VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Santhibastawad Road, Machhe Belagavi - 590018, Karnataka, India

DBMS LABORATORY WITH MINI PROJECT (18CSL58) REPORT

ON

“Canteen Database Management System”

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

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

For the Academic Year 2022-2023

Submitted by

Anushka Roy	**1JS20IS023**
G Sanjana Reddy	**1JS20IS036**

Under the Guidance of
Dr. Sowmya K N
Associate Professor, Dept. of ISE, JSSATEB

2022-2023

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060

JSS MAHAVIDYAPEETHA, MYSURU

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Dr.Vishnuvardhan Road, Bengaluru-560060

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that DBMS LABORATORY WITH MINI PROJECT (18CSL58) Report entitled "**Canteen Database Management System**" is a Bonafede work carried out by **Anushka Roy[1JS20IS0123]**, **G Sanjana Reddy [1JS20IS036]** in partial fulfilment for the award of degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University Belagavi during the year 2022- 2023.

Signature of the Guide
Dr. Sowmya K N
Associate Professor,
Dept. of ISE,
JSSATEB

Signature of the HOD
Dr. Rekha P M
Professor & HOD,
Dept. of ISE,
JSSATEB

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So, with gratitude, we acknowledge all those whose guidance and encouragement crowned our efforts with success.

First and foremost, we would like to thank his **Holiness Jagadguru Sri Shivarathri Deshikendra Mahaswamiji** and **Dr. Bhimasen Soragaon**, Principal, JSSATE, Bangalore for providing an opportunity to carry out the Project Work as a part of our curriculum in the partial fulfilment of the degree course.

We express our sincere gratitude for our beloved Head of the department, **Dr. Rekha P M**, for her co-operation and encouragement at all the moments of our approach.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guide **Dr. Sowmya K N**, Associate Professor for the constant encouragement, valuable help and assistance in every possible way.

We would like to thank all ISE Department Teachers, **Mrs. Sowmya G** ma'am our laboratory instructor and non-teaching staff for providing us with their valuable guidance and for being there at all stages of our work.

Anushka Roy [1JS20IS023]

G Sanjana Reddy [1JS20IS036]

ABSTRACT

Canteen management software is a type of ERP or customer relationship management (CRM) software that is specifically designed for managing the operations of canteens and cafeterias. The canteen management software normally consists of hardware, software, and consumables. A canteen is a place where students or employees can buy food for their personal consumption during work hours.

Canteen management system is to provide fast services to their college students, Staffs etc. Usually, People have to go to canteen and order the foods and they have to wait in queue for a long time to get the orders, but with the help of this you just have to follow a very simple process to order the food items. A canteen facility is a supplementary system that is provided by organizations for them employees/students. Organizations with large numbers of employees cannot handle a canteen with manual processes. Our canteen management system provides a friendly User Interface for numerous shop outlets in the JSSATE campus like menu design, scheduling pick-ups and lots more. Implementation of such a system makes the operation of the kitchen and the whole of the canteen as effective and quick as possible.

This system is generally advantageous for avoiding spending time waiting in the queue by posting orders directly to the kitchen without delay and also by scheduling orders ahead of time. It saves time and also the technique dealing with it is easy. Proposed Canteen Management System is an adept solution for chaos at college canteens.

The system will maintain location wise canteen details of particular organization. The system will also maintain available menus with their rates and quantity. The proposed system is going to maintain a calendar for canteen so that administrators can pre-plan the menus for particular timeslot.

CHAPTER 1

INTRODUCTION

INTRODUCTION TO CANTEEN DATABASE MANAGEMENT SYSTEM

Currently, the way college canteen works is, that you pay for the food and you wait for the food in line but the problem is that all the students in a facility have the same time slot for the break hence a group of people end up rushing towards the canteen at the same time this creates a lot of inconvenience for the canteen staff as well as the students and since the time is limited some students end up not eating food or wasting their food. This is where the Canteen Management system enters it helps in streamlining the whole process wherein students can order their food via their phones beforehand from the website in which the user has to enter their email id and then they are ready to order via e-menu.

As soon as a user order anything the website will alert about it the kitchen staff and they can start preparing it. Ultimately Canteen Management System will help streamline this whole cumbersome process and change the age-old ways of doing things.

OBJECTIVES

The Objectives of **CANTEEN DATABASE MANAGEMENT** are:

1. The main objective of the project on canteen management system is to manage the details of Canteen, customers of the college, food Items with prices.
2. The project is to build a web application program to reduce the manual work for managing the Canteen, Students, Customers, and Food Item etc. It tracks all the details essential for the management of the canteen system.
3. The project aims to keep record of the sales.
4. The mission targets to provide sorting and searching facilities based on various factors. Such as different shops in the Campus, customers, Food Items and Orders.
5. One of the important objectives is to Edit, add and update the records, which results in proper resource management of canteen data.
6. The project will have all fields such as customer sign-up and log-in to be validated so that they do not take invalid values.

The application ambitions to increase the efficiency of managing the canteen.

ORGANIZATION OF THE REPORT

Chapter 1 provides the information about the Introduction to the Canteen Database Management System and the objectives. In **chapter 2** we discuss the literature survey and the normalization. In **Chapter 3**, we discuss the software and hardware requirements to run the above applications. **Chapter 4** gives the idea of the Database design and wireframe of the Canteen Database Management System. **Chapter 5** gives a clear picture about the project and its actual implementation. In **Chapter 6** we discuss about testing the Canteen Database Management System. **Chapter 7** discusses the results and discussions of the program. **Chapter 8** concludes by giving the direction for future enhancement and the **Chapter 9** includes the references.

CHAPTER 2

LITERATURE SURVEY

INTRODUCTION

A database management system (DBMS) refers to the technology for creating and managing databases. Basically, DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of database. A datum is a unit of data. Meaningful data combined to form information. Hence, information is interpreted data – data provided with semantics. MS. ACCESS is one of the most common examples of database management software. The name indicates what database is. Database is one of the important components for many applications and is used for storing a series of data in a single set. In other words, it is a group / package of information that is put in order so that it can be easily access, manage and update.

MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is developed, marketed and supported by MySQL AB, which is a Swedish company.

MySQL is becoming so popular because of many good reasons −It is released under an open-source license. So, you have nothing to pay to use it, it is a very powerful program in its own right and handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc and works very quickly and works well even with large data sets.

PHP

PHP Hypertext Pre-processor is a scripting language that helps people make web pages moreinteractive by allowing them to do more things. PHP code is run on the web server.

A website programmed with PHP can have pages that are password protected. A website with noprogramming cannot do this without other complex things. Standard PHP file extensions are: .php .php3 or .phtml, but a web server can be set up to use any extension.

Its structure was influenced by many languages like C, Perl, Java, C++, and even Python. It isconsidered to be free software by the Free Software Foundation.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Bootstrap

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

Bootstrap is an HTML, CSS and JS library that focuses on simplifying the development of informative web pages (as opposed to web applications). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight. Bootstrap also comes with several JavaScript components which do not require other libraries like jQuery. They provide additional user interface elements such as dialog boxes, tooltips, progress bars, navigation drop-downs, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code.

JAVASCRIPT

JavaScript abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually via Node.js. They are also embedded in a variety of applications created with frameworks such as Electron and Cordova.

Windows 11

Windows 11 is the latest major release of Microsoft's Windows NT operating system, released in October 2021. It is a free upgrade to its predecessor, Windows 10 (2015), and is available for any Windows 10 devices that meet the new Windows 11 system requirements.

Windows 11 features major changes to the Windows shell influenced by the canceled Windows 10X, including a redesigned Start menu, the replacement of its "live tiles" with a separate "Widgets" panel on the taskbar, the ability to create tiled sets of windows that can be minimized and restored from the taskbar as a group, and new gaming technologies inherited from Xbox Series X and Series S such as Auto HDR and DirectStorage on compatible hardware. Internet Explorer (IE) has been replaced by the Chromium-based Microsoft Edge as the default web browser, like its predecessor, Windows 10, and Microsoft Teams is integrated into the Windows shell. Microsoft also announced plans to allow more flexibility in software that can be distributed via the Microsoft Store and to support Android apps on Windows 11 (including a partnership with Amazon to make its app store available for the function).

Mac OS X

Mac OS X was originally presented as the tenth major version of Apple's operating system for Macintosh computers; until 2020, versions of macOS retained the major version number "10". The letter "X" in Mac OS X's name refers to the number 10, a Roman numeral, and Apple has stated that it should be pronounced "ten" in this context. However, it is also commonly pronounced like the letter "X". Previous Macintosh operating systems (versions of the classic Mac OS) were named using Arabic numerals, as with Mac OS 8 and Mac OS 9.

NORMALIZATION

Database normalization is a database schema design technique, by which an existing schema is modified to minimize redundancy and dependency of data. Normalization split a large table into smaller tables and define relationships between them to increases the clarity in organizing data.

Database normalization types

First Normal Form (1NF)

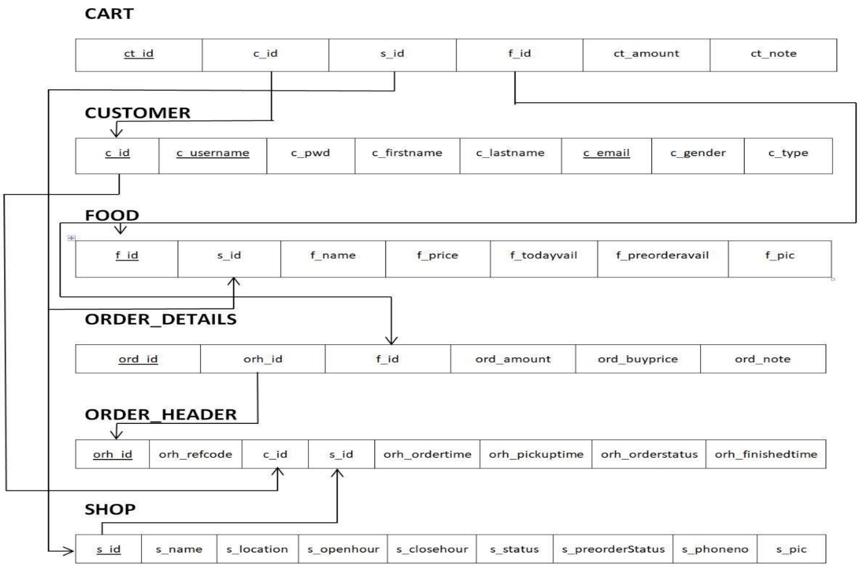
- First normal form (1NF) deals with the `shape' of the record type
- A relation is in 1NF if, and only if, it contains no repeating attributes or groups of attributes.
- Example: The Student table with the repeating group is not in 1NF

Second Normal Form (2NF)

- A relation is in 2NF if, and only if, it is in 1NF and every non-key attribute is fully functionally dependent on the whole key.

Third Normal Form (3NF)

- A relation is in 3NF if, and only if, it is in 2NF and there are no transitive functional dependencies.
- Transitive functional dependencies arise:
    - When one non-key attribute is functionally dependent on another non-key attribute
    - FD: non-key attribute -> non-key attribute and
    - When there is redundancy in the database



CART

ct_id	c_id	s_id	f_id	ct_amount	ct_note

CUSTOMER

c_id	c_username	c_pwd	c_firstname	c_lastname	c_email	c_gender	c_type

FOOD

f_id	s_id	f_name	f_price	f_todayvail	f_preorderavail	f_pic

ORDER_DETAILS

ord_id	orh_id	f_id	ord_amount	ord_buyprice	ord_note

ORDER_HEADER

orh_id	orh_refcode	c_id	s_id	orh_ordertime	orh_pickuptime	orh_orderstatus	orh_finishedtime

SHOP

s_id	s_name	s_location	s_openhour	s_closehour	s_status	s_preorderStatus	s_phoneno	s_pic

Boyce-Codd Normal Form (BCNF)

- When a relation has more than one candidate key, anomalies may result even though the relation is in 3NF.
- 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keys

  i.e. composite candidate keys with at least one attribute in common.
- BCNF is based on the concept of a determinant.

Fourth Normal Form (4NF)

- It is a normal form used in database normalization Introduced by Ronald Fagin in 1977, 4NF is the next level of normalization after Boyce–Codd normal form (BCNF).
- Whereas the second, third, and Boyce–Codd normal forms are concerned with functional dependencies, 4NF is concerned with a more general type of dependency known as a multivalued dependency.

Fifth Normal Form (5NF)

- It is also known as project-join normal form (PJ/NF) is a level of database normalization designed to reduce redundancy in relational databases recording multi-valued facts by isolating semantically related multiple relationships.
- A table is said to be in the 5NF if and only if every nontrivial join dependency in that table is implied by the candidate keys.

CHAPTER 3

REQUIREMENT SPECIFICATIONS

SOFTWARE SPECFICATION

- Project Type: Web-Based Application
- Front-end Tech: HTML, CSS, BOOTSTRAP
- Database Tool: MySQL
- Back-end Tech: PHP
- OS: Windows 11 and above, Linux and Mac compatible.
- Browser: Internet explorer, Chrome, Firefox, or Safari
- Software: XAMPP

HARDWARE SPECFICATION

- Processor: x86 compatible processor with 1.7 GHz Clock Speed
- RAM: 512 MB or greater
- Hard Disk: 20 GB or grater
- Monitor: VGA/SVGA
- Keyboard: 104 keys standard
- Mouse: 2/3 button. Optical/Mechanical.

USER CHARACTERISTICS

Every user:

- Should be comfortable with basic working of the computer.
- Must have basic knowledge of English.

FUNCTIONAL REQUIREMENTS

Users of the online canteen system, namely canteen customers, must be provided with the following functionalities:

Create an account. Manage their account. Log into the system. Navigating the menu of canteen system according to the presence of items at that period of time. Select an item from the menu. Options to customize the selected items. Add an item to their current order. Reviewing the current order of customer after the modification. Providing option 'Remove' to discard the items which added to the current order. We can remove items in selected manner if needed customer can remove all items too. Place an order by selecting the canteen. Customer will receive the confirmation in the form of an order number. We can check the preparation time by current order number.

NON-FUNCTIONAL REQUIREMENTS

These set of requirements are an extension to the functional requirements. Non-functional requirements are not the ones that a user demands, but is implicitly expected from the application developed so far. It can be thought of as the "efficiency" of the application in terms of its performance. This term can be further elaborated in the light of a simple fact.

When we order food on canteen system, the food has to be the one that is selected, but beyond that many companies offer delivery within 30 minutes, else, free delivery. This can be thought of as a non-functional requirement which extends beyond providing the basic functionality. In our case we can include non-functionalities like free food if the order is not prepared by pick-up time.

CHAPTER 4

DATABASE DESIGN

INTRODUCTION TO DATABASE DESIGN

Database design is the process of creating a structured and organized representation of data that can be used in a database management system. The goal of database design is to ensure that the database is efficient, reliable, and easy to use. A well-designed database should minimize data redundancy, minimize data inconsistencies, and provide data integrity.

The process of database design involves several steps, including requirements gathering, conceptual data modelling, logical data modelling, and physical database design. During the conceptual data modelling stage, the designer creates an abstract representation of the data and its relationships. Logical data modelling involves creating a more detailed and structured representation of the data, while physical database design involves creating a physical model of the database, including the actual data tables, indexes, and relationships.

Database Design is a collection of processes that facilitate the designing, development, implementation, and maintenance of enterprise data management systems. Properly designed databases are easy to maintain, improve data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored. The main objectives of database designing are to produce logical and physical designs models of the proposed database system.

ATTRIBUTES

Attributes define the properties of a data object and take on one of three different characteristics.

They can be used to:

- Name an instance of data object.
- Describe the instance.

The following figure shows the Relational model for Student Marks and Activity database management.

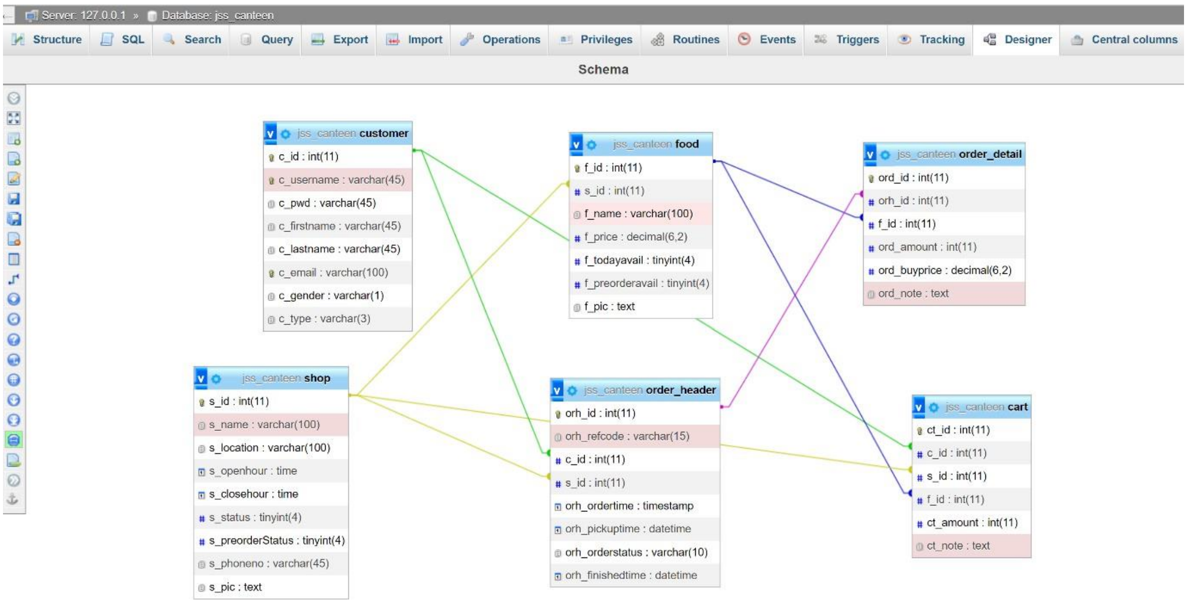


Fig4.1: - Relational model for Canteen database management.

Relational Model was proposed by E.F. Codd to model data in the form of relations or tables. After designing the conceptual model of Database using ER diagram, we need to convert the conceptual model in the relational model which can be implemented using any RDBMS languages like MySQL etc. So, we will see what Relational Model is. Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables).

SCHEMA DIAGRAM

A **database schema** is the skeleton structure that represents the logical view of the entire database. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

A **schema diagram** contains entities and the attributes that will define that **schema**. It only shows us the database design. It does not show the actual data of the database. **Schema** can be a single table or it can have more than one table which is related.

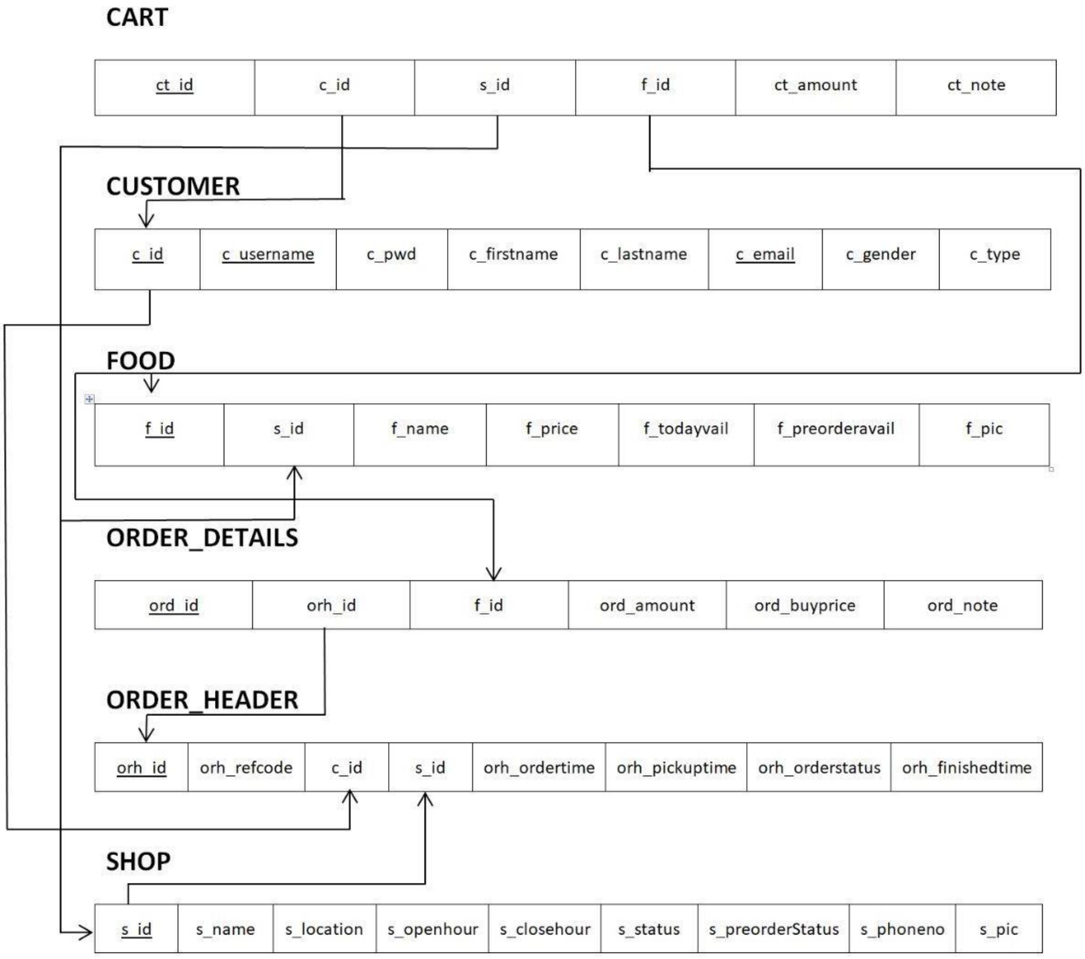


Fig4.2:- Schema Diagram for Canteen database management

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language.

The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modeled in the database. All the various table used are described in the following schema. The necessary Primary key's and the corresponding foreign keys are also represented.

ER DIAGRAM

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

The ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

Here are the geometric shapes and their meaning in an E-R Diagram.

Rectangle: Represents Entity sets.

Ellipses: Attributes

Diamonds: Relationship Set

Lines: They link attributes to Entity Sets and Entity sets to Relationship Set

Double Ellipses: Multivalued Attributes

Dashed Ellipses: Derived Attributes

Double Rectangles: Weak Entity Sets

Double Lines: Total participation of an entity in a relationship set

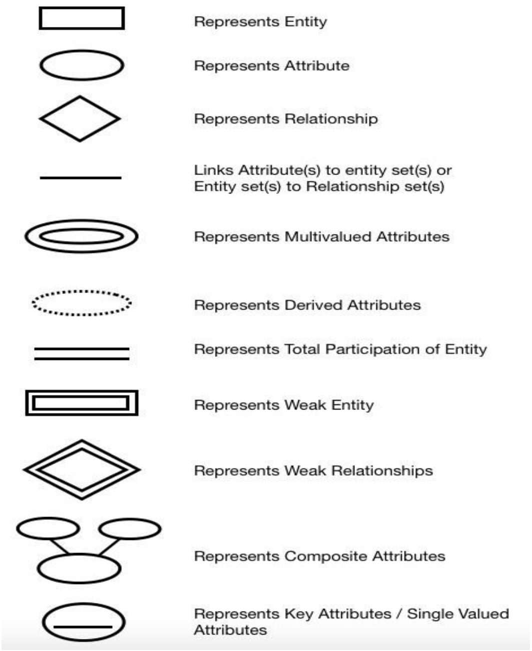

Represents Entity
Represents Attribute
Represents Relationship
Links Attribute(s) to entity set(s) or Entity set(s) to Relationship set(s)
Represents Multivalued Attributes
Represents Derived Attributes
Represents Total Participation of Entity
Represents Weak Entity
Represents Weak Relationships
Represents Composite Attributes
Represents Key Attributes / Single Valued Attributes

Fig4.3:- ER Diagram Symbols

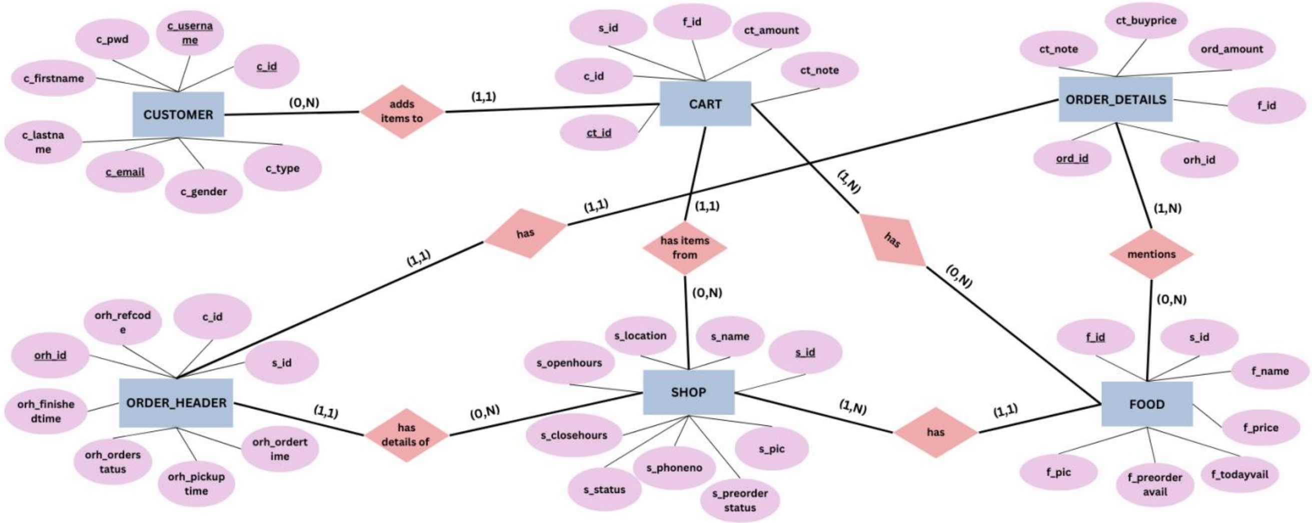


Fig4.4:- Canteen Database ER Diagram

An entity–relationship model or the ER Diagram describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types and specifies relationships that can exist between instances of those entity types.

WIREFRAME

A wireframe is a two-dimensional illustration of a page's interface that specifically focuses on space allocation and prioritization of content, functionalities available, and intended behaviours. For these reasons, wireframes typically do not include any styling, colour, or graphics. Wireframes also help establish relationships between a website's various templates.

The Value of Wireframes

Wireframes serve multiple purposes in the mini project by helping to:

- Connect the site's information architecture to its visual design by showing paths between pages.
- Clarify consistent ways for displaying particular types of information on the user interface in the canteen management system.
- Determine intended functionality in the interface to make our application operative.
- Prioritize content through the determination of how much space to allocate to a given food item and where that item is located.

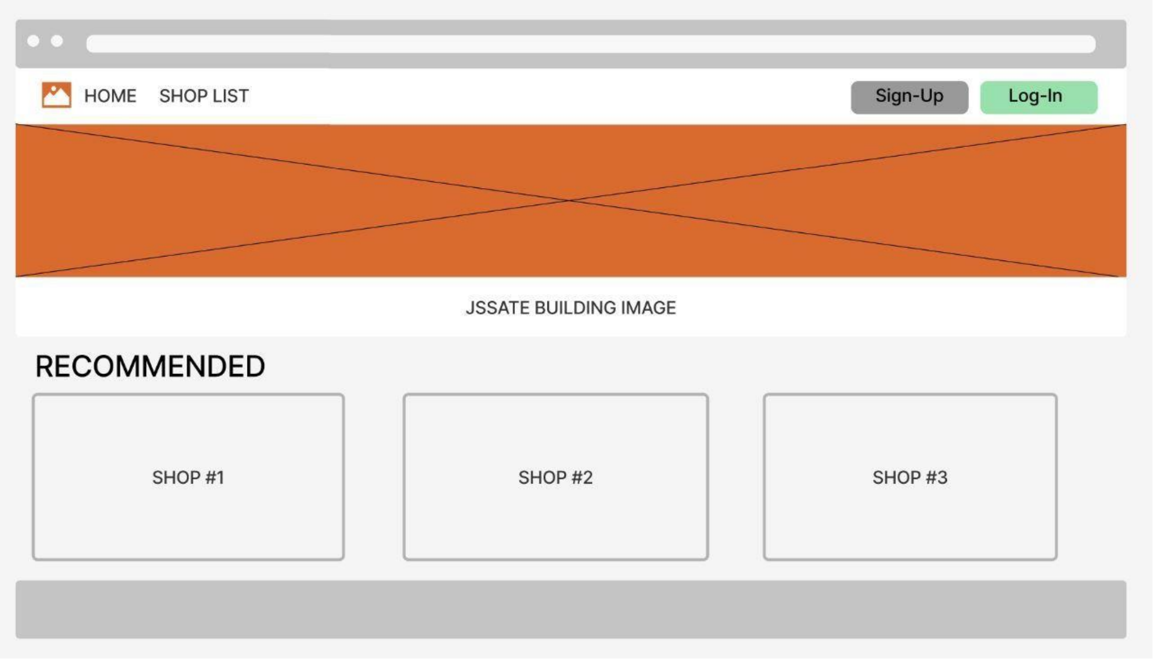


Fig4.5:- Home Page

The above wire frame is the simple block diagram that shows the placement of elements in the canteen management system's home page for the customers. We observe the demonstration of the intended layout of different shop outlets in the campus under recommendations. We have the logo on the top left corner of every page in the application. We also need two buttons "log-in" and "sign-up" that take us to the next corresponding pages.

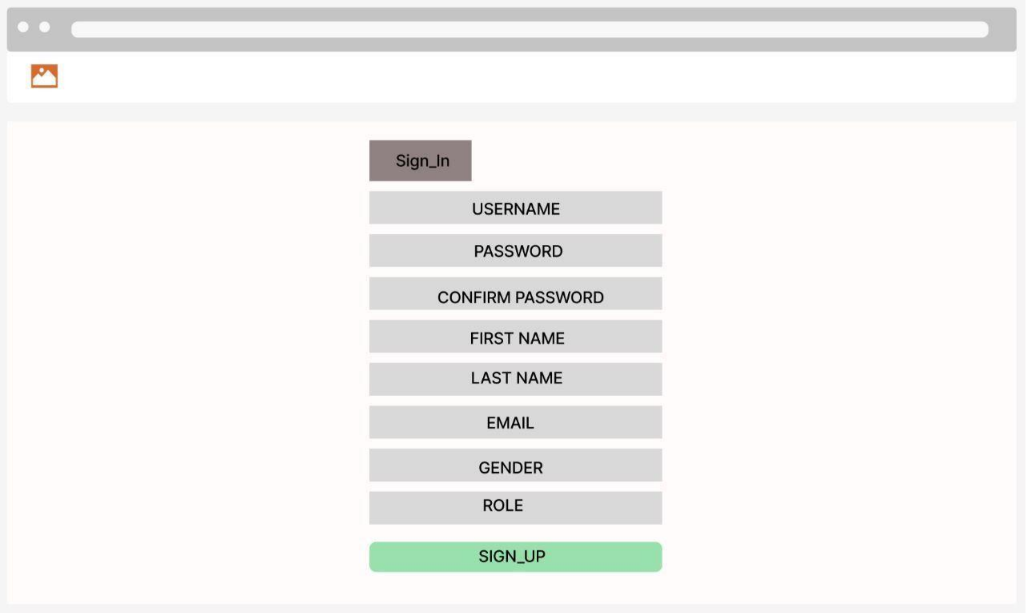
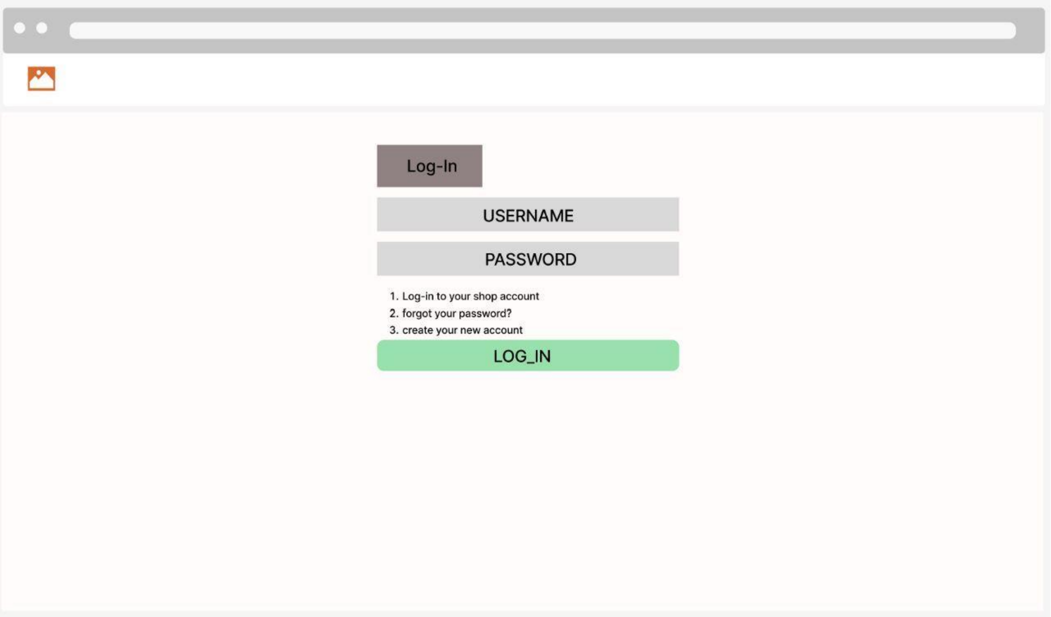


Fig4.6:- Sign-in Page

A signup page (also known as a registration page) enables users or customers to independently register and gain access to the canteen application system. There are different fields mentioned in the page to be filled which are validated and the text boxes are provided to take the input from the user.



1. Log-in to your shop account
2. forgot your password?
3. create your new account

Fig4.7:- Log-in Page

The login page allows a user to access a website or web application by entering their username and password or by authenticating with a social network login. In addition, the page allows you to enter both authorized users and those who first visited the site and need to register. There are various options given in the same page under the text fields in case, the registered user forgets the authenticating username or password.

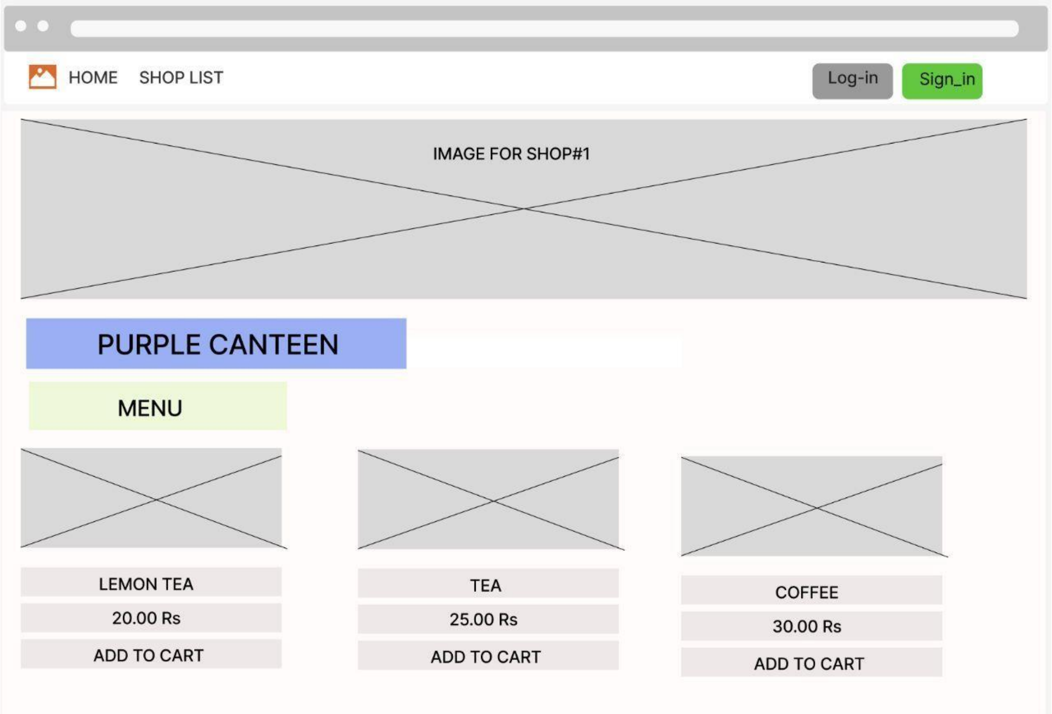


Fig4.8:- Shop#1 Page

The “Purple Canteen” page can be accessed by registered and non-registered users of the application, to scan through the items available in the canteen. The food items available are given, along with their prices and other basic details. When add to cart button is clicked, the application takes the user to the next page or if the user hasn’t logged in, the login page is extended to connect.

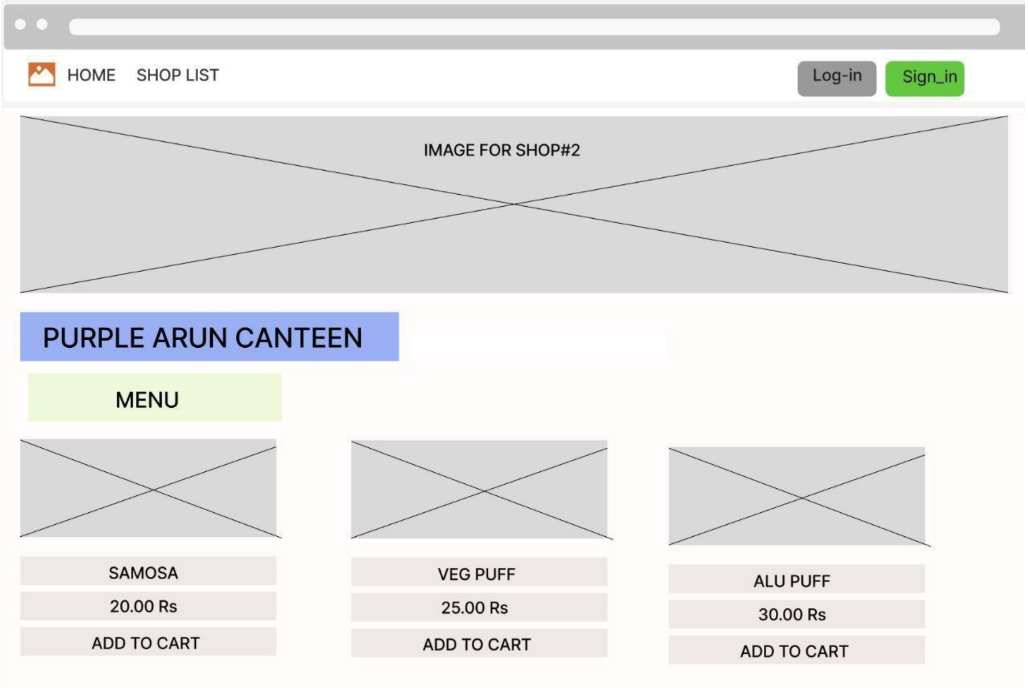


Fig4.9:- Shop#2 Page

The “Arun Canteen” page can be accessed by registered and non-registered users of the application, to scan through the items available in the canteen. The food items available are given, along with their prices and other basic details. When add to cart button is clicked, the application takes the user to the next page or if the user hasn’t logged in, the login page is extended to connect.

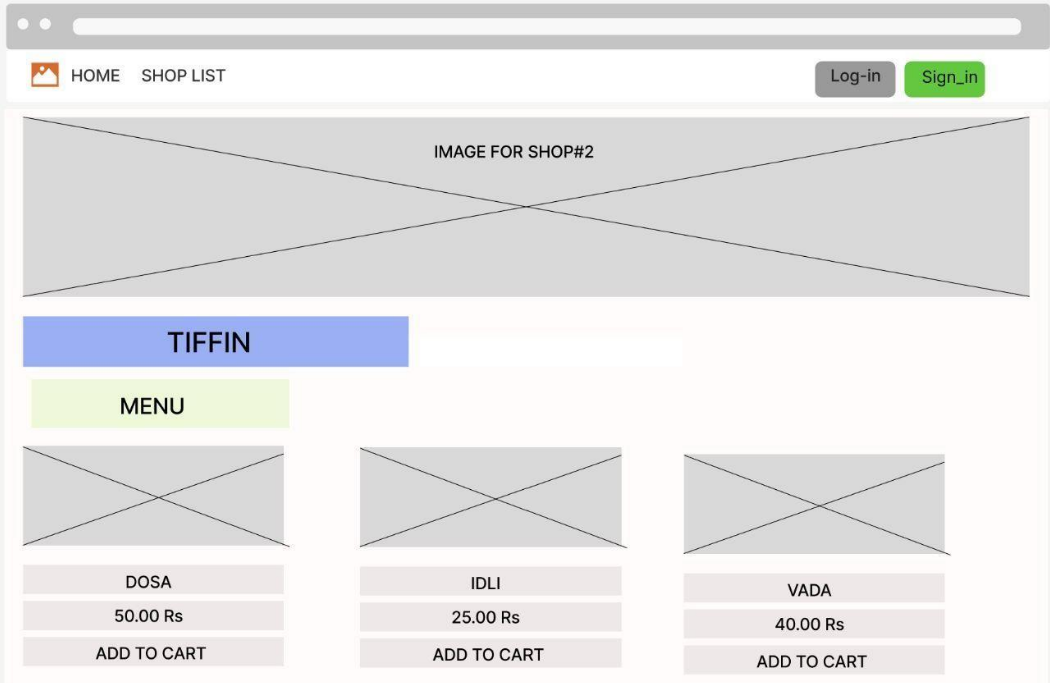


Fig4.10:- Shop#3 Page

The "Tiffin Canteen" page can be accessed by registered and non-registered users of the application, to scan through the items available in the canteen. The food items available are given, along with their prices and other basic details. When add to cart button is clicked, the application takes the user to the next page or if the user hasn't logged in, the login page is extended to connect.

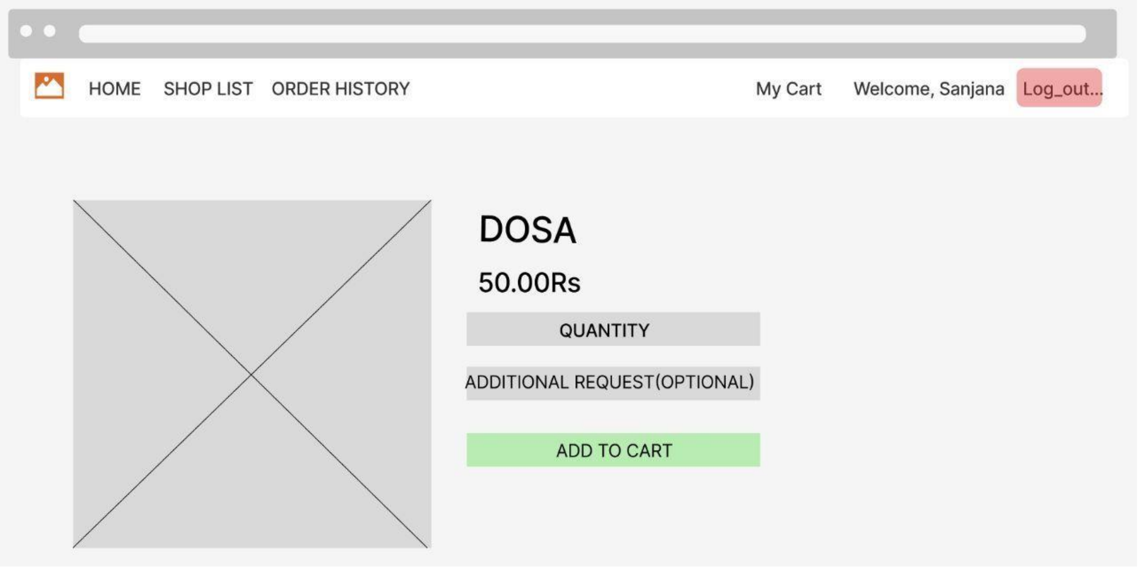
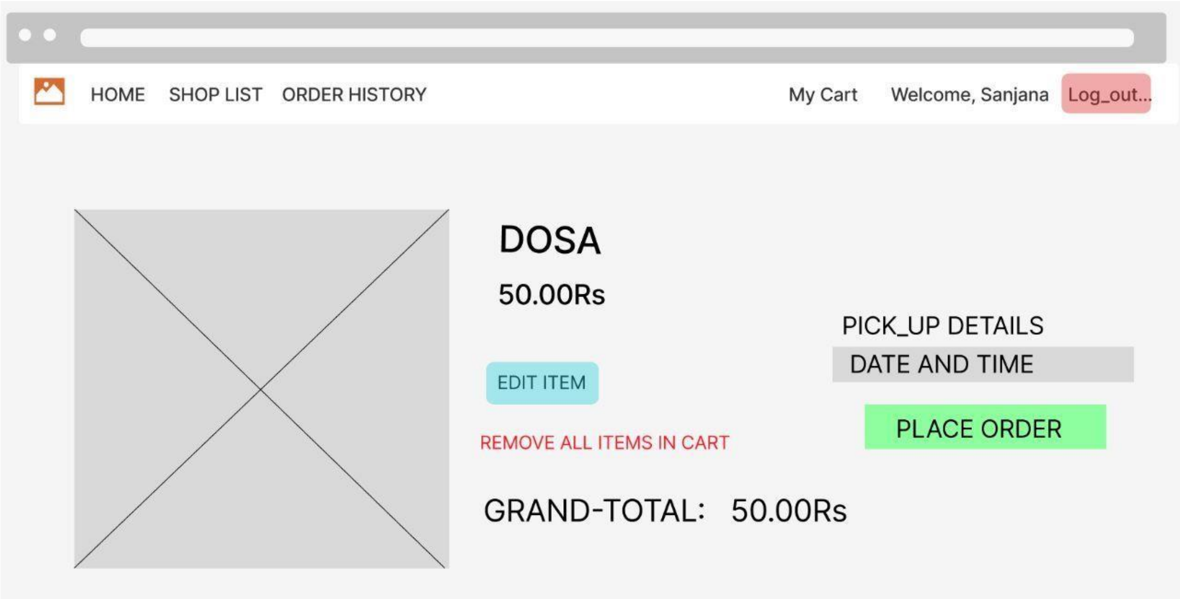


Fig4.11:- Item Selected Page

On clicking on any food items in any of the shops, the complete details of the food item are given, accompanied by the option to request the quantity of the food item to be added to the cart. The additional text field is given to mention if there are any requests to be made to the shops for preparing the food items, which is optional.



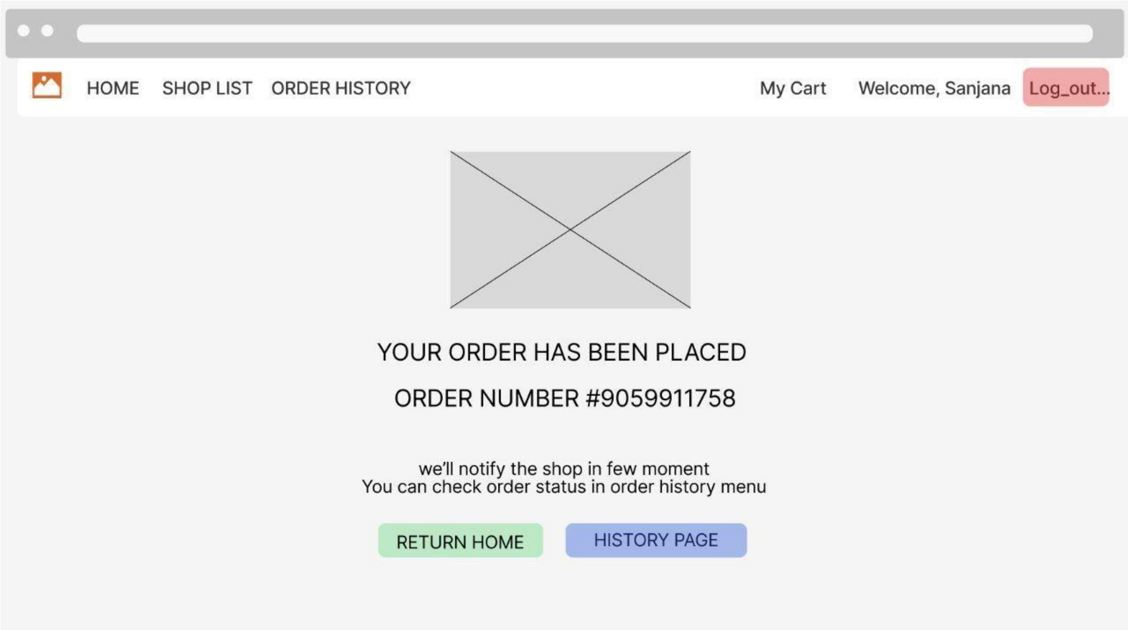
DOSA

50.00Rs

GRAND-TOTAL: 50.00Rs

Fig4.12:- Cart

A cart page is an essential part of an e-commerce website. It is the page where users can pile up what they want to buy from the website and then simply checkout by clicking on the button to place the order. To comprehend what a cart page does it is a normal shopping cart in a store. The information of the customer is given on the right side of the page to verify alongside pick-up details.



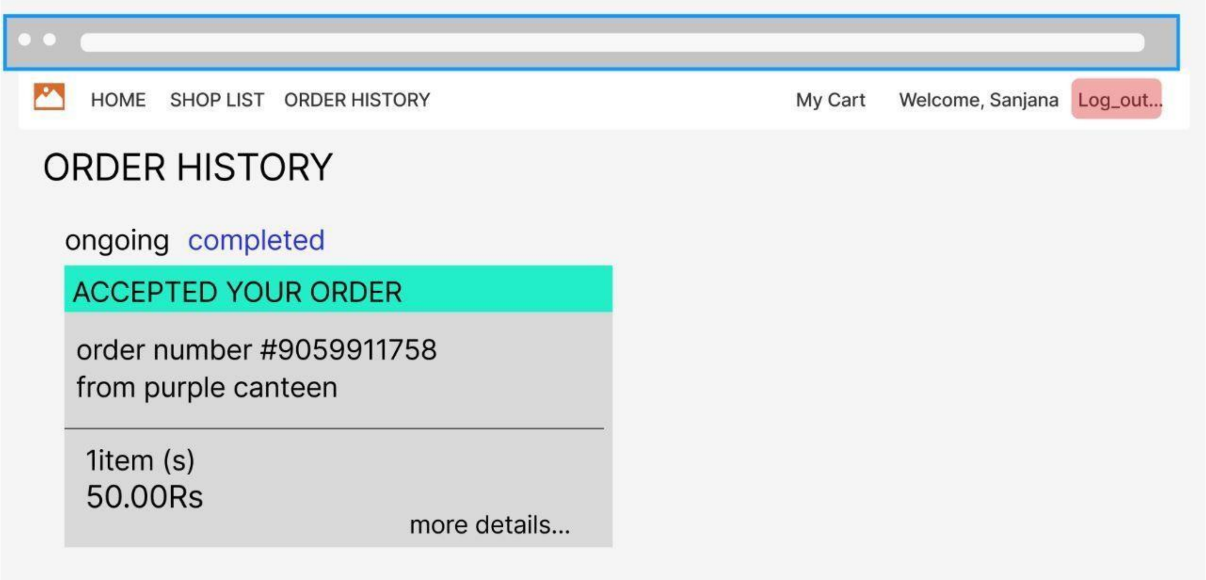
YOUR ORDER HAS BEEN PLACED

ORDER NUMBER #9059911758

we'll notify the shop in few moment
You can check order status in order history menu

Fig4.13:- Page for Order Confirmation

On clicking on place order, the user is taken to the next page where the user sees the message that says that the order has been placed along with the order id number that is auto-generated. This page is mapped to the next page that is to check the order history.



ORDER HISTORY

ongoing completed

ACCEPTED YOUR ORDER

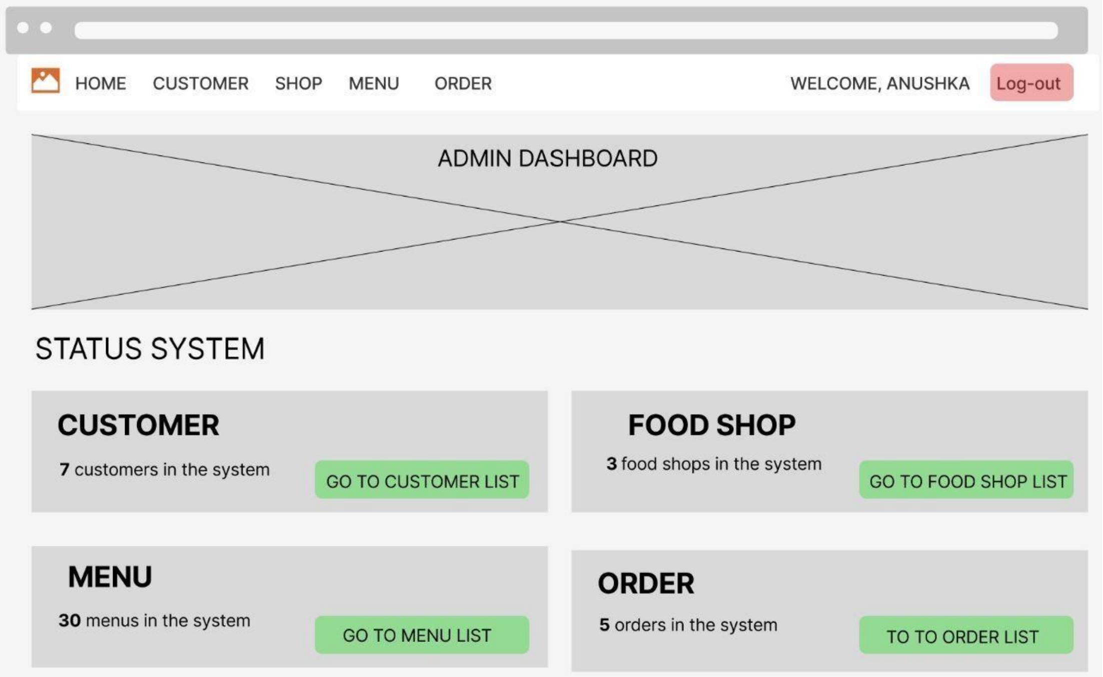
order number #9059911758
from purple canteen

1item (s)
50.00Rs

more details...

Fig14:- Order History Page

A customer's order history is a list of all the orders they've placed with the JSSATE canteen business in the past, beside the on-going orders. It includes the specific items they purchased and how much each cost, along with information about the customer and when they bought the products.



STATUS SYSTEM

CUSTOMER
7 customers in the system

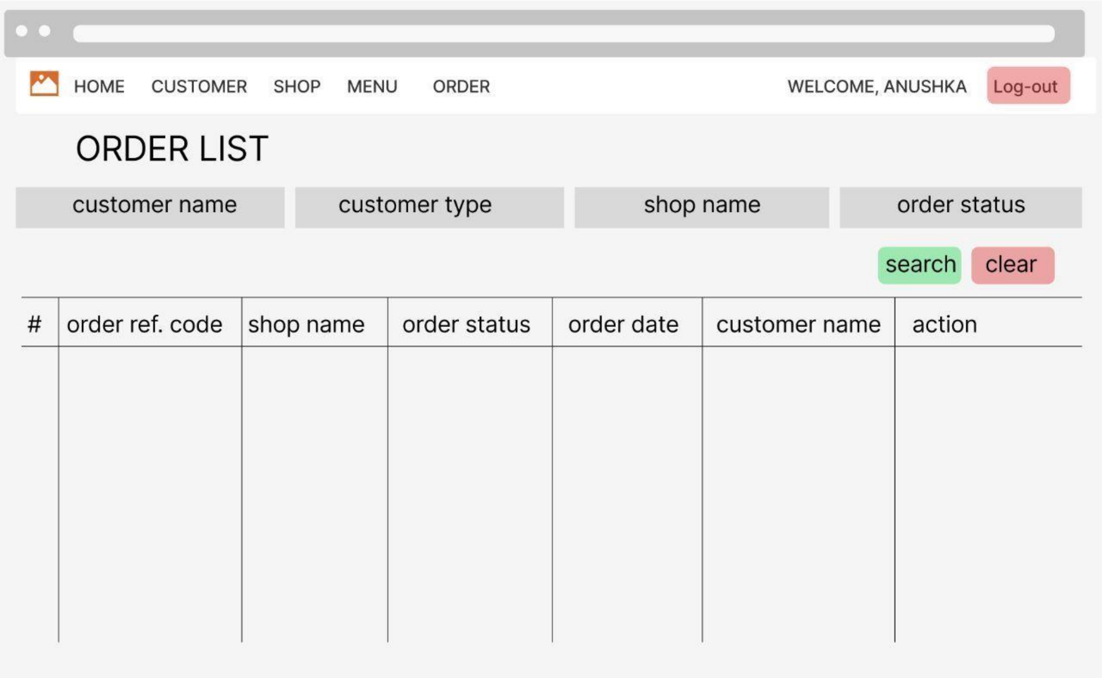
FOOD SHOP
3 food shops in the system

MENU
30 menus in the system

ORDER
5 orders in the system

Fig4.15:- Admin Home Page

The dashboard is the front page of the Administration UI. It provides convenient shortcuts for common management tasks. Review the order updates. The technical status of the platform, details about customer and user profiles, the status of business operations, and many other informational aspects of the system. This page in our application takes us to the system status of customers, shops, menus, and orders.

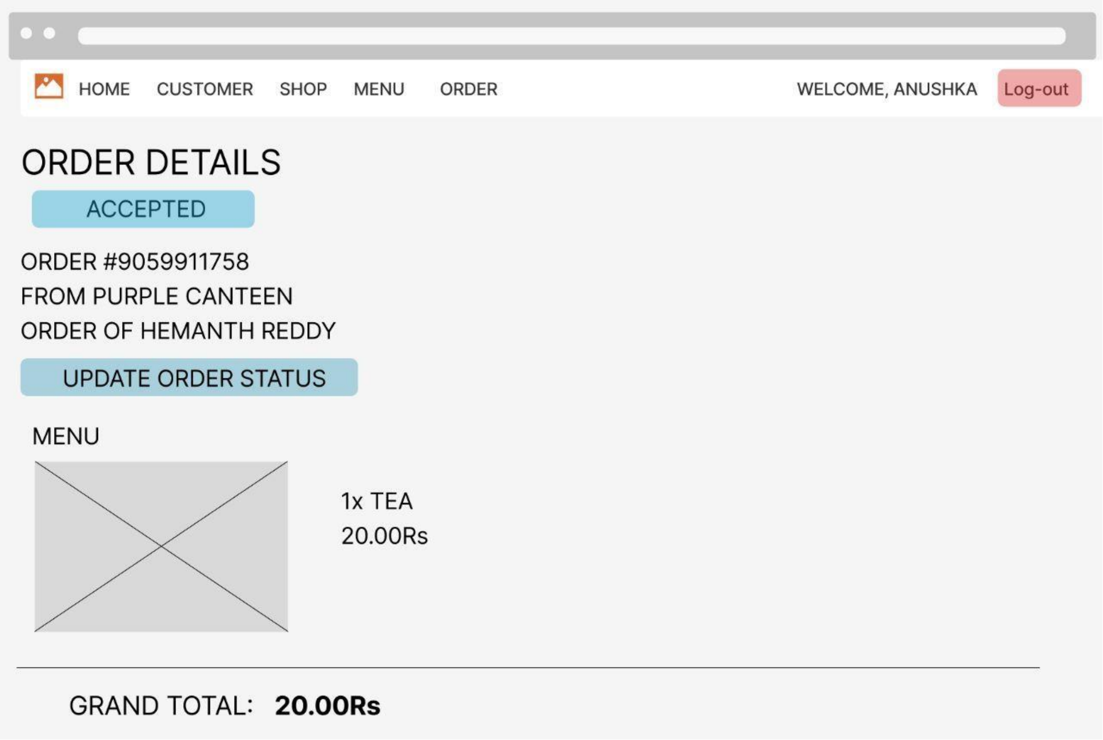


ORDER LIST

#	order ref. code	shop name	order status	order date	customer name	action

Fig4.16:- Order History Page

The orders Admin page will be the one we will visit most so it is important to know our way around. The interface here will provide you with all the information on orders as well as the ability to search your order database and change the order status. To refine the search options, you can select one or a combination of the parameters provided. Click on the order number or on the Name to bring up the full details of that particular order.



ORDER DETAILS

ORDER #9059911758
FROM PURPLE CANTEEN
ORDER OF HEMANTH REDDY

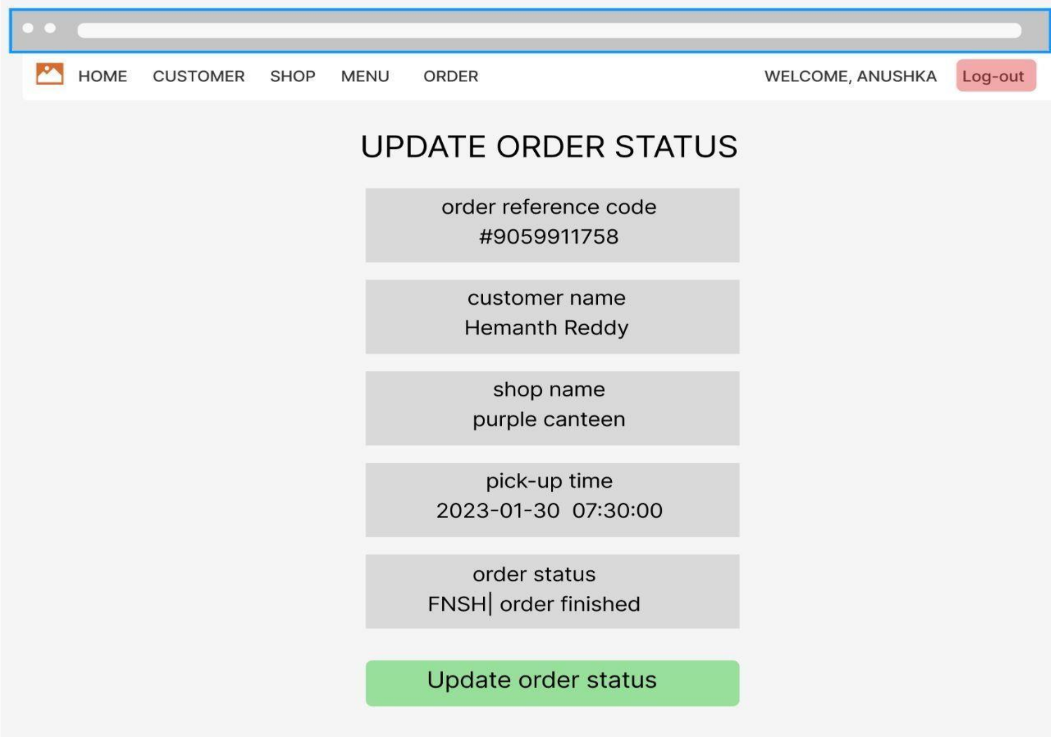
MENU

1x TEA
20.00Rs

GRAND TOTAL: **20.00Rs**

Fig4.17:- Order Details Page

The order details page provides the admin with details of the customer along with time the order was placed, the pick-up time selected and if the order is finished or being prepared.



UPDATE ORDER STATUS

order reference code
#9059911758

customer name
Hemanth Reddy

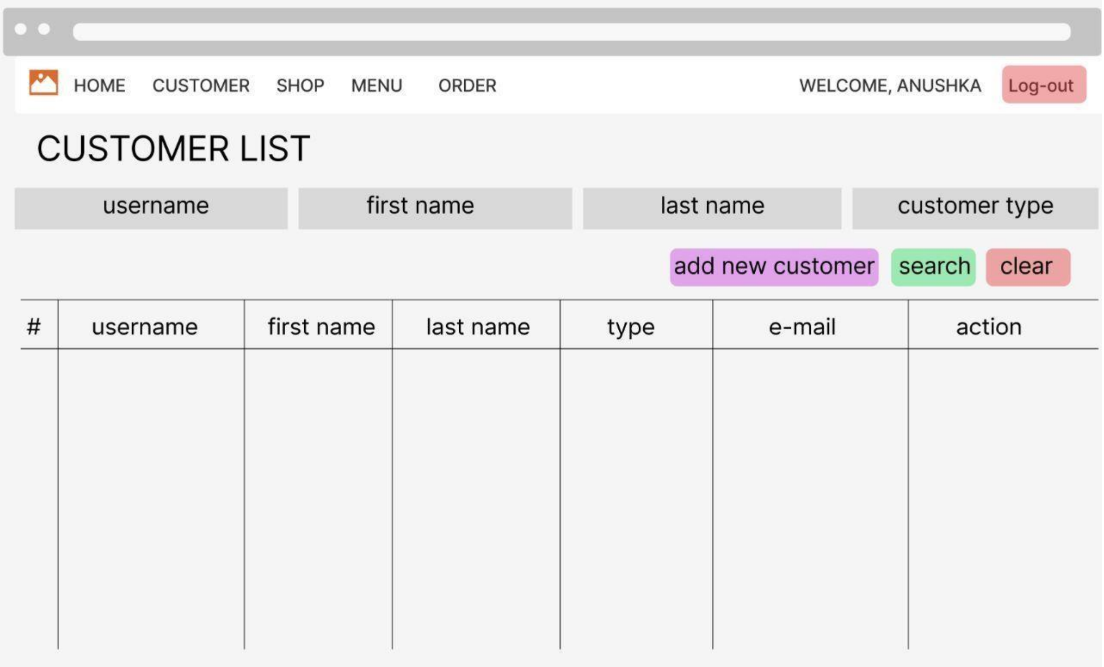
shop name
purple canteen

pick-up time
2023-01-30 07:30:00

order status
FNSH| order finished

Fig4.18:- Update Orders Status Page

This is where we can set the status of the customer's order and the options are finished or accepted. It is a good idea to check the page regularly in case the purchaser didn't make it back to your store after buying. It is possible to change these values in the Admin Order Status page to give a clearer message to the public if they choose to view the status of their order.



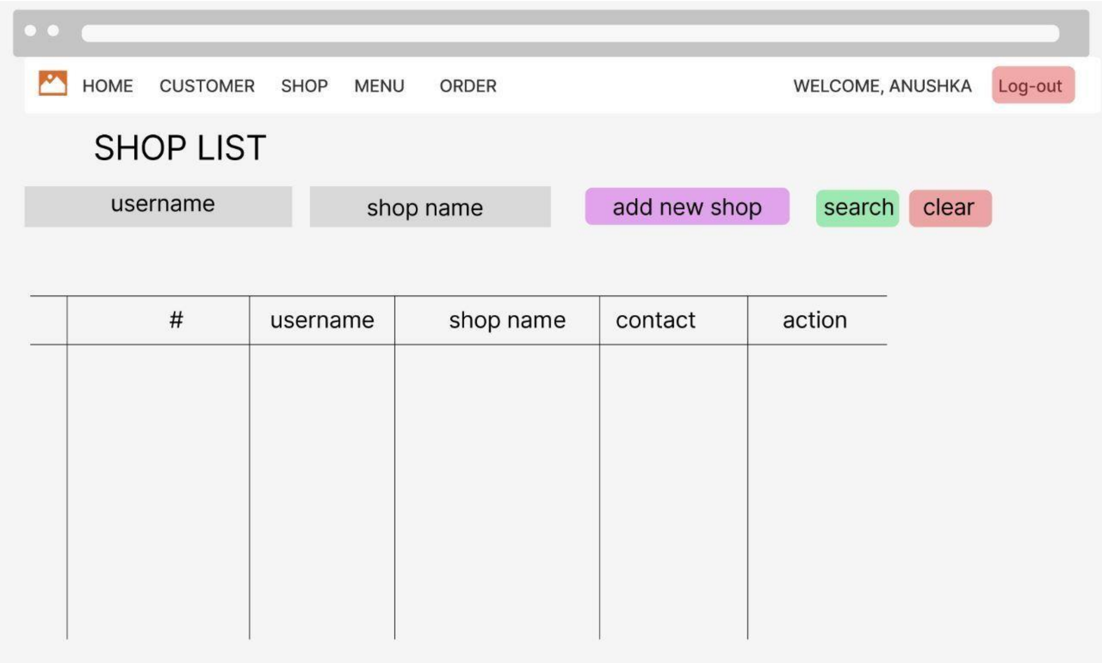
CUSTOMER LIST

username | first name | last name | customer type

#	username	first name	last name	type	e-mail	action

Fig4.19:- Customer List Page for admin

Store administrators can create and manage individual customers on our canteen management website. Doing so can help organize customers into specific groups, which can ease the distribution of price rules. As store administrator, we can view a customer's profile from the admin panel. We can view a variety of customer information, such as their account information or orders. This information can help you develop a better understanding of your customers' interests so that you can provide better customer service.

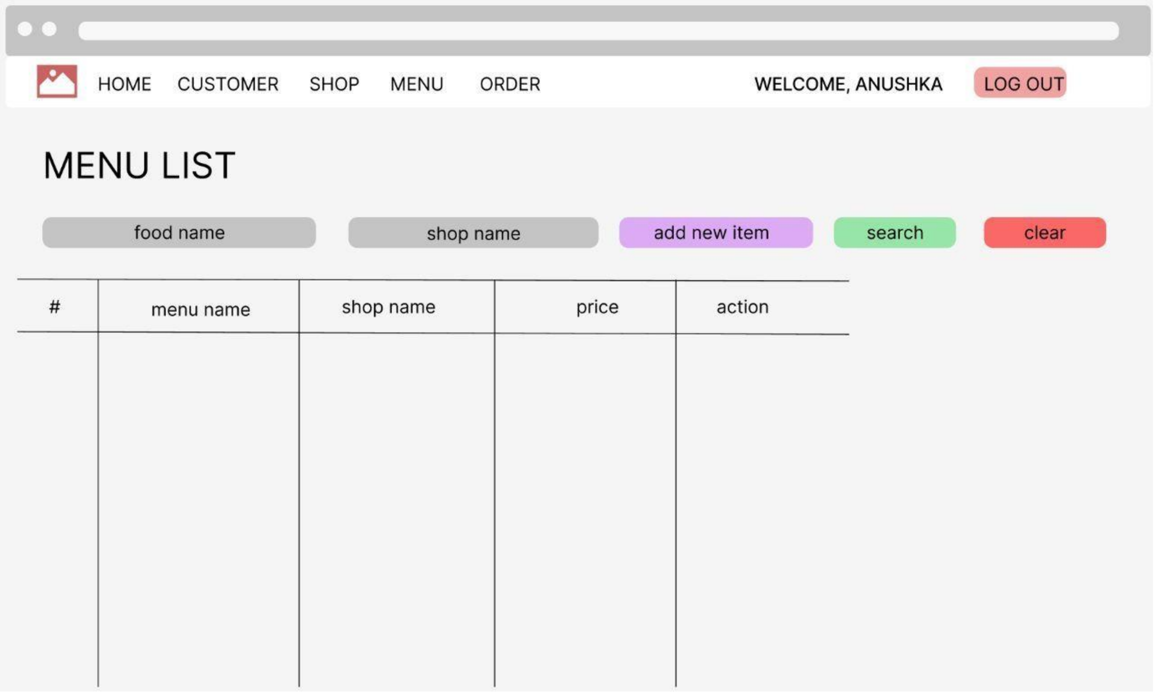


SHOP LIST

	#	username	shop name	contact	action

Fig4.20:- Shop List page for admin

Store administrators can create and manage individual shops on our canteen management website. Doing so can help organize shops in the campus into specific groups, which can ease the distribution of price rules. As store administrator, we can view the shop’s profile from the admin panel. We can view a variety of shop’s information, such as their account information or orders or the food items available in the particular shop. This information can help you develop a better understanding of your customers' interests so that you can provide better customer service.



MENU LIST

#	menu name	shop name	price	action

Fig4.21:- Menu List page for admin

Store administrators can create and manage individual menu items on our canteen management website. Doing so can help organize food items into specific groups, which can ease the distribution of price rules. As store administrator, we can view different food items in different shops from the admin panel. We can view a variety of food item information, such as its availability its price etc and also make these changes. The Store administrators also can add or delete different food items in different stores. This information can help you develop a better understanding of your customers' interests so that you can provide better customer service.

WIREFRAME FLOWCHART:

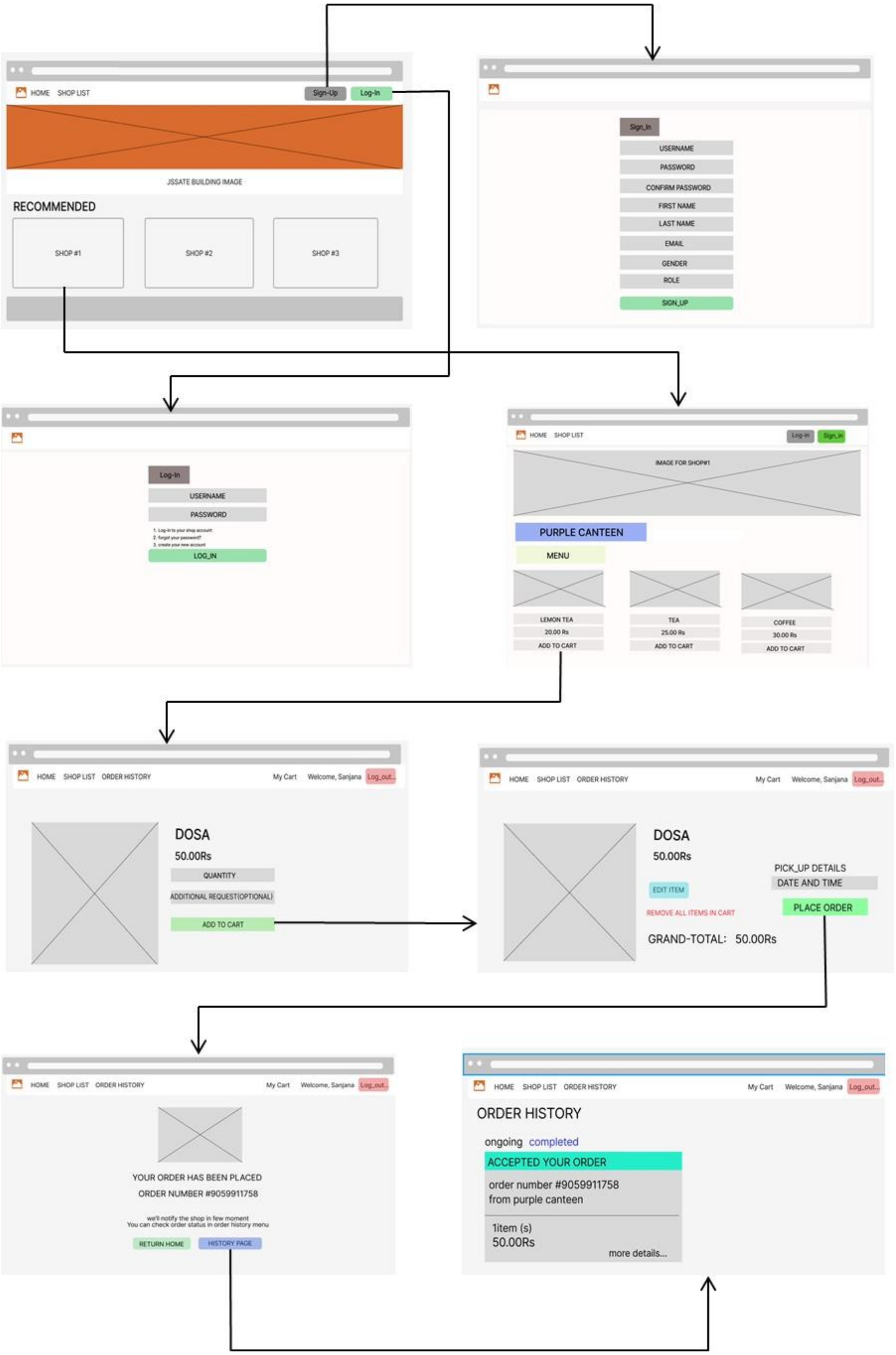
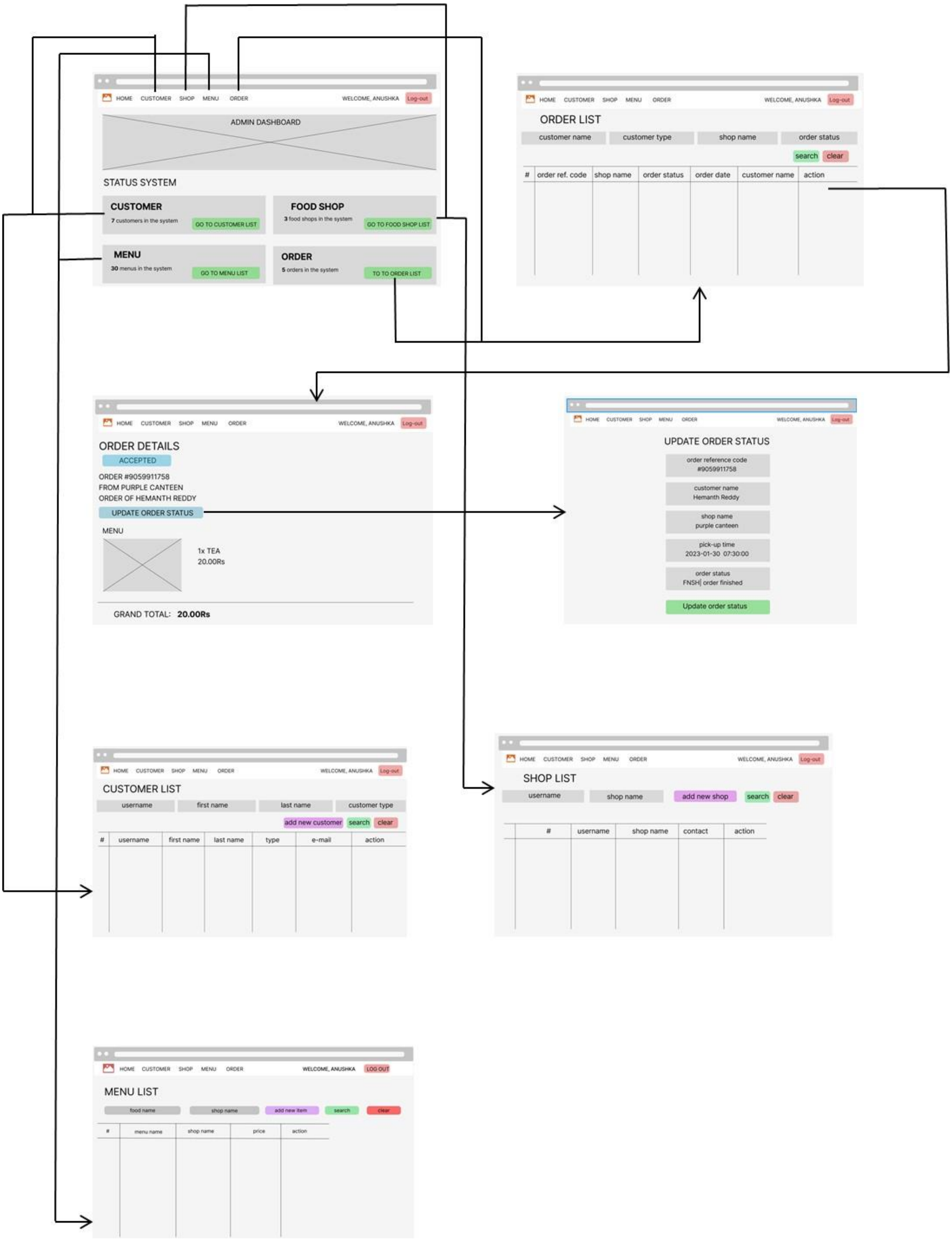


Fig4.22: - Wireframe Flowchart for customer page



ORDER #9059911758
FROM PURPLE CANTEEN
ORDER OF HEMANTH REDDY

1x TEA
20.00Rs

GRAND TOTAL: 20.00Rs

order reference code
#9059911758

customer name
Hemanth Reddy

shop name
purple canteen

pick-up time
2023-01-30 07:30:00

order status
FNSH| order finished

Fig2.22: - Wireframe Flowchart for admin page

CHAPTER 5

PROJECT IMPLEMENTATION

INTRODUCTION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigating of the current system and its constraints on implementation, design of methods to achieve the changeover and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The following codes will ensure the complete implementation of our design and the project.

CREATING TABLES

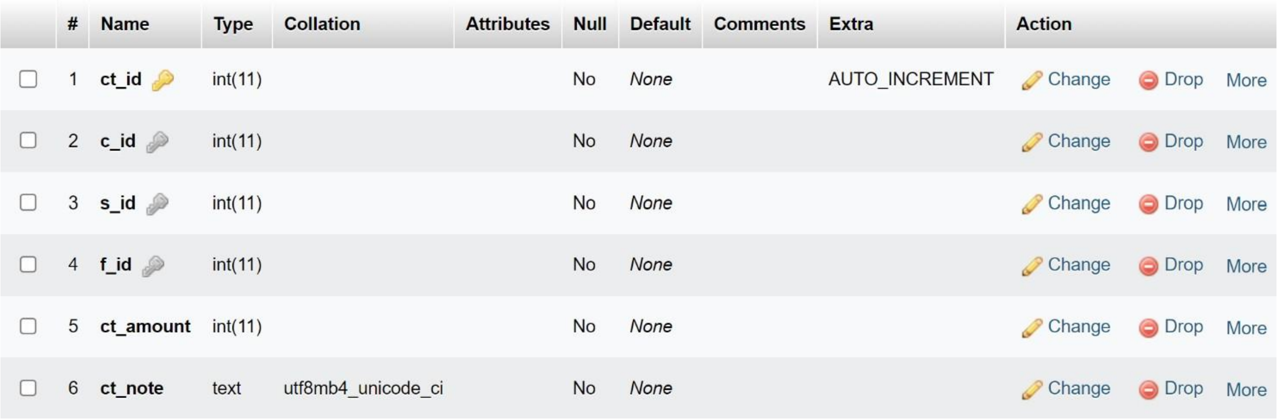
TABLE CART

```
CREATE TABLE `cart` (
  `ct_id` int(11) NOT NULL,
  `c_id` int(11) NOT NULL,
  `s_id` int(11) NOT NULL,
  `f_id` int(11) NOT NULL,
  `ct_amount` int(11) NOT NULL,
  `ct_note` text COLLATE utf8mb4_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

  ALTER TABLE `cart`
  ADD PRIMARY KEY (`ct_id`),
  ADD KEY `fk_ct_c_idx` (`c_id`),
ADD KEY `fk_ct_s_idx` (`s_id`),
 ADD KEY `fk_ct_f_idx` (`f_id`);

ALTER TABLE `cart`
 MODIFY `ct_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=56;

ALTER TABLE `cart`
ADD CONSTRAINT `fk_ct_c` FOREIGN KEY (`c_id`) REFERENCES `customer` (`c_id`)
ON DELETE CASCADE,
ADD CONSTRAINT `fk_ct_f` FOREIGN KEY (`f_id`) REFERENCES `food` (`f_id`) ON
DELETE CASCADE,
ADD CONSTRAINT `fk_ct_s` FOREIGN KEY (`s_id`) REFERENCES `shop` (`s_id`) ON
DELETE CASCADE;
```

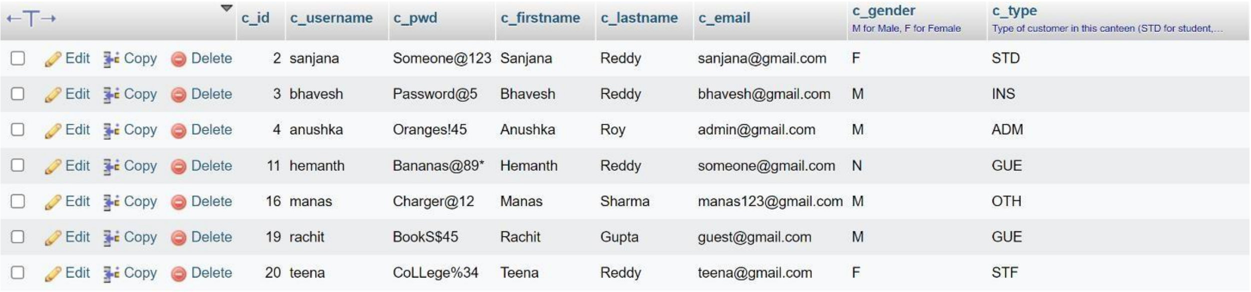


	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	**ct_id**	int(11)			No	*None*		AUTO_INCREMENT	Change	Drop	More
□	2	**c_id**	int(11)			No	*None*			Change	Drop	More
□	3	**s_id**	int(11)			No	*None*			Change	Drop	More
□	4	**f_id**	int(11)			No	*None*			Change	Drop	More
□	5	**ct_amount**	int(11)			No	*None*			Change	Drop	More
□	6	**ct_note**	text	utf8mb4_unicode_ci		No	*None*			Change	Drop	More

TABLE CUSTOMER

```
CREATE TABLE `customer` (
  `c_id` int(11) NOT NULL,
  `c_username` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `c_pwd` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `c_firstname` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `c_lastname` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `c_email` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  `c_gender` varchar(1) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT 'M for Male, F for Female',
  `c_type` varchar(3) COLLATE utf8mb4_unicode_ci NOT NULL COMMENT 'Type of customer in this canteen (STD for student, INS for instructor, STF for staff, GUE for guest, ADM for admin, OTH for other)'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

ALTER TABLE `customer`
  ADD PRIMARY KEY (`c_id`),
  ADD UNIQUE KEY `c_username` (`c_username`),
  ADD UNIQUE KEY `c_email` (`c_email`);
ALTER TABLE `customer`
  MODIFY `c_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;
```

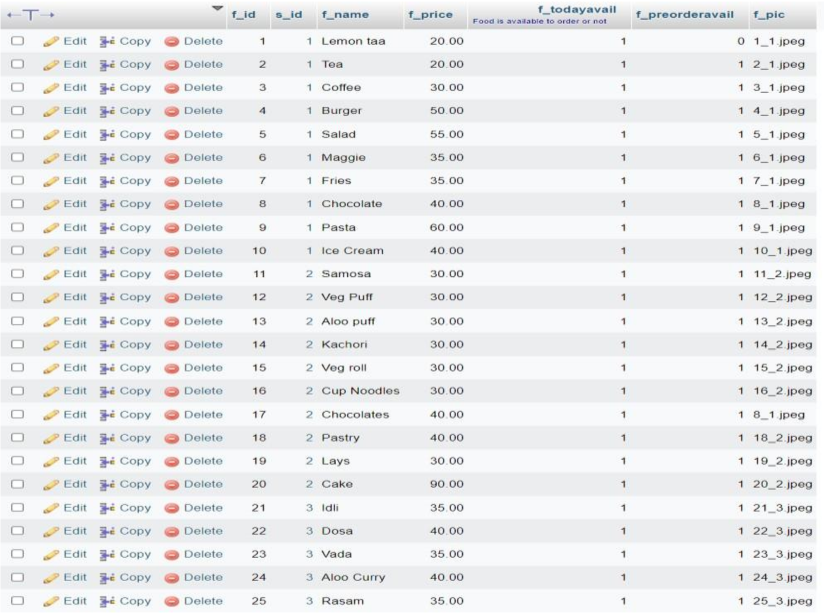


←T→	c_id	c_username	c_pwd	c_firstname	c_lastname	c_email	c_gender M for Male, F for Female	c_type Type of customer in this canteen (STD for student,...
□ Edit Copy Delete	2	sanjana	Someone@123	Sanjana	Reddy	sanjana@gmail.com	F	STD
□ Edit Copy Delete	3	bhavesh	Password@5	Bhavesh	Reddy	bhavesh@gmail.com	M	INS
□ Edit Copy Delete	4	anushka	Oranges!45	Anushka	Roy	admin@gmail.com	M	ADM
□ Edit Copy Delete	11	hemanth	Bananas@89*	Hemanth	Reddy	someone@gmail.com	N	GUE
□ Edit Copy Delete	16	manas	Charger@12	Manas	Sharma	manas123@gmail.com	M	OTH
□ Edit Copy Delete	19	rachit	BookS$45	Rachit	Gupta	guest@gmail.com	M	GUE
□ Edit Copy Delete	20	teena	CoLLege%34	Teena	Reddy	teena@gmail.com	F	STF

TABLE FOOD

```
CREATE TABLE `food` (
  `f_id` int(11) NOT NULL,
  `s_id` int(11) NOT NULL,
  `f_name` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  `f_price` decimal(6,2) NOT NULL,
  `f_todayavail` tinyint(4) NOT NULL DEFAULT 1 COMMENT 'Food is available to order or not',
  `f_preorderavail` tinyint(4) NOT NULL DEFAULT 1,
  `f_pic` text COLLATE utf8mb4_unicode_ci DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

ALTER TABLE `food`
  ADD PRIMARY KEY (`f_id`),
  ADD KEY `food_shop_s_id_idx` (`s_id`);
ALTER TABLE `food`
  MODIFY `f_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=35;
ALTER TABLE `food`
  ADD CONSTRAINT `fk_food_shop_id` FOREIGN KEY (`s_id`) REFERENCES `shop` (`s_id`) ON DELETE CASCADE ON UPDATE NO ACTION;
```

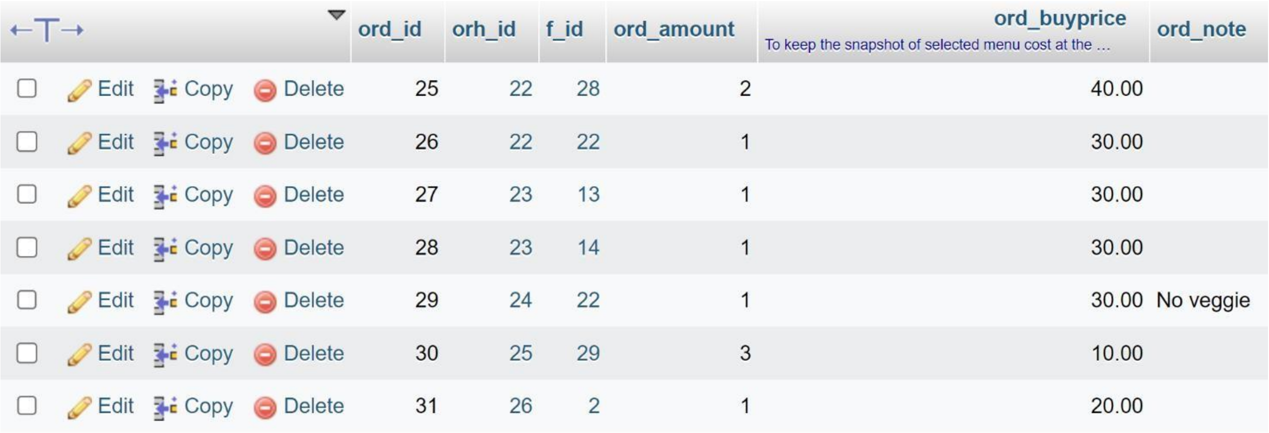


←T→				f_id	s_id	f_name	f_price	f_todayavail Food is available to order or not	f_preorderavail	f_pic
□	Edit	Copy	Delete	1	1	Lemon taa	20.00	1	0	1_1.jpeg
□	Edit	Copy	Delete	2	1	Tea	20.00	1	1	2_1.jpeg
□	Edit	Copy	Delete	3	1	Coffee	30.00	1	1	3_1.jpeg
□	Edit	Copy	Delete	4	1	Burger	50.00	1	1	4_1.jpeg
□	Edit	Copy	Delete	5	1	Salad	55.00	1	1	5_1.jpeg
□	Edit	Copy	Delete	6	1	Maggie	35.00	1	1	6_1.jpeg
□	Edit	Copy	Delete	7	1	Fries	35.00	1	1	7_1.jpeg
□	Edit	Copy	Delete	8	1	Chocolate	40.00	1	1	8_1.jpeg
□	Edit	Copy	Delete	9	1	Pasta	60.00	1	1	9_1.jpeg
□	Edit	Copy	Delete	10	1	Ice Cream	40.00	1	1	10_1.jpeg
□	Edit	Copy	Delete	11	2	Samosa	30.00	1	1	11_2.jpeg
□	Edit	Copy	Delete	12	2	Veg Puff	30.00	1	1	12_2.jpeg
□	Edit	Copy	Delete	13	2	Aloo puff	30.00	1	1	13_2.jpeg
□	Edit	Copy	Delete	14	2	Kachori	30.00	1	1	14_2.jpeg
□	Edit	Copy	Delete	15	2	Veg roll	30.00	1	1	15_2.jpeg
□	Edit	Copy	Delete	16	2	Cup Noodles	30.00	1	1	16_2.jpeg
□	Edit	Copy	Delete	17	2	Chocolates	40.00	1	1	8_1.jpeg
□	Edit	Copy	Delete	18	2	Pastry	40.00	1	1	18_2.jpeg
□	Edit	Copy	Delete	19	2	Lays	30.00	1	1	19_2.jpeg
□	Edit	Copy	Delete	20	2	Cake	90.00	1	1	20_2.jpeg
□	Edit	Copy	Delete	21	3	Idli	35.00	1	1	21_3.jpeg
□	Edit	Copy	Delete	22	3	Dosa	40.00	1	1	22_3.jpeg
□	Edit	Copy	Delete	23	3	Vada	35.00	1	1	23_3.jpeg
□	Edit	Copy	Delete	24	3	Aloo Curry	40.00	1	1	24_3.jpeg
□	Edit	Copy	Delete	25	3	Rasam	35.00	1	1	25_3.jpeg

TABLE ORDER_DETAILS

```
CREATE TABLE `order_detail` (
  `ord_id` int(11) NOT NULL,
  `orh_id` int(11) NOT NULL,
  `f_id` int(11) NOT NULL,
  `ord_amount` int(11) NOT NULL,
  `ord_buyprice` decimal(6,2) NOT NULL COMMENT 'To keep the snapshot of selected menu cost at the time of the purchase.',
  `ord_note` text COLLATE utf8mb4_unicode_ci DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

ALTER TABLE `order_detail`
  ADD PRIMARY KEY (`ord_id`),
  ADD KEY `fk_orh_ord_idx` (`orh_id`),
ADD KEY `fk_f_ord_idx` (`f_id`);
ALTER TABLE `order_detail`
  MODIFY `ord_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=31;
ALTER TABLE `order_detail`
  ADD CONSTRAINT `fk_f_ord` FOREIGN KEY (`f_id`) REFERENCES `food` (`f_id`),
  ADD CONSTRAINT `fk_orh_ord` FOREIGN KEY (`orh_id`) REFERENCES `order_header` (`orh_id`) ON DELETE CASCADE;
```



←T→				ord_id	orh_id	f_id	ord_amount	ord_buyprice To keep the snapshot of selected menu cost at the ...	ord_note
□	Edit	Copy	Delete	25	22	28	2	40.00	
□	Edit	Copy	Delete	26	22	22	1	30.00	
□	Edit	Copy	Delete	27	23	13	1	30.00	
□	Edit	Copy	Delete	28	23	14	1	30.00	
□	Edit	Copy	Delete	29	24	22	1	30.00	No veggie
□	Edit	Copy	Delete	30	25	29	3	10.00	
□	Edit	Copy	Delete	31	26	2	1	20.00	

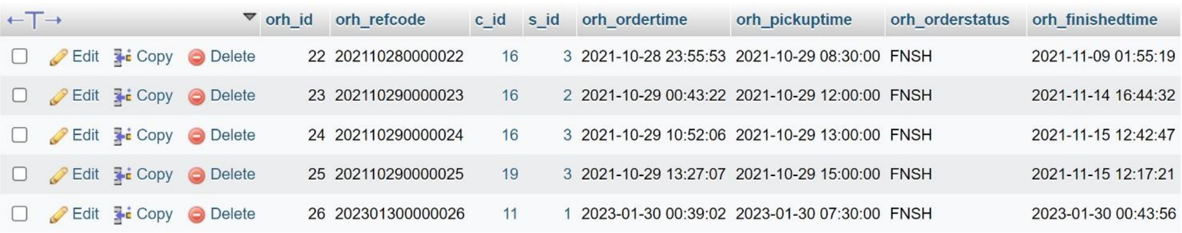
TABLE ORDER_HEADER

```
CREATE TABLE `order_header` (
  `orh_id` int(11) NOT NULL,
  `orh_refcode` varchar(15) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `c_id` int(11) NOT NULL,
  `s_id` int(11) NOT NULL,
  `orh_ordertime` timestamp NOT NULL DEFAULT current_timestamp(),
  `orh_pickuptime` datetime NOT NULL,
  `orh_orderstatus` varchar(10) COLLATE utf8mb4_unicode_ci NOT NULL,
  `orh_finishedtime` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

ALTER TABLE `order_header`
  ADD PRIMARY KEY (`orh_id`),
  ADD KEY `fk_orh_idx` (`c_id`),
  ADD KEY `fk_s_orh_idx` (`s_id`);

ALTER TABLE `order_header`
  MODIFY `orh_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=26;

ALTER TABLE `order_header`
  ADD CONSTRAINT `fk_c_orh` FOREIGN KEY (`c_id`) REFERENCES `customer`
(`c_id`) ON DELETE CASCADE,
  ADD CONSTRAINT `fk_s_orh` FOREIGN KEY (`s_id`) REFERENCES `shop` (`s_id`) ON
DELETE CASCADE;
```

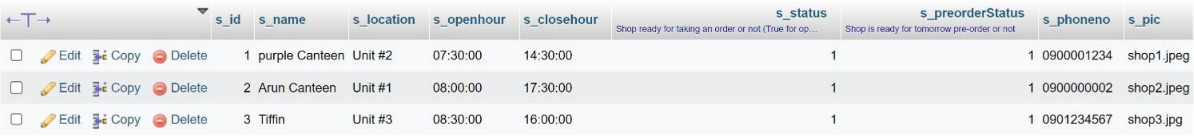


←T→			▼	orh_id	orh_refcode	c_id	s_id	orh_ordertime	orh_pickuptime	orh_orderstatus	orh_finishedtime
□	Edit	Copy	Delete	22	202110280000022	16	3	2021-10-28 23:55:53	2021-10-29 08:30:00	FNSH	2021-11-09 01:55:19
□	Edit	Copy	Delete	23	202110290000023	16	2	2021-10-29 00:43:22	2021-10-29 12:00:00	FNSH	2021-11-14 16:44:32
□	Edit	Copy	Delete	24	202110290000024	16	3	2021-10-29 10:52:06	2021-10-29 13:00:00	FNSH	2021-11-15 12:42:47
□	Edit	Copy	Delete	25	202110290000025	19	3	2021-10-29 13:27:07	2021-10-29 15:00:00	FNSH	2021-11-15 12:17:21
□	Edit	Copy	Delete	26	202301300000026	11	1	2023-01-30 00:39:02	2023-01-30 07:30:00	FNSH	2023-01-30 00:43:56

TABLE SHOP

```
CREATE TABLE `shop` (
  `s_id` int(11) NOT NULL,
  `s_name` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  `s_location` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  `s_openhour` time NOT NULL,
  `s_closehour` time NOT NULL,
  `s_status` tinyint(4) NOT NULL DEFAULT 1 COMMENT 'Shop ready for taking an order or not (True for open, False for close)',
  `s_preorderStatus` tinyint(4) NOT NULL DEFAULT 1 COMMENT 'Shop is ready for tomorrow pre-order or not',
  `s_phoneno` varchar(45) COLLATE utf8mb4_unicode_ci NOT NULL,
  `s_pic` text COLLATE utf8mb4_unicode_ci DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

ALTER TABLE `shop`
  ADD PRIMARY KEY (`s_id`);
ALTER TABLE `shop`
  MODIFY `s_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
```



←T→	s_id	s_name	s_location	s_openhour	s_closehour	s_status Shop ready for taking an order or not (True for op...	s_preorderStatus Shop is ready for tomorrow pre-order or not	s_phoneno	s_pic
□ Edit Copy Delete	1	purple Canteen	Unit #2	07:30:00	14:30:00	1	1	0900001234	shop1.jpeg
□ Edit Copy Delete	2	Arun Canteen	Unit #1	08:00:00	17:30:00	1	1	0900000002	shop2.jpeg
□ Edit Copy Delete	3	Tiffin	Unit #3	08:30:00	16:00:00	1	1	0901234567	shop3.jpg

QUERIES

The below mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update, joins and triggers. A short description of the query is also provided.

INNER JOIN QUERY:

Inner joins combine records from two tables whenever there are matching values in a field common to both tables.

Add_order:

```
$crt_query =
"SELECT ct.f_id,f.f_price,ct.ct_amount,ct.ct_note
FROM cart ct INNER JOIN food f ON ct.f_id = f.f_id
WHERE ct.c_id = {$_SESSION['cid']} AND ct.s_id = {$shop_id};\n";
```

Cust_cart:

```
$cartdetail_query=
"SELECTct.ct_amount,ct.f_id,f_pic,f.f_name,f.f_price,ct.ct_note,f_todayavail,f_preorderavai
l FROM cart ct INNER JOIN food f ON ct.f_id = f.f_id
WHERE ct.c_id = {$_SESSION['cid']}";
```

```
$gt_query =
"SELECT SUM(ct.ct_amount*f.f_price) AS grandtotal FROM cart ct INNER JOIN food f ON
ct.f_id = f.f_id WHERE ct.c_id = {$_SESSION['cid']} GROUP BY ct.c_id";
```

Cust_order_details:

```
$ord_query =
"SELECT f.f_name,f.f_pic,ord.ord_amount,ord.ord_buyprice,ord_note FROM order_detail
ord INNER JOIN food f ON ord.f_id = f.f_id WHERE ord.orh_id = {$orh_id}";
```

Food_item:

```
$query =
 "SELECT f.*,s.s_status,s.s_preorderstatus FROM food f INNER JOIN shop s ON f.s_id =
s.s_id WHERE f.s_id = {$s_id} AND f.f_id = {$f_id} LIMIT 0,1";
```

Admin_food_detail:

```
$query =
"SELECTs.s_id,s.s_name,f.f_name,f.f_price,f.f_todayavail,f.f_preorderavail,f.f_pic
FROM food f INNER JOIN shop s ON f.s_id = s.s_id WHERE f.f_id = $f_id LIMIT 0,1;";
$option_query =
"SELECT DISTINCT c.c_id, c.c_firstname,c.c_lastname
FROM order_header orh INNER JOIN order_detail ord ON orh.orh_id = ord.orh_id
INNER JOIN customer c ON orh.c_id = c.c_id WHERE ord.f_id = {$f_id};";
```

```
$query=
"SELECTorh.orh_id,orh.orh_refcode,orh.orh_ordertime,c.c_firstname,c.c_lastname,orh.orh_
orderstatus,ord.ord_amount
FROM order_header orh INNER JOIN order_detail ord ON orh.orh_id = ord.orh_id
INNER JOIN customer c ON orh.c_id = c.c_id WHERE ord.f_id = {$f_id};";
```

Admin_food_list:

```
 $search_query=
"SELECT   f.f_id,s.s_id,f.f_name,f.f_price,f.f_todayavail,f.f_preorderavail,s.s_name   FROM
food f INNER JOIN shop s ON f.s_id = s.s_id ORDER BY f.f_price DESC,f.s_id ASC;";
```

```
 $search_query=
"SELECT   f.f_id,s.s_id,f.f_name,f.f_price,f.f_todayavail,f.f_preorderavail,s.s_name   FROM
food f INNER JOIN shop s ON f.s_id = s.s_id
```

Admin_order_detail:

```
$ord_query =
"SELECT f.f_id,f.f_name,f.f_pic,ord.ord_amount,ord.ord_buyprice,ord_note FROM
order_detail ord INNER JOIN food f ON ord.f_id = f.f_id WHERE ord.orh_id =
{$orh_id}";
```

Admin_order_list:

```
$option_query =
"SELECT DISTINCT c.c_id, c.c_firstname,c.c_lastname
FROM order_header orh INNER JOIN customer c ON orh.c_id = c.c_id;";
```

```
$option_query =
"SELECT DISTINCT s.s_id, s.s_name
FROM order_header orh INNER JOIN shop s ON orh.s_id = s.s_id;";
```

```
$query=
"SELECTorh.orh_id,orh.orh_refcode,orh.orh_ordertime,c.c_firstname,c.c_lastname,orh.orh_
orderstatus,s.s_name
FROM order_header orh INNER JOIN customer c ON orh.c_id = c.c_id
INNER JOIN shop s ON orh.s_id = s.s_id WHERE c.c_type LIKE '%{$_GET['ut']}%'
AND  orh_orderstatus  LIKE  '%{$_GET['os']}%'".$cid_clause.$sid_clause."  ORDER  BY
orh.orh_ordertime DESC;";
```

```
$query=
"SELECTorh.orh_id,orh.orh_refcode,orh.orh_ordertime,c.c_firstname,c.c_lastname,orh.orh_
orderstatus,s.s_name
FROM order_header orh INNER JOIN customer c ON orh.c_id = c.c_id INNER JOIN shop s
ON orh.s_id = s.s_id ORDER BY orh.orh_ordertime DESC;";
```

```
 $query=
"SELECTorh.orh_refcode,orh.orh_ordertime,c.c_firstname,c.c_lastname,orh.orh_orderstatus,
orh.orh_pickuptime,s.s_name
FROM order_header orh INNER JOIN customer c ON orh.c_id = c.c_id
INNER JOIN shop s ON orh.s_id = s.s_id WHERE orh.orh_id = {$orh_id};";
```

TRIGGERS

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

Syntax:

```
create trigger [trigger_name]
[before | after]
{insert | update | delete}
on [table_name]
[for each row]
[trigger_body]
```

Trigger for CANTEEN SYSTEM

In this trigger we aim to create a backup of the customer ids that will be deleted. Here, we have used BEFORE DELETE which will insert the values of CUSTOMER into a separate table as shown below.

```
CREATE TRIGGER `deleteLog` BEFORE DELETE ON `customer` FOR EACH ROW
INSERT INTO logs VALUES(null, OLD.c_id, 'Deleted', NOW());
```

In this trigger we aim to create a insertion record of the customer ids that will be inserted. Here, we have used AFTER INSERT which will insert the values of CUSTOMER into a separate table as shown below.

```
CREATE TRIGGER `insertLog` AFTER INSERT ON `customer` FOR EACH ROW INSERT
INTO logs VALUES(null, NEW.c_id, 'Inserted', NOW());
```

In this trigger we aim to create a update record of the customer ids that will be updated. Here, we have used AFTER UPDATE which will insert the values of CUSTOMER into a separate table as shown below.

```
CREATE TRIGGER `updateLog` AFTER UPDATE ON `customer` FOR EACH ROW
INSERT INTO logs VALUES(null, NEW.c_id, 'Updated', NOW());
```

PROCEDURES

A stored procedure is a prepared SQL code that we can save, so the code can be reused over and over again. So, if we have an SQL query that we write over and over again, we save it as a stored procedure, and then just call it to execute it. we can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

Syntax:

```
DELIMITER &&
CREATE PROCEDURE procedure_name [[IN | OUT | INOUT] parameter_name datatype[,
parameter datatype])]
BEGIN
   Declaration_section
   Executable_section
END &&
DELIMITER;
```

Procedures for CANTEEN SYSTEM

Database Procedures in canteen system are subroutines that can contain more than one SQL statements that perform a specific task. 'customer_order', `customer_order_history` are the procedures in the canteen management system which are pre-compiled SQL statements into a logical unit. They are stored as an object inside the database server.

```
CREATE PROCEDURE `customer_order` (IN `order_id` INT(11))  BEGIN
         SELECT orh.orh_refcode AS reference_code, CONCAT(c.c_firstname,' ',c.c_lastname) AS customer_name, s.s_name AS shop_name,f.f_name AS food_name,ord.ord_buyprice AS buy_price, ord.ord_amount AS amount ,ord.ord_note AS order_note, orh.orh_ordertime AS order_time , orh.orh_pickuptime AS pickup_time
   FROM order_header orh
   INNER JOIN order_detail ord ON orh.orh_id = ord.orh_id
   INNER JOIN food f ON f.f_id = ord.f_id
   INNER JOIN customer c ON orh.c_id = c.c_id
INNER JOIN shop s ON orh.s_id = s.s_id
   WHERE orh.orh_id = order_id;
END$$

CREATE PROCEDURE `customer_order_history` (IN `customer_id` INT(11)) BEGIN
         SELECT orh.orh_refcode AS reference_code, CONCAT(c.c_firstname,'
',c.c_lastname) AS customer_name,
   s.s_name AS shop_name, orh.orh_ordertime AS order_time, orh.orh_pickuptime AS
pickup_time,
   p.p_amount AS order_cost, orh.orh_orderstatus AS order_status
   FROM order_header orh INNER JOIN customer c ON orh.c_id = c.c_id
   INNER JOIN shop s ON orh.s_id = s.s_id
   WHERE c.c_id = customer_id;
END$$
```

HTML CODE

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <?php session_start(); include("conn_db.php"); include('head.php');?>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="css/main.css" rel="stylesheet">
    <style>
        html {
            height: 100%;
        }
    </style>
    <title>Welcome | JSS Cafeteria</title>
</head>
<body class="d-flex flex-column h-100">

    <?php include('nav_header.php')?>

    <div class="position-relative d-flex text-center text-white promo-banner-bg py-3">
        <div class="p-lg-2 mx-auto my-5">
            <h1 class="display-5 fw-normal">Welcome to JSS Cafeteria</h1>
            <p class="lead fw-normal">Food ordering system of JSSATE-B Campus Canteen</p>
            <span class="xsmall-font text-muted"></span>
        </div>
    </div>
    <div class="container p-5" id="recommended-shop">
        <h2 class="border-bottom pb-2"><i class="bi bi-shop align-top"></i> Recommended</h2>
        <!-- GRID SHOP SELECTION -->
        <div class="row row-cols-1 row-cols-lg-3 align-items-stretch g-4 py-3">
```

```
      <?php
      $query="SELECT s_id,s_name,s_openhour,s_closehour,s_status,s_preorderstatus,s_pic FROM shop
      WHERE (s_preorderstatus = 1) OR (s_preorderstatus = 0 AND (CURTIME() BETWEEN s_openhour AND s_closehour));";
      $result = $mysqli -> query($query);
      if($result -> num_rows > 0){
      while($row = $result -> fetch_array()){
    ?>
      <!-- GRID EACH SHOP -->
      <div class="col">
        <a href="<?php echo "shop_menu.php?s_id=".$row["s_id"]?>" class="text-decoration-none text-dark">
          <div class="card rounded-25">
            <img <?php
              if(is_null($row["s_pic"])){echo "src='img/default.png'";}
              else{echo "src=\"img/{$row['s_pic']}\"";}
            ?> style="width:100%; height:175px; object-fit:cover;"
              class="card-img-top rounded-25 img-fluid" alt="<?php echo $row["s_name"]?>">
            <div class="card-body">
              <h4 name="shop-name" class="card-title"><?php echo $row["s_name"]?></h4>
              <p class="card-subtitle">
              <?php
                  $now = date('H:i:s');
                  if((($now < $row["s_openhour"])||($now > $row["s_closehour"]))||($row["s_status"]==0)){
                ?>
                <span class="badge rounded-pill bg-danger">Closed</span>
                <?php }else{ ?>
                <span class="badge rounded-pill bg-success">Open</span>
                <?php }
                  if($row["s_preorderstatus"]==1){
                ?>
                <span class="badge rounded-pill bg-success">Pre-order avaliable</span>
                <?php }else{ ?>
                <span class="badge rounded-pill bg-danger">Pre-order
Unavaliable</span>
                <?php } ?>
              </p>
              <?php
                $open = explode(":",$row["s_openhour"]);
                $close = explode(":",$row["s_closehour"]);
              ?>
              <p class="card-text my-2">Open hours:
                <?php echo $open[0].":".$open[1]." - ".$close[0].":".$close[1];?></p>
              <div class="text-end">
                <a href="<?php echo "shop_menu.php?s_id=".$row["s_id"]?>"
                  class="btn btn-sm btn-outline-dark">Go to shop</a>
              </div>
            </div>
          </div>
        </a>
      </div>
      <!-- END GRID EACH SHOP -->
      <?php }
    }else{
      ?>
      <div class="row row-cols-1 w-100">
        <div class="col mt-4 pt-3 px-3 bg-danger text-white rounded text-center">
          <svg xmlns="http://www.w3.org/2000/svg" width="32" height="32"
fill="currentColor"
            class="bi bi-x-circle" viewBox="0 0 16 16">
            <path d="M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0
16z" />
            <path
              d="M4.646 4.646a.5.5 0 0 1 .708 0L8 7.293l2.646-2.647a.5.5 0 0 1
.708.708L8.707 8l2.647 2.646a.5.5 0 0 1-.708.708L8 8.707l-2.646 2.647a.5.5 0 0 1-.708-
.708L7.293 8 4.646 5.354a.5.5 0 0 1 0-.708z" />
```

```
                    </svg>
                    <p class="ms-2 mt-2">No shop currently avaliable to order.</p>
                </div>
            </div>
            <?php
        }
        $result -> free_result();
        ?>
        </div>
        <!-- END GRID SHOP SELECTION -->
    </div>
    <footer
        class="footer d-flex flex-wrap justify-content-between align-items-center px-5 py-3 mt-auto bg-secondary text-light">
        <span class="smaller-font">&copy; 2023 JSSATE-B FoodGroup <br /><span class="xsmall-font">G.Sanjana Reddy, Anushka Roy</span></span>
        <span class="xsmall-font"> +91-9953411751 <br /><span class="xsmall-font">+91-9465911750</span></span>
    </footer>
</body>
</html>
```

CODE TO ESTABLISH CONNECTION BETWEEN FRONTEND AND DATABASE

Conn_db.php:

```
<?php
    $mysqli = new mysqli("localhost","root","","jss_canteen");
    if($mysqli -> connect_errno){
        header("location: db_error.php");
        exit(1);
    }
    define('SITE_ROOT',realpath(dirname(_FILE_)));
    date_default_timezone_set('Asia/Kolkata');
?>
```

CHAPTER 6

SYSTEM TESTING

INTRODUCTION

System Testing is a level of testing that validates the complete and fully integrated software product which in our case is **canteen management system.** The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is defined as a series of different tests whose sole purpose is to exercise the full computer-based system.

There are mainly two widely used methods for software testing, one is White box testing which uses internal coding to design test cases and another is black box testing which uses GUI or user perspective to develop test cases.

- White box testing
- Black box testing

System testing falls under Black box testing as it includes testing of the external working of the software. Testing follows user's perspective to identify minor defects.

System Testing includes the following steps.

1. Verification of input functions of the application to test whether it is producing the expected output or not.
2. Testing of integrated software by including external peripherals to check the interaction of various components with each other.
3. Testing of the whole system for End-to-End testing.
4. Behavior testing of the application via a user's experience.

TYPES OF TESTING

Unit testing

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other.

Unit testing is a software development process that involves a synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Unit testing aims to eliminate construction errors before code is promoted to additional testing; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development process.

Sl.no	Test Case	Expected Result	Test Result
1	Enter valid username and password and click login (both admin and user)	Software should display the dashboard.	Successful
2	Enter invalid username or password and click login (both admin and user)	Software should display invalid username or password.	successful

Table 6.1: Unit testing case for user validation

Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

Sign-up page

Sl. No.	Test Case	Excepted Result	Test Result
1	On clicking sign-up	i. The user needs to fill all the details that are compulsory. If the fields are left empty, a message should pop up saying "The particular field not filled"	Successful
		ii. The user needs to fill valid email address and password in the specified format. Else, invalid password message should be popped"	Successful

Login page

Sl. No.	Test Case	Excepted Result	Test Result
1	On clicking submit	i. The user needs to fill all the details that are compulsory that are username and password correctly. If, not, invalid username or	Successful

		password message should be popped.	

Cart page

Sl. No.	Test Case	Excepted Result	Test Result
1	On clicking Add to Cart	The user needs to select a specified date and time for pick-up, else the place order button should not proceed to the next page	Successful

System Testing

System testing tests a completely integrated system to verify that the system meets its requirements. For example, a system test might involve testing a logon interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then logoff.

Menu List page

Sl. No.	**Test Case**	**Excepted Result**	**Test Result**
1	On clicking Search	On selecting username and shop list parameters, the respective values should be selected which clicked on search	Successful

Order List page

Sl. No.	**Test Case**	**Excepted Result**	**Test Result**
1	On clicking Search	On selecting customer name, customer type and order status parameters, the respective values should be selected	Successful

		which clicked on search	

Shop List page

Sl. No.	**Test Case**	**Excepted Result**	**Test Result**
1	On clicking Search	On selecting username and shop list parameters, the respective values should be selected which clicked on search	Successful

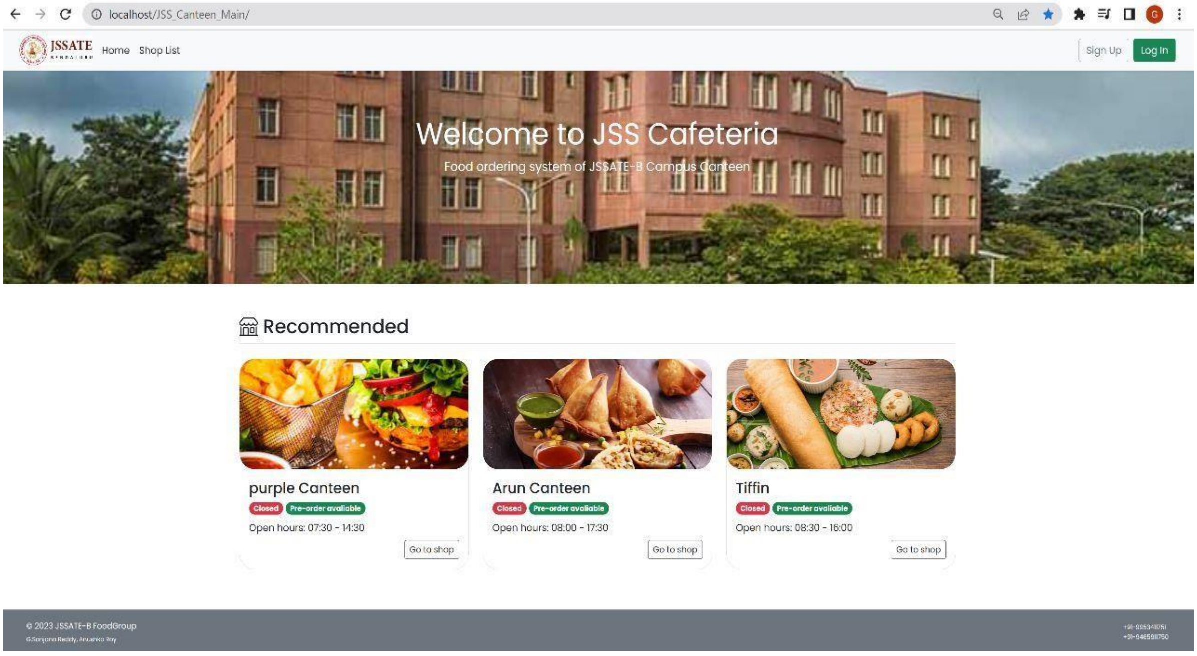
Customer List page

Sl. No.	**Test Case**	**Excepted Result**	**Test Result**
1	On clicking Search	On selecting customer name and customer type parameters, the respective values should be selected which clicked on search	Successful

CHAPTER 7

RESULTS AND DISCUSSIONS

Home page of CANTEEN DATABASE MANAGEMENT SYSTEM



Welcome to JSS Cafeteria

Food ordering system of JSSATE-B Campus Canteen

purple Canteen
Open hours: 07:30 - 14:30

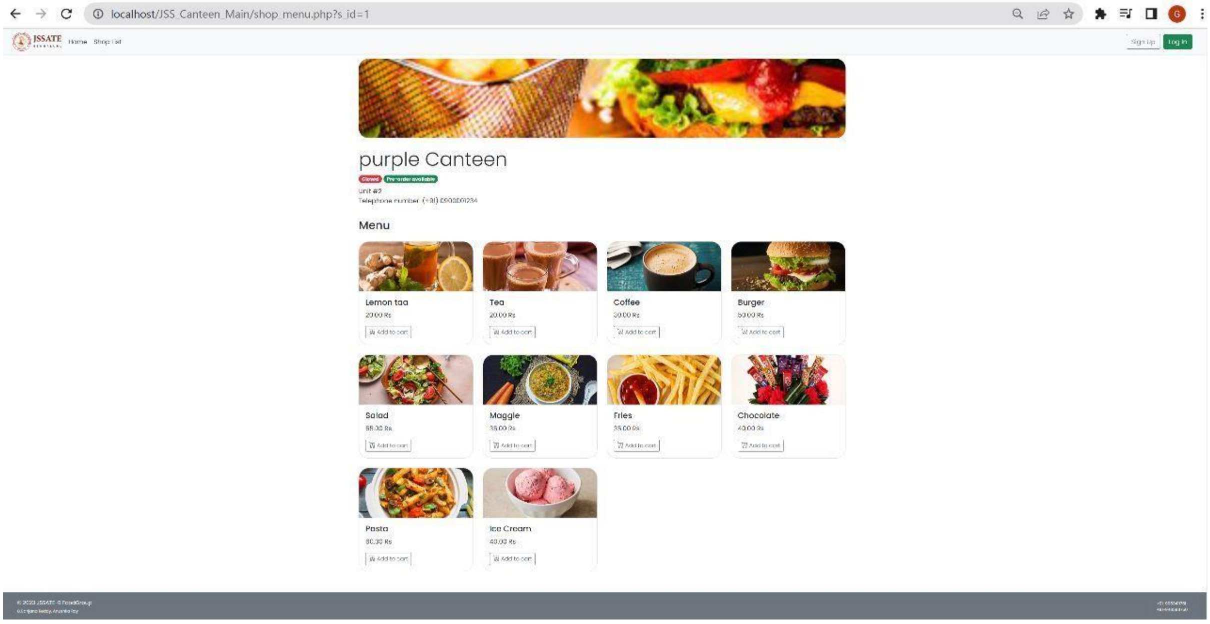
Arun Canteen
Open hours: 08:00 - 17:30

Tiffin
Open hours: 08:30 - 16:00

© 2023 JSSATE-B FoodGroup
G.Sanjana Reddy, Anushka Roy

+91-9353411751
+91-9465911750

Fig7.1:- Home Page of Canteen Database Management



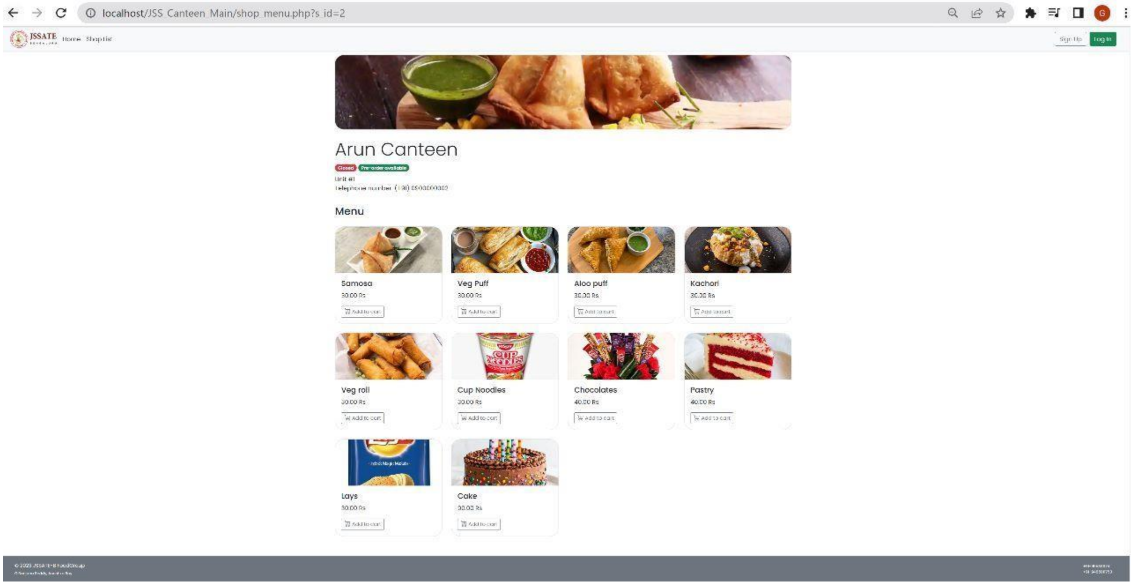
purple Canteen

Unit #2
Telephone number: (+91) 0900001234

Lemon taa 20.00 Rs
Tea 20.00 Rs
Coffee 30.00 Rs
Burger 50.00 Rs
Salad 55.00 Rs
Maggie 35.00 Rs
Fries 35.00 Rs
Chocolate 40.00 Rs
Pasta 80.00 Rs
Ice Cream 40.00 Rs

© 2023 JSSATE-B FoodGroup
G.Sanjana Reddy, Anushka Roy

Fig7.2: - Shop#1



localhost/JSS_Canteen_Main/shop_menu.php?s_id=2

Arun Canteen

Unit #1
Telephone number: (+91) 0900000002

Menu

Samosa
30.00 Rs

Veg Puff
30.00 Rs

Aloo puff
30.00 Rs

Kachori
30.00 Rs

Veg roll
30.00 Rs

Cup Noodles
30.00 Rs

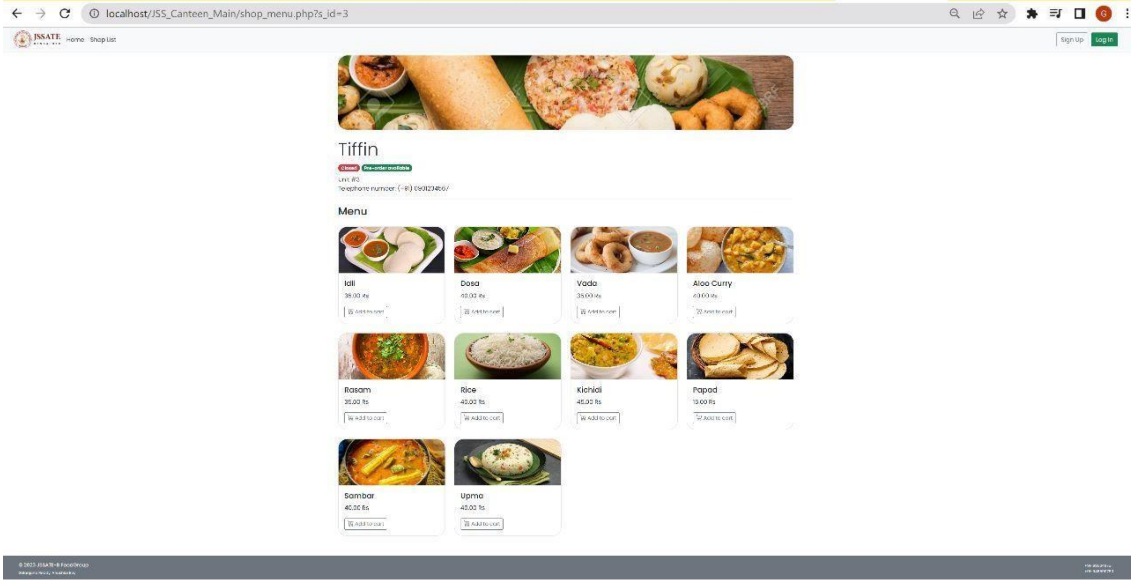
Chocolates
40.00 Rs

Pastry
40.00 Rs

Lays
30.00 Rs

Cake
30.00 Rs

Fig7.3:- Shop#2



localhost/JSS_Canteen_Main/shop_menu.php?s_id=3

Tiffin

Unit #3
Telephone number: (+91) 0901234567

Menu

Idli
35.00 Rs

Dosa
40.00 Rs

Vada
35.00 Rs

Aloo Curry
40.00 Rs

Rasam
35.00 Rs

Rice
40.00 Rs

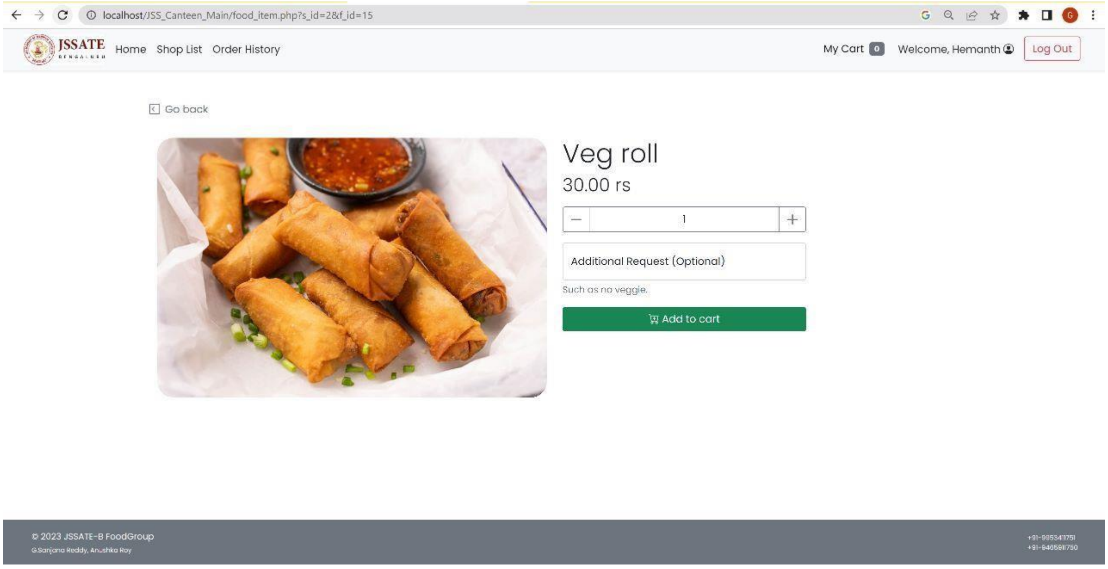
Kichidi
45.00 Rs

Papad
15.00 Rs

Sambar
40.00 Rs

Upma
40.00 Rs

Fig7.4: - Shop#3



localhost/JSS_Canteen_Main/food_item.php?s_id=2&f_id=15

Veg roll
30.00 rs

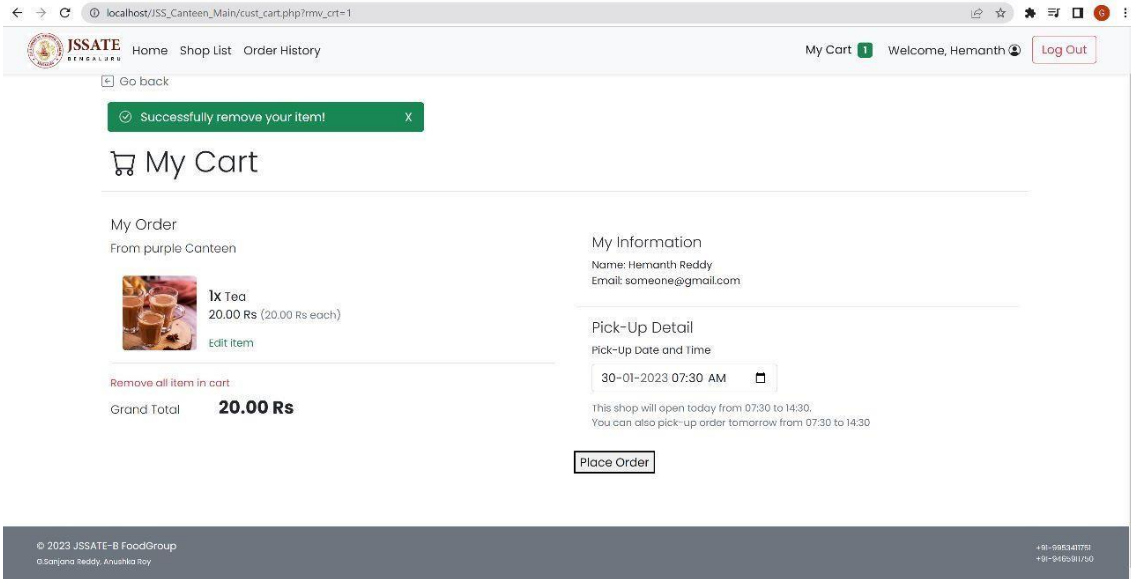
Additional Request (Optional)

Such as no veggie.

© 2023 JSSATE-B FoodGroup
G.Sanjana Reddy, Anushka Roy

+91-9853411751
+91-9465911750

Fig7.5:- Items to add to cart



localhost/JSS_Canteen_Main/cust_cart.php?rmv_crt=1

Successfully remove your item! X

My Cart

My Order

From purple Canteen

1x Tea

20.00 Rs (20.00 Rs each)

Grand Total 20.00 Rs

My Information

Name: Hemanth Reddy

Email: someone@gmail.com

Pick-Up Detail

Pick-Up Date and Time

30-01-2023 07:30 AM

This shop will open today from 07:30 to 14:30.

You can also pick-up order tomorrow from 07:30 to 14:30

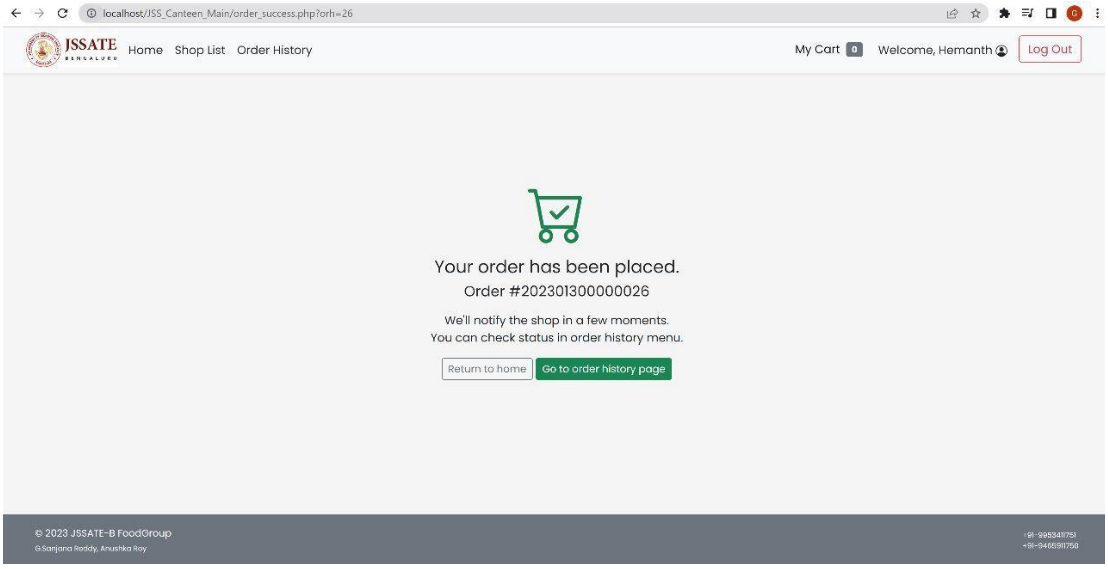
© 2023 JSSATE-B FoodGroup

G.Sanjana Reddy, Anushka Roy

+91-9953411751

+91-9465911750

Fig7.6:- Items added to cart



localhost/JSS_Canteen_Main/order_success.php?orh=26

Your order has been placed.

Order #202301300000026

We'll notify the shop in a few moments.

You can check status in order history menu.

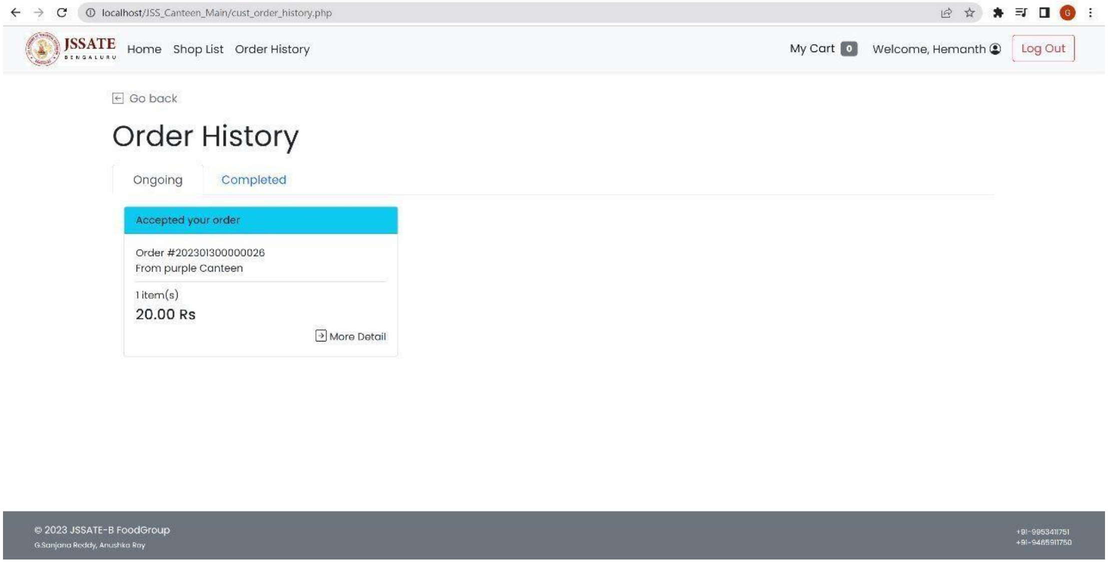
© 2023 JSSATE-B FoodGroup

G.Sanjana Reddy, Anushka Roy

+91-9953411751

+91-9465911750

Fig7.7:- Ordered placed page



localhost/JSS_Canteen_Main/cust_order_history.php

Order History

Accepted your order

Order #202301300000026

From purple Canteen

1 item(s)

20.00 Rs

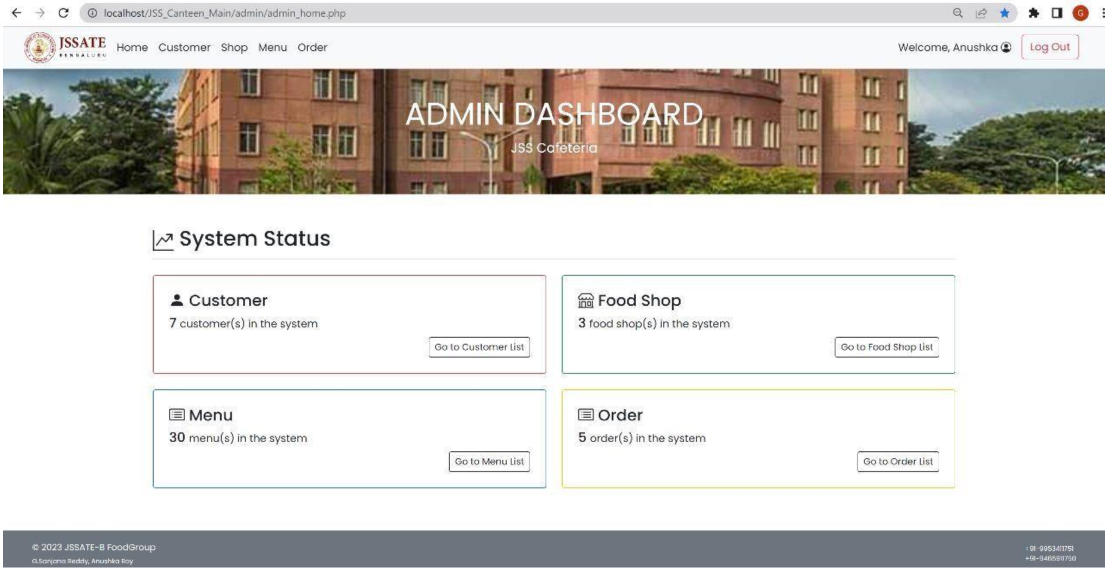
© 2023 JSSATE-B FoodGroup

G.Sanjana Reddy, Anushka Roy

+91-9953411751

+91-9465911750

Fig7.8:- Order history page



localhost/JSS_Canteen_Main/admin/admin_home.php

ADMIN DASHBOARD

JSS Cafeteria

System Status

Customer

7 customer(s) in the system

Food Shop

3 food shop(s) in the system

Menu

30 menu(s) in the system

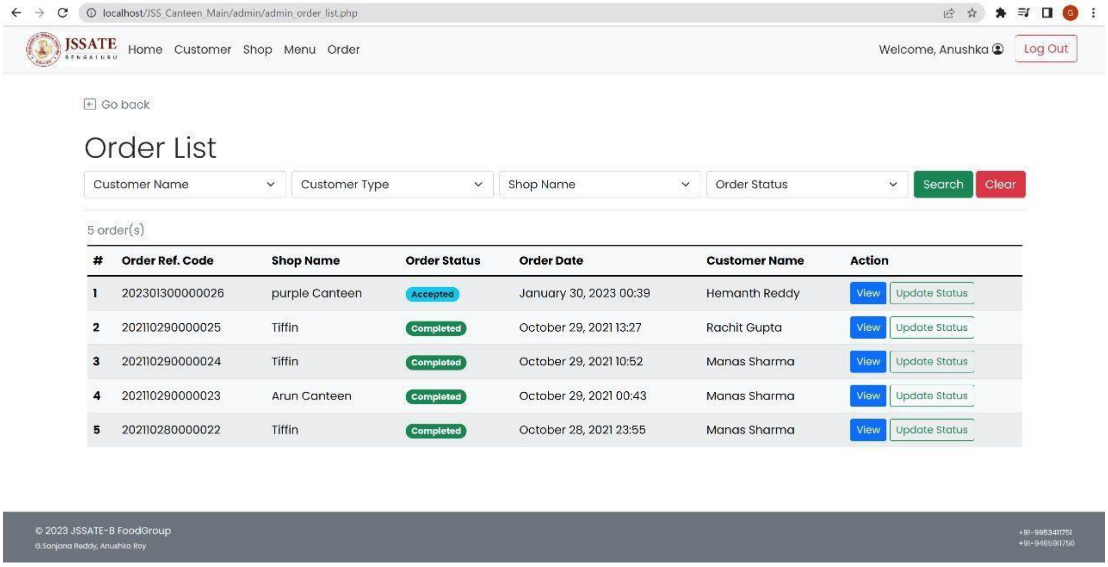
Order

5 order(s) in the system

© 2023 JSSATE-B FoodGroup
G.Sanjana Reddy, Anushka Roy

+91-9953411751
+91-9465911750

Fig7.9:- Admin home page



localhost/JSS_Canteen_Main/admin/admin_order_list.php

Order List

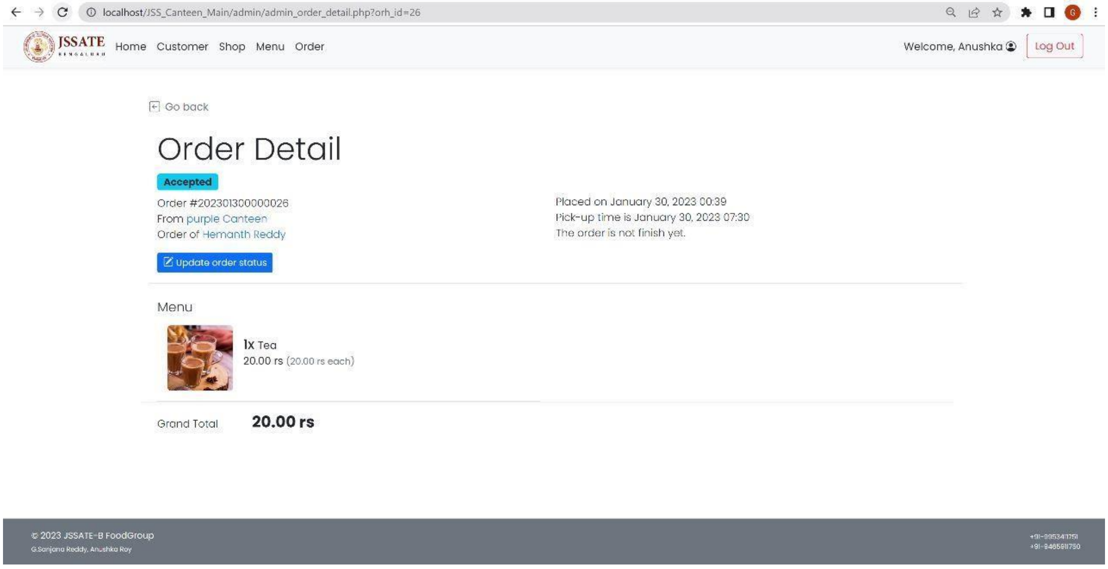
5 order(s)

#	Order Ref. Code	Shop Name	Order Status	Order Date	Customer Name	Action
1	202301300000026	purple Canteen	Accepted	January 30, 2023 00:39	Hemanth Reddy	View Update Status
2	202110290000025	Tiffin	Completed	October 29, 2021 13:27	Rachit Gupta	View Update Status
3	202110290000024	Tiffin	Completed	October 29, 2021 10:52	Manas Sharma	View Update Status
4	202110290000023	Arun Canteen	Completed	October 29, 2021 00:43	Manas Sharma	View Update Status
5	202110280000022	Tiffin	Completed	October 28, 2021 23:55	Manas Sharma	View Update Status

© 2023 JSSATE-B FoodGroup
G.Sanjana Reddy, Anushka Roy

+91-9953411751
+91-9465911750

Fig7.10:- Received placed order on admin page



localhost/JSS_Canteen_Main/admin/admin_order_detail.php?orh_id=26

Order Detail

Accepted

Order #202301300000026
From purple Canteen
Order of Hemanth Reddy

Placed on January 30, 2023 00:39
Pick-up time is January 30, 2023 07:30
The order is not finish yet.

Menu

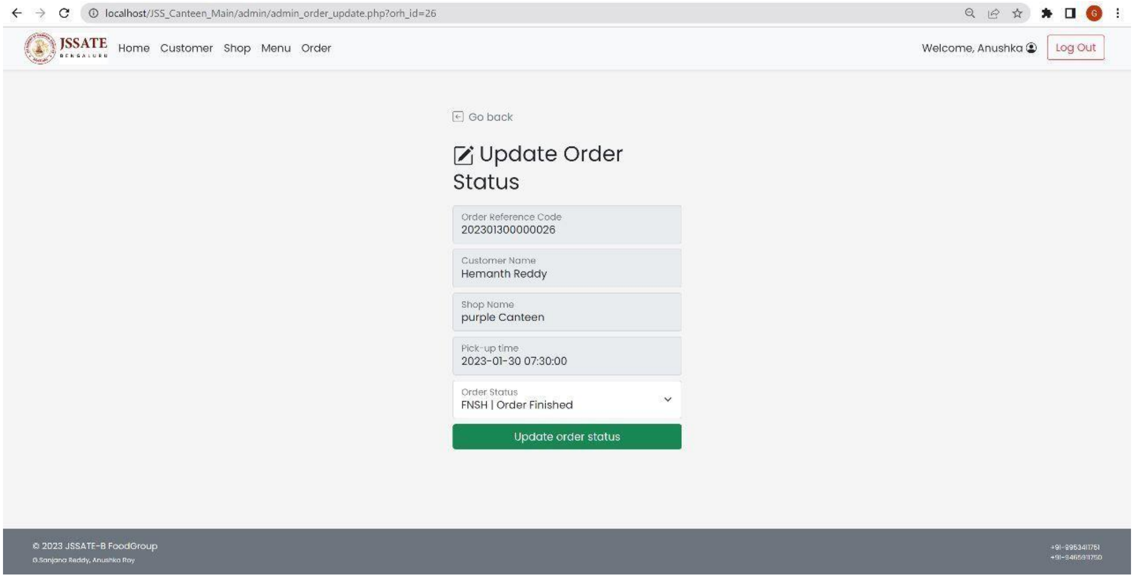
1x Tea
20.00 rs (20.00 rs each)

Grand Total 20.00 rs

© 2023 JSSATE-B FoodGroup
G.Sanjana Reddy, Anushka Roy

+91-9953411751
+91-9465911750

Fig7.11:- Checking order details on admin page


localhost/JSS_Canteen_Main/admin/admin_order_update.php?orh_id=26
JSSATE BENGALURU
Home
Customer
Shop
Menu
Order
Welcome, Anushka
Log Out
Go back
Update order status
© 2023 JSSATE-B FoodGroup
G.Sanjana Reddy, Anushka Roy
+91-9953411751
+91-9465911750

Update Order Status

Order Reference Code
202301300000026

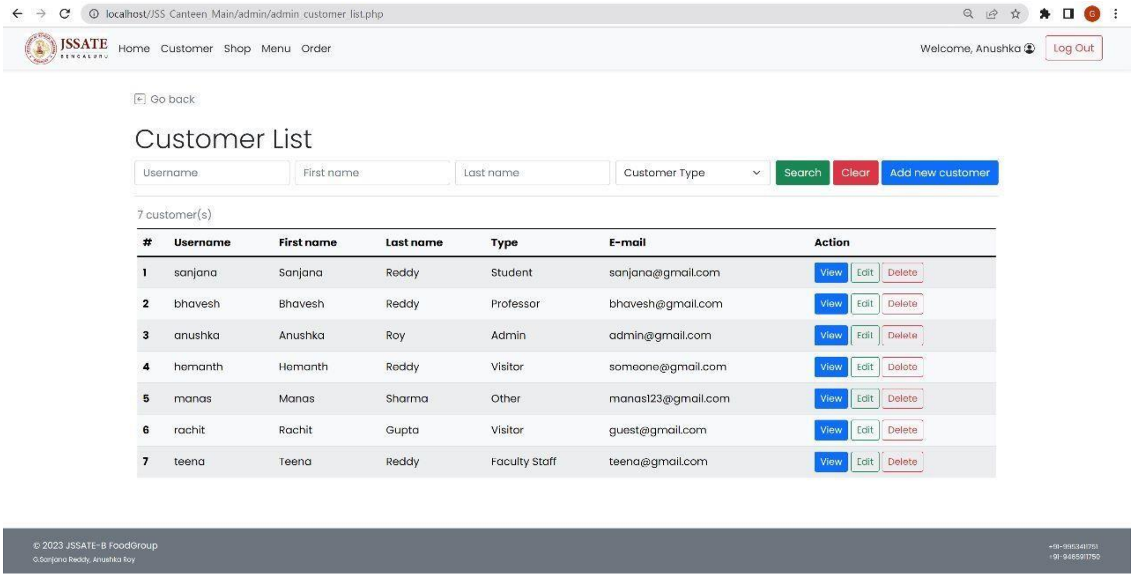
Customer Name
Hemanth Reddy

Shop Name
purple Canteen

Pick-up time
2023-01-30 07:30:00

Order Status
FNSH | Order Finished

Fig7.12:- Making the order status as finished on admin page

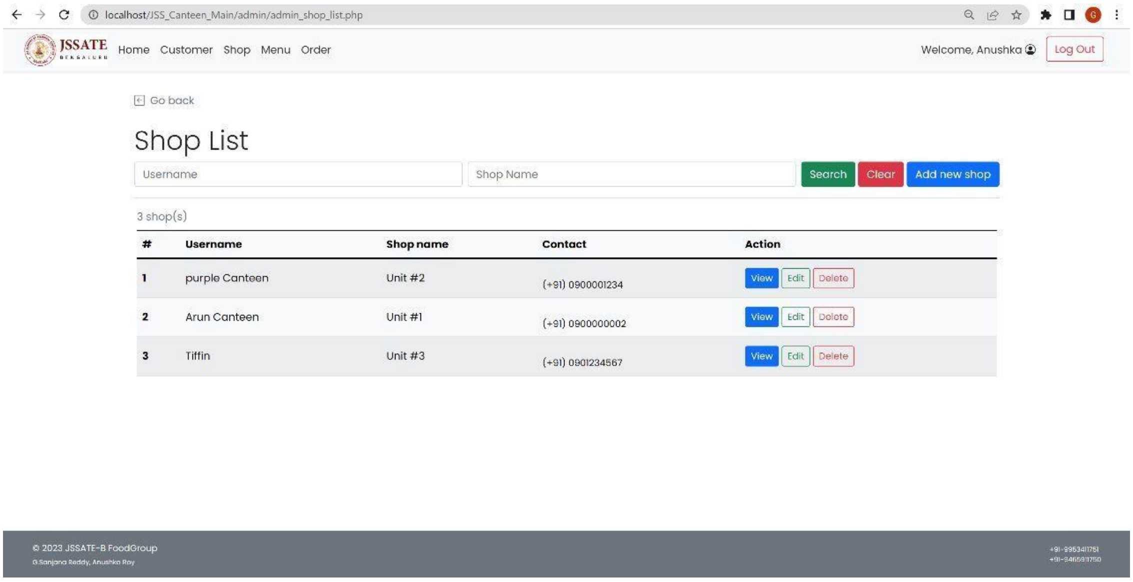


Customer List

7 customer(s)

#	Username	First name	Last name	Type	E-mail	Action
1	sanjana	Sanjana	Reddy	Student	sanjana@gmail.com	View Edit Delete
2	bhavesh	Bhavesh	Reddy	Professor	bhavesh@gmail.com	View Edit Delete
3	anushka	Anushka	Roy	Admin	admin@gmail.com	View Edit Delete
4	hemanth	Hemanth	Reddy	Visitor	someone@gmail.com	View Edit Delete
5	manas	Manas	Sharma	Other	manas123@gmail.com	View Edit Delete
6	rachit	Rachit	Gupta	Visitor	guest@gmail.com	View Edit Delete
7	teena	Teena	Reddy	Faculty Staff	teena@gmail.com	View Edit Delete

Fig7.13:- Customers list on admin page

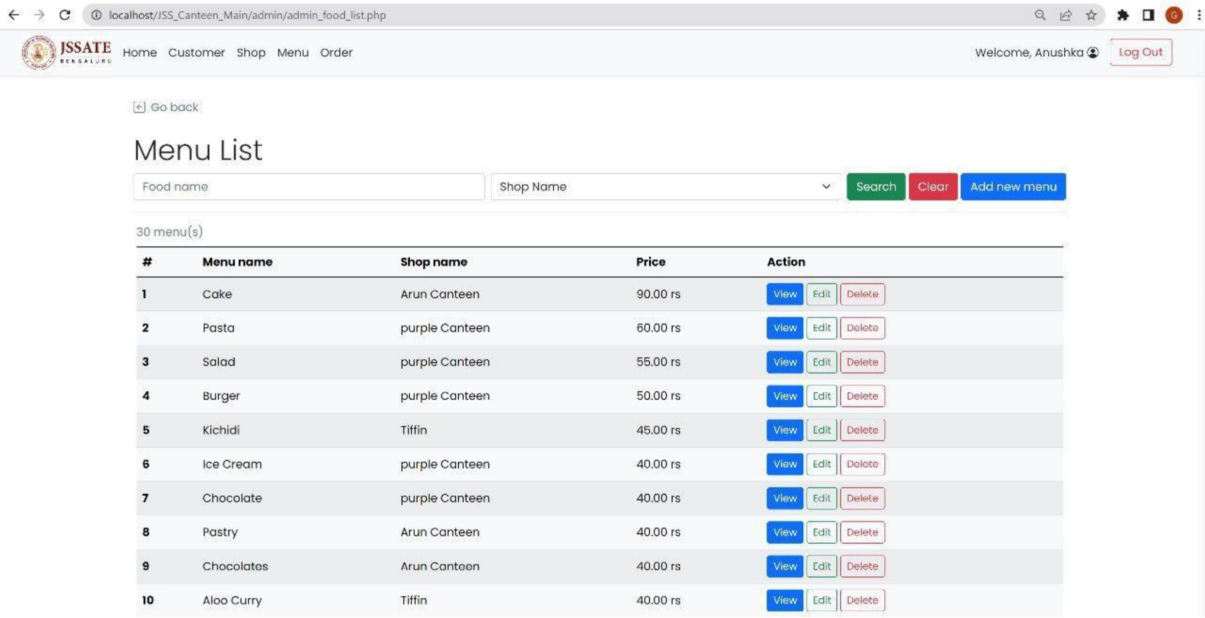


Shop List

3 shop(s)

#	Username	Shop name	Contact	Action
1	purple Canteen	Unit #2	(+91) 0900001234	View Edit Delete
2	Arun Canteen	Unit #1	(+91) 0900000002	View Edit Delete
3	Tiffin	Unit #3	(+91) 0901234567	View Edit Delete

Fig7.14:- Shop list on admin page



Menu List

30 menu(s)

#	Menu name	Shop name	Price	Action
1	Cake	Arun Canteen	90.00 rs	View Edit Delete
2	Pasta	purple Canteen	60.00 rs	View Edit Delete
3	Salad	purple Canteen	55.00 rs	View Edit Delete
4	Burger	purple Canteen	50.00 rs	View Edit Delete
5	Kichidi	Tiffin	45.00 rs	View Edit Delete
6	Ice Cream	purple Canteen	40.00 rs	View Edit Delete
7	Chocolate	purple Canteen	40.00 rs	View Edit Delete
8	Pastry	Arun Canteen	40.00 rs	View Edit Delete
9	Chocolates	Arun Canteen	40.00 rs	View Edit Delete
10	Aloo Curry	Tiffin	40.00 rs	View Edit Delete

Fig7.14:- Menu list on admin page

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION

1. Web based canteen management system project is an application that is developed in PHP and MySQL involved many phases.
2. The approach used is a top-down one concentrating on what first then how and moving to successive levels of details.
3. The first phase started with a detailed study of the problems and prospects of ordering in Foods.
4. This Software is efficient in maintaining customer’s details and can easily perform operations on platform
5. Therefore, we understand the need for the system since it helps in advance planning, Reducing Administrative works, saving time & improving employee efficiency, Reducing Wastage of food items, providing a fast and efficient service and Increasing customer satisfaction.

FUTURE ENHANCEMENT

1. In the future we will create more reliable long databases for fetching records. We will also try to provide best services and also online payment methods.
2. In future we will try to updated student profile and also admin profile. Some more interesting features will be added so that students will find it as more useful.
3. The machine learning algorithms can also be used for the prediction of the most preferred item by the customers. The customers will give feedback and this will be sent to the database. Using the machine learning algorithms this feedback will be analysed and preferred food items will be displayed to the regular users on the online system.
4. The menu list can also be updated according to the admin's choice by the shortcut methods used in the learning algorithms.

CHAPTER 9

REFERENCES

BOOK REFERENCES

1. Database systems Models, Languages, Design and Application Programming, Ramez
2. Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
3. An Introduction to Database Systems" by Bipin Desai.
4. Database System Concepts" by Abraham Silberschatz and S Sudarshan.
5. "Database Management Systems" by Raghu Ramakrishnan.
6. Automate the Boring stuff with Python by Al Sweigart.

WEB REFERENCES

https://creately.com/er-diagram-tool-online

https://www.researchgate.net

https://www.academia.edu

https://www.geeksforgeeks.org