KARNATAK UNIVERSITY DHARWAD

Janata Shikshana Samiti's

Shri Manjunatheshwara Institute of UG & PG Studies, Vidyagiri, Dharwad-580004.

A PROJECT REPORT ON

"FISH_TO_HOME"

BACHELOR OF COMPUTER APPLICATIONS (BCA)

OF

KARNATAK UNIVERSITY, DHARWAD

PROJECT GUIDED BY

Prof. UMESH AMBIGER

Submitted By

VASANT NAIK

BCA V SEMESTER

REG NO:20U10145

VIGNESHWAR NAIK

BCA V SEMESTER

REGNO:20U11167

DEPARTMENT OF COMPUTER SCIENCE 2022-2023

JANATA SHIKSHANA SAMITI'S

SHRI MANJUNATHESHWARA INSTITUTE OF UG & PG STUDIES, VIDYAGIRI,

DHARWAD - 580004

CERTIFICATE

This is to certify that Mr. Vasanat Naik & Mr.Vigneshwar Naik has satisfactorily completed Project Work entitled "Fish to Home" for the partial fulfillment of BCA prescribed by Karnataka University Dharwad during the academic year 2022-2023.

Prof. Umesh Ambiger	**Prof. Vivek M Laxmeshwar**	**Dr. Ajith Prasad**
Project Guide	[HOD] Computer Department	Principal

Examiners:

1)

2)

ACKNOWLEDGEMENT

The successful presentation of this project is acknowledgements of the immense support extended by JSS SMI UG & PG STUDIES, DHARWAD, which has provided us opportunity to fullfill the most cherished & desired way to reach our goal.

We would like to express our heartfelt thanks to our President Shri Shri Vishwaprasanna Theertha Swamiji of Sri Pejavarmath, Udupi, Parama Pujya Dr. D. Veerendra Heggade, the Chairman of Janata Shikshana Samiti & Dharmadhikari of Dharmasthala and the Secretary of Janata Shikshana Samiti Dr. Ajith Prasad.

We would like to express our sincere and hearty thanks of gratitude to our beloved Principal Dr. Ajith Prasad and our Head of the Computer Science Department Prof. Vivek Laxmeshwar who gave us the golden opportunity to do this wonderful project on the topic "Fish to Home", which also helped us in doing a lot of research and exposed to lot of new information that would help us in our mere future.

We would also take this opportunity to offer our sincere gratitude to our Project guide Prof. Umesh Ambiger for his excellent support throughout the development of this project and providing the necessary information on our demand at all the times.

Vigneshwar Naik & Vasanta Naik

DECLARATION

*We, **Vigneshwar Naik & Vasant Naik** students of **Fifth Semester BCA, Department of Computer Science, JSS SHREE MANJUNATHESHWARA INSTITUTE OF UG & PG STUDIES, VIDYAGIRI, DHARWAD** affiliated to Karnataka University, Dharwad here by declare that the Project entitled **"FISH TO HOME"** submitted in partial fulfilment of the course requirement for the award of degree in **Bachelor of Computer Application, Karnataka University, Dharwad** during the academic year 2022-2023. We have not submitted the matter embodied to any other university or institution for the award of any other degree.*

Date: Vigneshwar Naik

Place: Dharwad Vasanta Naik

CONTENTS

1 SYNOPSIS

INTRODUCTION

The Fish to Home is located at Dharwad. The agency will communicate with cuatomers regarding the product requirements periodically. Customers will send the required fishes order. Payment may in the form of cheque, cash or DD it is done manually. To overcome this we introduce a computerized manner for their maintenance by developing a Web Application.

the Fish to home serves you ready to cook seafood which is ‘As good as Live’ with all the goodness of nutrients stored in it, promoters of Daily Fish and one of the leading exporters of marine products from Dharwad

Objectives:

-Customers can order their favorite fish anywhere and anytime whenever they want.

i.e customers can save time shopping through online web application.

- User can get cart system
- This software helps in easy maintaining and

And updating daily fishes in the website for the administrator

- Orders delivered on the same day.
- Standardized packing system available.
- Two business gained after corona,delivered to restaurant and directly to customers.
- Set customer base with regular customers and good reputation

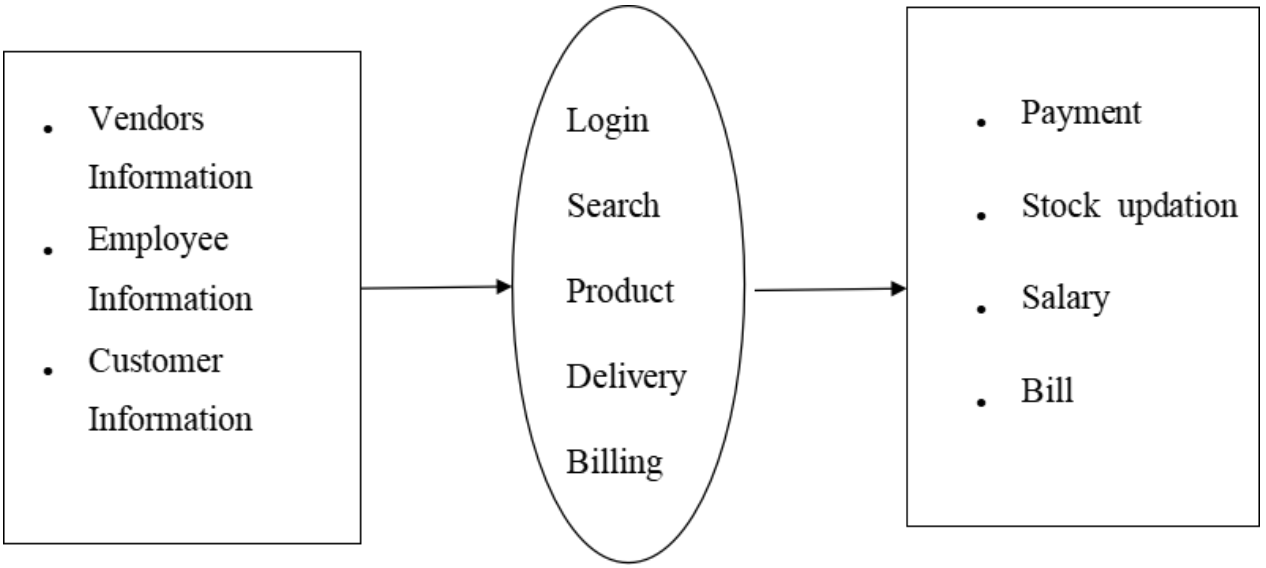
INPUT OF THE PROJECT:

- Vendors Information
- Employee Information
- Customer Information

OUTPUT OF THE PROJECT:

- Payment and stock updation
- Salary
- Bill

PROCESS LOGIC



Tools/Platforms, languages to be used:

Hardware requirements:

- Processor: Core i3 intel processor
- RAM : 2GB and above
- Hard Disk : 20GB
- Operating System: Windows 11

Software requirement:

- Front End: HTML, Css
- Middle ware: PHP
- Scripting: Java script
- Back End: MYSQL

Are you doing this Project for any industry/Client? If yes acceptance of the Project:

- No

Duration of the Project:

- 2 Months

Members of the Project:

Vigneshwar Naik.

Vasant Naik.

Limitations of the Project:

- Live tracking of order is not possible.
- Order quantity is limited.
- Location of ordered delivery is restricted.

SCOPE OF THE PROJECT:

- Development of Android and IOS application.
- GPS system can be connected.

2 FRAMEWORK

Introduction:

Implementation is the process of converting a new revised system design into operation. The objective is to put the new revised system ,which has been tested into operation while holding costs, risks and personal irritation to the minimum. A critical aspect of the implementation process is to ensure that there will be no description in the function of the organization.

Introduction to technologies used in this project:

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy and it is a process of having the systems personnel check out and put new products into use and construct any files of data needed to use it.

Why you need XAMP, MySql and PHP?

PHP is a powerful scripting language that can be run by itself in the command line of any computer with PHP installed. However PHP alone is not enough in order to build dynamic web sites. To use PHP on a web site, you need a server that can process PHP scripts. WAMP server allows developers to test PHP scripts locally. Additionally dynamic websites are dependent on stored information that can be accessed easily; this is the main difference between a dynamic site and a static HTML site. However PHP does not provide a simple, efficient way to store data. This is where a relational database management system like MySql comes into play.

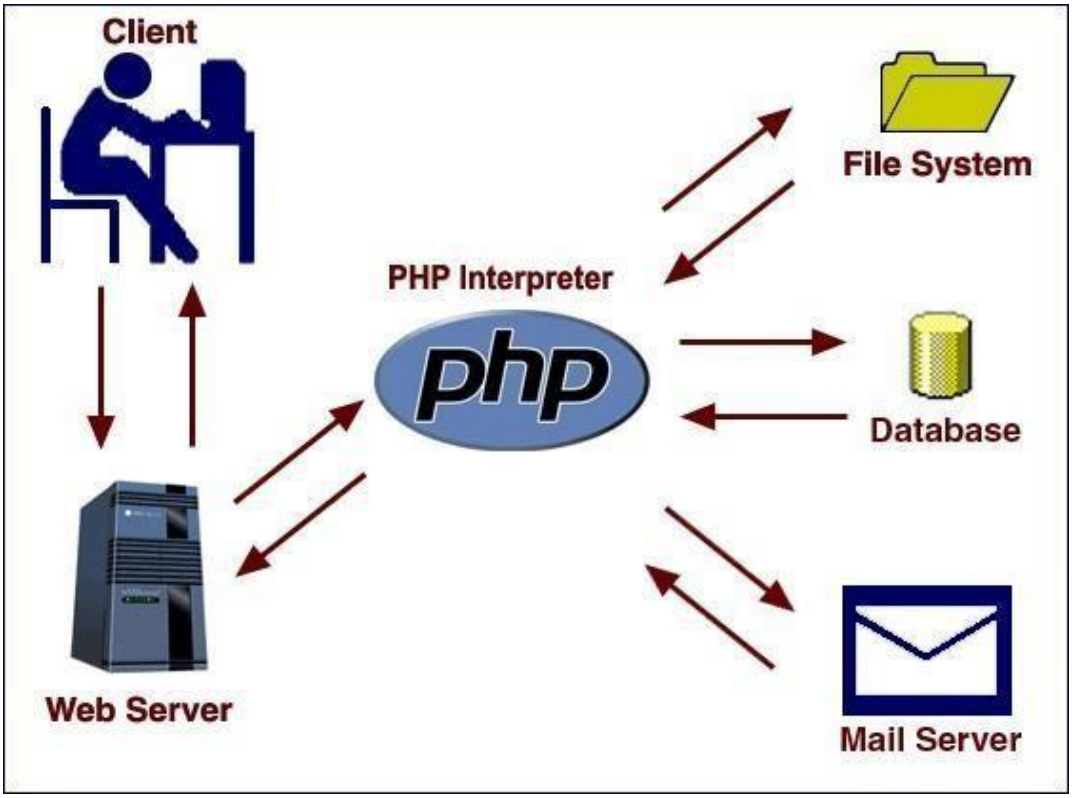
PHP:

PHP is a scripting language originally designed for producing dynamic webpages. It has evolved to include a command line interface capability and can be used in standalone graphical application. While PHP was originally created by RasmusLerdorf in 1995, the main implementation of PHP is now produced by the

PHP Groups and serves as the de facto standard for PHP as there is no formal specification.

PHP is a scripting language under the PHP License; however, it is incompatible with the GNU General Public License (GPL). Due to restrictions on the usage of the term PHP. It is widely used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as its input, I am creating web pages as out puts. It can be deployed on web servers and on almost every operating system and platform free of charge. PHP in installed on more the twenty million web sites and one million web servers.

PHP Architecture:



Usage:

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. It can automatically detect the language of the user. From PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor. Originally designed to create dynamic web pages, PHP's principal focus is server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's Active Server Pages, Sun Microsystems JavaServer Pages, and mod_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include CakePHP, Symfony, CodeIgniter, and Zend Framework, offering features similar to other web application frameworks.

The XAMP architecture has become popular in the web industry as a way of deploying web applications. PHP is commonly used as the PHP in this bundle alongside Linux, Apache and MySQL, although they may also refer to Python or Perl. As of April 2007, over 20 million Internet domains were hosted on servers with PHP installed, and PHP was recorded as the most popular Apache module. Significant websites are written in PHP including the user-facing portion of Facebook, Wikipedia (MediaWiki), Yahoo!, MyYearbook, Wordpress.

In addition to server-side scripting, PHP can be used to create stand-alone, compiled applications and libraries, it can be used for shell scripting.

Dreamweaver:

A website authoring program originally developed by Macromedia. It does many things including allowing designer to move back and forth between visual and HTML modes **HTML:**

HTML means Hypertext Markup Language. HTML is a method of describing the format of document, which allows them to be viewed on computer screen. Web browsers display HTML documents, program which can navigate across networks and

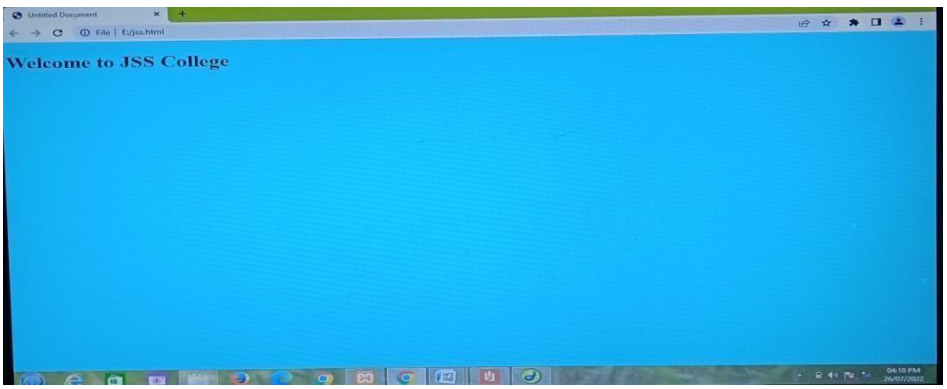
display a wide variety of types of information. HTML pages can be developed to be simple text or to be complex multimedia extra advantages containing, moving images, virtual reality, and java applets. The global publishing format of the Internet is HTML.

It allows authors to use not only text but also format that text with headings, list and tables, and also includes still images videos, and sound within text. Readers can access pages information from any where in the world at the click of mouse button information can be downloaded to readers own PC or workstations HTML pages can also be used for entering a data and as a front end for commercial transaction.

Example to display message using HTML tags

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Untitled Document</title>
</head>
<body>
<h1>Welcome to JSS College</h1>
</body>
</html>
```

Output:



MY SQL:

INTRODUCTION TO SQL SERVER

SQL Server is a Relational Database Management System (RDBMS) that runs exclusively under the Windows operating system. One benefit of using Windows exclusively is that you can send and receive E-mail messages based on SQL Server "events" and you can also let the operating system handle login security. The data base is an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

MySql is a multithreaded, multi-user SQL database management system (DBMS). The basic program runs as a server providing multi-user access to a number of databases. Originally financed in a similar fashion to the JBoss model, MySql was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB now a subsidiary of Sun Micro system , which holds the copyright to most of the codebase.

Database Evolution:

SQL was invented back in the 1960's by E.F. Cod of IBM. In order to increase data integrity and reduce repetitive data. RDBMS systems didn't appear until the late 70's when Sybase and Oracle introduced systems. These systems existed on mainframes at the time.

ANSI-SQL came to be in the 1980's. This was important because it meant that disparate systems could communicate through an agreed set of standards. There are different levels of ANSI-SQL compliance. Almost every major RDBMS today is entry level compliant, including SQL Server 2000. Every RDBMS has its own flavour

of SQL that complements ANSI-SQL with proprietary elements. SQL Server's flavour of SQL is known as Transact SQL (T-SQL).

SQL Server was originally a Sybase product. Microsoft bought the product outright from Sybase and by version 7.0, the version prior to 2000, all the code had been rewritten by Microsoft's programming gurus

Features of sql:

- It is simple English like language and uses simple commands such as SELECT, CREATE, DROP etc.
- It is not having condition loops, variables and most of the commands are single line commands.
- as PL/SQL (Procedural language of sql).
- One of the key features of sql server is the XML support. XML has Grown to be standard technology for organizations that share data on the web.
- Now with sql server 2000 XML documents can be retrieved directly from the database and it provides various ways to retrieve data in XML format.
- The entire SQL has been divided into 4 major categories.

    1. Data Manipulation Language.
    2. Data Definition Language.
    3. Transaction Control language.
    4. Data Control Language

Security:

View are basically used as a part of security, means in many organizations, the end user will never be given original tables & all data entry will be done with the help of views only. But the data base administrator will be able to see everything because all the operations done by the different users will come to the same table.

Queries:

A query is a question or a request. With MySql, we can query a database for specific information and have a record set returned. Create a connection to a database:

Before you can access data in a database, you must create a connection to the database. In PHP, this is done with the mysql_connect() function.

Syntax:

mysql_connect (server name, username, password);

Server name: Optional Specifies the Server to connect .Default values is localhost: 3306 **MySql Functions:**

What is a database? Quite simply, it's an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or SQL Server provides you with the software tools you need to organize that data in a flexible manner. It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected

contents. MySql is a multithreaded, multi-user SQL database management system (DBMS). The basic program runs as a server providing multi-user access to a number of databases. Originally financed in a similar fashion to the JBoss model, MySql was owned and sponsored by a single for-profit firm, the Swedish company MySQLAB now a subsidiary of Sun Micro system , which holds the copyright to most of the codebase.

The project's source code is available under terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySql is a database. The data in MySql is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows. Databases are useful when storing information categorically.

Create a connection to a database:

Before you can access data in a database, you must create a connection to the database.

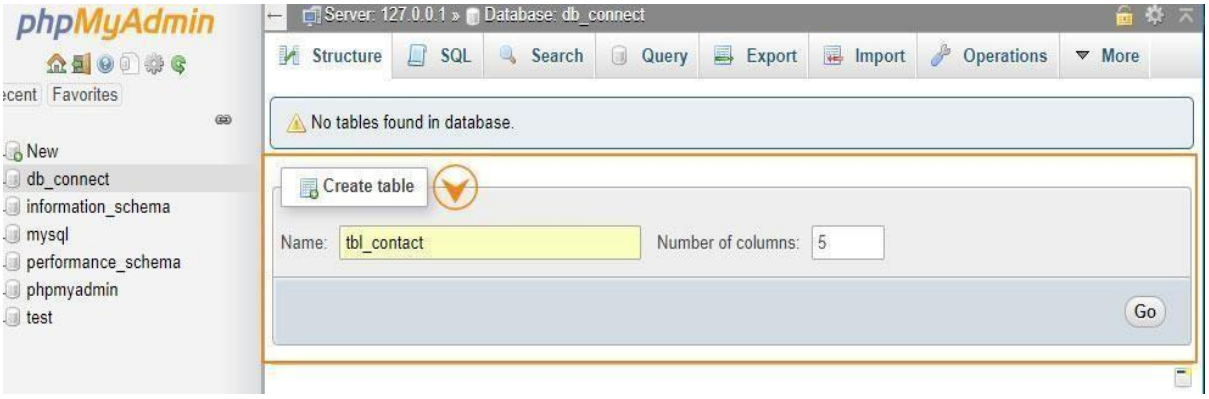
In PHP, this is done with the mysql_connect() function.

Syntax: Mysql connect (server name, username, password);

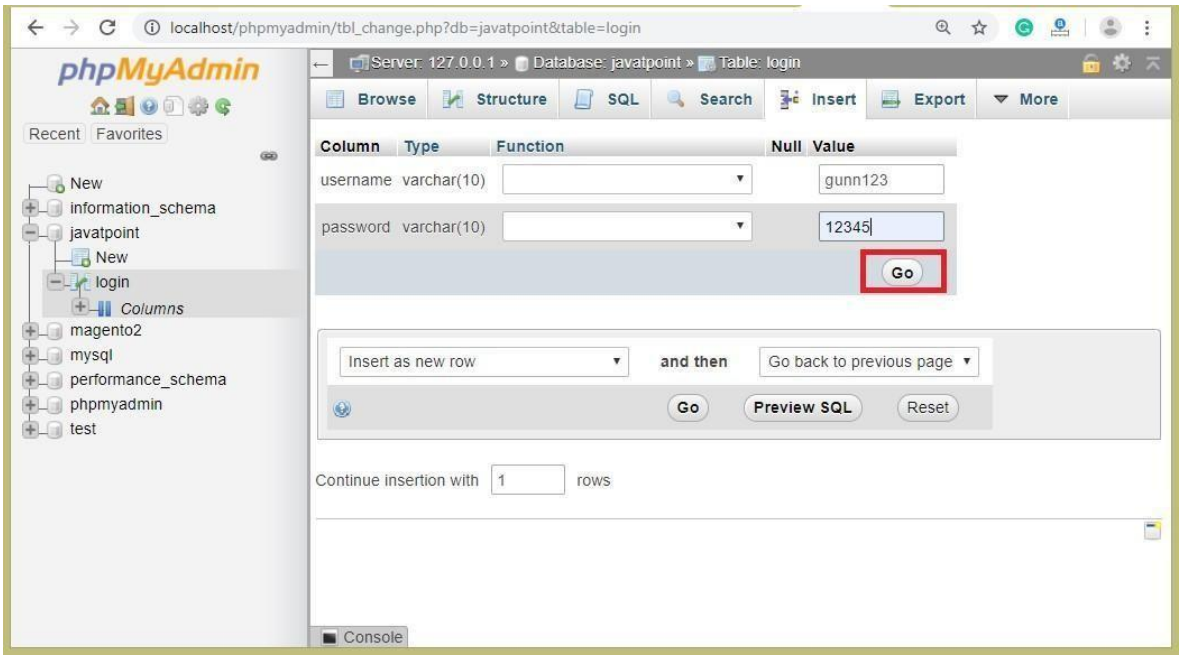
Server name: Optional Specifies the Server to connect .Default values is localhost: 3306

Steps to create a database in PHPMyAdmin:

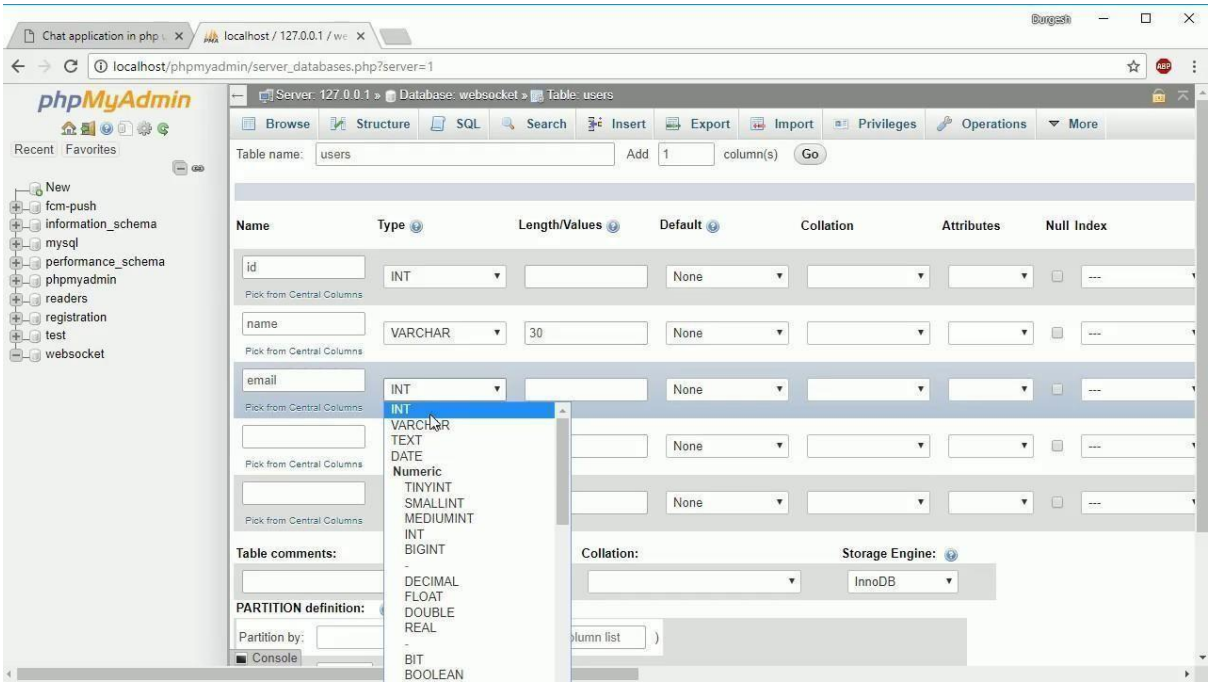
1) The following figure shows your PHPMyAdmin interface, just enter your database name and click the 'Create' button to create your database.



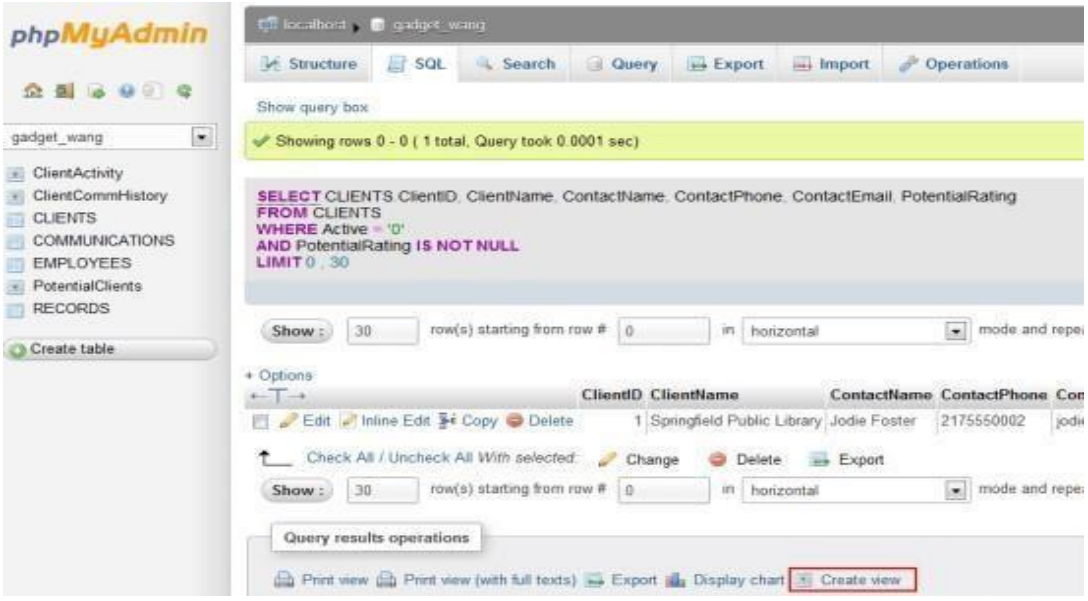
2) Now to create a new table enter your table name and the number of fields in the table, then click the 'Go' Button.



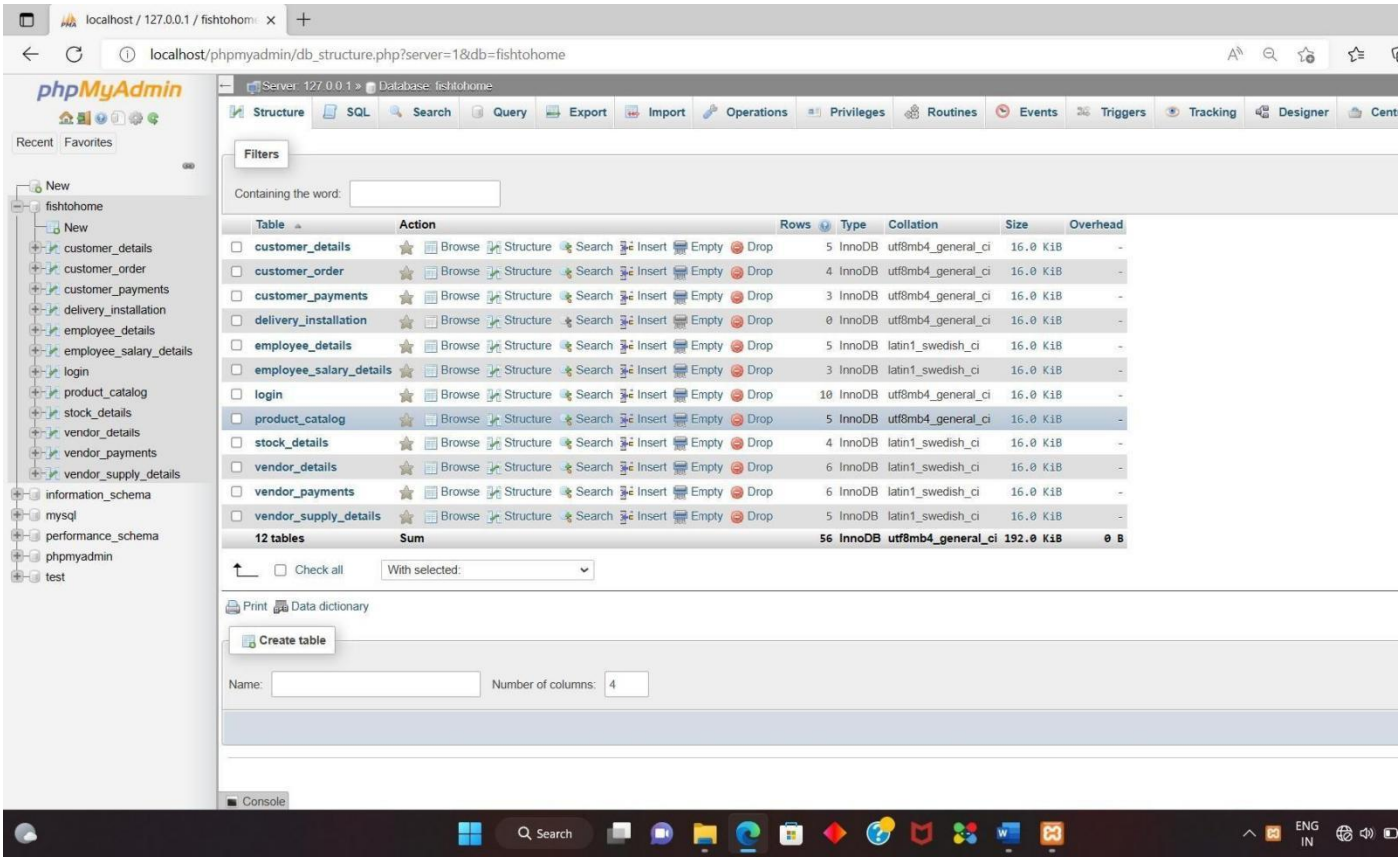
The next step is to create the fields, just enter values for each field name, type, length of the field, null option and mention whether it is a primary key or not. Then click the 'Save' button to complete your table creation.



3) The following figure is displayed upon successful creation of your table



The following figure to Drop Table in PHPMyAdmin:



3.PROJECT SUBJECT

The Fish to Home shop is located Dharwad. The agency will communicate with customers regarding the product requirement daily. Customers will send the requirements through the indent slip or customers may call and tell the order. Payment may in the form of cash or Cheque or DD it is done manually. To overcome this we introduce an computerized manner for their maintenance by developing an Web based Application.

The main aim of developing this application is to distribute the equipments and instruments. It provides the information of transactions and accounts etc. If any customers wants the information about the current transactions made in shop distribution of equipments which is not available. It would be better if all details are available in one location.

The proposed system includes following modules:

- Admin module
- Billing module
- Customer module

ADMIN MODULE:

- Login:

  In login page admin can login with his own username and password.

- Add Category:

  In the admin panel add category facility is provided. Where admin can add the category details

- Add and Remove Products:

  In the admin panel add and remove products facility is provided. Where admin can add or remove the product details or else he can edit the information.

- View Customer Order:

  In the admin panel admin can view the customer orders facility is provided. Where admin can get the details of the customer orders.

2. Customer Module:

- Registration:

In Registration page user can register with his own username and password.

Login:

After Registration user can get logged in using his username and password.

- View and order products:

In the customer panel customer can view the products and order facility is provided.

- View the offer details:

In the customer panel customer can view the offers on specific brands facility is provided.

- View the Bill :

In the customer panel customer can view the bill or the total amount to paid facility is provided.

3 . Billing Module:

- View the products:
  In the Billing panel admin can view the product details facility is provided.

- Generate the bill:
  In the Billing panel admin can generate the bill facility is provided.

4. Vendor module:

This module mainly stores the information about any supplier. Supplier stock, and give information about newly arrived products

4. SOFTWARE REQUIREMENTS SPECIFICATION

INTRODUCTION

SRS is the official statement of what is required by the system developers; it includes both user requirements for the system and detailed specification of the system requirements. This document is used while designing the proposed system and can also be used in the future if the system is to be enhanced.

Purpose

The purpose of this Requirements Elicitation document is to provide a clear understanding as to what actually the Industry Management Syatem is and to identify the critical requirements essential for the project's successful completion.

This document explains our team architecture, our teams's initial understanding of the user needs.

Document Conventions

Main Section Title: Font: Times New Roman, Bold: Size 16

Sub Section Title: Font: Times New Roman, Bold: Size 14

Other Text Matter: Font: Times New Roman, Bold: Size 12

Scope

This document is intended for providing an abstract overview of the system and general overview of the entire project. The scope of the document:

- Team Architecture,
- System Functional and Non-Functional Requirements
- Prototype of the System,
- Online payment acceptance for wide range of services.

Reference

- IEEE SRS Format
- Software Engineering 9th Edition, Ian Somerville.
- https://www.w3schools.com/

General Description

This section will give an overview of the whole application. The explanation of the application will be in its context to show the application interacts with other systems

and introduce the basic functionality of it. It will also describe all the constraints and assumptions for the application. **System Perspective**

As per the requirements of the client, to build a customized application that facilitates to maintain the entire details of the e-postal service to get through the demerits of existing manual system with new implementations such as:

- It takes very less time for completion of process.
- The application and requests can be verified by the officers wherever they are and can take necessary actions.
- It is very easy for giving complaints and suggestions.
- It is very easy for giving complaints and suggestions.

Existing System:

- In the present system the user has to do the transaction manually.
- It takes lot of time for transaction as they are maintaining the records in the book.

Proposed System

The proposed system is E-POST. Here everything is done online. The activities like depositing can be done through online. No waiting is required. And feedbacks can be given through online. Even if the officer is not available in the village, he can verify application online. It is very quick process than the existing system.

Overview:

- Design the proposed system
- Implementing the system
- Installing the software
- Testing the system for compliances

- Maintaining the system

Overall Description:

Product perspective

The fishes web application system is intended to replace the manual model of distribution record keeping by means paper records of distribution of fishes. It will be replaced with a single interaction between the distributor and distribution of fishes distributor will be able to view details of medicines on their pc and quickly maintain records. The features expressed in this Software Requirements Specification document are intended to be fully implemented. The system will be developed in such a way to provide easy addition of enhanced features, which may be desired in subsequent versions.

Product functions

- Provides statistical data regarding the daily updates of transactions.
- Authentication of the users.
- Updates the fishes data base daily or weekly.
- Adding new fishes , updating list.

User characteristics

- The user should be familiar with the transactions and security and their management system
- The user should know all the accounting details of the system.

Design and implementation constraints

- The working of fishes is through only Web connection

Specific Requirement

Functional Requirements

Functional requirements specify how this application supposes to work. The application provides the user with interactive data entry screens

- User must provide correct username and password.
- Admin is provided with User Id and Password to avoid unauthorized access
- Basic and advance admin facilities like add/update/delete Product details are provided.
- Manages details about the suppliers and customers.
- Manages product details.
- Back up/recovery of data such as product details.

NONFUNCTIONAL REQUIREMENTS

These are the requirements which are not concerned with the functionality of the system but they include the standards which are used in domain constraints.

Performance Requirement

The response time of the application should be less than 1 second. Response time to load information for billing should be less than 90 percentage of total time.

Safety Requirement

Application should not be disturbed by the hang up or overload of computer system and it should not harm to running processes.

Security Requirement

The files in which the information regarding day to day transactions and fishes and instruments details should be secured against malicious deformations.

Software Quality Attributes

- Interoperability: the system must be compatible with different windows operating system.
- Reliability user by providing user Id and password options: The system must

- accurately display whether the selected user information or not.
- Availability: The system is available 24 hours a day once it is installed.
- Usability: The system should be 99 percentage users interactive.

Hardware Specification:

- Processor: Core i3 intel processor
- RAM :2GB and above
- Hard Disk :20GB
- Operating System: Windows 7 and above

Software Specification:

- Front End: HTML, CSS
- Middle ware: PHP
- Scripting : Java script
- Back End: MYSQL

5. DESIGN OF THE SYSTEM

Introduction

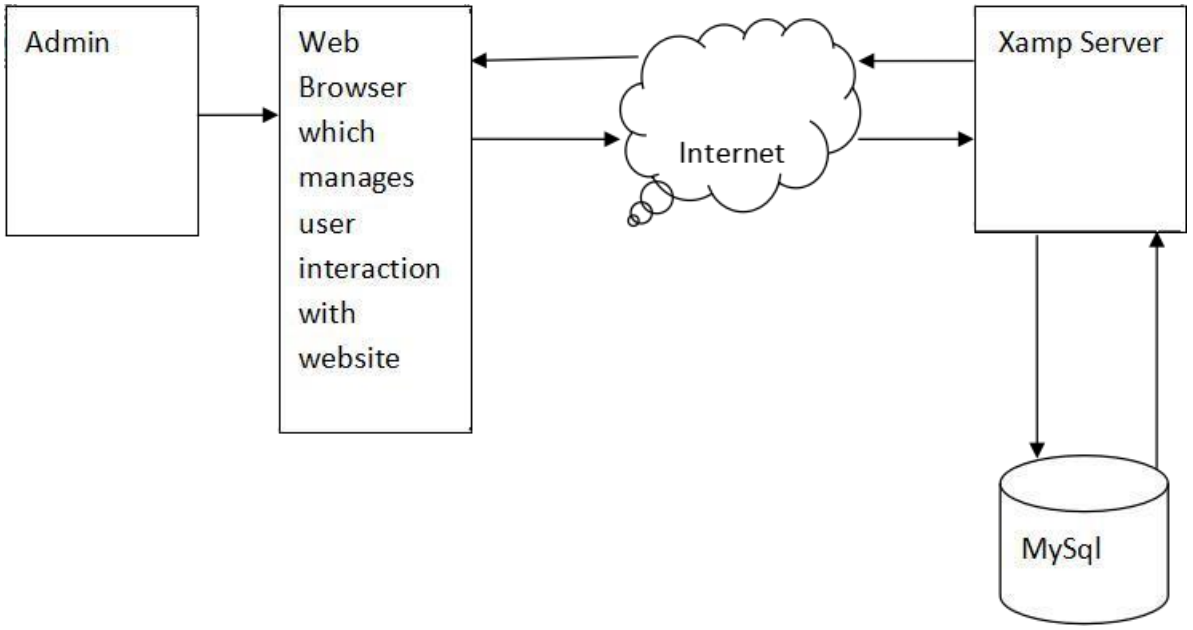
The purpose of the decision phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving the problem domain to the solution domain. It involves the process, in which conceiving, planning and carrying out the plan generating the necessary report, In other words, the design phase act as a bridge between SRS and implementation phase. The design of the system is perhaps the most critical factor affecting the quality of the software, and as a major impact on the later phase, particularly the testing and maintenance. Design is the key phase of any project. It is the first step in moving from the problem domain

to the solution domain. The input to the design phase is the specifications of the system to be designed.

The output of the top-level design is the architectural design, or the system design for the software system to be built. A design should be very clear, verifiable, complete, traceable, efficient and simple.

Architecture Design

The architecture design defines the relationship among major structural element of the program.Fishes diagram shows the relationship between different components of system. This diagram helps to understand the overall concept of system.



Logical design

The graphical representation of systems' data and how the process transforms the data is known as Data Flow Diagram. It shows the logical flow of the data.

The logical design describes the detailed specification for the system, describing its features, an effective communication and the user interface requirements. The logical system of proposed system should include the following.

1. External system structure.
2. Relationship between all the activities.
3. The physical construction and all the activities.
4. Global data.
5. Control flow.
6. Derived program structure.

Design Principles

Basic design principles that enable the software engineer to navigate the design process are:

- The design process should not suffer from “Tunnel vision”.
- The design should be traceable to analysis model.
- The design should not reinvent the wheel.
- The design should minimize the intellectual distance between the software and the problem, as it exists in the real world.
- The design should exhibit uniformity and integrity.
- The design should be structured to accommodate changes.
- The design not coding, the coding is not a design.
- The design should be assessed for the quality, as it is being created, not after the fast.
- The design should be reviewed to minimize the conceptual errors.

DATA FLOW DIAGRAM

The data flow diagram(DFD) is one of the important model tools. It shows the user of the data pictorially. DFD represents the flow of the data between different transformations and processes in the systems. The data flow diagram shows logical

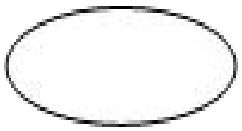
flow of the data. It represents the functional dependencies within a system. It shows output values in a computation or derived from input values. It is a simple pictorial representation or model for system behavior. It specifies, “what is to be done but not how is to be done”. It describes the logical structure of the system. It relates data information to various processes of the system. It follows top-down approach.

Data Flow Diagram Notations

Data Flow:

It may be from file-to-file or file-to-process or process-to-process .It is generally in terms of attributes. There may be either an input data flow or output data flow.

Functional processing:



The process is nothing but the transformation of data . It starts with the subject and has the verb followed by the subject.

Data store:

It includes file, data base and repository. To parallel lines represent it one end closed rectangle.

Actor/source/sink:

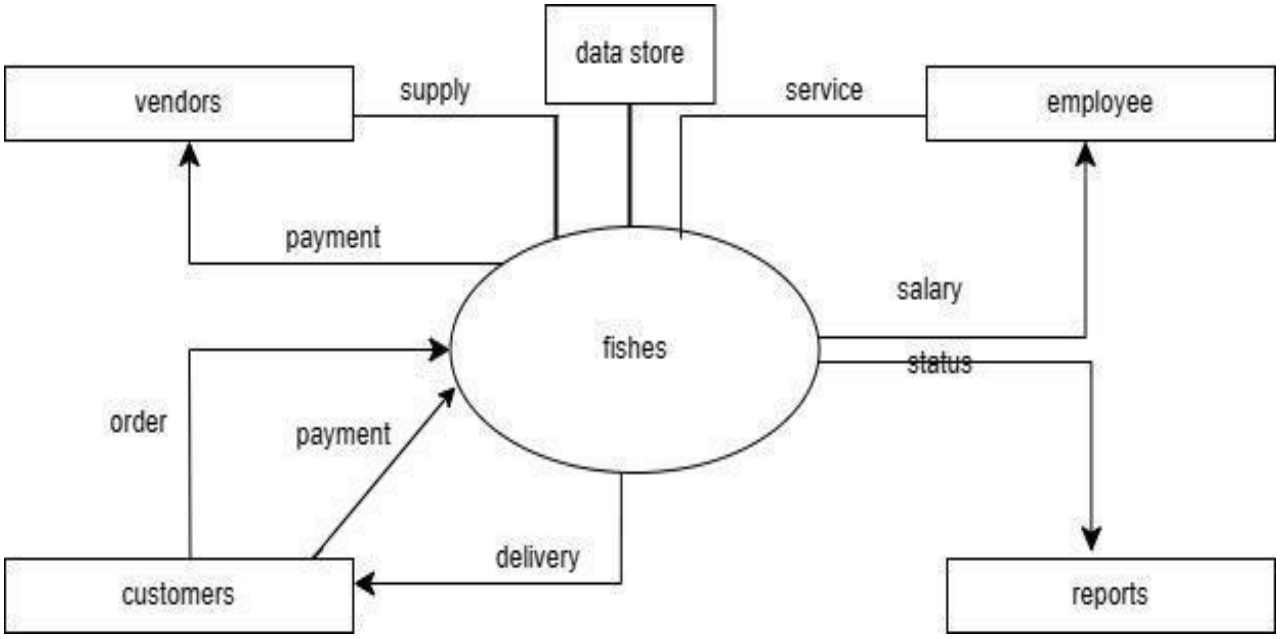
The files which are outside the system and used by the process or processes of the system. Generally Source/sink in the actor.

Objectives:

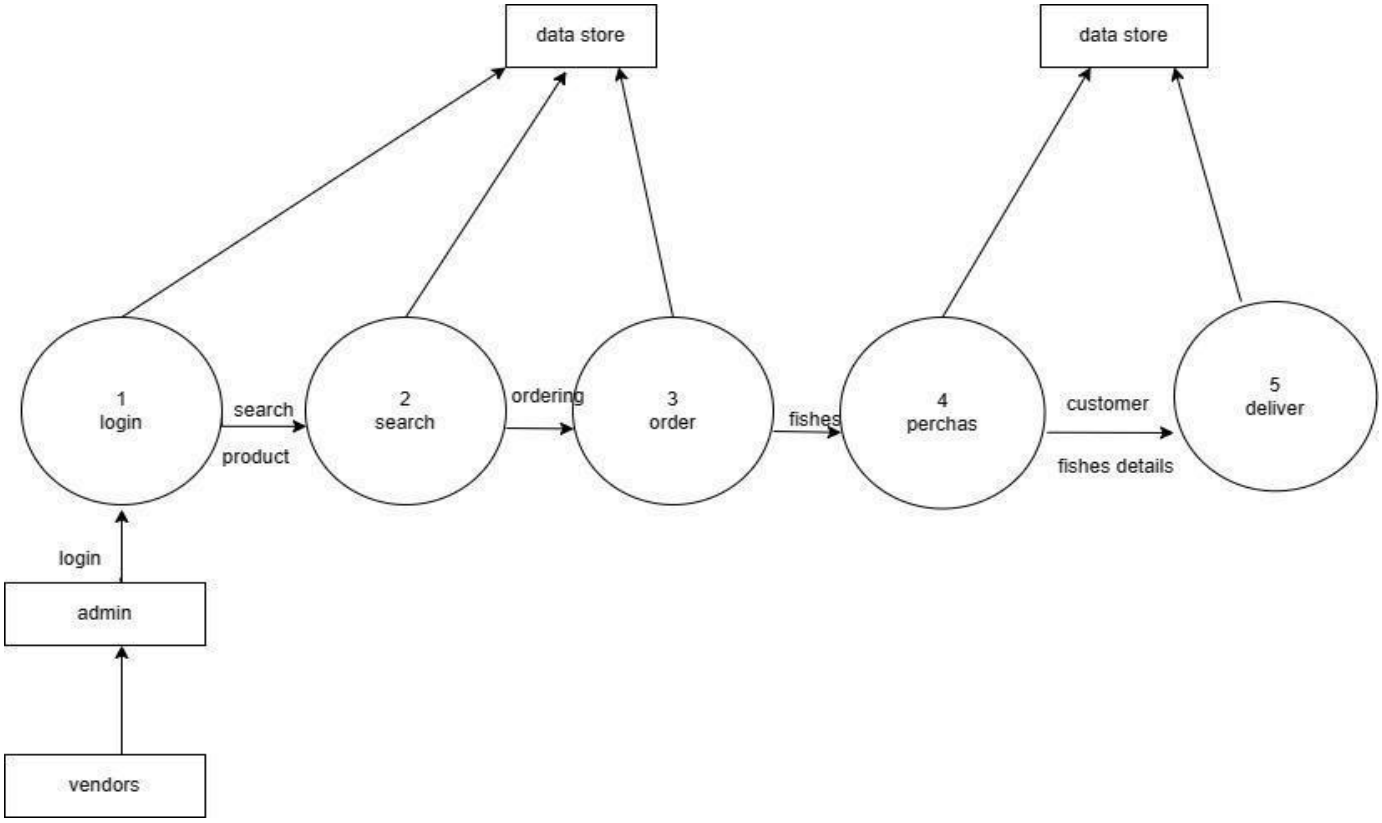
- To graphically document boundaries of a system.

- To provide hierarchy breakdown of the system.
- To show movement of information between a system and its environment.
- To document information flows within the system.
- To aid communication between users and developers.

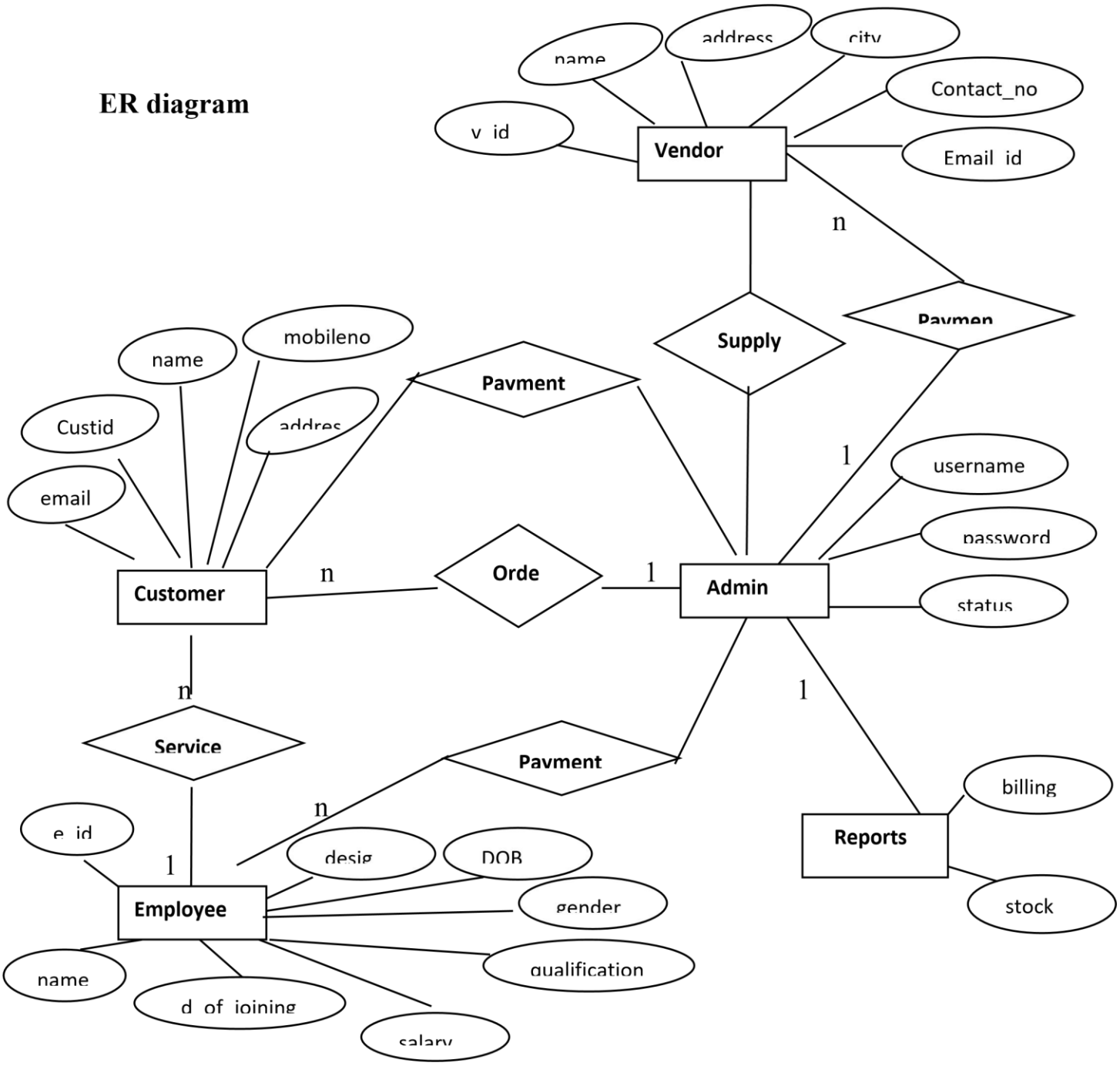
Context level diagram(zero level DFD):



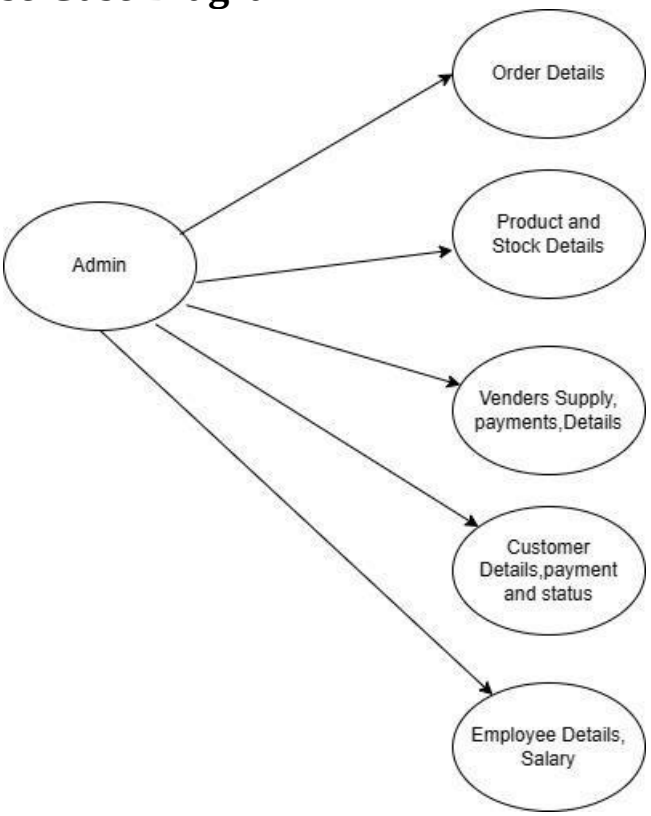
Context level diagram (First Level DFD):



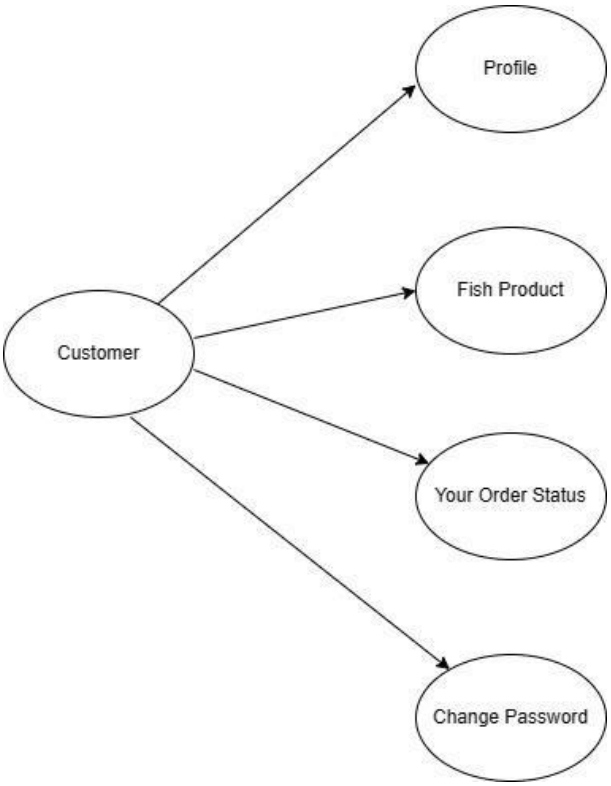
ER diagram


name
address
city
Contact_no
v_id
Vendor
Email id
n
Supply
Pavmen
mobileno
name
Pavment
Custid
addres
email
1
username
password
Customer
n
Orde
1
Admin
status
n
1
Service
Pavment
billing
n
e_id
desig
DOB
Reports
1
Employee
gender
stock
qualification
name
d_of_joining
salary

Use Case Diagram



Customer:



6. TABLES USED IN DATABASE

Table structure for CUSTOMER DETAILS table:

Primary Key: customer_id

Field	**Data Type**	**Length**	**Allow Null**	**Constraint**
Customer_id	Int	11	Not Null	Primary
customer_name	Varchar	100	Not Null	
Customer Address	Varchar	200	Not Null	
pincode	Varchar	100	Not Null	
city	Varchar	20	Not Null	
email_id	Varchar	100	Not Null	

Table structure for STOCK details table:

Primary Key:customer_billing_id

Field	**Data Type**	**Length**	**Allow Null**	**Constraint**
stock_id	Int	100	Not Null	Primary Key
bill_date	Date	50	Not Null	
r_id	Int	100	Not Null	Foreign Key

Table structure for CUSTOMER PAYMENT table:

Primary Key: customer_bill_master_id

Field	**Data Type**	**Length**	**Allow Null**	**Constraint**
Cust_pay_id	Int	100	Not Null	Primary Key
Cust_order _id	Int	100	Not Null	Foreign Key
Total_amount	Int	100	Not Null	
status	Int	100	Not Null	

date	Int	100	Not Null	
Transc_id	Int	100	Not Null	

Table structure for CUSTOMER ORDER table:

Primary Key: cust_oreder_id

Field	Data Type	Length	Allow Null	Constraint
Cust_order_id	Int	100	Not Null	Primary Key
customer_id	Int	100	Not Null	Foreign Key
Product_id	Int	100	Not Null	Foreign Key
quantity	Varchar	60	Not Null	
rate	Varchar	12	Not Null	
Order_status	Varchar	100	Not Null	
Order_date	Int	60	Not Null	

Table structure for DELIVERY INSTALLATION table:

Primary Key: Delivery_installation_id

Field	Data Type	Length	Allow Null	Comnstraint
Delivery_installatiom_id	Int	11	Not Null	Primary Key
Cust_order_id	Int	11	Not Null	Foreign Key
Delivery description	Int	11	Not Null	Foreign Key
Delivery status	Int	11	Not Null	
delivery_date	Date		Not Null	
Status	Varchar	11	Not Null	

Table structure for EMPLOYEE DETAILS table:

Primary Key: employee_id

Field	Data Type	Length	Allow Null	Constraint
employee_id	Int	111	Not Null	Primary Key
employee_name	Varchar	60	Not Null	
Designation	Varchar	100	Not Null	
date_of_birth	Date		Not Null	
Gender	Varchar	10	Not Null	
Qualification	Varchar	100	Not Null	
date_of_joining	Date			
basic_salary	Int	11	Not Null	

Table structure for EMPLOYEE SALARY DETAILS table:

Primary Key: employee_salary_id

Field	Data Type	Length	Allow Null	Constraint
Employee_salary_id	Int	11	Not Null	Primary Key
employee_id	Int	11	Not Null	Foreign Key
da_ta	Float		Not Null	
Pf	Float		Not Null	
Year	Year	11	Not Null	
Month	Int	20	Not Null	
given_date	Date		Not Null	

Table structure for LOGIN table:

Field	Data Type	Length	Allow Null	Constraint
Username	Varchar	200	Not Null	
Password	Varchar	200	Not Null	
user_type	Varchar	50	Not Null	
Hint_q	Varchar	500	Not Null	
Hint_a	Varchar	200	Not Null	
Status	Varchar	50	Not Null	

Table structure for PRODUCT CATALOG table:

Primary Key: product_id

Field	Data Type	Length	Allow Null	Default
product _id	Int	11	Not Null	Constraint
product_name	Varchar	60	Not Null	
product_desc	Varchar	60	Not Null	
product_type	Varchar	11	Not Null	
Product_price	Int	11	Not Null	

Table structure for STOCK DETAILS table:

Primary Key: stock_id

Field	Data Type	Length	Aloow Null	Constraint
stock_id	Int	11	Not Null	Primary Key
product_id	Int	11	Not Null	Foreign Key
Stock	Int	60	Not Null	

Table structure for VENDORS DETAILS table:

Primary Key: vendor_id

Field	Data Type	Length	Allow Null	Constraint
vendor_id	Int	11	Not Null	Primary Key
vendor_name	Varchar	60	Not Null	
vendor_city	Varchar	100	Not Null	
vendor_address	Varchar	200	Not Null	
contact_no	bigInt	20	Not Null	
email_id	Varchar	100	Not Null	

Table structure for VENDORS SUPPLY DETAILS table:

Primary Key: vendor_supply_id

Field	Data Type	Length	Null	Constraint
vendor_supply_id	Int	11	Not Null	Primary Key
vendor_id	Int	11	Not Null	Foreign Key

Product_id	Int	11	Not Null	Foreign Key
Quantity	Int	11	Not Null	
supply_id	Date	11	Not Null	
Status	Varchar		Not Null	

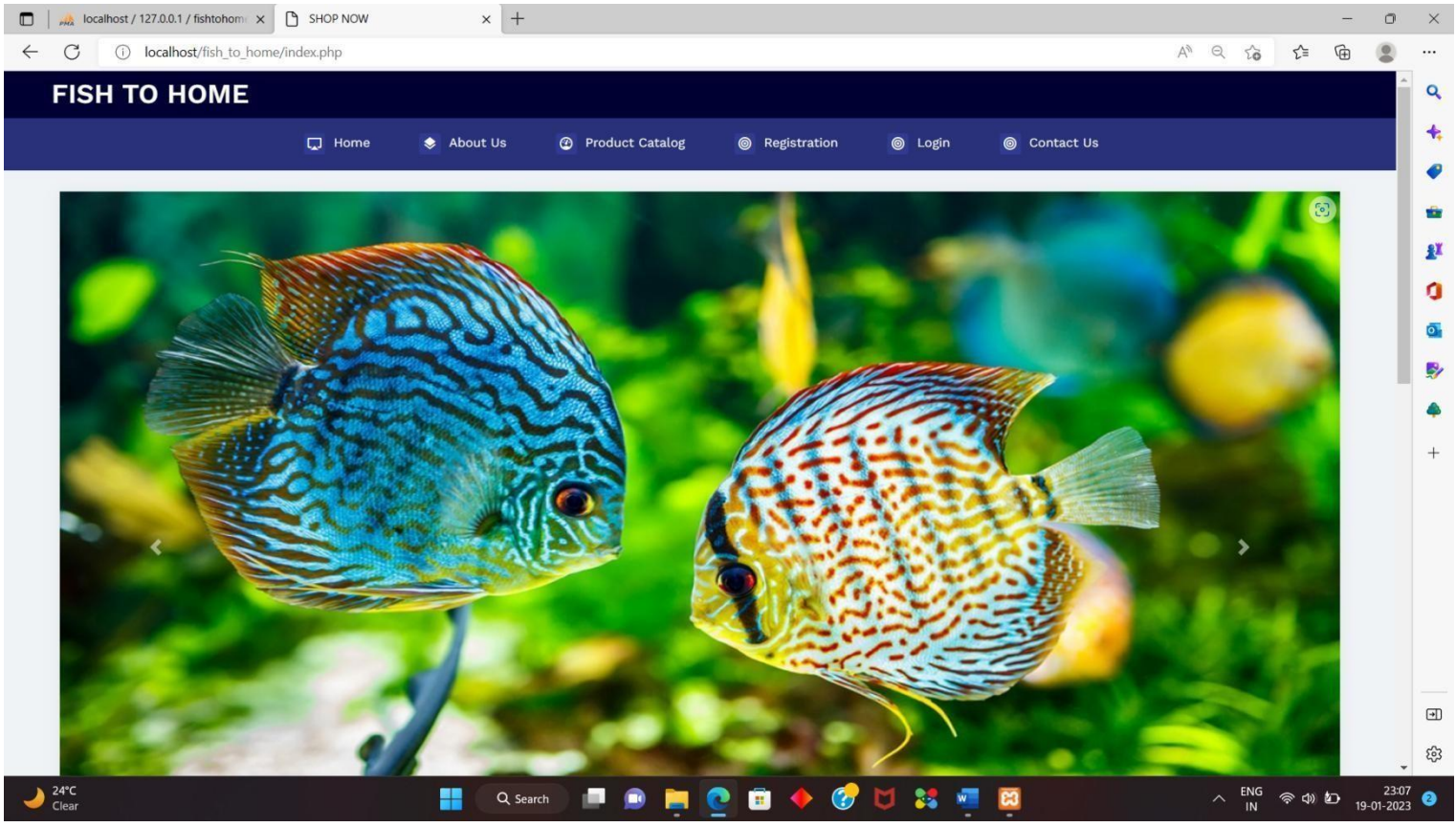
Table structure for VENDORS PAYMENTS table:

Primary Key: vendor_payment_id

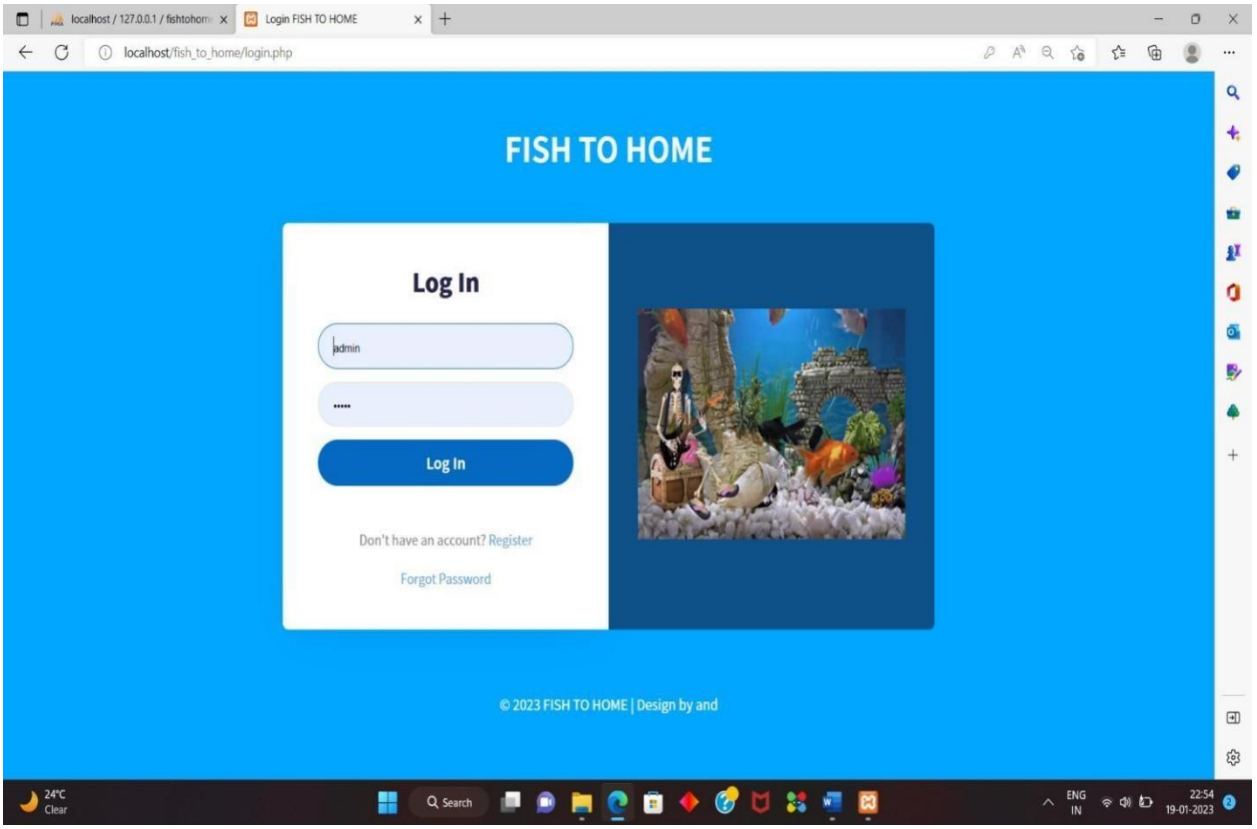
Field	**Data Type**	**Length**	**Allow Null**	**Default**
vendor_payment_id	Int	11	Not Null	Primary Key
vendor_id	Int	11	Not Null	Foreign Key
payment_no	Int	11	Not Null	
Amount	Int	11	Not Null	
mode_of_payment	Varchar	20	Not Null	
bank_name	Varchar	100	Not Null	
cheque_dd_no	Int	60	Not Null	
given_date	Date		Not Null	

7 SCREEN SHOTS

A VIEW OF HOMEPAGE

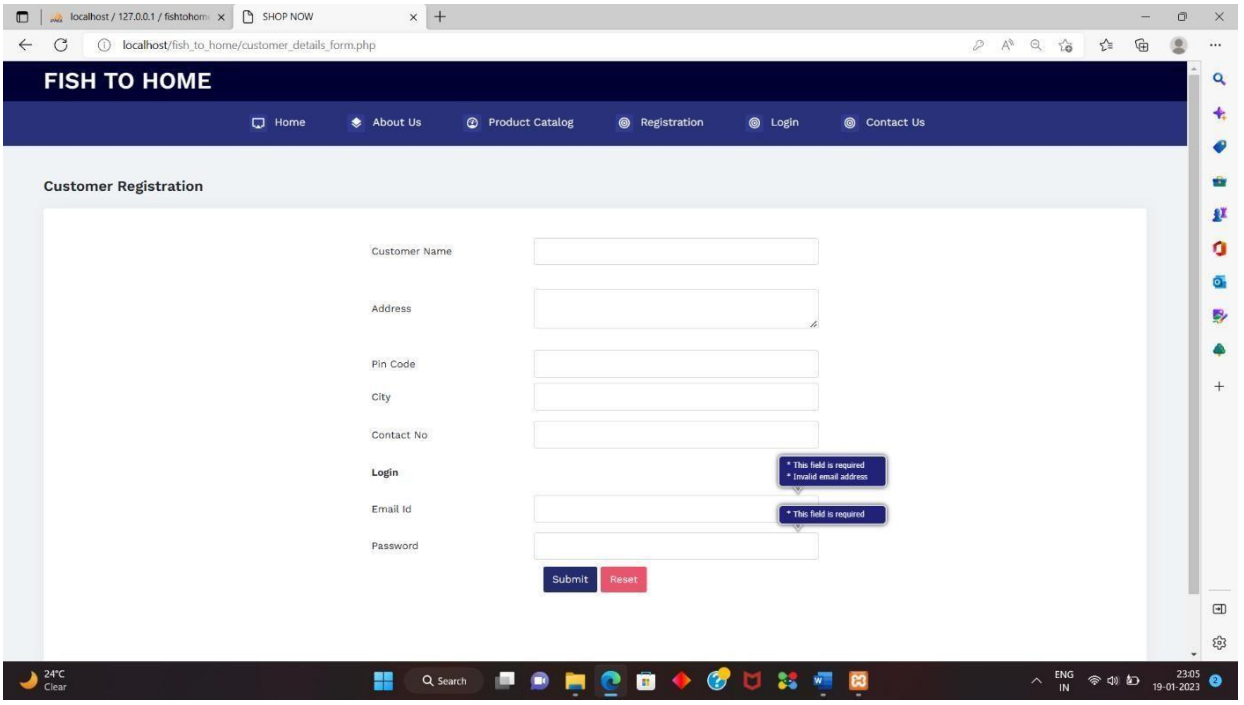


A VIEW OF LOGIN PAGE

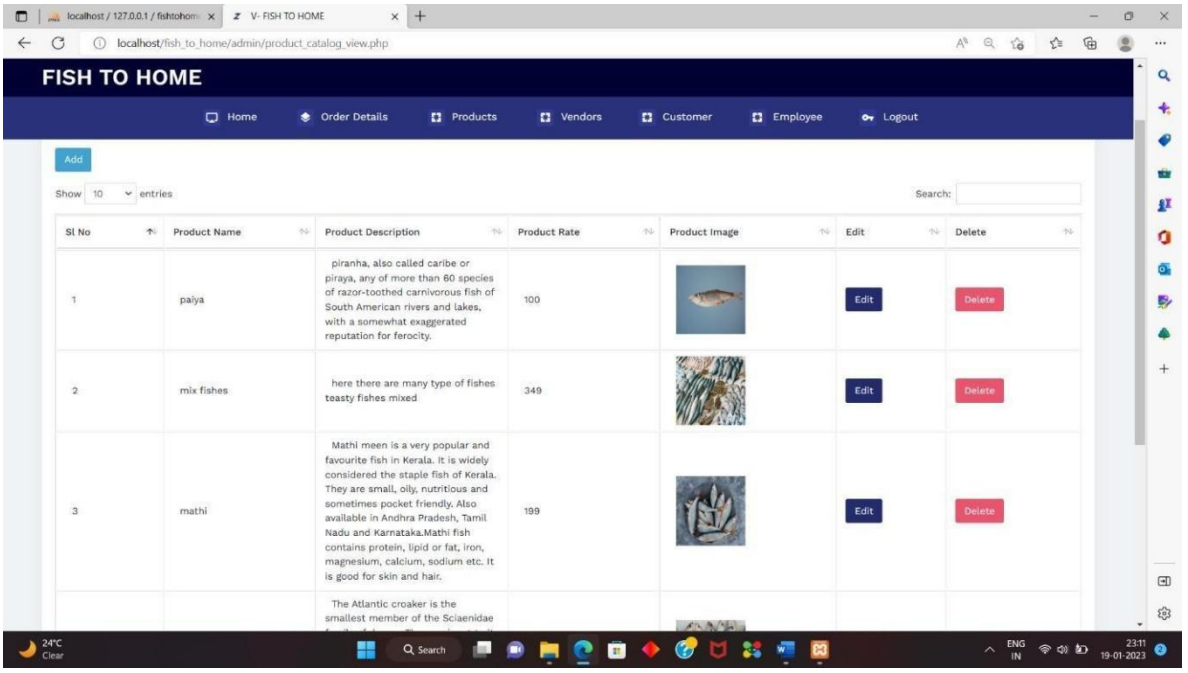


A VIEW OF

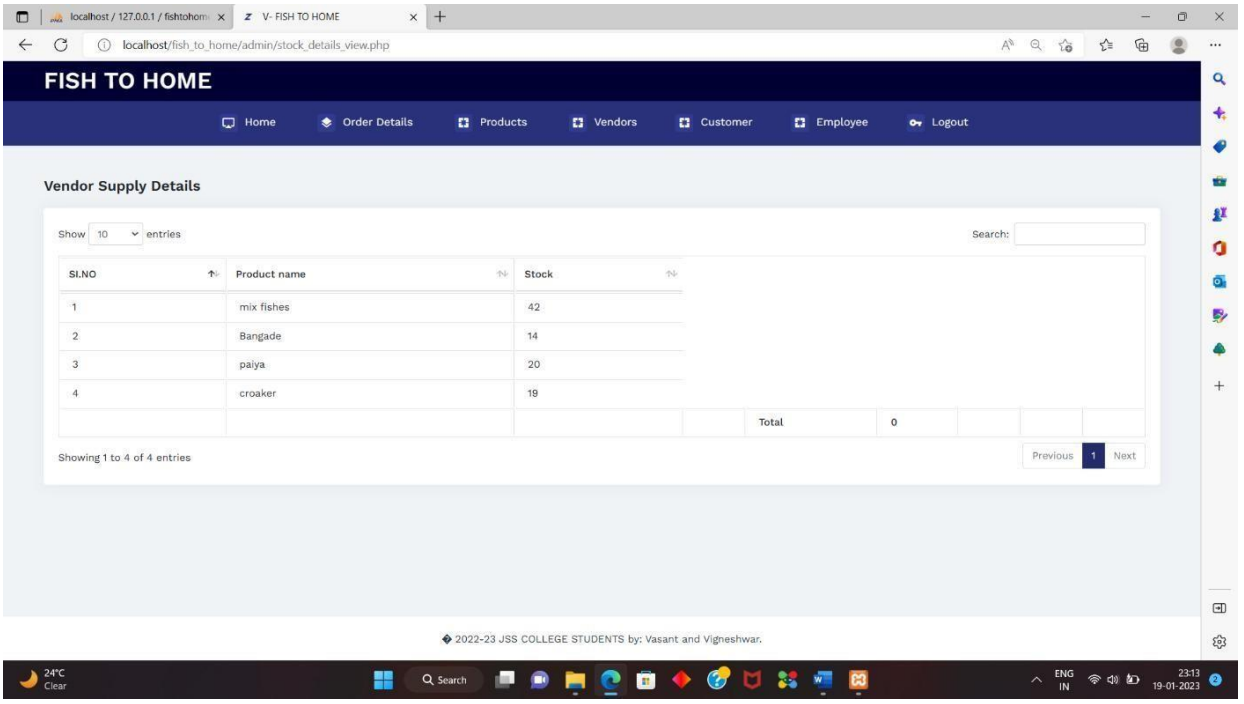
CUSTOMER REGISTRATION PAGE



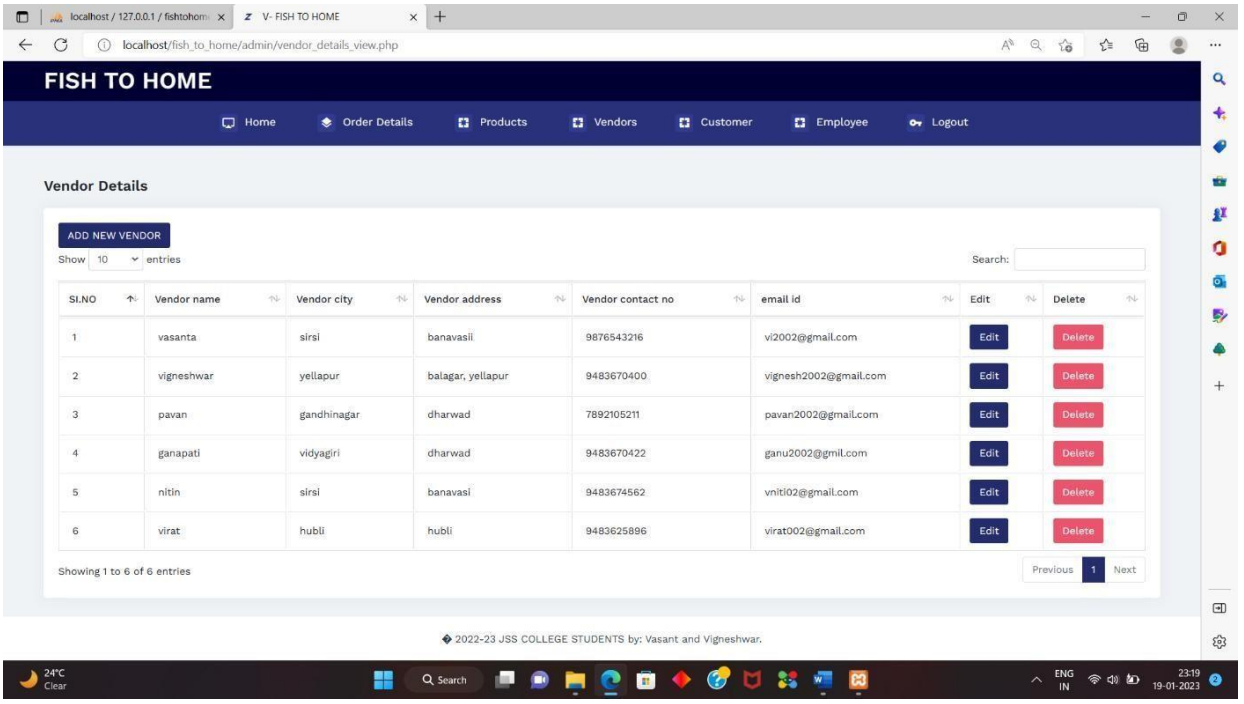
A VIEW OF PRODUCT DETAILS TABLE



A VIEW OF PRODUCT STOCK TABLE

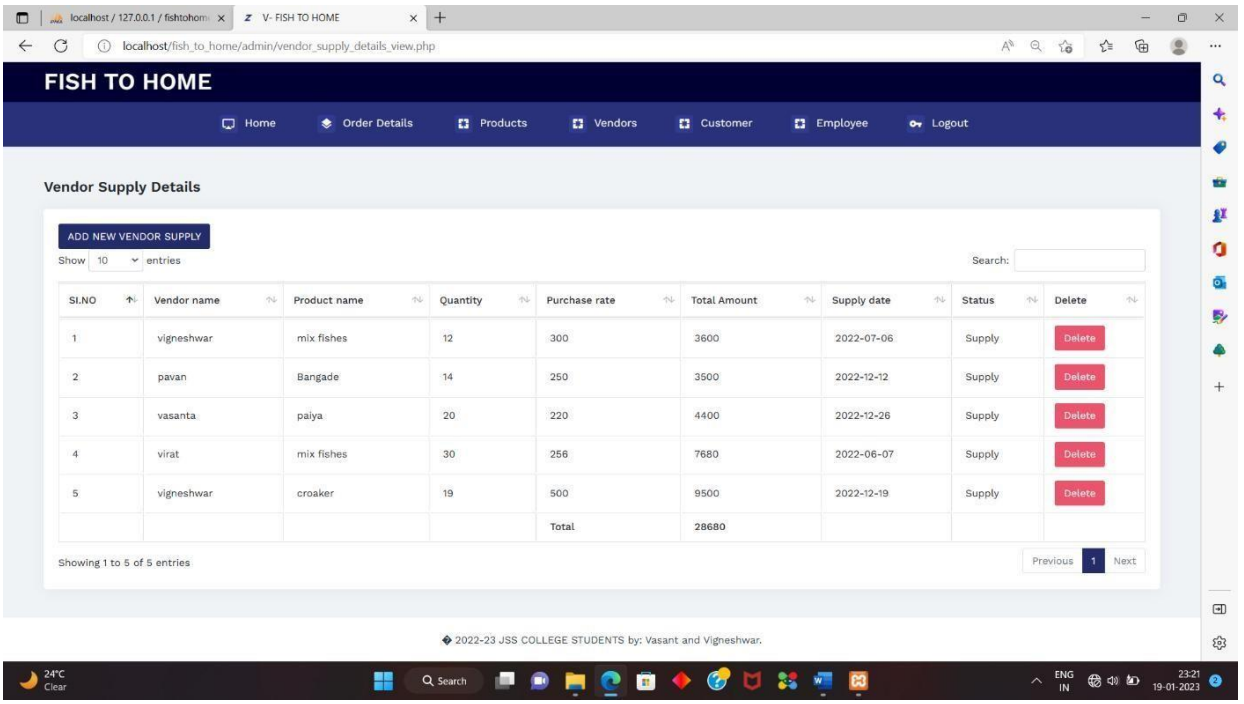


A VIEW OF VENDOR DETAILS TABLE

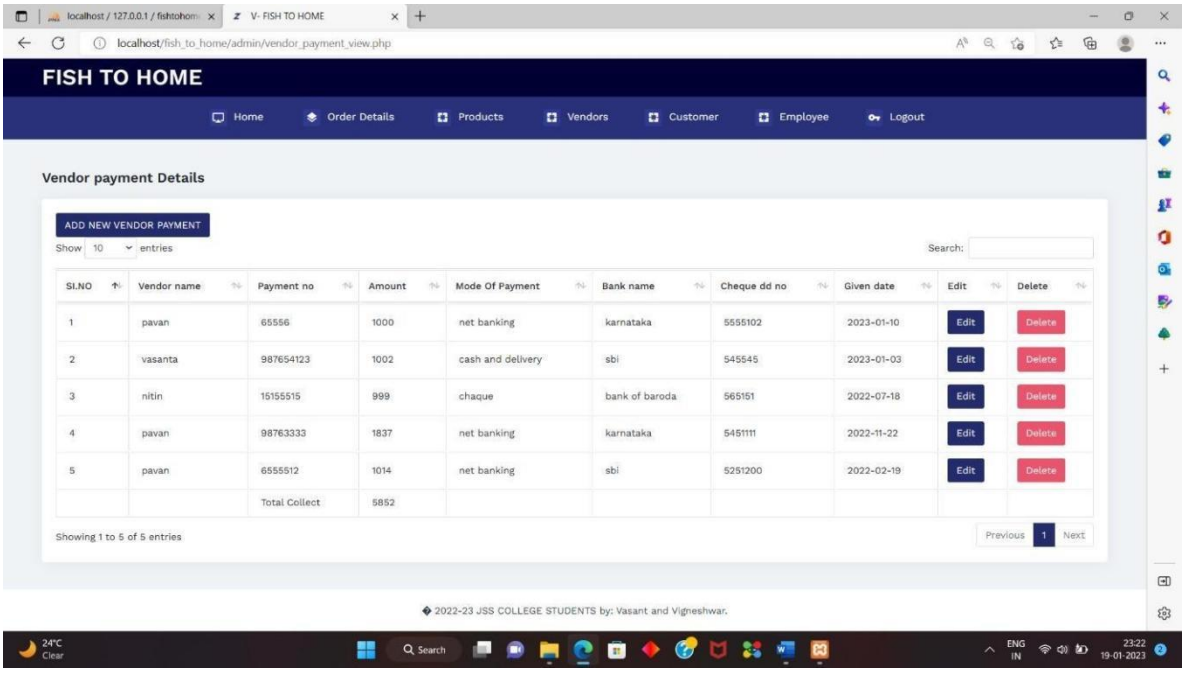


VENDOR SUPPLY DETAILS TABLE

A VIEW OF

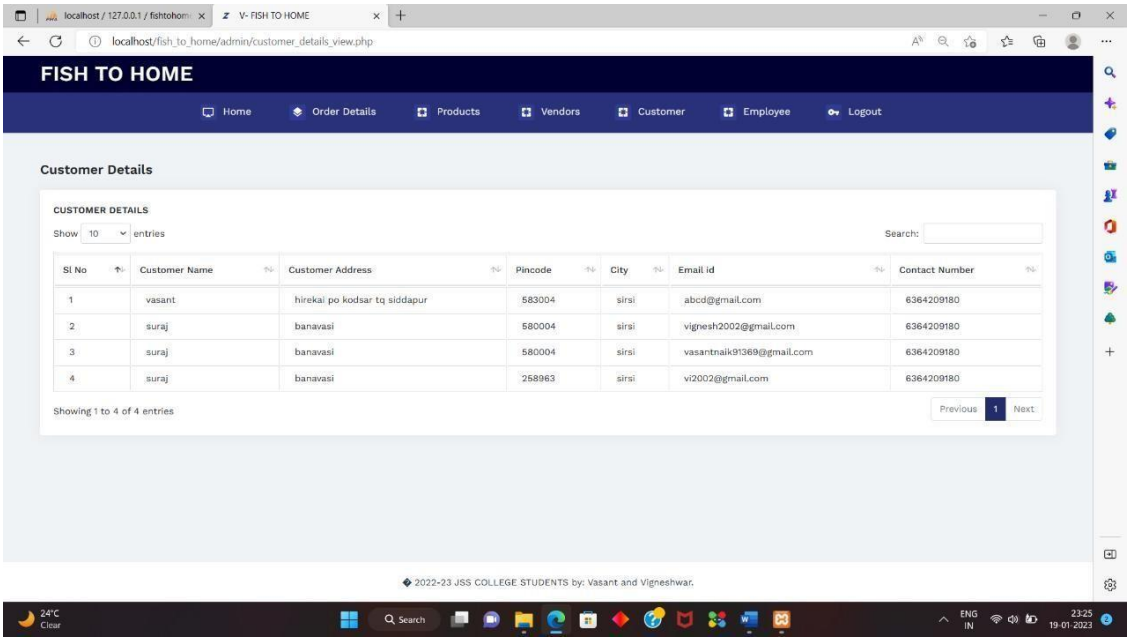


A VIEW OF VENDOR PAYMENT DETAILS TABLE



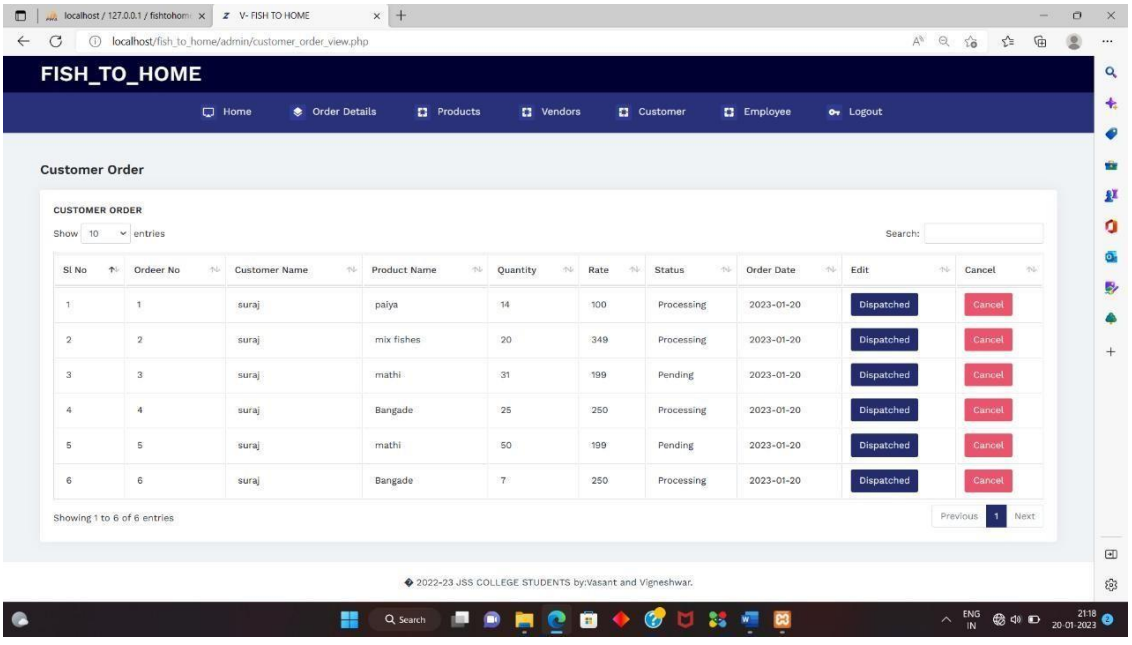
A VIEW OF

CUSTOMER DETAILS FOR ADMIN



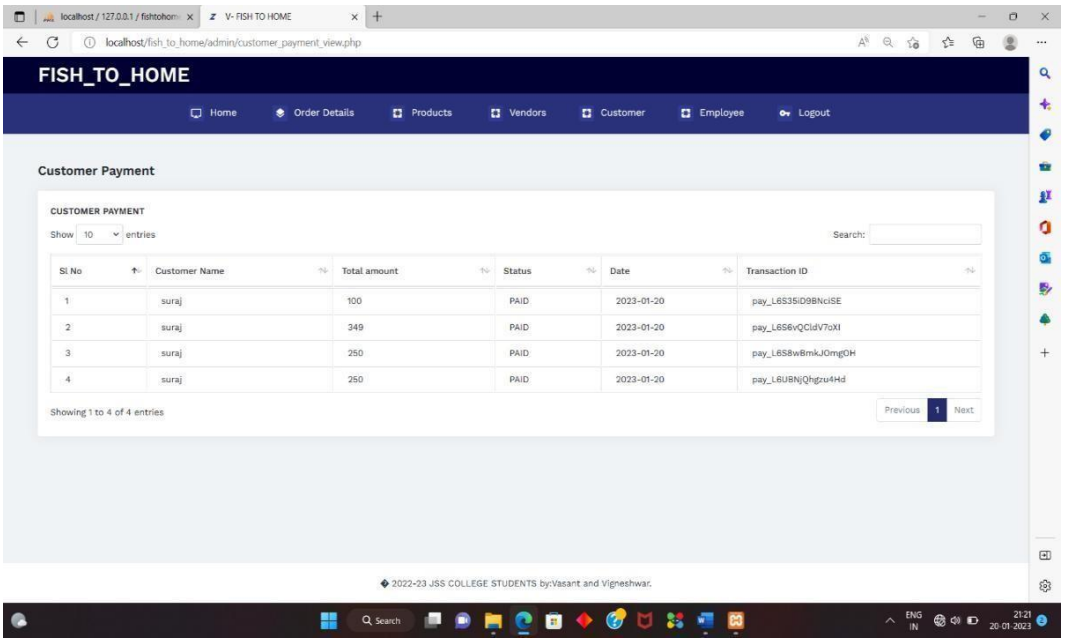
A VIEW OF CUSTOMER ORDER DETAILS FOR ADMIN

CUSTOMER BILLING FOR ADMIN

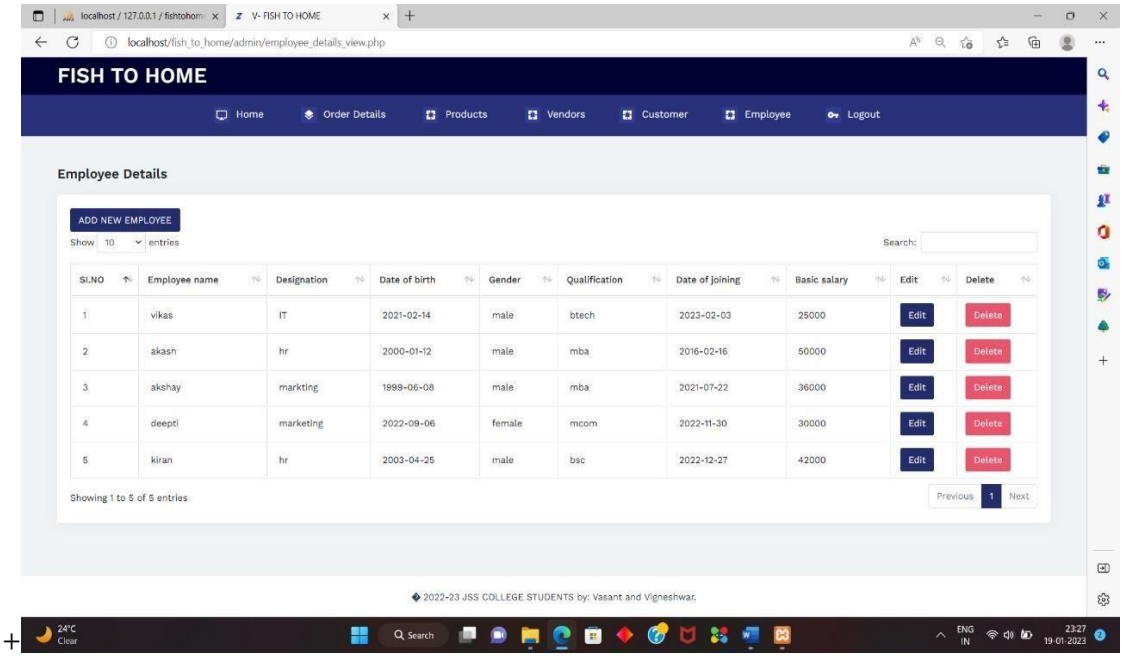


CUSTOMER PAYMENT

A VIEW OF

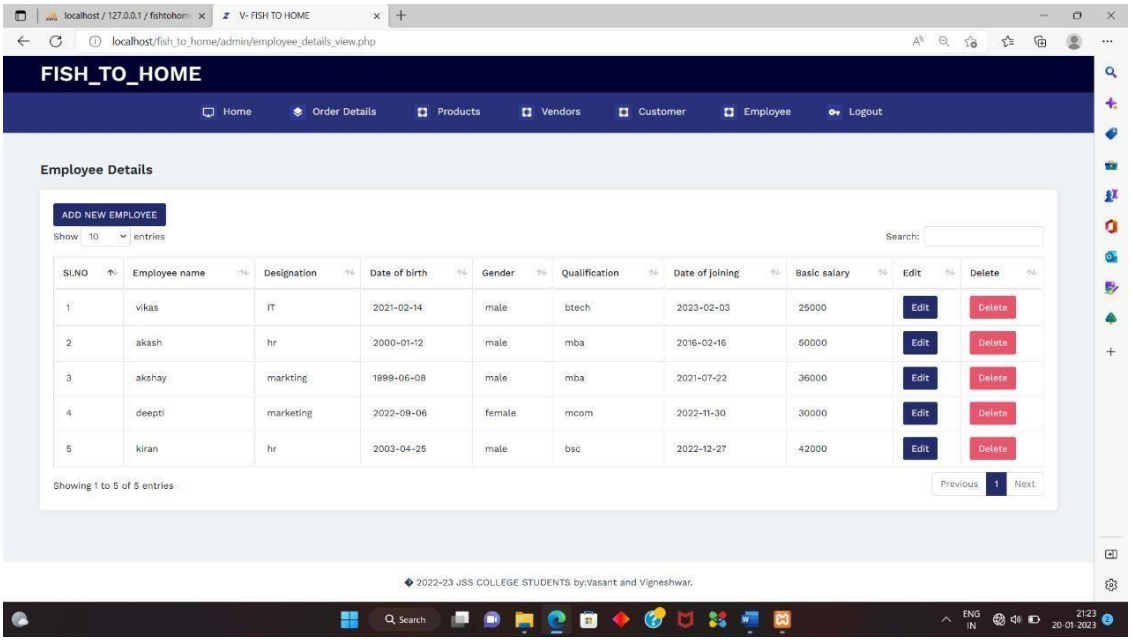


A VIEW OF EMPLOYEE DETAILS FORM FOR ADMIN

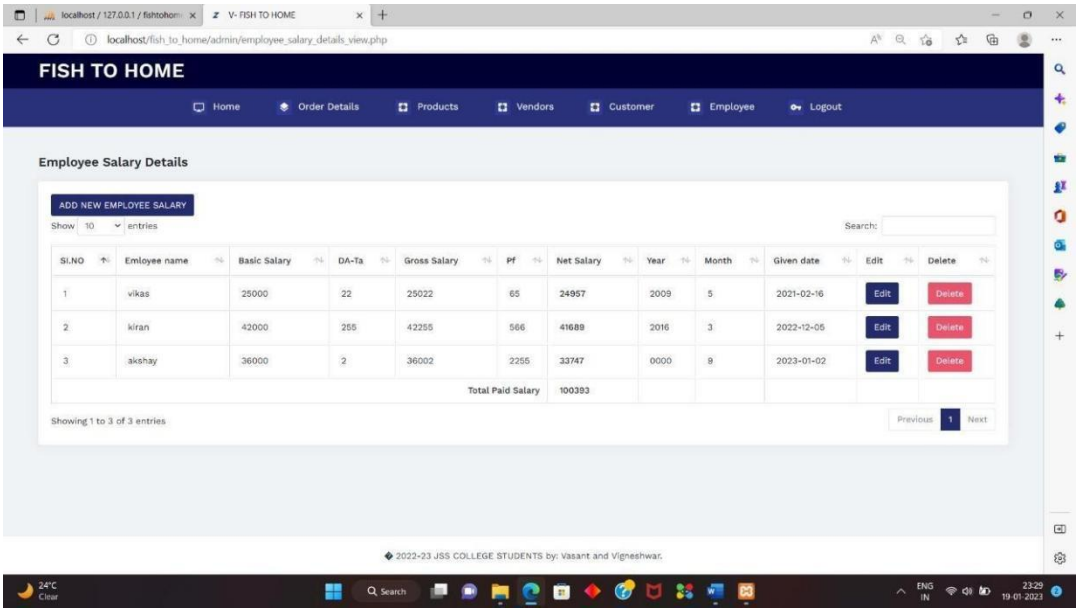


EMPLOYEE DETAILS FOR ADMIN

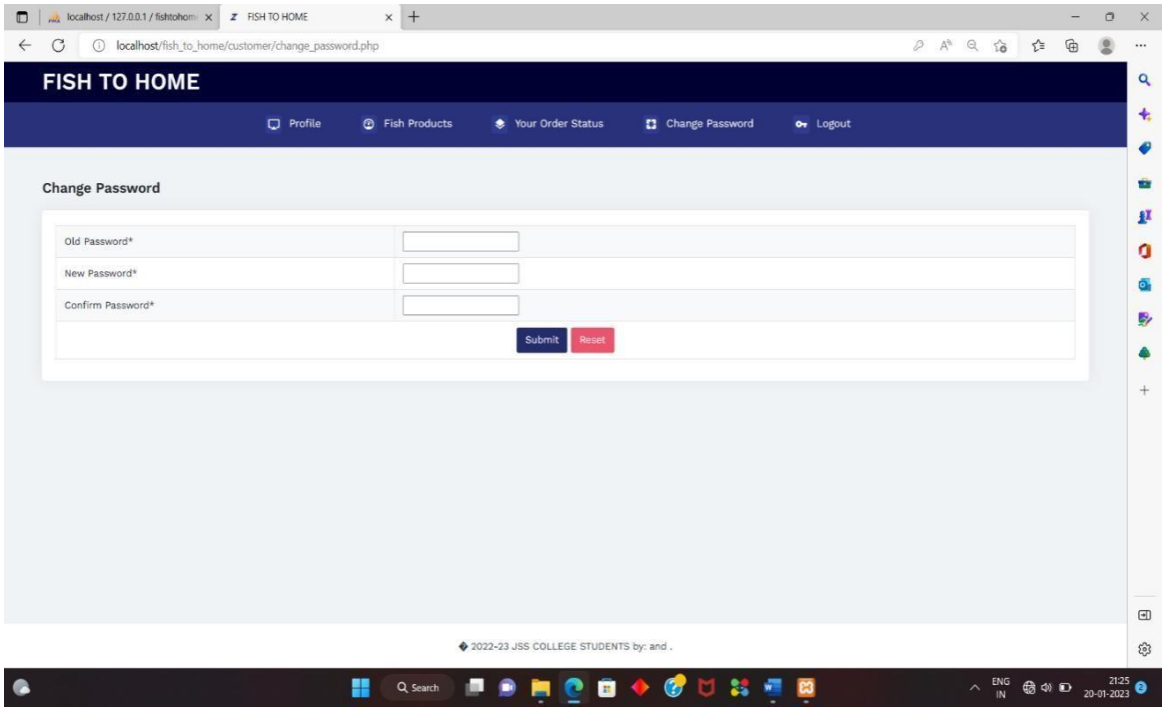
A VIEW OF



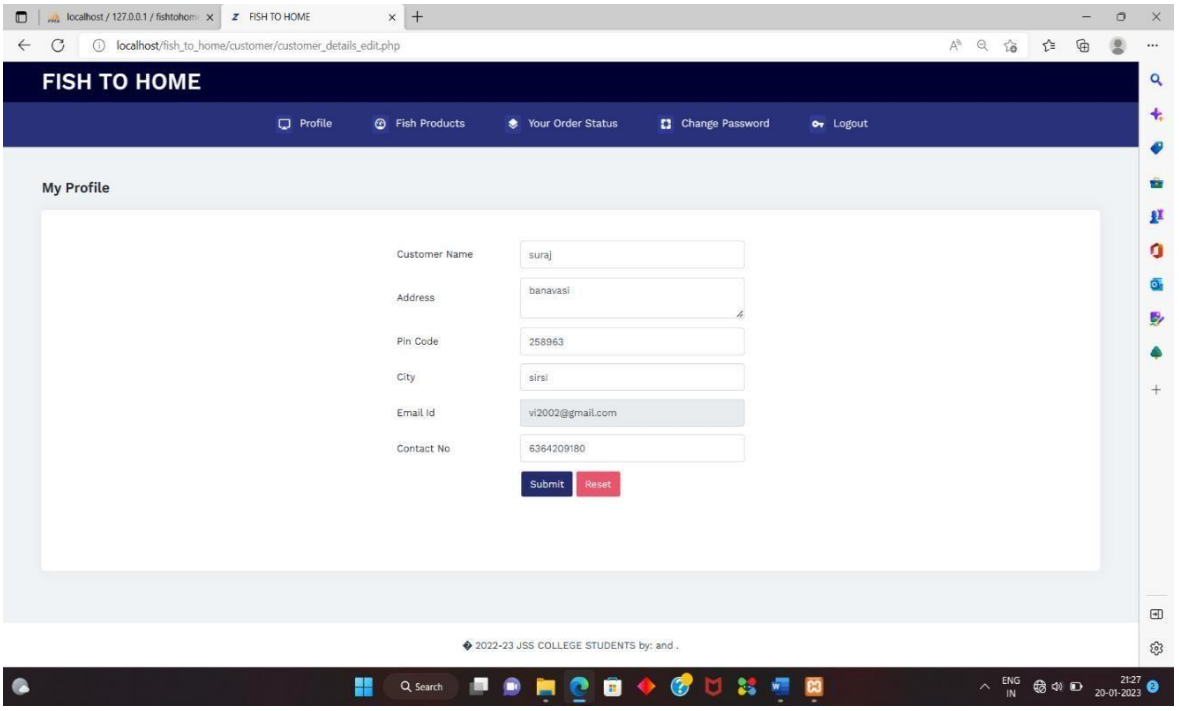
A VIEW OF EMPLOYEE SALARY DETAILS FOR ADMIN



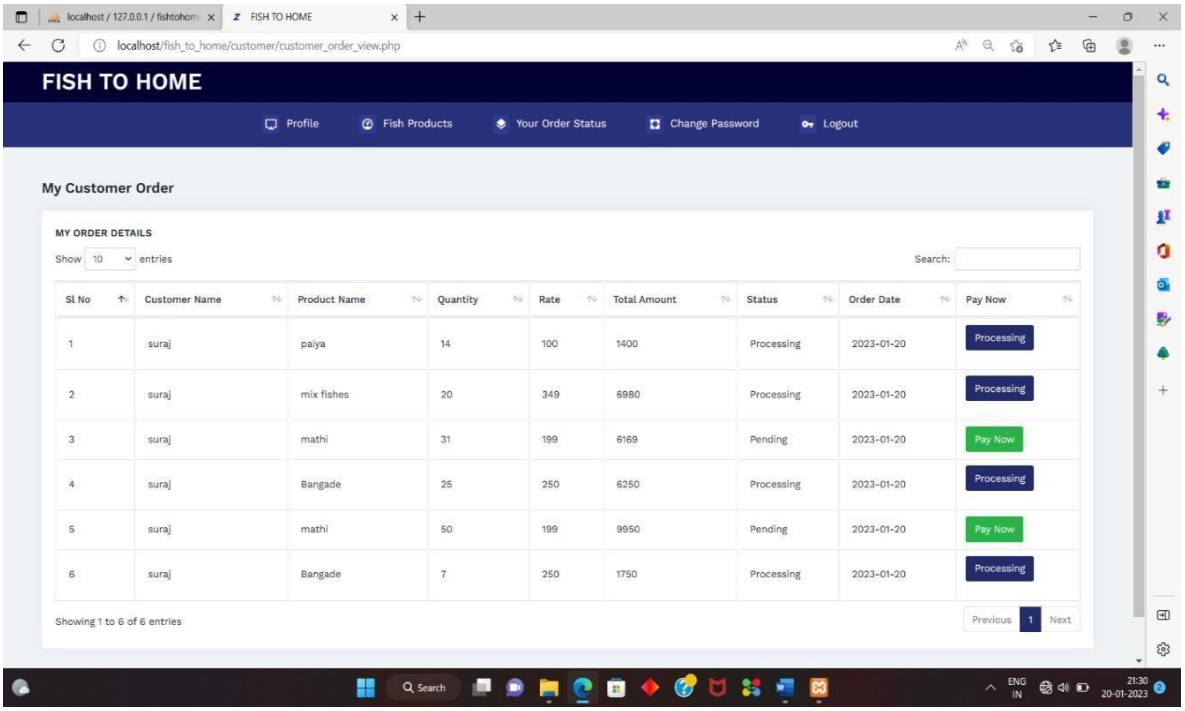
A VIEW OF CHANGE PASSWORD TABLE



A VIEW OF CUSTOMER PROFILE



A VIEW OF BILLING FOR CUSTOMER



8. SOURCE CODE

DATABASE CONNECTION:

```
<?php
$server="localhost";
$user="root";
$pass="";
$db="fish_to_home";
$conn=new mysqli($server,$user,$pass,$db);
?>
```

Customer Details: Form

Code:

```
<?php
include('meta_tag.php'); ?>
```

```
<?php
include('top_bar.php'); ?>

 <!-- MENU Start -->
   <?php
include('menu_bar.php');?>
<div class="wrapper">
<div
class="containerfluid">
  <!-- Page-Title -->
  <div class="row">
  <div class="col-sm-12">
<div class="page-titlbox">
 <div class="btn-group
pull-right">
 </div>
 <h4
class="pagetitle">Custome
r Details</h4>
  </div>
  </div>
 </div>
 <div class="row">
<div class="col-12">
<div class="card">
 <div class="card-body">
```

```
<h4 class="mt-0
headertitle">CustomerDeta
ls</h4>
<form name="form1"
id="formID"
method="post"
action="customer_details_i
nsert.php">

 <table width="620"
height="375" border="0"
align="center">
 <tr>
 <td
width="228">Customer
Name </td>
<td width="406"><input
name="customer_name"
type="text"
id="customer_name"
class="validate[required,cu
stom[onlyLetter]] form-
control"></td>
 </tr>
 <tr>
```

```
 <td
height="112">Address</td
>
<td><textarea
name="address"
id="address"
class="validate[required]
form-
control"></textarea></td>
  </tr>
  <tr>
   <td height="37">Pin
Code </td>
 <td><input
name="pin_code"
type="text" id="pin_code"
class="validate[required,cu
stom[pinCode]] form-
control"></td>
  </tr>
  <tr>
   <td
height="51">City</td>
 <td><input name="city"
type="text" id="city"
class="validate[required,cu
```

```
stom[onlyLetter]] form-
control"></td>
  </tr>
  <tr>
   <td height="41">Email
Id </td>
 <td><input
name="email_id"
type="text" id="email_id"
class="validate[required,cu
stom[email]] form-
control"></td>
  </tr>
  <tr>
  <td
height="54">ContactNo
</td>
<td><inputname="contactn
o" type="text"
id="contact_no"
class="validate[required,cu
stom[mobile]]
formcontrol"></td>
</tr>
<tr>
```

```
 <td
colspan="2"><divalign="c
enter">
<inputtype="submit"name=
"Submit" value="Submit"
class="btn btn-primary">
 <input
type="reset"name="Reset"
value="Reset" class="btn
btn-danger">
</div></td>
  </tr>
 </table>
 <p> </p>
 <p> </p>
</form>
 </div>
 </div> <!-- end col -->
 </div> <!-- end row </div>
<!-- end container -->
 </div>
 <!-- end wrapper -->

<?phpinclude('footer.php'); ?
>
<?php include('val.php');?>
```

Insert Code

```
<?php
include('../db_connect/dbconnect.php');
$customer_name=$_POST['customer_name'];
$address=$_POST['address'];
$pin_code=$_POST['pin_code'];
$city=$_POST['city'];
$email_id=$_POST['email_id'];
$contact_no=$_POST['contact_no'];
$sql="insert into customer_details
values(null,'$customer_name','$address','$pin_code','$city','$email_id','$contact_no')";
mysqli_query($conn,$sql);
?>
<script>
alert("Inserted...");
document.location="customer_details_form.php";
</script>
```

Delete Code:

```
<?php
  include('../db_connect/dbconnect.php');
 $customer_id=$_REQUEST['customer_id'];

 $sql="delete from customer_details where customer_id='$customer_id'";
 mysqli_query($conn,$sql);

?>
<script>
alert('Record Deleted..');
document.location="customer_details_view.php";
```

```
</script>
```

View Code:

```
<?php include('meta_tag.php'); ?>
<?php include('top_bar.php'); ?>
<?php include('menu_bar.php'); ?>
<div class="wrapper">
<div class="container-fluid">
 <!-- Page-Title -->
 <div class="row">
 <div class="col-sm-12">
 <div class="page-title-box">
 <div class="btn-group pull-right">
 </div>
 <h4 class="page-title">Customer Details</h4>
 </div>
 </div>
  </div>
   <!-- end page title end breadcrumb -->
   <div class="row">
   <div class="col-12">
   <div class="card">
   <div class="card-body">
   <h4 class="mt-0 header-title">Customer Details</h4>
   <table id="datatable" class="table table-bordered">
   <thead>
    <tr>
    <th>Sl No</th>
    <th>Customer Name</th>
    <th>Customer Address</th>
    <th>Pincode</th>
    <th>City</th>
    <th>Email id</th>
    < th>Contact Number </th>
   </tr>
  </thead>
  <tbody>
   <?php
  include('../db_connect/dbconnect.php');
$sl=1;
$sql="select * from customer_details";
$res=mysqli_query($conn,$sql);
```

```
while($row=mysqli_fetch_array($res))
{
?>
<tr>
  <td>  <?php echo $sl++; ?> </td>
  <td>  <?php echo $row['customer_name']; ?> </td>
  <td>  <?php echo $row['customer_address']; ?> </td>
  <td>  <?php echo $row['pincode']; ?> </td>
  <td>  <?php echo $row['city']; ?> </td>
  <td>  <?php echo $row['email_id']; ?> </td>
  <td>  <?php echo $row['contact_no']; ?> </td>
</tr>
<?php
}
?>
 </tbody>
 </table>
 </div>
 </div>
 </div> <!-- end col -->
 </div> <!-- end row -->
 <!-- Footer -->
 <?php include('footer.php'); ?>
```

Update Code

```
<?php

include('../db_connect/dbconnect.php');

$cust_order_id=$_POST['cust_order_id'];

$customer_id=$_POST['customer_id'];

$product_id=$_POST['product_id'];

$quantity=$_POST['quantity'];

$rate=$_POST['rate'];

$status=$_POST['status'];

$dat=$_POST['dat'];
```

```
$sql="update customer_order set customer_id='$customer_id',
product_id='$product_id',quantity='$quantity',rate='$rate',order_status='$st
atus',order_date='$dat' where cust_order_id='$cust_order_id'";

mysqli_query($conn,$sql);

?>

<script>

alert('updated...');

document.location="customer_order_view.php";

</script>
```

Edit Code:

```
<!DOCTYPE html>

<html lang="en">

<?php include('metatag.php'); ?>

<body>

 <div class="wrapper sidebar_minimize">

 <div class="main-header">

 <!-- Logo Header -->

 <!-- End Logo Header -->

 <!-- Navbar Header -->

 <?php include('header.php'); ?>

 <!-- End Navbar -->

 </div>

 <!-- Sidebar -->

 <?php include('sidebar.php');  ?>

<!-- Sidebar End -->

 <div class="main-panel">

 <div class="content">

 <div class="page-inner">
```

```
<div class="page-header">
<h4 class="page-title">Customer Details</h4>
<ul class="breadcrumbs">
<li class="nav-home"
<a href="#">
  <i class="flaticon-home"></i>
</a>
</li>
<li class="sepa
<i class="flaticon-right-arrow"></i>
</li>
</ul>
</div>
<div class="row">
<div class="col-md-12"> <div
class="card">
<div class="card-header">
<div class="card-title">Customer Details</div>  </div>
<div class="card-body"<div class="row">
<?php
include('dbconnect.php');
$a_id=$_REQUEST['a_id'];
$sql="select * from customer where cust_id='$a_id'";
$res=mysqli_query($conn,$sql); $row=mysqli_fetch_array($res);
?>
<form name="formID" ID="formID" method="post"
action="customer_details_update.php">
<input type="hidden" name="a_id" value="<?php echo $row['cust_id'];?>">
```

```
<form name="form1" method="post" action="customer_details_insert.php">
<table width="423" border="0">
 <tr>
<td width="160" height="49">Customer Name </td>
<td width="247"><input name="customer_name" type="text" id="customer_name" value="<?php echo $row['customer_name'];?>" class="form-control validate[required,custom[onlyLetter]]">
</td>
</tr>
<tr>
<td>Address</td>
 <td><textarea name="address" id="address" class="form-control validate[required]"><?php echo $row['address'];?></textarea>
</td>
</tr>
<tr>
 <td>City</td>
 <td><input name="customer_city" type="text" id="customer_city" value="<?php echo $row['customer_city'];?>" class="form-control validate[required,custom[onlyLetter]]">
 </td>
 </tr>
 <tr>
 <td>Contact No </td>
 <td><input name="contact_no" type="text" id="contact_no" value="<?php echo $row['contact_no'];?>" class="form-control validate[required,custom[mobile]]">
  </td>
  </tr>
  <tr>
  <td>Email ID </td>
```

```
<td><input name="email_id" type="text" id="email_id" value="<?php echo $row['email_id'];?>" class="form-control validate[required,custom[email]]">
</td>
</tr>
<tr>
<td>Customer Code </td>
<td><input name="customer_code" type="text" id="customer_code" value="<?php echo $row['customer_code'];?>" class="form-control validate[required,custom[onlyNumber]]">
</td>
</tr>
<tr>
<td height="58" colspan="2"><input type="submit" name="Submit" value="Submit" class="btn btn-primary">
<input type="reset" name="Reset" value="Reset" class="btn btn-danger"></td>
</tr>
</table>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<?php include('footer.php'); ?>
</div>
<!-- Custom template | don't include it in your project! -->
<?php include('setting.php'); ?>
<!-- End Custom template -->
```

```
</div>
<?php include('script.php'); ?>
<?php include('val.php'); ?>
</body>
</html>
```

Login Page Code:

```
<html>
<head>
<title>Login From Design</title>
<link rel="stylesheet" type="text/css" href="style.css">
<body>
<div class="loginbox">
<img src="img3.png" class="img3">
<h1>Login Here</h1>
<form method="post" action="logcheck.php">
<p>User name</p>
<input type="text" name="username" placeholder="Enter Username required>
<p>Password</p>
<input type="password" name="password" placeholder="Enter Password" maxlength="10" required>
<input type="submit" name="" value="Login">
<a href="forgot_password.php">Lost your Password?</a><br>
<a href="reg_form.php">Don't have an account?</a>
</form>
</div>
</body>
</head>
</html>
```

9 SYSTEM TESTING AND RESULTS TESTING:

Introduction

Testing is a process of executing a program with the indent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding. System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing. The code is tested at various levels in software testing. Unit, system and user acceptance testings are often performed.

Testing Objectives

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an undiscovered error.

Testing Principles

- All tests should be traceable to end user requirements.
- Tests should be planned long before testing begins.
- Testing should begin on a small scale and progress towards testing in large.
- Exhaustive testing is not possible.
- To be most effective testing should be conducted by a independent third party.

The primary objective for test case design is to derive a set of tests that has the highest livelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are:

- White box testing.
- Black box testing.

White Box Testing: White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Black Box Testing: Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing functions, interface errors, errors in data structures, error in functional logic are the errors falling in this category.

Testing strategies: A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

There are two general strategies for testing software. They are as follows:

Code Testing: This examines the logic of the program. To follow this test, cases are developed such that every path of program is tested.

Specification Testing: Specification Testing examines the specification, starting what the program should do and how it should perform under various conditions. Then test cases are developed for each condition and combinations of conditions and to be submitted for processing.

Levels of Testing

The stages of Testing Process are:

Unit Testing: Individual components are tested to ensure that they operate correctly. Each component tested independently without other system components. Ex. Check for Username and Password with the table, after the next module is loaded.

Integration Testing: Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing.

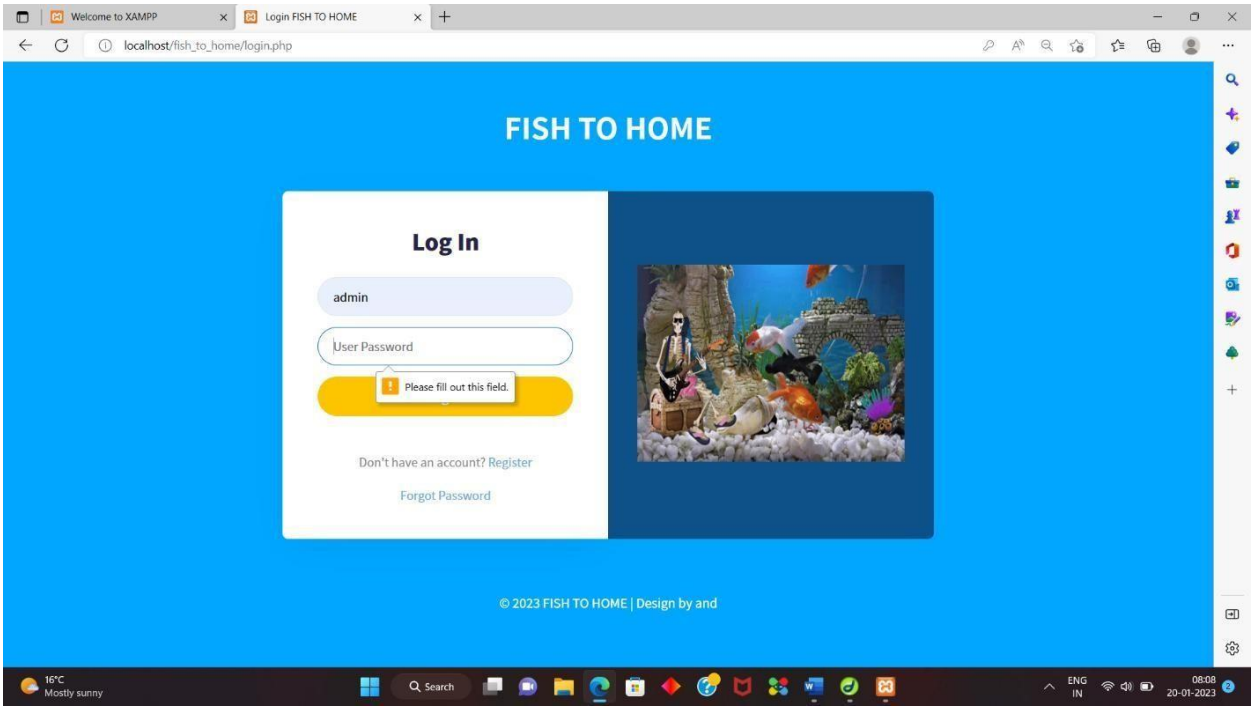
This testing is done using the bottom-up approach to integrate the software components of the software system in to functioning whole.

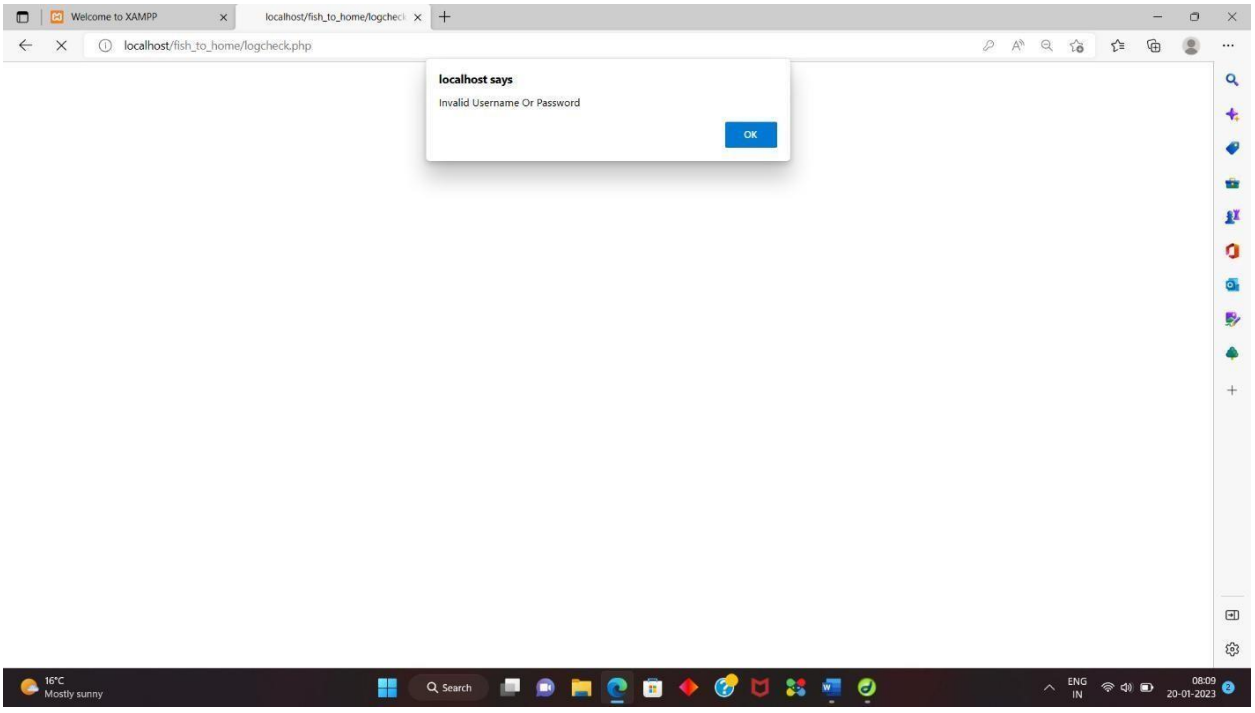
System Testing: System testing is actually a series of different tests whose primary purpose is fully to exercise the computer-based system. The system tests that where applied are recovery testing and performance testing. Finally, a review or audit is conducted which is a final evaluation that occurs only after operating the system long enough for user to have gained a familiarity with it. System testing was done by the inspection team to verify that all the functionality identified is the software requirement specification has been implemented. Defects that crept in the system has been found defect free and is working well. System testing is concerned with interfaces, design logic, control flow recovery, procedures throughput, capacity and timing characteristics of the entire system. For blank field, alphabets, number and special character validation.

Acceptance Testing: User acceptance of the system is the key factor for the success of any system. This is done by user. The system is given to the user and they test it with live data. Acceptance testing involves the planning and execution of functional test. Performance tests, stress tests in order to demonstrate that the implemented system satisfies its requirements. Two sets of acceptance test can be run, those developed by the customer. The system has been tested for its performance at unit level by the individuals through performance testing that is designed to test the run time performance of the software. The performance of the fully integrated system is tested and was found good.

Validating the tables :-

Login Form:





The previous screenshot is showing the error message that we should enter valid username and password.

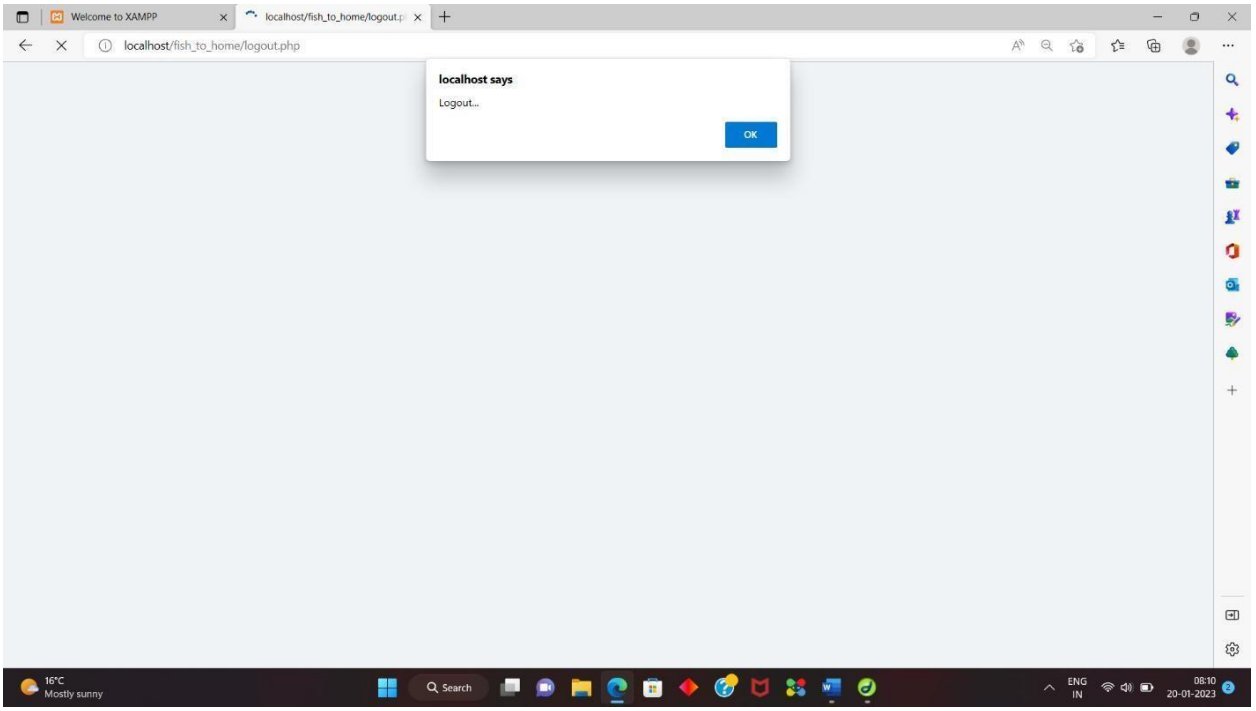
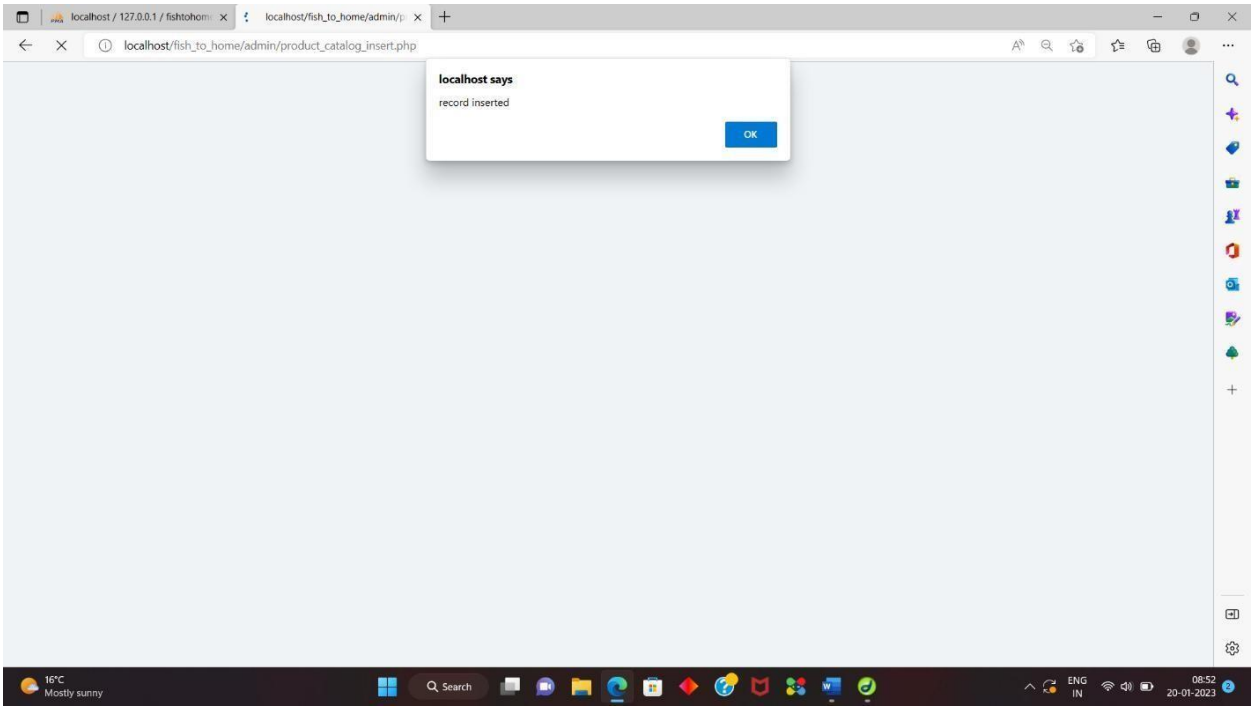


Fig Logout

CUSTOMER TABLE

Adding a new value to the table



Functional Testing:

Test case:

Test No	Test Case	Expected Result	Pass
1	Leave the Username and Password Fields blank	Message Stating that fill the feilds	✓
2	Invalid Username and Password	Message stating that invalid username and password	✓
3	Mismatch New password and confirm password	Message stating that new password and confirm password donot match	✓
4	Leave all the text boxes blank	Message staring that all these fields should not be empty	✓
5	Leave the customer name empty	Message stating that fill all the fields	✓
6	Leave amount field empty	Message stating that fill the fields	✓
7	Order quantity less than current stock	Message stating that No Stock	✓
8	Invalid E-mail	Message stating that invalid e-mail	✓

10. CONCLUSION

Software is said to have attained its objective only when it need all requirements of the user, further the user himself is the person to judge the success of the system. Every attempt has been made to ensure that the system is fully functional and works effectively and efficiently. The system has been tested with simple data to cover all possible options and checked for all outputs. Since the system is flexible and modular, further modification of this package can be easily incorporated.

Importance of the system

- Less manual work.
- Increased efficiency.
- Decreases the rate of errors.
- It reduces the time consumption.
- Quick (instant) result.

11 .FUTURE ENHANCEMENT OF PROJECT

- Development of Android and IOS application.
- GPS system can be connected.
- Supports multiple language.

11. BIBLIOGRAPHY

Books

1. Fundamentals of database system-B.Navathe

2. Ian Summerville “Software Engineering” pearson Education Ltd 6th edition

3. Ali Bahrami Object Oriented Systems Development, Mcgraw hill.

4. Elias M Awad, System Analysis and Design, Golgotia

5. HTML How to Program by Rohit Khurana.

Websites:

1. http://en.wikipedia.org/wiki/PHP for php
2. http://www.mysql.com/click.php?e=35050 for MySQL
3. http://stackoverflow.com for Validation
4. www.tutorialspoint.com for HTML
5. https://www.programiz.com/javascript/examples/string-comparison
6. https://www.w3schools.com/jsref/jsref_length_string.asp
7. https://developer.chrome.com/docs/devtools/console/javascript/