**WINTER SEMESTER 2020-2021**

TECHNISCHE UNIVERSITÄT
CHEMNITZ

# Study Course: Advanced Management of Data

# REPORT: Project Report on Pizza Baker System

**REPRESENT BY: Shariful Islam**
**Matriculation no: 625878**

Table of Contents

## 1. Our Topic

Building an application that compose and order pizzas configurations to our needs and send them to the pizza baker.

### 1.1 Objective

Our goal is to provide a database with a user interface (website) where customers can choose different ingredients for their own pizza and place their order. The order will be sent to the "kitchen" where the pizza is to be made. The aim is to create a "simple" website which will allow the customer to finish their orders for the first time with ease.

### 1.2 Existing solutions

There are a lot of different web-based ordering systems. These systems do not offer the full functionality that is required for an Application Pizza Order. Ordering systems usually allow people to add products and separate them into different categories and subcategories. "Wow Pizza's" sells pizzas, and most of them also offer customers the opportunity to customize their own pizzas by accumulating their own preferable ingredients.

### 1.3 Challenges

AS our work going further we face a lot of challenges especially with the database of postgresql.

## 2. Planning Phase

### 2.1 Use Cases

Three main use cases are identified: ordering, order processing and administration. The following paragraphs describe each use case in more detail.

### 2.1 Ordering:

The ordering is the first use case. To order, the potential customer must first know what to choose from ("What's on the menu?"). For ordering custom pizzas, the customer will need to know what ingredients he can choose from and how the choice of ingredients will influence the overall price. Customers pick ingredients for their pizza, and sometimes they want a double portion of the ingredients. Also the customer needs to register for ordering by providing his name, address and telephone number.

### 2.2 Order processing:

When an order has been confirmed, processing starts with a customer order. The status of the order will be updated and notified to the customer throughout the processing of the order. New orders will appear in the Admin's order page list. They must state which products are ordered and which ingredients are chosen for the custom pizzas. The order status will be updated after the order has been prepared.

### 2.3 Administration:

By choosing multiple ingredients, custom pizzas are made, the list of available ingredients and their prices are handled by an administrator, such as the manager. Administration involves the ingredients added, modified, and removed. The administrative side is also responsible for adding the supplier responsible for supplying the custom pizza ingredients. The administrator can add, upgrade, and remove the supplier. Administrators also need to have logs of previous orders.

### 2.4 Users

Three types of users should be able to use the system: Customer, Supplier and administrator.

1. Customers are users who visit the website and can create orders by customizing pizzas, selecting products and entering customer details.
2. Suppliers are the group of users that responsible for the ingredients .Suppliers will have their own accounts to log on to.
3. The administrator, or super user, has the ultimate control of the system, he can add, change or delete ingredients and products, as well as add, change, or delete Suppliers accounts.

## 3 Functional requirements:

### 3.1 Customer

1.1 The user must be able to create a new order.
1.2 1.2 The user must be able to customize a pizza by:
    1.2.1 The user must be able to view a list of available ingredients.
    1.2.2 The user must be able to add an ingredient to a custom pizza
    1.2.3 The user must be able to remove an ingredient from a custom pizza
1.3 The user must be able to add a custom pizza to an order.
1.4 The user must be able to see a list of custom pizzas that are added to the order
1.5 The user must be able to see the total price of an order.
1.6 The user must be able to add the name and address of the customer in the time of register

### 3.2 Suppliers

2.1 The Suppliers must be able to log in and out.
2.2 the suppliers must be able to add newer ingredients.
2.3 they can make the ingredient visible or hide if any ingredients stock out or in

### 3.3 Administrators

3.1 The administrator must be able to log in and out.
3.2 The administrator must be able to add/delete/edit pizzas.
3.3 The administrator must be able to add/delete/edit ingredients.
3.4 The administrator must be able to add/delete/edit suppliers.
3.5 The administrator must be able to view an order log.
3.6 The administrator must be able to update the order status

## 4 Technical Approach

During the process of selecting the tools and methods to be used during the process, we took a look at the requirements of the project. As a programming paradigm, Php has an excellent integration with PLsql. Because php is an open source system, there are a huge amount of hosts available. For managing the PLsql database, phpmyadmin was used. This was the logical choice because it is free, and it integrates so well. Creating the code, the program SUBLIME TEXT Editor was used.

## 4.1 ERD DIAGRAM

The data model consists of an Entity Relationship Diagram (ERD) and relational Schemas.

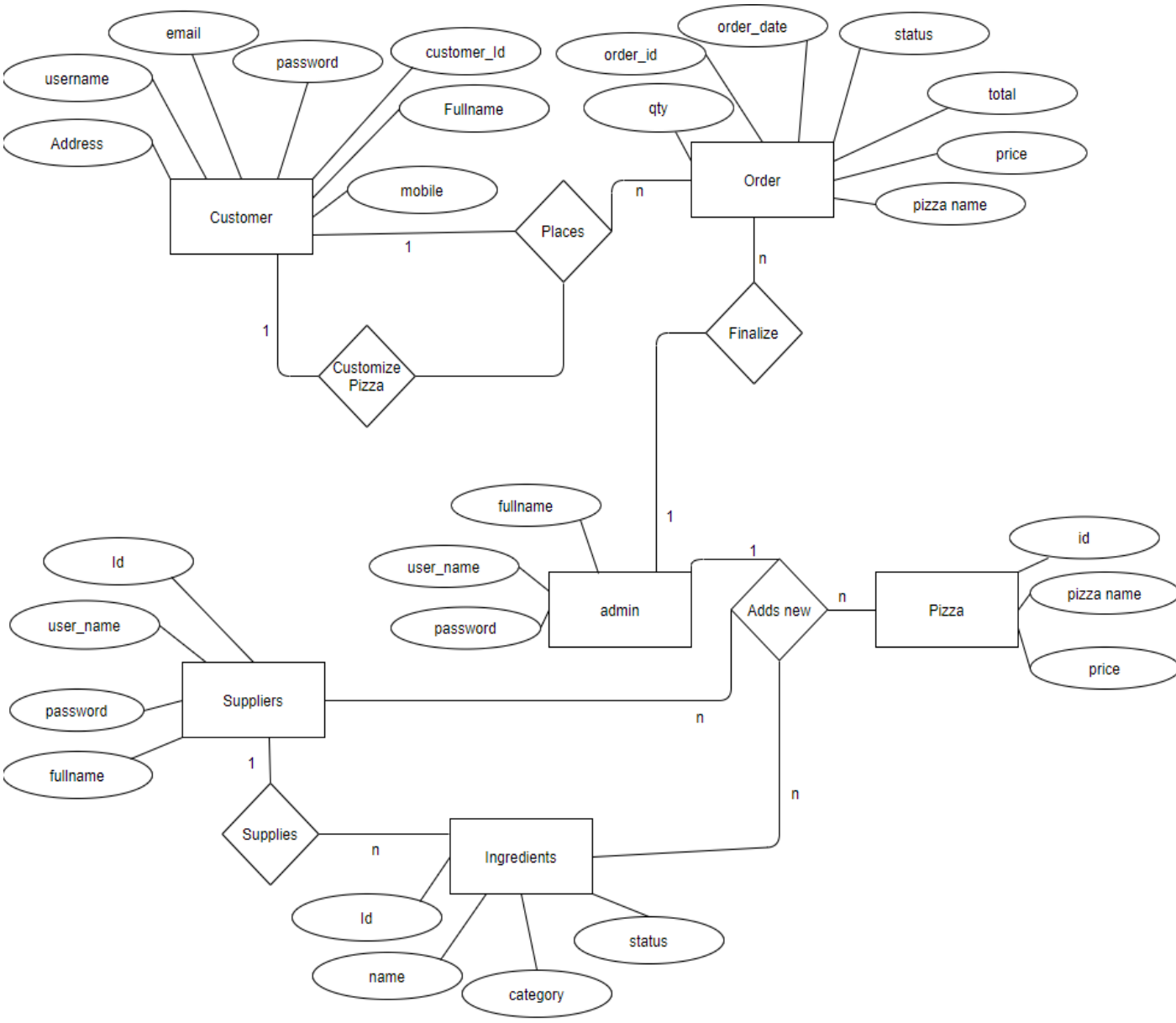



Figure-1: shows the Entity-Relationship Diagram (ERD).

In this paragraph every relation between entities is described.

### Customer --- Places --- Order

A customer can place several orders; the customer entity holds attributes describing properties concerning the customer. The order entity holds attributes such as OrderStatus to track the status of the order (new/prepared/delivered/failed to deliver). Order entity contains several attributes such as order date, price, which type of pizza is ordered and so many things.

**Customer --- Customize Pizza --- Order**

Customer can chose or customize pizza with according to his own taste. He can choose different ingredients on based pizza and give order to that.

**Admin --- finalizes --- Order**

An Admin is responsible for the finalizing the order. Admin can see the entire order list and change the status of the order.

**Admin --- Adds New --- Pizza**

Admin can add/delete/ update new pizza's in the shop and look out how much quantity is left.

**Admin --- Adds New --- Ingredients**

Admin can add new ingredients in shop; he can manage and see which supplier is responsible for supplying the ingredients. Also he can enable or disable a ingredients based on the quantity available in the shop.

**Admin --- Adds New --- Supplier**

Admin can add new supplier as their supplier providing by username and password he can manage them by adding, deleting or updating supplier.

**Supplier --- supplies --- ingredients**

Supplier is responsible for supplying the ingredients according to their category. Through this relationship we track which ingredient has been added by which supplier.

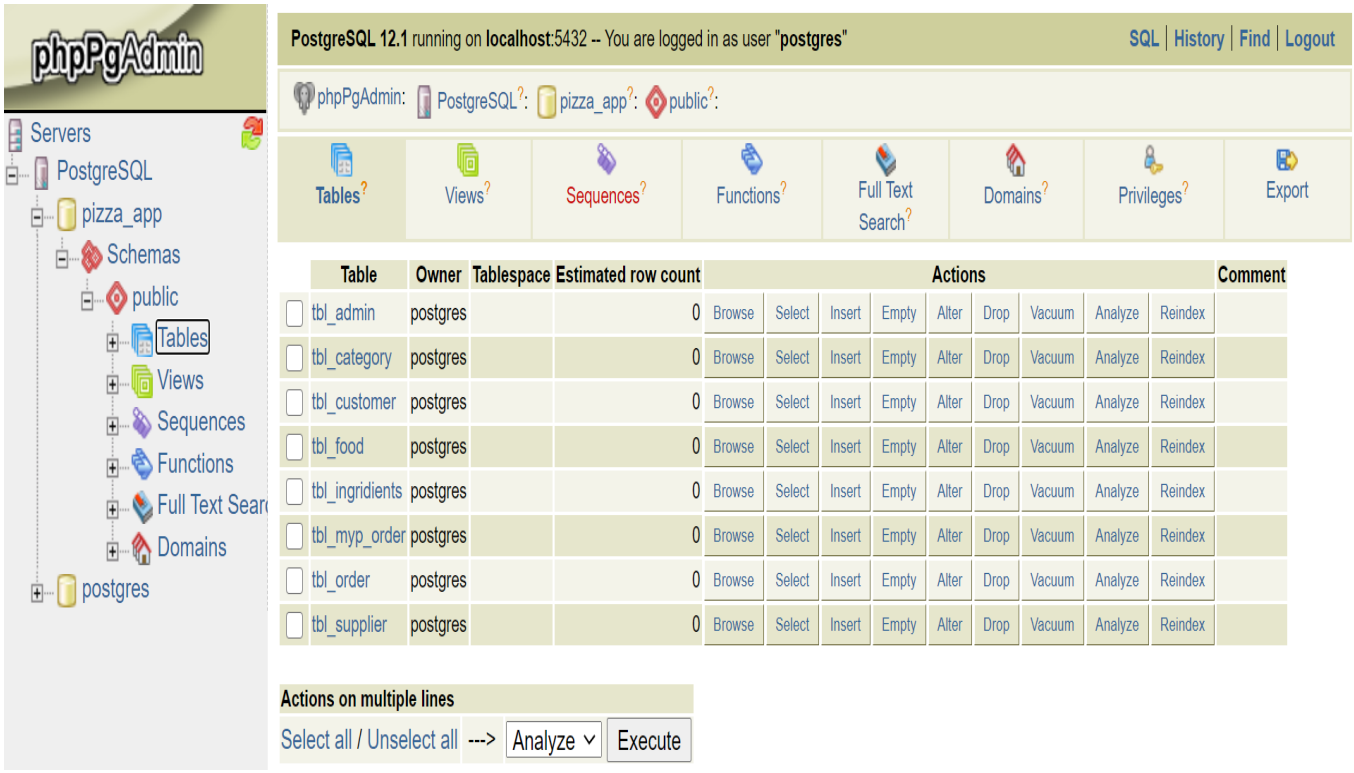
## 4.2 Implementation and Display

### 4.2.1 UI Design:

For User interface design, working with raw Php and develop from sketch. PHP is a general-purpose web development language that is one of the most popular server-side scripting tools based on HTML. This platform is quite fast and considered as the best option for building dynamic web pages. Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

### 4.2.2 Database:

For database implementation and for data passing Postgresql is used in this project. PostgreSQL is a powerful, open source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.In this project mostly used CRUD function of Postgresql. There are numbers of tables according to the entity of that project which shown on previously on ERD diagram.




### 4.2.3 Testing and Results

**Tools:**

**Bitnami WAMP stack:**

For storing database we use Bitnami wamp stack server. Bitnami WAMP stack is an easy to install and easy to use open source Web Platform. It combines leading open source projects, such as Apache, MySQL and PHP with BitRock's extensive open source expertise to provide a consistent, painless way to deploy WAMP in any Linux environment. We use there Postgresql server and making our own database.

**Sublime text Editor:**

For development purpose we use Sublime text editor for both frontend design and backend database operation. Sublime Text 3 (ST3) is a lightweight, cross-platform code editor known for its speed, ease of use, and strong community support. It's an incredible editor right out of the box, but the real power comes from the ability to enhance its functionality using Package Control and creating custom settings.

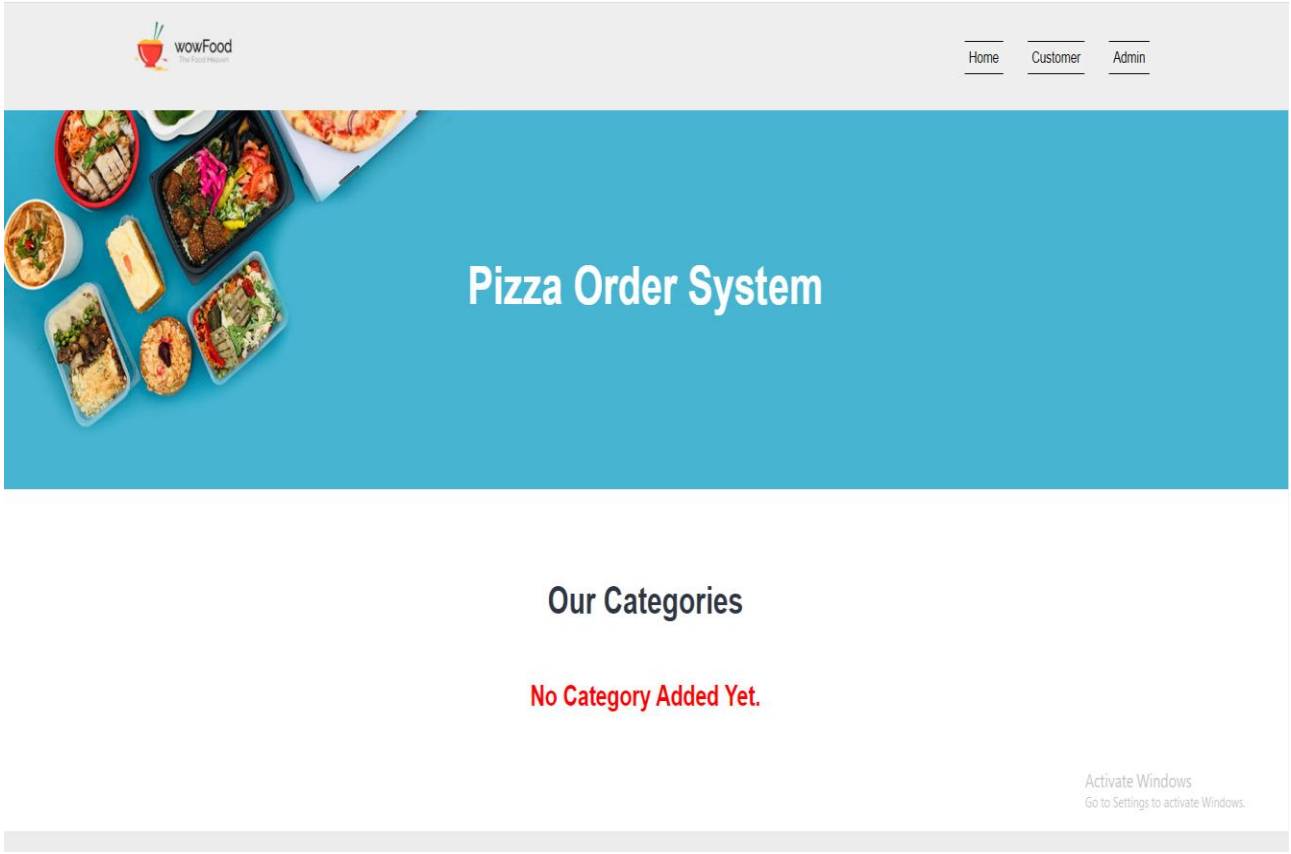
**MD5 Hash Generator:**

We use MD5 Hash generator in order to set the user and password of Admin or super user. The MD5 hash function was originally designed for use as a secure cryptographic hash algorithm for authenticating digital signatures. MD5 has been deprecated for uses other than as a non-cryptographic checksum to verify data integrity and detect unintentional data corruption.

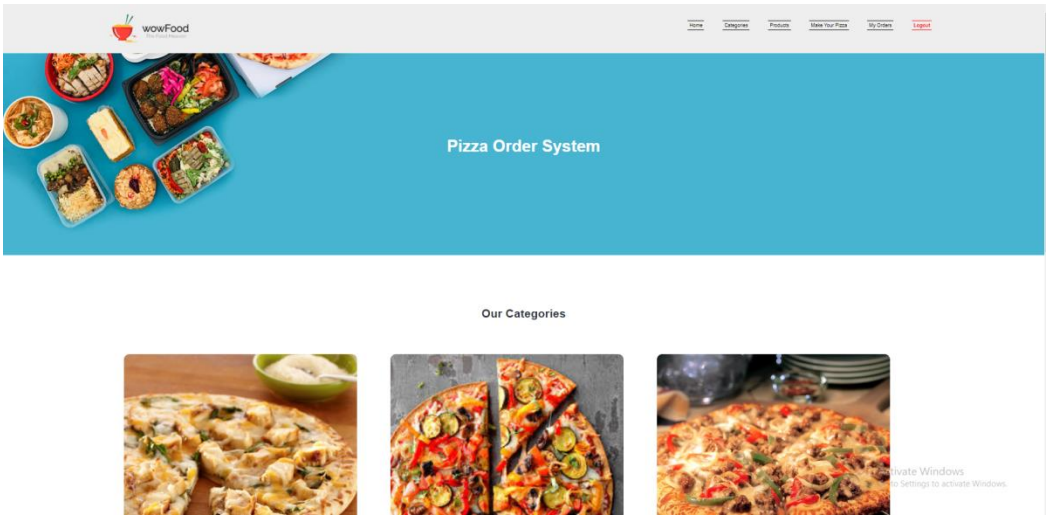
**Result:**

**Home Page:**

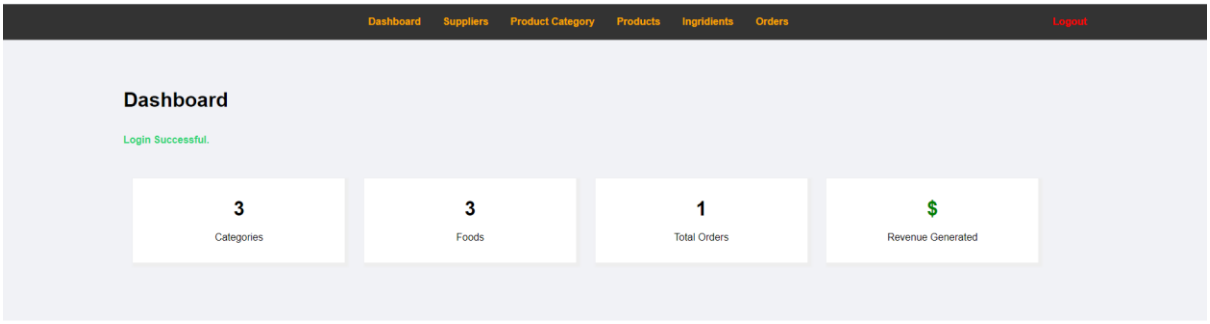
In the home page we set two views as a customer or as admin. There is also showing all available category of pizza is available in the shop. Later we will see the functionality of Customer and Admin.



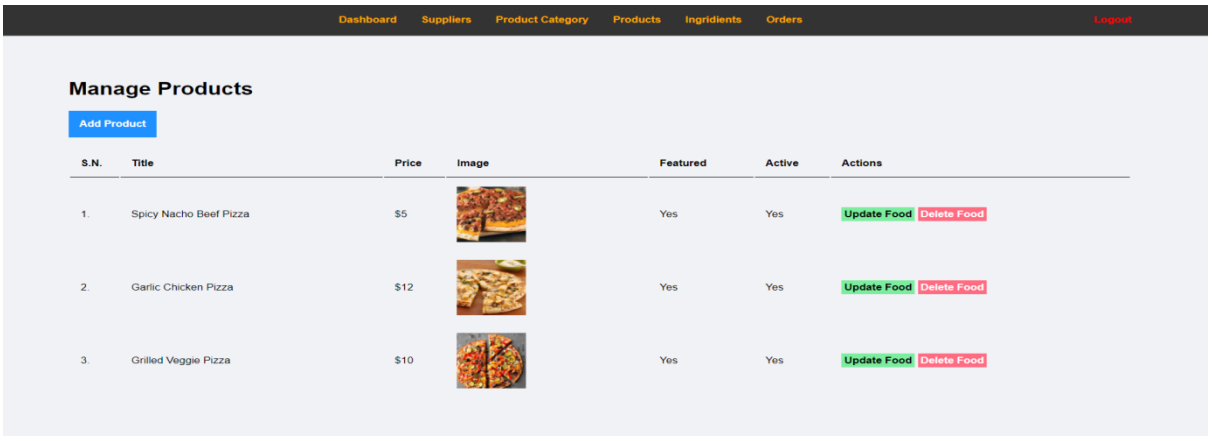
## Customer View Page:



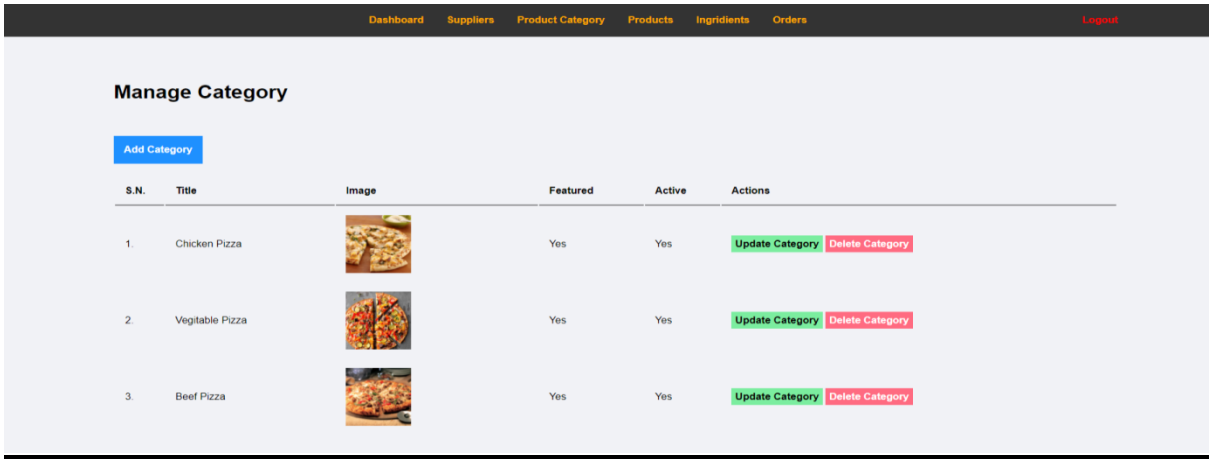

## Admin dashboard Page:



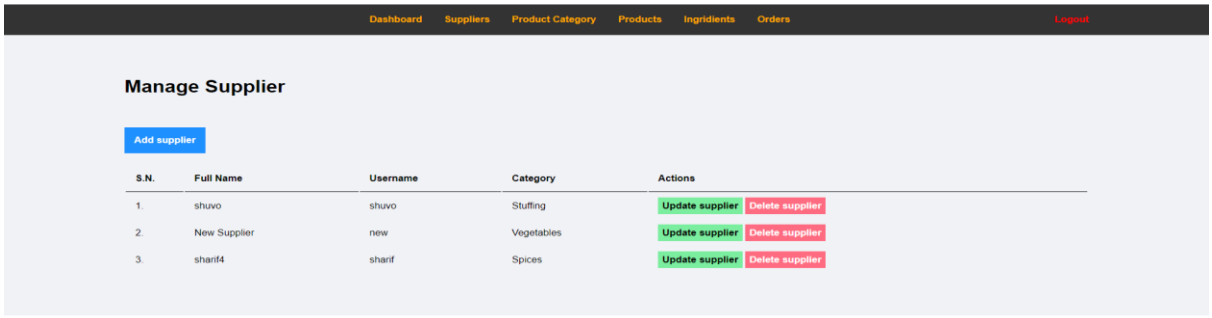

## Manage Food Page:



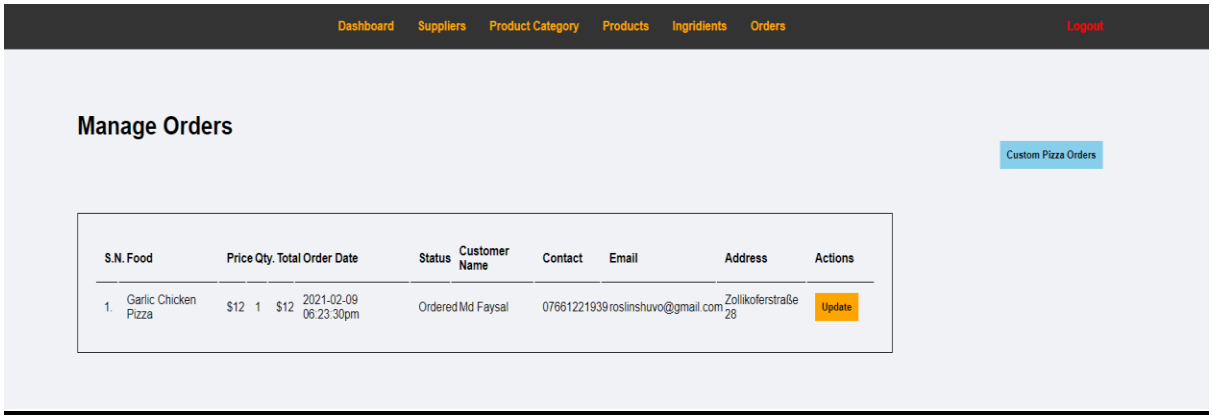
## Manage Category Page:



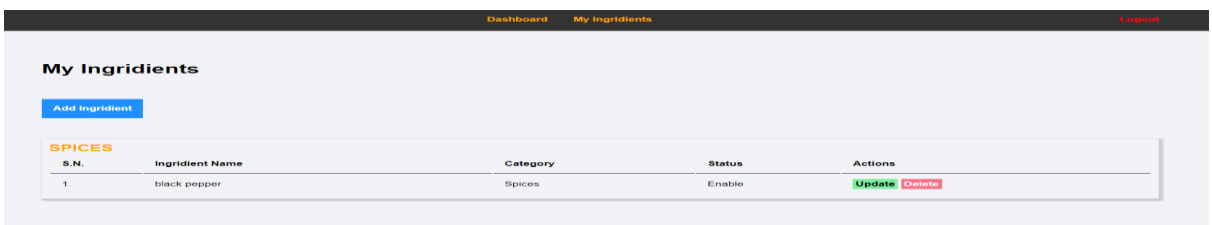

## Manage Supplier Page:




## Order list Page:




## Supplier Dash board Page for ingredients:



# Reference

*PostgreSQL with php - a thorough introduction*. (2020). From w3resource: https://w3resource.com/PostgreSQL/PostgreSQL-with-php-a-through-introduction.php

ANDATI, S. S. (2017). *ONLINE PIZZA ODERING SYSTEM.*

*Get Started With WAMP*. (n.d.). From bitnami.com: https://docs.bitnami.com/installer/infrastructure/wamp/get-started/get-started/

*MD5 hash generator*. (n.d.). From danstools.com: https://www.md5hashgenerator.com/

Postgresql. (n.d.). *Postgresql*. From php.net: https://www.php.net/manual/en/book.pgsql.php

Tyrone Vriesde, T. C. (n.d.). *Pizzeria Ordering System.*

w3resource. (2020). *PostgreSQL with php - a thorough introduction*. From w3resource: https://w3resource.com/PostgreSQL/PostgreSQL-with-php-a-through-introduction.php

# Table of Figures