-agile input[type="tel"],div#signup-agile input[type="email"],div#signup-agile
input[type="password"]
   {
  padding: 10px;
    width: 100%;
    border: 1px solid rgba(30, 0, 41, 0);
    color: white;
    font-size: 16px;
    background-color:rgba(1,0,0,0.44);/*--type space color --*/
    margin-top: 1em;
    margin-bottom: 2em;
   font-family:Lucida Calligraphy;
   }

   div#signin-agile .sign-in, div#signin-agile .join-us
{
  padding:10px;
  background-color:rgba(1,0,0,0.43);
  width:100%;
  border:0;
  margin-top: 1em;
    margin-bottom: 2em;
    color:white;
    letter-spacing:1px;
  font-family:Lucida Calligraphy;
    font-size:18px;
    text-transform:uppercase;
}
/*--//sing in button color and letter color--*/
div#signin-agile .sign-in:hover,div#signin-agile .join-us:hover {
  background-color :black;
  color:black;

  }
  div#signin-agile .button{
padding:10px;
  background-color:rgba(1,0,0,0.43);
  width:100%;
  border:0;
    color:;
```

```
   letter-spacing:1px;
   font-size:20px;
    text-transform:uppercase;}
font-family:Lucida Calligraphy;
  div#signin-agile .login100-form-btn:hover {
  background-color :gray;
  color:black;

  }
</style>
<title>KSWA</title>
</head>
 <?php
 include 'Connect.php';
 $id2=$_SESSION['id2'];
 $sql= "SELECT * FROM `kswatable` where id='$id2'";
 $result=mysqli_query($conn,$sql);
 $row=mysqli_fetch_array($result);?>
<body>
<center>
<img src="images\Banner_Ed1.gif" style="width:100%">
<ul>

 <li> <a href="ContactUs.php">Contact Us</a></li>
  <li><a href="Staff.php">Meter Related</a></li>

  <li><a href="about.php">About</a></li>
  <li><a href="logout.php">Logout</a></li>

  <li><a href="metre reading.php">Meter Readings</a></li>
</ul>
<body background="images\1.jpg">
<div class="form-w3ls">

    <div class="tab-content">
        <div id="signin-agile">

   <form  action="payment2.php" method="POST" > <table align="center">
  <tr><td>
  <div id="metre readingbox" align="center"><font size="5
px">     <b>METER READING</b></font></div>

   <input type="text" id="username" name="username" value="<?php echo $row['name'];?>" readonly />
                  <br><br>
                </td>
              </tr>

<tr>
   <td><input type="text" id="cnum" name="cnum" maxlength="13" required placeholder="Enter your
Consumer No: ">
 <br><br></td> </tr><tr><td>
<input type=text" id="meterpay" name="meterpay" required placeholder="Enter your meter reading" />
 <br><br></td>
```

```
<td><input type="date" name="date" required placeholder="date"/>
</td></tr><tr><td>NB: 10 Rupee Per Litre</td></tr>
<script >
  function amt(){
  var first_number = Number(document.getElementById("meterpay").value);
  var result = first_number * 10;
  alert(result);
  Number(document.getElementById("cal").value) = result;
  }
 </script>
 </table> <br><br>
 <input type="submit" name="bill" value="BILL">
  </form></div></div></center>
</body>

</html>

 <?php
 include 'Connect.php';

//$yy=$_SESSION['consumerno'];
if(isset($_POST['bill']))
{
$uname=$_SESSION['uname']=$_POST['username'];
$cno=$_SESSION['cno']=$_POST['cnum'];
$pay=$_SESSION['pay']=$_POST['meterpay'];
$date=$_SESSION['date']=$_POST['date'];


}
?>
```

Payment Gateway

```
<?php
session_start();
require('stripe-php-master/config.php');
?>
<!DOCTYPE html>
<html>
 <title>Payment Gateway</title>
<head>
<style>
body {
 display: flex;
 justify-content: center;
 align-items: center;
 background: #242d60;
 font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
 'Helvetica Neue', 'Ubuntu', sans-serif;
```

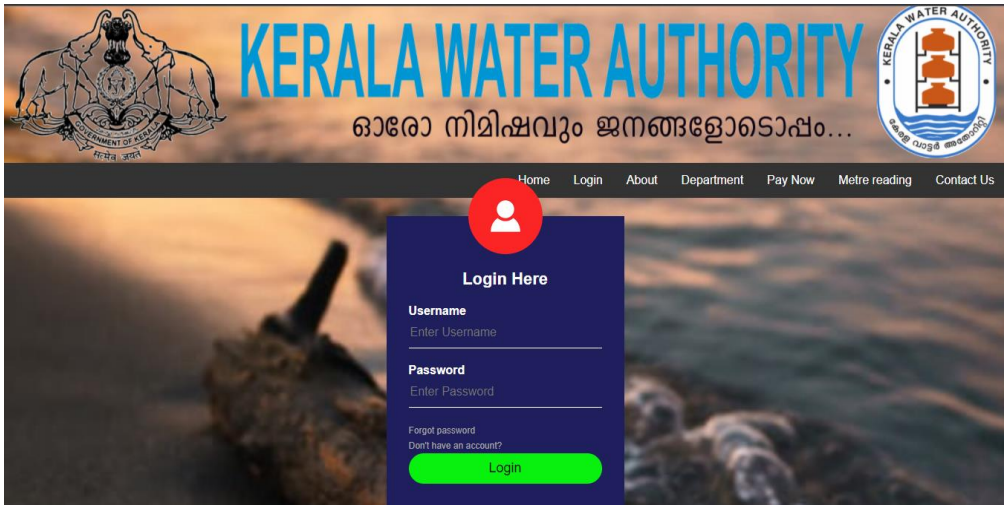
```
  height: 100vh;
  margin: 0;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
section {
  background: #ffffff;
  display: flex;
  flex-direction: column;
  width: 400px;
  height: 112px;
  border-radius: 6px;
  justify-content: space-between;
}
.product {
  display: flex;
  padding: 24px;
}
.description {
  display: flex;
  flex-direction: column;
  justify-content: center;
}
p {
  font-style: normal;
  font-weight: 500;
  font-size: 14px;
  line-height: 20px;
  letter-spacing: -0.154px;
  color: #242d60;
  height: 100%;
  width: 100%;
  padding: 0 20px;
  display: flex;
  align-items: center;
  justify-content: center;
  box-sizing: border-box;
}
h3,
h5 {
  font-style: normal;
  font-weight: 500;
  font-size: 14px;
  line-height: 20px;
  letter-spacing: -0.154px;
  color: #242d60;
  margin: 0;
}
```

```
h5 {
  opacity: 0.5;
}

button {
  height: 36px;
  background: #556cd6;
  color: white;
  width: 100%;
  font-size: 14px;
  border: 0;
  font-weight: 500;
  cursor: pointer;
  letter-spacing: 0.6;
  border-radius: 0 0 6px 6px;
  transition: all 0.2s ease;
  box-shadow: 0px 4px 5.5px 0px rgba(0, 0, 0, 0.07);
}
.cbutton
{
  height: 36px;
  background: #556cd6;
  color: white;
  width: 100%;
  font-size: 14px;
  margin-top:20px;
  border: 0;
  font-weight: 500;
  cursor: pointer;
  letter-spacing: 0.6;
  border-radius: 0 0 6px 6px;
  transition: all 0.2s ease;
  box-shadow: 0px 4px 5.5px 0px rgba(0, 0, 0, 0.07);
}
button:hover {
  opacity: 1;
}

.detailsection{
    display: block;
}
.anim{
    width:25%;
    margin: 0% 4%;
}
</style></head>
<div class="anim" id="anim"></div>
<section>
```

```
    <div class="product">
      <div class="description">
        <h3>The Amount to Be Pay By: <?php echo $_SESSION['n'];?></h3>
        <h5>₹<?php echo 1600;?></h5>
      </div>
    </div>
    <form action="order-history.php" method="POST">
    <button
        type="submit"
        value="Pay with Card"
        data-key="<?php echo $publishableKey?>"
        data-amount="<?php echo 1600  * 100;?>"
        data-currency="inr"
        data-name="KSWA"
        data-description="Water Supply"
      >Procced to Pay</button>
      <input type="button" id="delete" class="cbutton" name="cancelbutton" value="Cancel
Payment">


      <script src="https://checkout.stripe.com/v2/checkout.js"></script>
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.js"></script>
      <script src="js/delete.js"></script>
      <script>
      $(document).ready(function() {
        $(':submit').on('click', function(event) {
          event.preventDefault();

          var $button = $(this),
            $form = $button.parents('form');

          var opts = $.extend({}, $button.data(), {
            token: function(result) {
              $form.append($('<input>').attr({ type: 'hidden', name: 'stripeToken', value: result.id
})).submit();
            }
          });

          StripeCheckout.open(opts);
        });
      });
      </script>
      <script src="./js/lottie.js"></script>
      <script src="js/app.js"></script>
</form>
   </section>

</body>
</html>
```

SCREENSHOTS

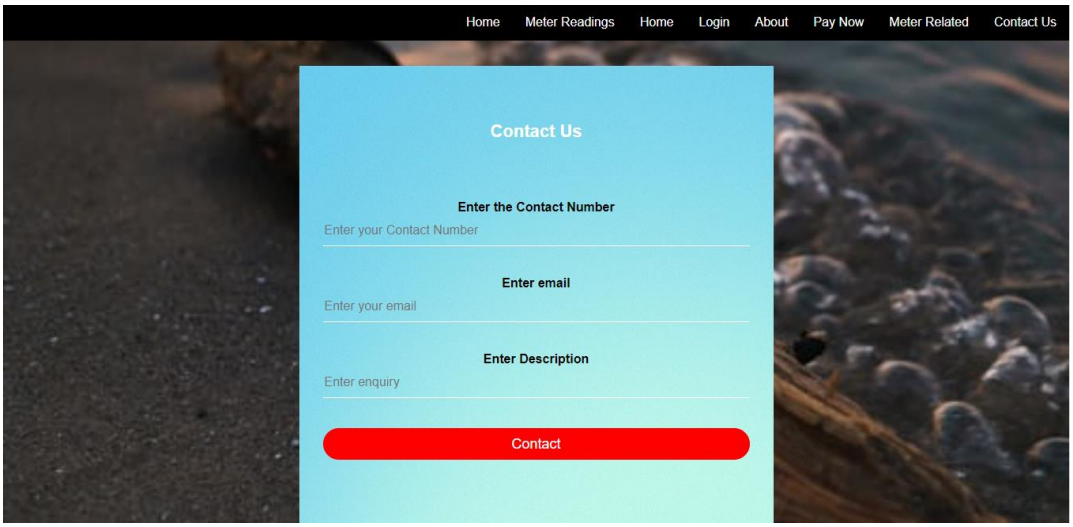
Login page



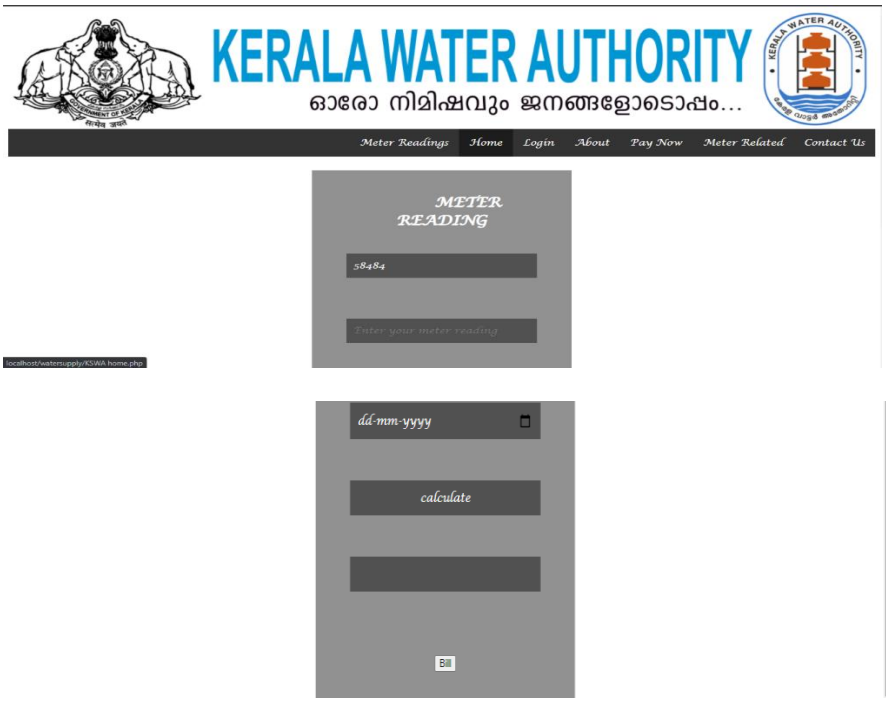
Meter related Page

Home Page

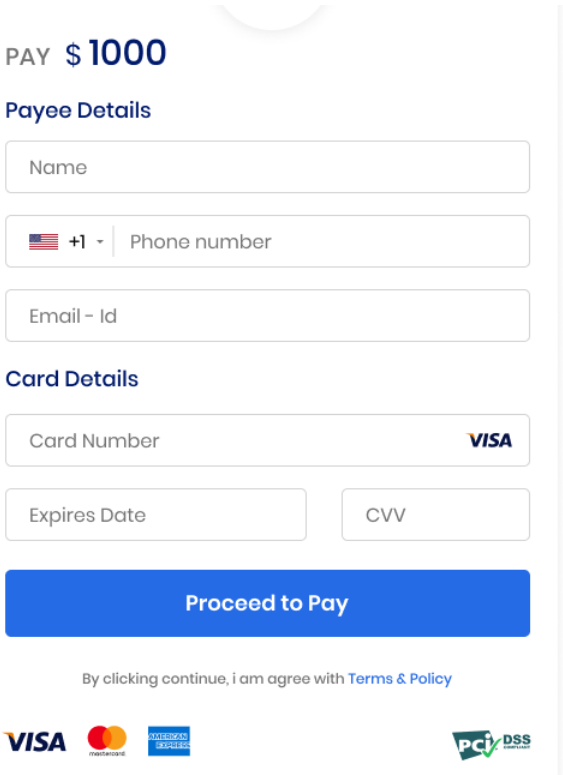
Contact us page



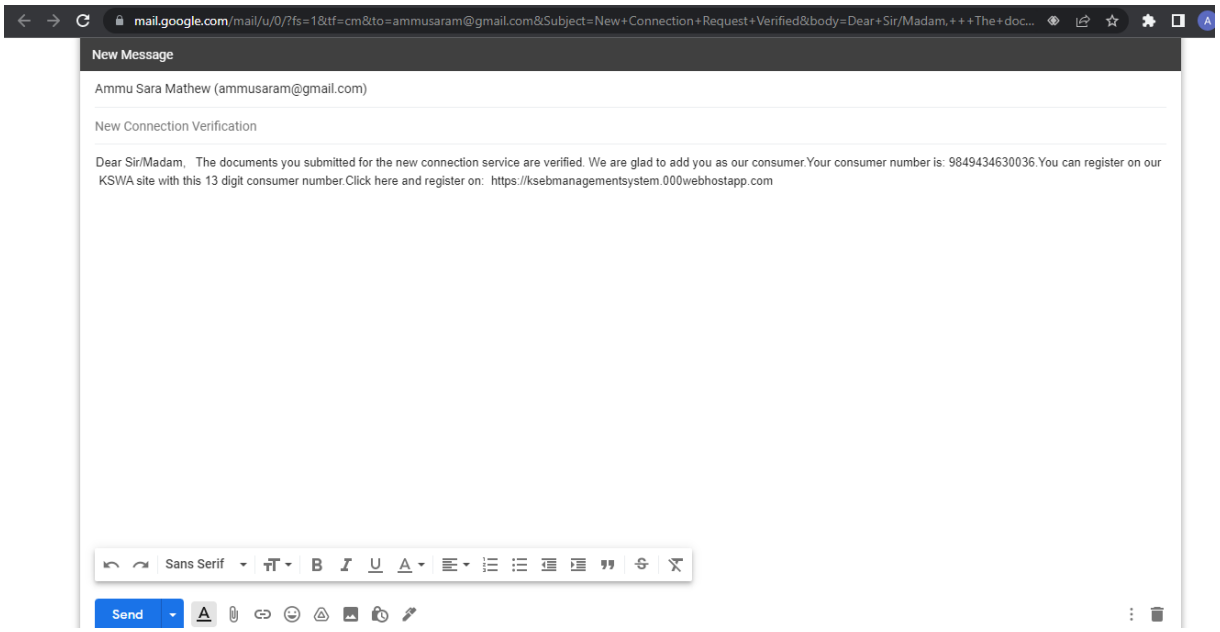
Meter Reading page



Payment page



Consumer Mail page



Customer table