FINAL PROJECT REPORT

ONLINE PIZZA DELIVERY SYSTEM

UTD PIZZA STORE

WEB PROGRAMMING LANGUAGES (CS-6314.002) Under Guidance of Prof. Nurcan Yuruk

TEAM MEMBERS

PANCHAMI RUDRAKSHI (pgr150030)

RAGHUNANDAN N R (rxn150230)

INDEX

- 1. Project Description
- 2. Database Design
- 3. Languages and Framework
- 4. Results
- 5. Team details and roles
- 6. Conclusion and Key learning

1. PROJECT DESCRIPTION

1.1 INTRODUCTION

In today's age of fast food and take-out, many restaurants have chosen to focus on quick preparation and speedy delivery of orders rather than offering a rich dining experience. Until very recently, all of these delivery orders were placed over the phone, but there are many disadvantages to this system, including the inconvenience of the customer needing to have a physical copy of the menu, lack of a visual confirmation that the order was placed correctly, and the necessity for the restaurant to have an employee answering the phone and taking orders.

Online Pizza System is a new approach for the food delivery industry. The main advantage of this system is that it greatly simplifies the ordering process for both the customer and the restaurant. When the customer visits the ordering webpage, they are presented with an interactive and up-to-date menu, complete with all available options and dynamically adjusting prices based on the selected options. After making a selection, the item is then added to their order, which the customer can review the details of at any time before checking out. This provides instant visual confirmation of what was selected and ensures that items in the order are, in fact, what was intended.

This system also greatly lightens the load on the restaurant's end, as the entire process of taking orders is automated. Within this application, all items in the order are displayed, along with their corresponding options and delivery details, in a concise and easy to read manner. This allows restaurant employees to quickly go through the orders as they are placed and produce the necessary items with minimal delay and confusion.

The "UTD Pizza Store" website allows the users to order pizza online sitting at home. The portal helps the users to select from wide range of pizzas and also allows customization of them with different pizza base type, toppings and so on.

The online pizza store has the following features:

The system consists of two kinds of users:

- 1. Admin
- 2. Customer

• The "admin" user can:

- Create an account.
- Log in to the system.
- Navigate the restaurant's menu.
- Add a new Item in the menu.
- o Delete an Item from the menu.
- o Update/Modify the item from the menu.
- View all the items available.
- o Search for an item based on keyword and category.

• The system allows the "**customer**" user to:

- Create an account.
- O Log in to the system.
- o Navigate the restaurant's menu.
- Search for an item based on keyword and category.
- View all the items available.
- O Select items from the wide range of menu.
- Add an item to their current order.
- o Review their current order.
- o Remove an item/remove all items from their current order.
- View order history.

• Login and Sign Up Validation

- o The signup form validates if all mandatory fields are filled out.
- There are different signup forms provided for admin and customer. Each has to register before he can access the system.
- o The signup form checks if the password is strong enough.

The rules of having a strong password are:

- At least one character.
- At least one capital letter.
- At least one number.
- At least 8 characters.

 The signup form checks the new username or email address if it's already registered to the system.

2. DATABASE DESIGN

```
The tables are created as follows:
     create database wplproject ;
     use wplproject;
     LOGIN
     CREATE TABLE login (
     UserName varchar(50) NOT NULL,
     Password varchar(16) NOT NULL,
     UNIQUE KEY UserName ( UserName )
     ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
     USERDATA
     CREATE TABLE userdata (
     userId varchar(20) NOT NULL,
     userFirstName varchar(20) NOT NULL,
     userLastName varchar(20) NOT NULL,
     userEmail varchar(50) NOT NULL,
     userAddress varchar(150) NOT NULL,
     userContactNumber varchar(10) NOT NULL DEFAULT 0 ,
     userGender varchar(1) NOT NULL DEFAULT 0 ,
     userdob date NOT NULL,
     PRIMARY KEY ( userId ),
     UNIQUE KEY userId ( userId ),
     UNIQUE KEY userEmail ( userEmail )
```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```
TYPE
CREATE TABLE type (
typeId varchar(20) NOT NULL,
typeDescription varchar(20) NOT NULL,
PRIMARY KEY ( typeId )
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
PIZZA
drop table pizza;
Create table pizza (
pizzaId varchar(20),
pizzaName varchar(50),
pizzaDescription varchar(1000),
basePrice varchar(20),
PRIMARY KEY( pizzaId )
);
DRINK TYPE
select * from drinkType;
drop table drinkType;
Create table drinkType (
drinkId varchar(20),
drinkName varchar(200),
drinkDescription varchar(1000),
drinkPrice varchar(20),
PRIMARY KEY( drinkId )
);
DESSERT TYPE
Create table desertType (
desertId varchar(20),
dessertName varchar(200),
desertDescription varchar(1000),
desertPrice varchar(20), PRIMARY KEY( desertId ));
```

```
PRODUCTS
```

```
CREATE TABLE IF NOT EXISTS products (
id int(11) NOT NULL AUTO INCREMENT,
product code varchar(60) NOT NULL,
product name varchar(60) NOT NULL,
product desc]tinytext NOT NULL,
product img name varchar(60) NOT NULL,
price decimal(10,2) NOT NULL,
PRIMARY KEY (id),
UNIQUE KEY product code (product code)
) AUTO INCREMENT=1 ;
SEARCH
Create table searchengine(
id int(11),
title varchar(100),
description text,
url text,
keywords varchar(100),
PRIMARY KEY( id )
);
ORDERS
Create table orders (
orderId varchar(20) NOT NULL,
userId varchar(20) NOT NULL,
dates date NOT NULL,
orderPrice varchar(10) NOT NULL DEFAULT 0 ,
typeId varchar(20),
PRIMARY KEY ( orderId )
);
```

3. LANGUAGES AND FRAMEWORKS

Implementation Languages:

Client Side: Bootstrap, HTML5, CSS, JavaScript, AJAX, JQuery.

Server Side: PHP.

Database: MySQL.

Implementation Tools:

Server Side: MAMP Server (PHP)

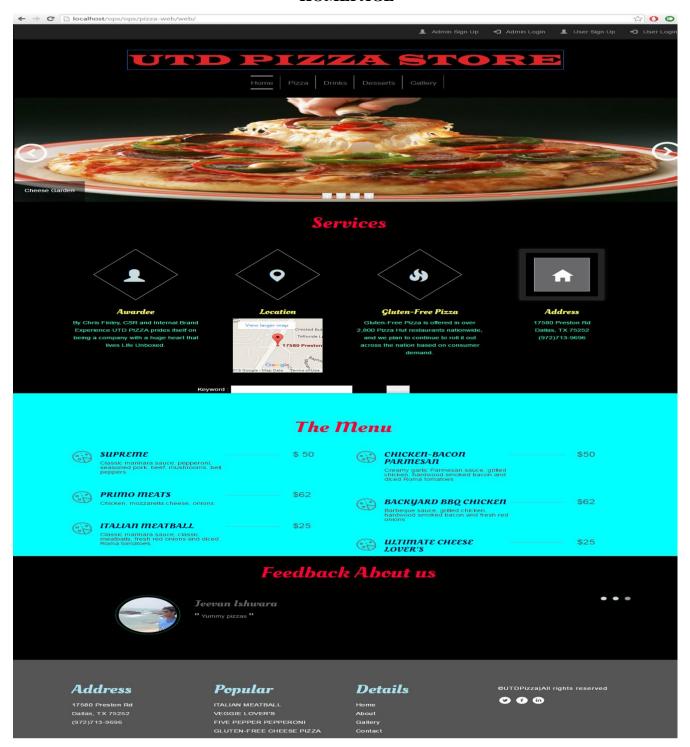
Database: MySQL Workbench.

Debugging Tools:

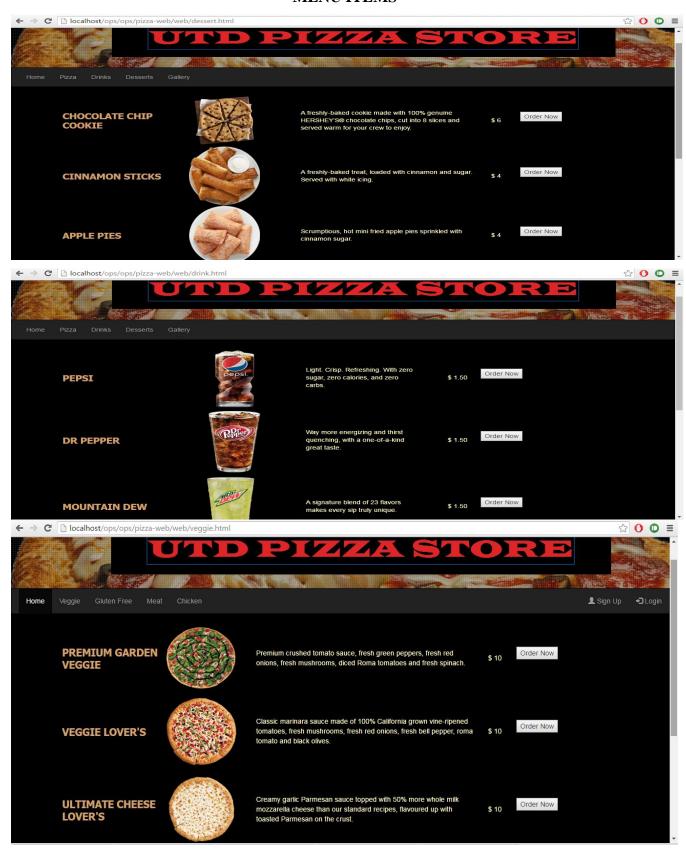
Client Side: Web Browser's Developer Tools Debugger.

4. SCREENSHOTS

HOMEPAGE



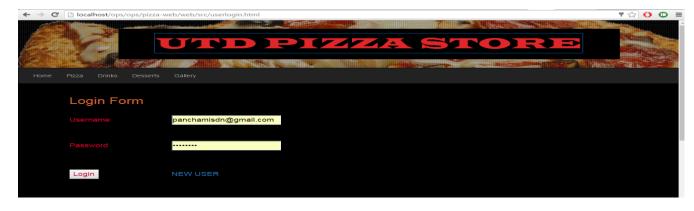
MENU ITEMS



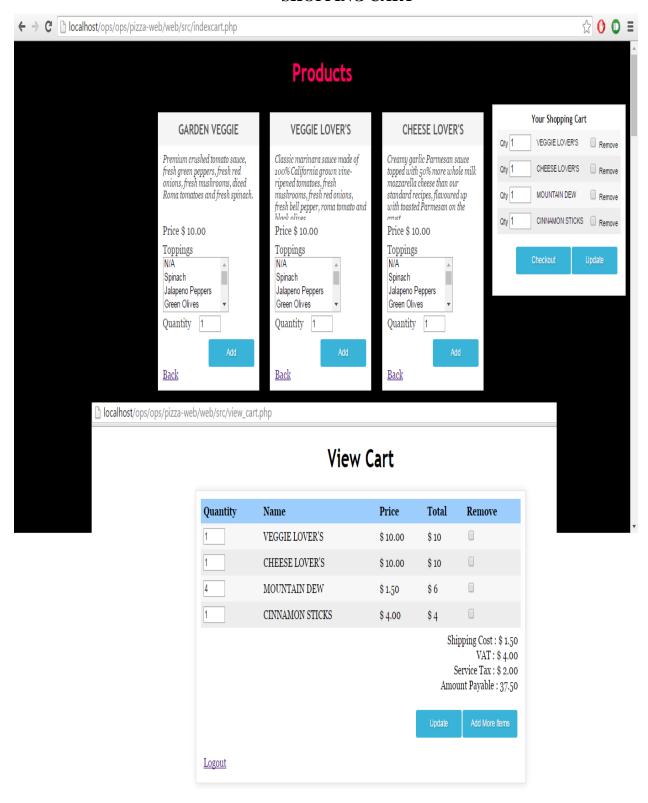
USER SIGN UP FORM



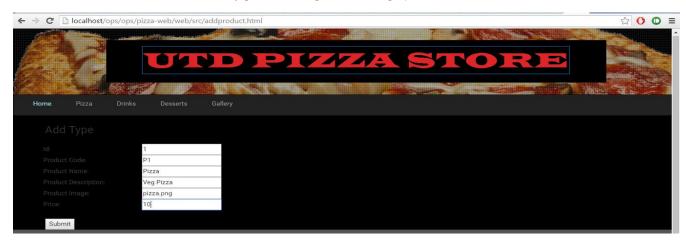
USER LOGIN FORM



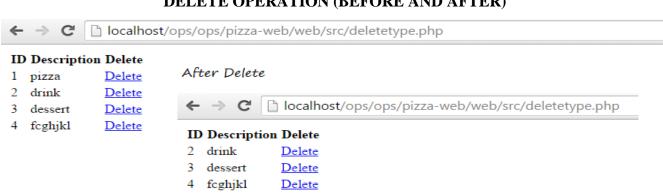
SHOPPING CART



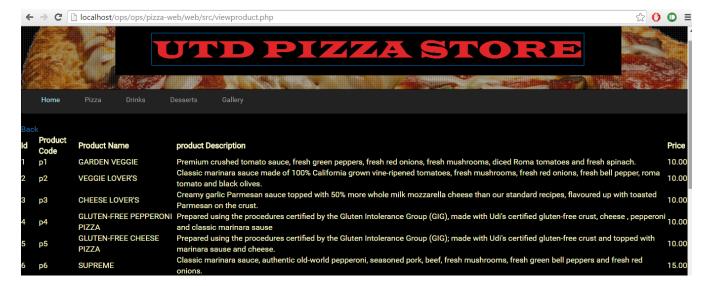
ADD/CREATE OPERATION



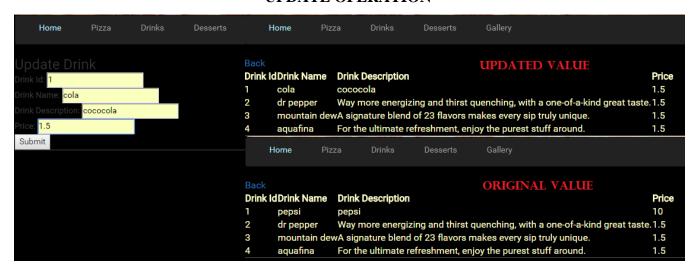
DELETE OPERATION (BEFORE AND AFTER)



VIEW OPERATION



UPDATE OPERATION



SEARCH OPERATION



SEARCH BY KEYWORD RESULT



SEARCH BY CATEGORY RESULT



5. TEAM DETAILS AND WORK DIVISION

Team Members: The team consists of 2 members

- Panchami Rudrakshi pgr150030
- Raghunandan Nuggehalli Ramesha rxn150230

Work Details:

The database design was carried out by all the team mates in collaboration with the professor's feedback.

The project was divided into 4 modules

- 1. Signup / Login
- 2. Admin Data
- 3. User Data
- 4. Implementation of other required features

Module	Functionality	Worked By
Login	Admin and user signup, User/Admin authentication, Login	Raghunandan
	forms, Validating the form in real time for user signup and	
	login, Database design.	
Other features	Home page design, database design, Implementation of	Panchami
	Shopping cart, Search features, session handling.	
User Data	Adding data and implementing features for the user's view	Panchami
	and search. Shopping cart and order history	
	implementation.	
Admin Data	CRUD operations and Pizza inventory design.	Raghunandan

6. CONCLUSION AND KEY LEARNING

The online pizza store website was successfully developed. All required functionalities namely form validations using JQuery and Ajax, features of adding/updating/viewing/deleting data, shopping cart, searching for data based on keywords and filtering the results based on different options, were implemented. Proposed features were like implementing an attractive user interface and shopping cart were implemented. Thus our website can be used as a framework for Online Pizza store. We made sure that our project had no more than 20% creeping rate of including new functions and excluding some mentioned functions based on the feasibility as proposed. Some of the future improvements that we can propose are implementing the email notification feature when the order is made or cancelled.