



Visvesvaraya Technological University
Belagavi, Karnataka

ವಿಶ್ವೇಶ್ವರಯ್ಯ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ
ಬೆಳಗಾವಿ, ಕರ್ನಾಟಕ

DATABASE MANAGEMENT SYSTEM MINI-PROJECT REPORT
ON

“ WATER REFILLING MANAGEMENT SYSTEM ”

Submitted by

Name

USN

1. PRAJWAL P KASHYAP
2. SUDEEP P

1KI20CS070
1KI20CS106

Submitted in partial fulfillment of the requirements for the award of degree

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

Dr. CHETHAN CHANDRA S BASAVARADDI -B.E., M.Tech., PhD.,D.Litt.,

Associate Professor, Dept of CSE

KIT, Tiptur.



Department of Computer Science & Engineering

Kalpataru Institute of Technology

Tiptur-572201

Kalpataru Vidya Samsthe®



KALPATARU INSTITUTE OF TECHNOLOGY
COMPUTER SCIENCE AND ENGINEERING
CERTIFICATE

This is to certify that the database management system Mini Project Entitled

“WATER REFILLING MANAGEMENT SYSTEM”

Submitted by

1. PRAJWAL P KASHYAP

1KI20CS070

2. SUDEEP P

1KI20CS106

Students of 5th Semester B.E. CSE, in partial fulfillment of the requirement for the award of degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2022-23.

Signature of Guide

Signature of HOD

**Dr. CHETHAN
CHANDRA S BASAVA
RADDI Assoc. Professor
Dept of CSE
KIT, Tiptur.**

**Prof. SHASHIDHARA M S
Assoc. Professor &
Head Dept of CSE
KIT, Tiptur.**

Name of Examiners

Signature with date

1. _____

ACKNOWLEDGEMENT

On presenting the Database Management System Laboratory Mini Project report on “**WATER REFILLING MANAGEMENT SYSTEM**”, we feel great to express my humble feelings of thanks to all those who have helped us directly or indirectly in the successful completion of the mini-project work.

We would like to thank the Principal **Dr G.D. Gurumurthy** , KIT-Tiptur. Also, we would like to thank **Prof. Shashidhara MS**, Associate Professor and HOD, Dept. of CSE KIT-Tiptur for all their support and encouragement.

We would like to thank our respected guide **Dr. Chethan Chandra S Basavaraddi** Dept. of CS, for helping us a lot in completing this task, for his continuous encouragement and guidance throughout the mini-project work.

We are grateful to **Department of Computer Science and Engineering** and our institution, **Kalpataru Institute of Technology** for imparting us the knowledge with which we could do our best.

At the End, we would also like to thank the whole teaching and non-teaching staff Of CSE Department.

Thank You

Prajwal P Kashyap - 1KI20CS070

Sudeep P - 1KI20CS106

ABSTRACT

The Purpose of “**WATER REFILLING MANAGEMENT SYSTEM**” Design is to overcome difficulties in manual operation in refilling station. The difficulty in the manual system is one of the reasons why the efficiency in availing services of the clients is not satisfying and keeping of records is often misplaced and not secure.

This system manages to display the data to be filled by the user according to the information of the customer in organize manner, such that their personal details, and the services they want to avail as well as the payment on the transaction they purchased. The system keeps the information of the customer and the details of what they purchased.

The system coordinates the arrangement on the delivery of products. It consists all the records for the location of the clients, date of transaction, schedule of delivery, contact number and the person assigned to deliver and the payment of customer to the quantity of product that about to deliver.

TABLE OF CONTENTS

1. INTRODUCTION-----	01 - 02
1.1 Introduction	
2. REQUIREMENTS SPECIFICATION-----	03 - 04
2.1 Introduction	
2.2 Hardware requirements	
2.3 Software requirements	
3. ANALYSIS-----	05 - 08
3.1 Existing System	
3.2 Proposed System	
3.3 Software Specification	
4. DESIGN-----	09 - 12
4.1 System Design	
4.2 Introduction to UML	
4.3 UML Diagrams of	
5. SYSTEM IMPLEMENTATION-----	13 - 19
5.1 Introduction	
5.2 Sample code	
6. TESTING-----	20 - 22
6.1 Introduction	
6.2 Testing Methods	
7. SAMPLE SCREENSHOTS-----	23 - 25
8. CONCLUSION-----	26
• BIBLIOGRAPHY-----	27

1. INTRODUCTION

1.1 INTRODUCTION

One of the fastest and expanding businesses today is water refilling station. Waterrefilling station is small water system that has its own water purification facility producing a portable drinking water. The aqua water refilling system has their own water tank and equipment that intend on their business.

So, we design a system in this kind of business in order to be on top and align on the fast-growing business that is demand now a day.

The purpose of water refilling management system is to overcome difficulties in manual operation in refilling station. The difficulty in manual system are one of the reasons why the efficiency in availing services of the clients is not satisfying and keeping of records is often misplaced and not secure.

This system will be programmed to java that can enable the user to record things that are being purchased by the clients and it be created using MySQL database.

This system manages to display the data to be filled by the user according to the information of the customer in organize manner, such that their personal details, and the services they want to avail as well as the payment on the transaction they purchased. The system keeps the information of the customer and the details of what they purchased.

The system coordinates the arrangement on delivery of products. It consists all the records for the location of the clients, date of transaction, schedule of delivery, contact number and the person assign to deliver and the payment of customer to the quantity of product that about to deliver.

The system also views the information about the availability of the products as well as the containers. The system views the available containers to provide stocks again

This system also manages the information of the employees that a refilling station must have just like front liner, cashier, technical assistant, and delivery an. It stores the information in organize so that it easy to the owner to access on the detail of his/her employee.

Upon having this system, it will provide the capacity to the owner and clients to transact without spending time and effort.

1.1.1 Objective:

- 1) To identify the existing system used by Aijem's Water Refilling Station.
- 2) To identify the problems encountered in the existing system.
- 3) To describe the features of the system.
- 4) To test the acceptability of the system

1.1.2 Scope of the Project:

- 1) The system administrator will have access to the whole system including all administrative rights these includes; changes of passwords, addition, deletion and updating of information, reports viewing and access to transaction. Cashier can only perform transaction on the system it does not have administrative rights except adding and updating customer information.
- 2) Customer Information
- 3) Ordering and billing
- 4) Sales summary report
- 5) Summary of expenses report
- 6) Reports for the list of customers for follow up

2. REQUIREMENTS SPECIFICATION

2.1 INTRODUCTION:

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer. These pre-requisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

2.2 HARDWARE REQUIREMENTS:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

HARDWARE REQUIREMENTS FOR PRESENT PROJECT:

- ➔ PROCESSOR : i 5 / ryzen 5
- ➔ RAM : 4 GB
- ➔ HARD DISK : 250 MB

2.3 SOFTWARE REQUIREMENTS:

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

SOFTWARE REQUIREMENTS FOR PRESENT PROJECT :

- ➔ **OPERATING SYSTEM :** Windows XP /10 / 11
- ➔ **FRONT END :** HTML , CSS, JS
- ➔ **SERVER SIDE SCRIPT :** PHP
- ➔ **DATABASE :** MySQL
- ➔ **TOOL :** Visual Studio

IMPLEMENTATION is the stage of the project when the ‘Theoretical Design’ is turned out into a ‘Working System’. Thus, it can be considered to be the “Most Critical Stage” in achieving a ‘Successful New System’ ; and in giving User the Confidence, that the new system will work, be effective and satisfies the need of developing it.

The Implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. However, the Implementation can be achieved only upto 90% when compared to that of the Theoretical Design. This doesn’t mean that the working requirement can be compromised. Though we cannot implement the Theoretical Design as it is, we should make sure that the desired working nature of the Project should get implemented to the most.

To Implement this “Water Refilling Management System”, we have used Front-end Technologies like HTML, CSS and JavaScript ; PHP as the Back-end Technology ; and MySQL as the Database Query Language. By Default, to integrate our codes written in different languages, we are making use of “Xampp Server” to run our Project Code on the “Google Chrome” Web Browser

3. ANALYSIS

3.1 EXISTING SYSTEM:

Water refilling plant currently use a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the water refilling management infrastructure. Often information is incomplete or does not follow management standards. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost.

3.2 PROPOSED SYSTEM:

The Water Refilling Management System is designed for any water refilling plants to replace their existing manual paper based system. The new system is to control the information of users. Room availability, sales and sales items and user invoices. These services are to be provided in an efficient, cost effective manner, with the goal of reducing the time and resources currently required for such tasks .

3.3 SOFTWARE SPECIFICATION:

HTML:

HTML or Hypertext Markup Language is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets(like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example . The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

CASCADING STYLE SHEETS(CSS):

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation. CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. [1]

This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content. CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

MySQL:

MySQL is developed, distributed, and supported by Oracle Corporation. MySQL is a database system used on the web it runs on a server. MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use. It supports standard SQL. MySQL can be compiled on a number of platforms. The data in MySQL is stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful when storing information categorically. MySQL is the world's most popular open-source database. Despite its powerful features, MySQL is simple to set up and easy to use. Below are some instructions to help you get MySQL up and running in a few easy steps. We also explain how to perform some basic operations with MySQL using the **mysql** client.

FEATURES OF MySQL:

Internals and portability:

- Written in C and C++.
- Tested with a broad range of different compilers.
- Works on many different platforms.
- Tested with Purify (a commercial memory leakage detector) as well as with Val grind, a GPL tool.
- Uses multi-layered server design with independent modules.

Security:

- A privilege and password system that is very flexible and secure, and that enables
- host-based verification.
- Password security by encryption of all password traffic when you connect to a server.

CONNECTIVITY:

- Clients can connect to MySQL Server using several protocols:
- Clients can connect using TCP/IP sockets on any platform.
- On Windows systems in the NT family (NT, 2000, XP, 2003, or Vista), clients can connect using named pipes if the server is started with the --enable-named-pipe option. In MySQL 4.1 and higher, Windows servers also support shared-memory connections if started with the --shared-memory option.
- Clients can connect through shared memory by using the --protocol=memory option.
- On UNIX systems, clients can connect using Unix domain socket files.

LOCALIZATION:

- The server can provide error messages to clients in many languages.
- All data is saved in the chosen character set.

CLIENTS AND TOOLS:

- MySQL includes several client and utility programs. These include both command-line programs such as mysqldump and mysqladmin, and graphical programs such as MySQL Workbench.
- MySQL Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the mysqlcheck client. MySQL also includes myisamchk , a very fast command-line utility for performing these operations on MyISAM tables.
- MySQL programs can be invoked with the --help or -? option to obtain online assistance.

WHY TO USE MySQL:

- Leading open source RDBMS
- Ease of use— No frills
- Fast
- Robust
- Security
- Multiple OS support
- Free
- Technical support
- Support large database – up to 50 million rows, file size limit up to 8 Million TB

4. DESIGN

4.1 SYSTEM DESIGN:

4.2 INTRODUCTION TO UML:

UML Design :

The Unified Modeling Language (UML) is a standard language specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

- Visualizing
- Specifying
- Constructing
- Documenting

➤ **Visualizing :**

Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

➤ **Specifying :**

Specifying means building, models that are precise, unambiguous and complete. UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

➤ **Constructing :**

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

➤ **Documenting :**

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, design, source code, project plans, tests, prototypes, releases, etc...

4.3 UML Approach:

UML Diagram :

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs. You draw a diagram to visualize a system from different perspectives, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams, or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

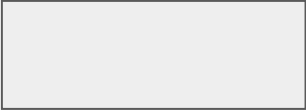

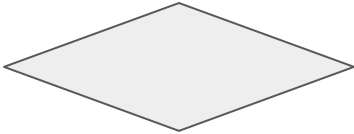

1. Class diagram
2. Object diagram
3. Use case diagram
4. Sequence diagram
5. Collaboration diagram
6. State chart diagram
7. Activity diagram
8. Component diagram
9. Deployment diagram

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

ENTITY-RELATIONSHIP DIAGRAMS :

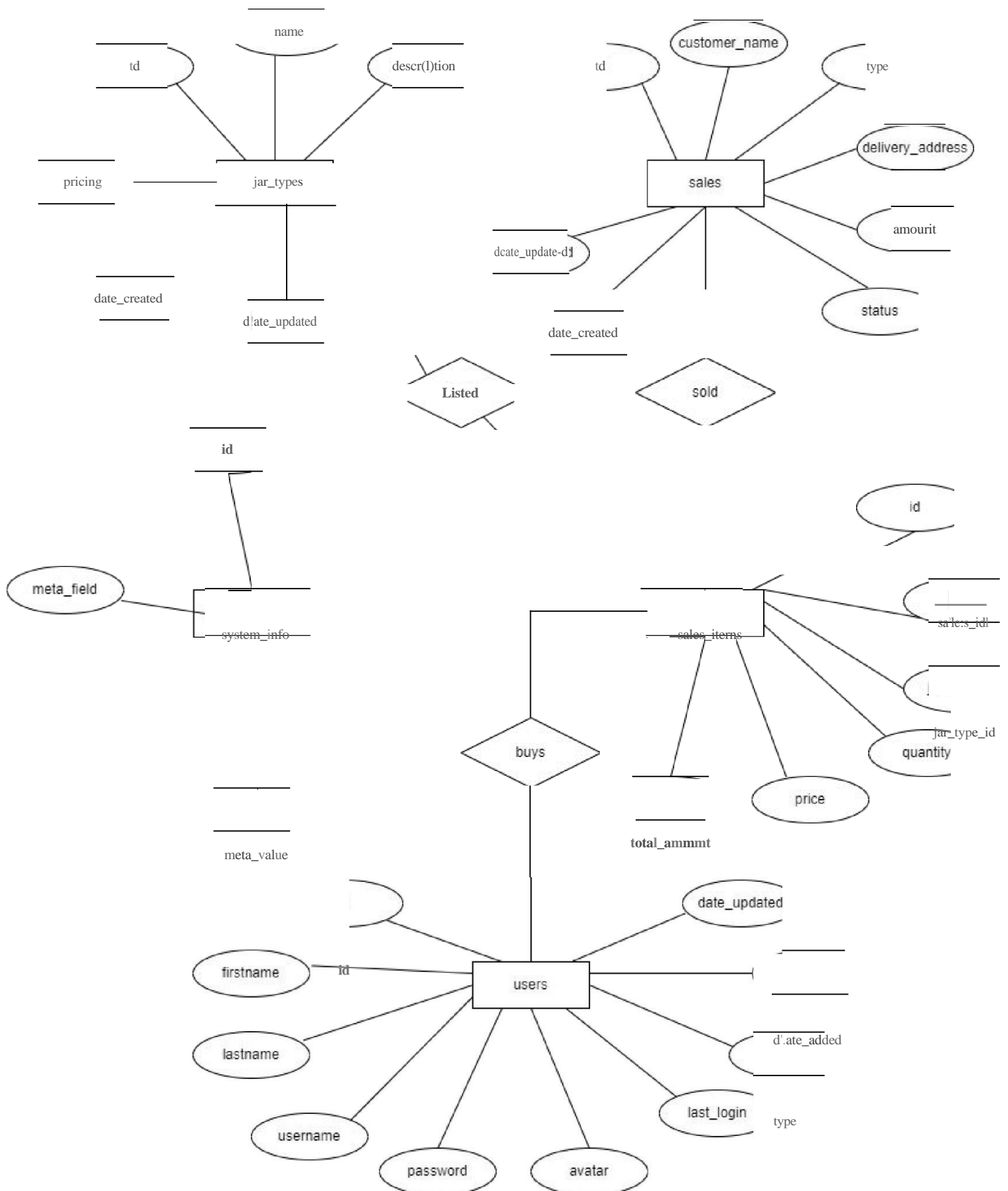
E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table. The symbols used in E-R diagrams are:

<u>SYMBOL</u>	<u>PURPOSE</u>
	Represents Entity sets
	Represent attributes
	Represent Relationship Sets
	Line represents flow

Structured analysis is a set of tools and techniques that the analyst.

To develop a new kind of a system:

The traditional approach focuses on the cost benefit and feasibility analysis, Project management, and hardware and software selection a personal considerations.



5. SYSTEM IMPLEMENTATION

5.1 Introduction:

IMPLEMENTATION is the stage of the project when the ‘Theoretical Design’ is turned out into a ‘Working System’. Thus, it can be considered to be the “Most Critical Stage” in achieving a ‘Successful New System’ ; and in giving User the Confidence, that the new system will work, be effective and satisfies the need of developing it.

The Implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. However, the Implementation can be achieved only upto 90% when compared to that of the Theoretical Design. This doesn’t mean that the working requirement can be compromised. Though we cannot implement the Theoretical Design as it is, we should make sure that the desired working nature of the Project should get implemented to the most.

To Implement this “Water Refilling Management System”, we have used Front-end Technologies like HTML, CSS and JavaScript ; PHP as the Back-end Technology ; and MySQL as the Database Query Language. By Default, to integrate our codes written in different languages, we are making use of “Xampp Server” to run our Project Code on the “Google Chrome” Web Browser

5.2 SAMPLE CODE :

```
SET SQL_MODE =
"NO_AUTO_VALUE_ON_ZERO"; START
TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `water_refilling_db`
--
```

```
-- -----
```

```
--
-- Table structure for table `jar_types`
--
```

```
CREATE TABLE `jar_types` (
  `id` int(30) NOT NULL,
  `name` text NOT NULL,
  `description` text NOT NULL,
  `pricing` float NOT NULL,
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
-- Dumping data for table `jar_types`
--
```

```
INSERT INTO `jar_types` (`id`,`name`,`description`,`pricing`,
`date_created`,`date_updated`) VALUES
(1, 'Slim Container with cap and faucet', '<span style="color: rgb(0, 0, 0); font-family: \'Open Sans\',
Arial, sans-serif; font-size: 14px; text-align: justify;">
```

Integer a risus enim. Mauris justo erat, tempus eu mauris sed, scelerisque tincidunt diam. Nam eget augue aliquam, commodo ligula consequat, maximus tellus. Suspendisse elit eros, pellentesque nec enim non, tincidunt pharetra magna. Vestibulum vel ex nunc. Nam semper diam et diam efficitur blandit. In ullamcorper dolor nec mauris vulputate, vel blandit purus elementum.

(2, 'Round Container with Cap', '

Nunc a massa id ligula varius convallis in non augue. Sed feugiat blandit mattis. Mauris pulvinar fringilla tellus a accumsan. Nunc pharetra semper posuere. Ut rutrum odio at lectus maximus suscipit. Sed feugiat turpis a auctor malesuada. Integer ante quam, suscipit eu aliquet eget, ullamcorper ac justo. Maecenas nec orci non ipsum cursus pellentesque quis eget ligula. Nulla facilisi. Etiam tincidunt felis id maximus interdum. Curabitur non neque non sapien rhoncus tristique.

--
 -- Table structure for table `sales`
 --

```
CREATE TABLE `sales` (
  `id` int(30) NOT NULL,
  `customer_name` text NOT NULL,
  `type` tinyint(4) NOT NULL DEFAULT 1 COMMENT '1 = walk-in, 2 = for delivery',
  `delivery_address` text NOT NULL,
  `amount` float NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0= Unpaid, 1=Paid',
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--
 -- Dumping data for table `sales`
 --

```

INSERT INTO `sales` (`id`, `customer_name`, `type`, `delivery_address`,
`amount`, `status`, `date_created`, `date_updated`) VALUES
(1, 'John Smith', 1, '', 360, 1, '2021-08-14 15:41:36', '2021-08-14
15:50:29'),
(2, 'Claire Blake', 2, 'Sample Address', 150, 1, '2021-08-14 15:51:44',
'2021-08-14 15:55:17');

```

```

--
-- Table structure for table `sales_items`
--

```

```

CREATE TABLE `sales_items` (
  `id` int(30) NOT NULL,
  `sales_id` int(30) NOT NULL,
  `jar_type_id` int(30) NOT NULL,
  `quantity` float NOT NULL,
  `price` float NOT NULL,
  `total_amount` float NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```

--
-- Dumping data for table `sales_items`
--

```

```

INSERT INTO `sales_items` (`id`, `sales_id`, `jar_type_id`, `quantity`,
`price`, `total_amount`) VALUES
(3, 1, 1, 10, 30, 300),
(4, 1, 2, 2, 30, 60),
(7, 2, 2, 5, 30, 150);

```

```

--
-- Table structure for table `system_info`
--

```

```

CREATE TABLE `system_info` (
  `id` int(30) NOT NULL,
  `meta_field` text NOT NULL,
  `meta_value` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

--

-- Dumping data for table `system_info`

```
INSERT INTO `system_info` (`id`,`meta_field`,`meta_value`) VALUES (1, 'name', 'Simple
Water Refilling Management System'),
(6, 'short_name', 'Water Refilling System - PHP'),
(11, 'logo', 'uploads/1628916900_water_refilling.png'),
(13, 'user_avatar', 'uploads/user_avatar.jpg'),
(14, 'cover', 'uploads/1626249540_dark-bg.jpg');
```

--

-- Table structure for table `users`

--

```
CREATE TABLE `users` (
  `id` int(50) NOT NULL,
  `firstname` varchar(250) NOT NULL,
  `lastname` varchar(250) NOT NULL,
  `username` text NOT NULL,
  `password` text NOT NULL,
  `avatar` text DEFAULT NULL,
  `last_login` datetime DEFAULT NULL,
  `type` tinyint(1) NOT NULL DEFAULT 0,
  `date_added` datetime NOT NULL DEFAULT current_timestamp(),
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--

-- Dumping data for table `users`

--

```
INSERT INTO `users` (`id`,`firstname`,`lastname`,`username`,`password`,
`avatar`,`last_login`,`type`,`date_added`,`date_updated`) VALUES
(1, 'Adminstrator', 'Admin', 'admin', '0192023a7bbd73250516f069df18b500', 'uploads/1624240500_avatar.png', NULL,
1, '2021-01-20 14:02:37', '2021-06-21
09:55:07');
```

--

-- Indexes for dumped tables

--

```
-- Indexes for table `jar_types`
--
ALTER TABLE `jar_types` ADD
    PRIMARY KEY (`id`);

--
-- Indexes for table `sales`
--
ALTER TABLE `sales`
    ADD PRIMARY KEY (`id`);

--
-- Indexes for table `sales_items`
--
ALTER TABLE `sales_items` ADD
    PRIMARY KEY (`id`),
    ADD KEY `sales_id` (`sales_id`);

--
-- Indexes for table `system_info`
--
ALTER TABLE `system_info` ADD
    PRIMARY KEY (`id`);

--
-- Indexes for table `users`
--
ALTER TABLE `users`
    ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `jar_types`
--
ALTER TABLE `jar_types`
    MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

--
-- AUTO_INCREMENT for table `sales`
--
ALTER TABLE `sales`
    MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `sales_items`
--
ALTER TABLE `sales_items`
    MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;

--
-- AUTO_INCREMENT for table `system_info`
--
ALTER TABLE `system_info`
    MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;
```

```
--  
-- AUTO_INCREMENT for table `users`  
--  
ALTER TABLE `users`  
    MODIFY `id` int(50) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;  
  
--  
-- Constraints for dumped tables  
--  
  
--  
-- Constraints for table `sales_items`  
--  
ALTER TABLE `sales_items`  
    ADD CONSTRAINT `sales_items_ibfk_1` FOREIGN KEY (`sales_id`) REFERENCES  
`sales` (`id`) ON DELETE CASCADE; COMMIT;
```


6. TESTING:

6.1 INTRODUCTION TO SYSTEM TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING:

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Organizations and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or one step up software applications at the company level interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing:

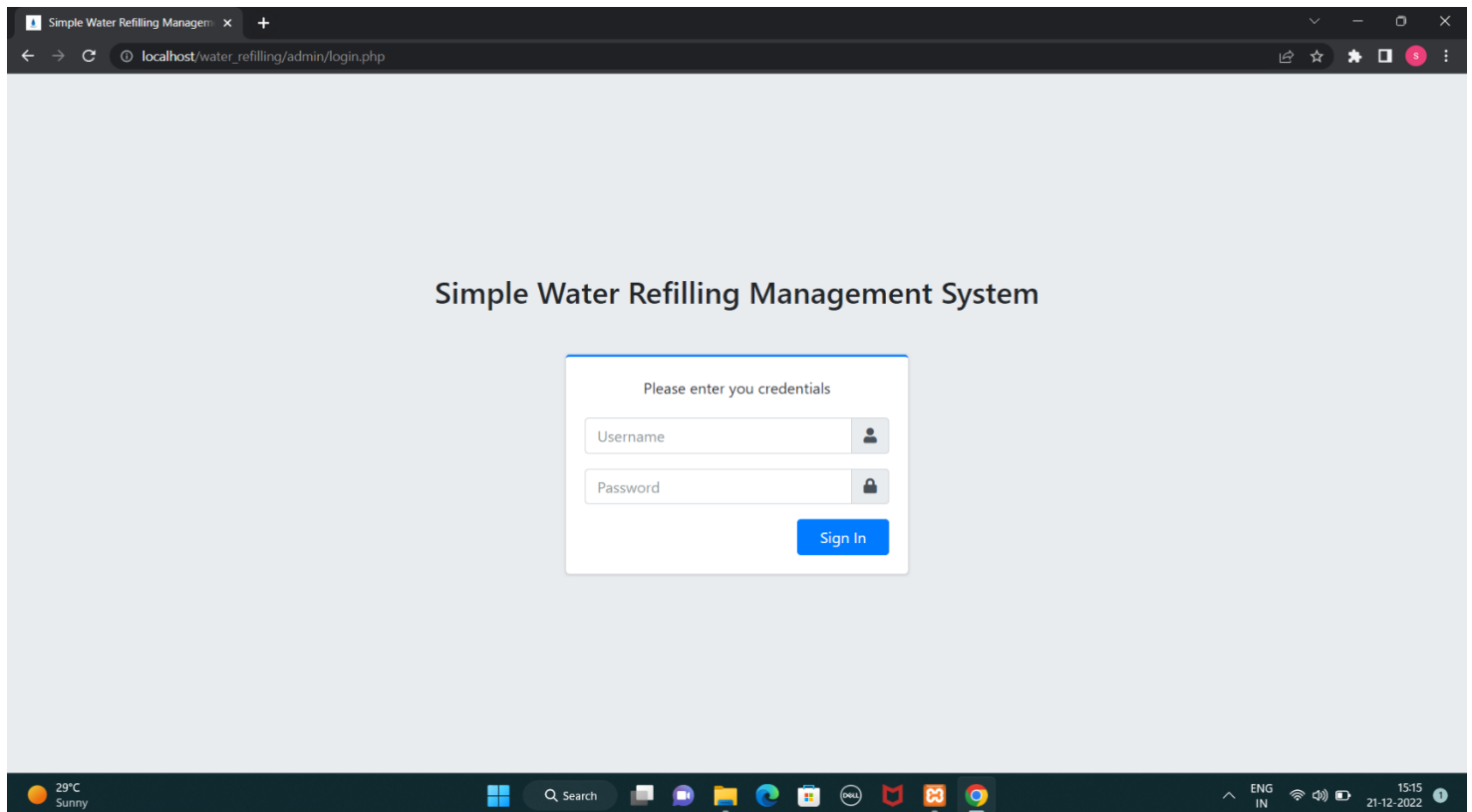
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

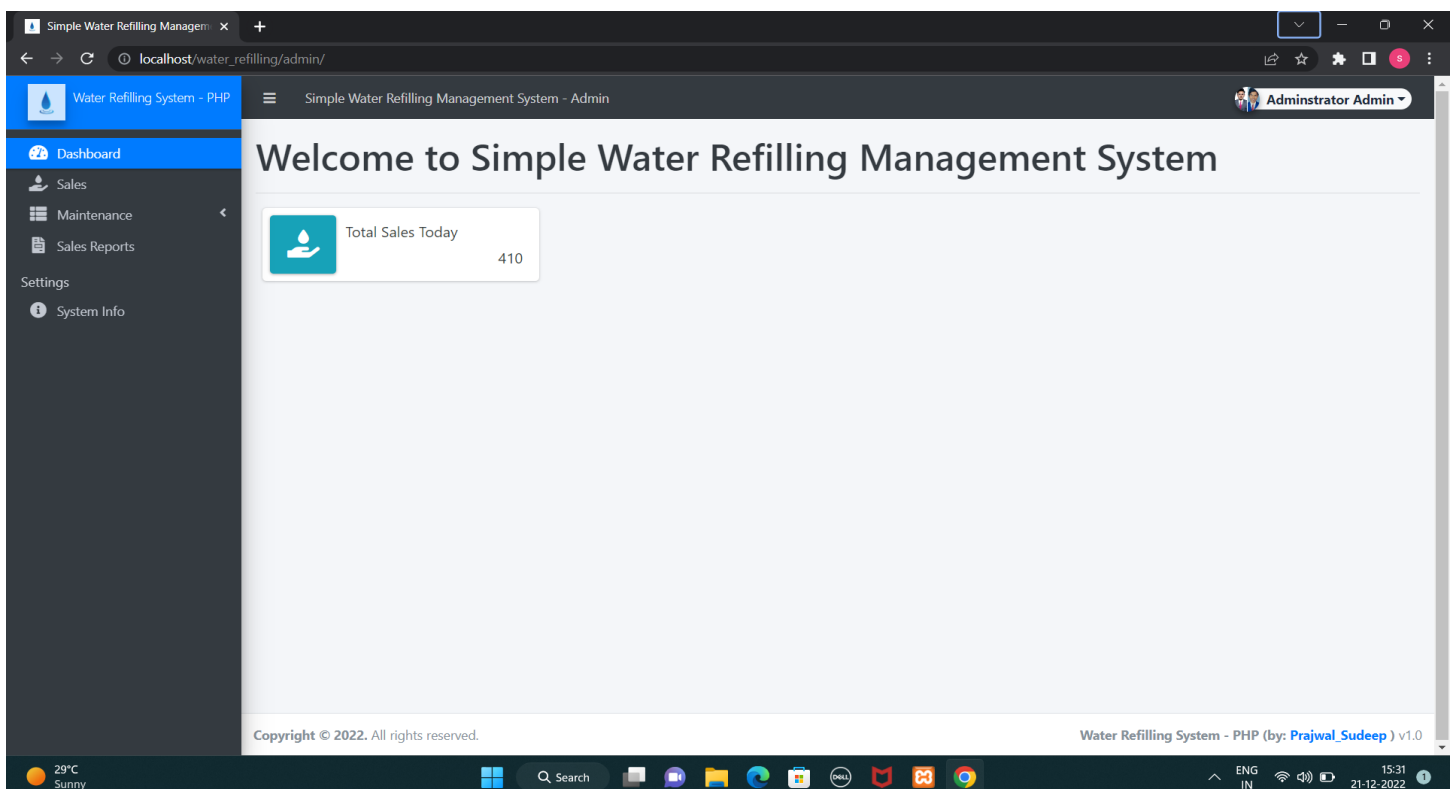
All the test cases mentioned above passed successfully. No defects encountered.

7. SAMPLE SCREENSHOTS :

➤ LOGIN PAGE :



➤ HOME PAGE :



WATER REFILLING MANAGEMENT SYSTEM

➤ SALES PAGE :

Water Refilling System - PHP

Simple Water Refilling Management System - Admin

Administrator Admin

Sales

+ Create New

Show 10 entries

Search:

#	Date Created	Customer	Type	Status	Total Items	Amount	Action
1	2022-12-21 15:28:36	Prajwal	for delivery	Paid	3	90.00	Action
2	2022-12-21 15:27:34	Siddesh	walk-in	Paid	2	80.00	Action
3	2022-12-21 15:27:00	Sudeep	for delivery	Unpaid	5	180.00	Action
4	2022-12-21 15:26:01	Aditya	walk-in	Unpaid	2	60.00	Action

Showing 1 to 4 of 4 entries

Previous 1 Next

Copyright © 2022. All rights reserved.

Water Refilling System - PHP (by: Prajwal Sudeep) v1.0

➤ MAINTENANCE PAGE :

Water Refilling System - PHP

Simple Water Refilling Management System - Admin

Administrator Admin

List of Jar Types & Pricing

+ Create New

Show 10 entries

Search:

#	Date Created	Name	Description	Price	Action
1	2021-08-14 14:32:18	Round Container with Cap	Adds some Curves to your Packaging with smooth glossy dome lids on these clear PET (Polyethylene terephthalate) Plastic Jars from Speciality Bottle	30	Action
2	2021-08-14 14:29:40	Slim Container with cap and faucet	Uniquely designed Container with Slim body and faucet (tap) design, you can get drinking water by gently pressing the faucet. The faucet has a rubber ring, so dont worry about leaking.	40	Action

Showing 1 to 2 of 2 entries

Previous 1 Next

Copyright © 2022. All rights reserved.

Water Refilling System - PHP (by: Prajwal Sudeep) v1.0

➤ SALES REPORTS PAGE :



8. CONCLUSION:

The main purpose of water refilling management system of overcoming difficulties at manual operation in refilling station is fulfilled. The difficulty in manual system is one of the reasons, why the efficiency in availing services of the clients is not satisfying and keeping of records is often misplaced and not secure. This Problem is resolved by this System.

This system is programmed to java that enable the user to record things that are being purchased by the clients, created using MySQL database.

This system manages to display the data to be filled by the user according to the information of the customer in organize manner, such that their personal details, and the services they want to avail as well as the payment on the transaction they purchased. The system keeps the information of the customer and the details of what they purchased.

The system coordinates the arrangement on delivery of products. It consists all the records for the location of the clients, date of transaction, schedule of delivery, contact number and the person assigned to deliver and the payment of customer to the quantity of product that about to be delivered.

The system also views the information about the availability of the products as well as the containers. The system views the available containers to provide stocks again.

We hereby conclude with the Guarantee that, Details of the users i.e., electronically entered in the "Water Refilling Management System" will be secured. Using this application we can retrieve user's history with a single click; thus, processing information will be faster. This System ensures accurate maintenance of user details and easily reduces the book keeping task, thereby reducing the human effort and increasing accuracy as well as speed.

BIBLIOGRAPHY:

➤ **For MySQL :**

<https://www.mysql.com/>

<http://www.mysqltutorial.org>

➤ **For XAMPP :**

[Download XAMPP \(apachefriends.org\)](http://www.apachefriends.org)

➤ **For PHP :**

<https://www.php.net/manual/en/index.php>

➤ **For HTML and CSS :**

<https://www.w3schools.com/html/>

<https://www.w3schools.com/css/>

► **REFERENCE PAPERS :**

• Domingo, M. D., Karelia, G. C., Maria Eugenia, T. R., Mayda, C. R., & Maureen, L. S. (2017, November). Estrategias antivectoriales con deltametrina en Santiago de Cuba para el control de Aedes aegypti (Diptera: Culicidae). In Cuba Salud 2018.

• Bandao, L. C., Gano, M. M., Babaran, Richard. P., Sagario, M. T. (2007). Don Bosco High School student record management system. (Unpublished undergraduate thesis). Saint Mary's University, Bayombong, Nueva Vizcaya, Philippines.

• Bayawon, D.D., Belington, K.G., Dama-On, L.M., Garcia, L.G. (2016). Information technology investment in relation to financial performance. (Unpublished undergraduate research paper). Saint Mary's University, Bayombong, Nueva Vizcaya, Philippines.

• Sánchez-Muros, M. J., Barroso, F. G., & Manzano-Agugliaro, F. (2014). Insect meal as renewable source of food for animal feeding: a review. Journal of Cleaner Production, 65, 16-27.

• Codd, E. F. (1990). The relational model for database management: version 2. Addison-Wesley Longman Publishing Co., Inc.

