DBMS Lab Project Report

## Online Food Delivery System



Database Management System Lab

CSE-403L

Submitted By:

* Afreen Nasrullah
* Hoorish Ahmad
* Sidrah Ishtiaq

Registration No:

* 22PWCSE2195
* 22PWCSE2198
* 22PWCSE2159

Section: B

“On my honor, as a student of the University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work”

Submitted To: Engr. Sumayyea Salahuddin

*Department of Computer Systems Engineering*

*University of Engineering and Technology Peshawar*

## Project Title: Online Food Delivery System

1.1 INTRODUCTION:

The Online Food Delivery System is a web-based application developed using the Laravel framework, designed to allow users to order ready-to-eat meals from restaurants conveniently and efficiently. This system replaces traditional food ordering methods with a modern, automated platform that simplifies the process for both customers and restaurant staff.

Using Laravel’s MVC architecture, the application ensures clean code organization, secure routing, and robust backend functionality. Customers can browse the food menu, add items to their cart, place orders, and track delivery status, while restaurant managers can monitor and update order statuses in real time. Laravel’s built-in features like Eloquent ORM and Blade templating make the development process faster and more efficient, ensuring a scalable and secure application.

1.2 OBJECTIVE:

The main objective of this Laravel-based Online Food Delivery System is to streamline the food ordering and delivery workflow through a structured and maintainable web application. It aims to:

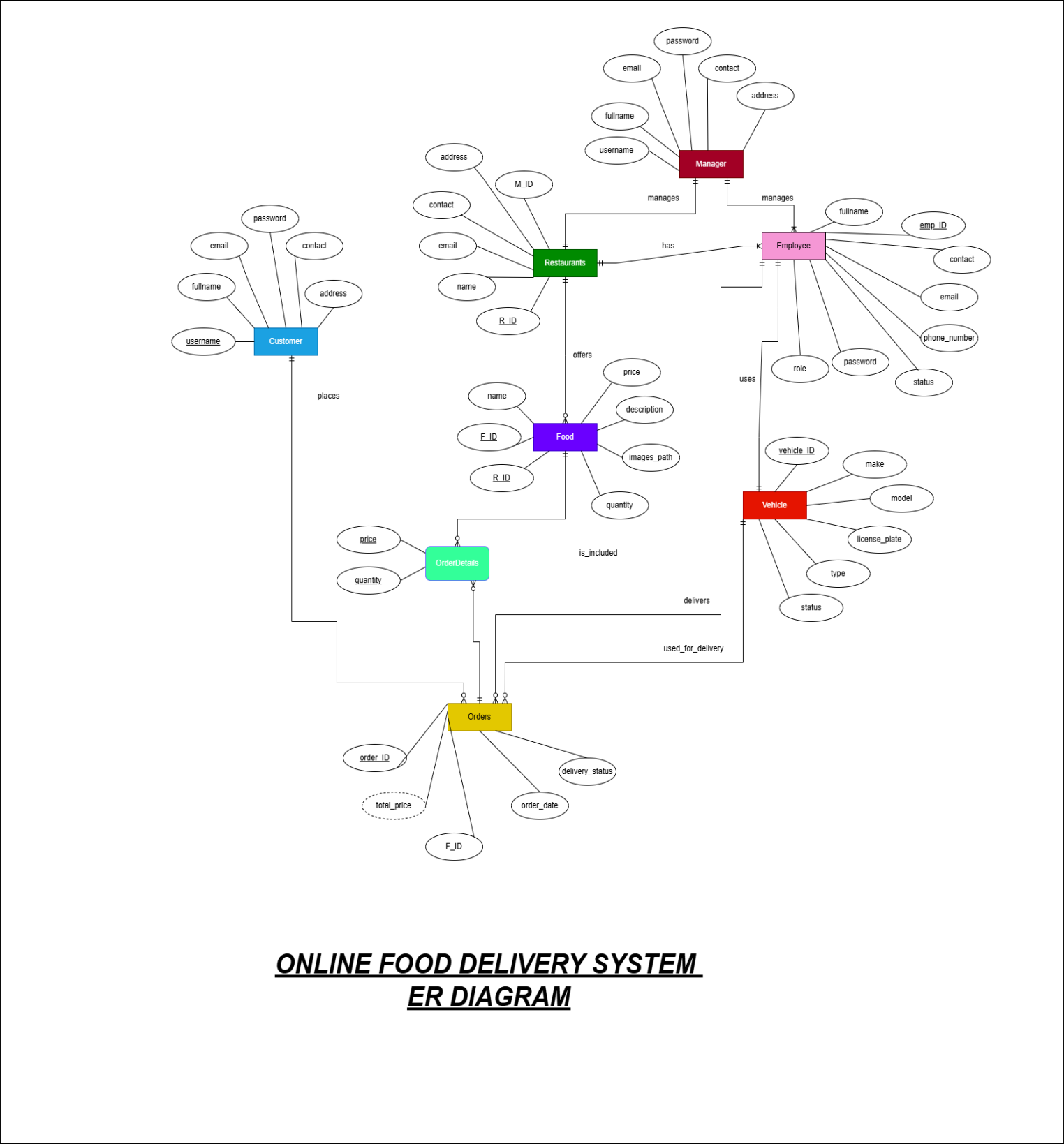
* Enable customers to browse food items, place orders, and view delivery status.
* Allow restaurant managers to view, update, and manage pending and completed orders.
* Improve data management through Laravel’s Eloquent ORM and database migrations.
* Enhance system security and performance by utilizing Laravel’s built-in tools.
* Deliver a user-friendly interface using Laravel Blade templates for both customers and admins.

1.3 SOFTWARE REQUIREMENTS:

|  |  |
| --- | --- |
| Software | Purpose |
| Laravel Framework (v12) | Backend web application framework based on PHP |
| PHP (v7.4 or above) | Server-side scripting language required to run Laravel |
| Composer | Dependency manager for PHP to install Laravel and its packages |
| MySQL | Relational database to store user, food, and order data |
| XAMPP | Local server environment for running Apache and MySQL |
| Visual Studio Code | Code editor used for writing and editing Laravel code |

1.4 DATABASE DESIGN:

Conceptual Schema:



Normalization:

USERS Table:

1NF: Remove repeating groups

Already in 1NF — no multivalued attributes.

2NF: Remove partial dependencies

Already in 2NF — id is the primary key; all non-key columns fully depend on it.

3NF: Remove transitive dependencies

Still valid — no non-key column depends on another non-key column.

Final users table (3NF) remains unchanged:

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Notes |
| id | BIGINT | PK | Auto-increment |
| name | VARCHAR |  |  |
| email | VARCHAR | UNIQUE | Must be unique |
| email\_verified\_at | TIMESTAMP |  | Nullable |
| isAdmin | BOOLEAN |  | Default: false |
| password | VARCHAR |  |  |
| address | VARCHAR |  |  |
| remember\_token | VARCHAR |  | Nullable |

FOOD Table:

1NF: No repeating groups

Already in 1NF

2NF: All columns fully depend on PK (id)

Yes — name, price, etc. depend on id

3NF: No transitive dependencies

No field is derived from another non-key (e.g., type is not derived from description)

Final food table (3NF):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Notes |
| id | BIGINT | PK | Auto-increment |
| name | VARCHAR | UNIQUE | Must be unique |
| price | FLOAT |  |  |
| description | VARCHAR |  |  |
| type | VARCHAR |  | e.g., FastFood, Beverage |
| picture | VARCHAR |  | Path to image |

ORDER Table:

1NF: No repeating groups

Already in 1NF

2NF: Fully functional dependency on PK

All fields depend only on id

3NF: No transitive dependency

We assume type and deliveryAddress are atomic and not derived from other fields.

Final order table (3NF):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Notes |
| id | BIGINT | PK | Auto-increment |
| user\_id | BIGINT | FK | References users(id) |
| date | DATETIME |  | Order time |
| type | VARCHAR |  | 'online', 'pickup', etc. |
| deliveryAddress | VARCHAR |  | Nullable for pickup orders |

FOOD\_ORDER Table:

1NF: Split rows per food item

|  |  |  |
| --- | --- | --- |
| order\_id | food\_id | quantity |
| 1 | 1 | 1 |
| 1 | 2 | 2 |

2NF: Composite PK (order\_id, food\_id)

quantity depends on the full PK, not just part of it

3NF: No transitive dependencies

All columns depend only on the whole PK

Final food\_order table (3NF):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Notes |
| order\_id | BIGINT | PK, FK | References order(id) |
| food\_id | BIGINT | PK, FK | References food(id) |
| quantity | INTEGER |  | Quantity of that food in the order |

PASSWORD\_RESETS Table:

Simple table, no normalization issues and Composite candidate key:

email + token

Final password\_resets table (3NF):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Notes |
| email | VARCHAR | INDEX | Indexed for fast lookup |
| token | VARCHAR |  | Reset token |
| created\_at | TIMESTAMP |  | Nullable timestamp |

PERSONAL\_ACCESS\_TOKENS Table:

Already normalized:

* All attributes depend on id
* No repeating groups or transitive dependencies

Final personal\_access\_tokens table (3NF):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Notes |
| id | BIGINT | PK | Auto-increment |
| tokenable\_type | STRING |  | Model type (e.g., User) |
| tokenable\_id | BIGINT |  | ID of the related model |
| name | STRING |  | Token name |
| token | STRING(64) | UNIQUE | Unique access token |
| abilities | TEXT |  | Nullable |
| last\_used\_at | TIMESTAMP |  | Nullable |
| created\_at | TIMESTAMP |  |  |
| updated\_at | TIMESTAMP |  |  |

SQL Database Tables and Queries:

Table for food:

A screenshot of a computer

AI-generated content may be incorrect.

Table for food\_order:

A screenshot of a computer

AI-generated content may be incorrect.

Table for migrations:

A screenshot of a computer

AI-generated content may be incorrect.

Table for orders:

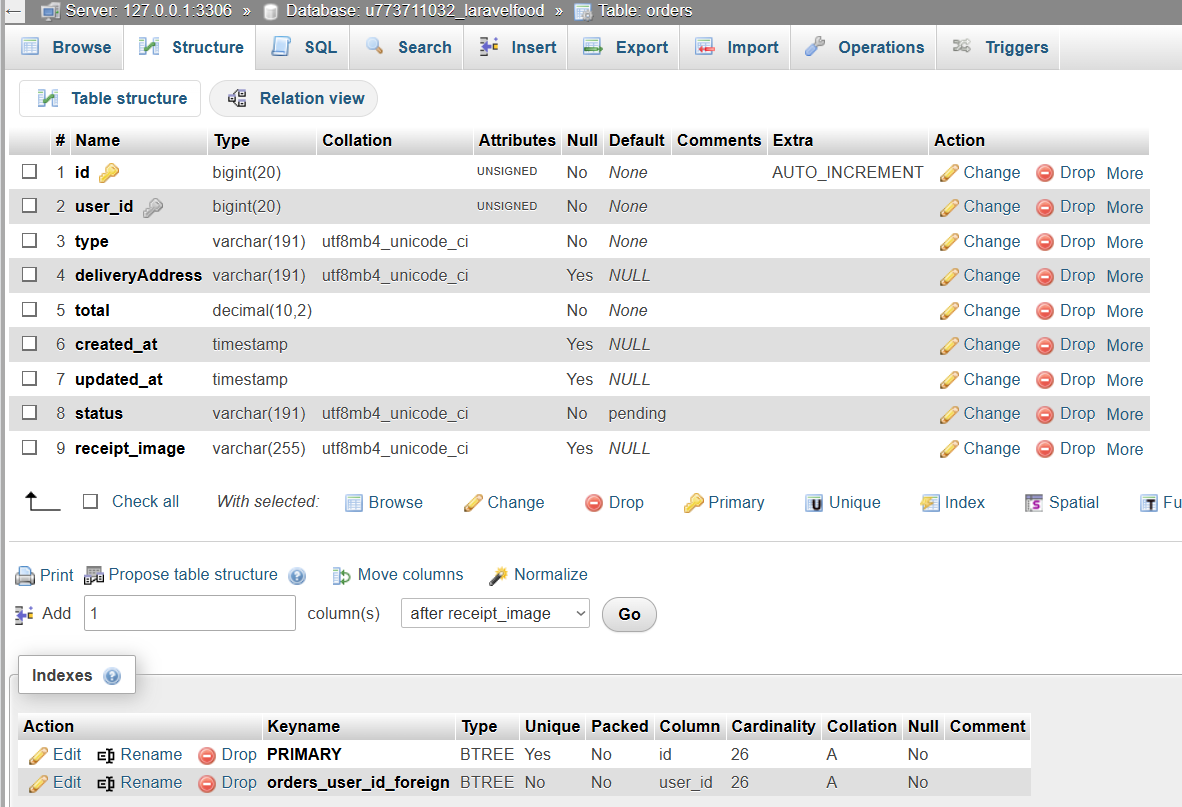


Table for order\_items:

A screenshot of a computer

AI-generated content may be incorrect.

Table for password\_resets:

A screenshot of a computer

AI-generated content may be incorrect.

Table for personal\_access\_tokens:

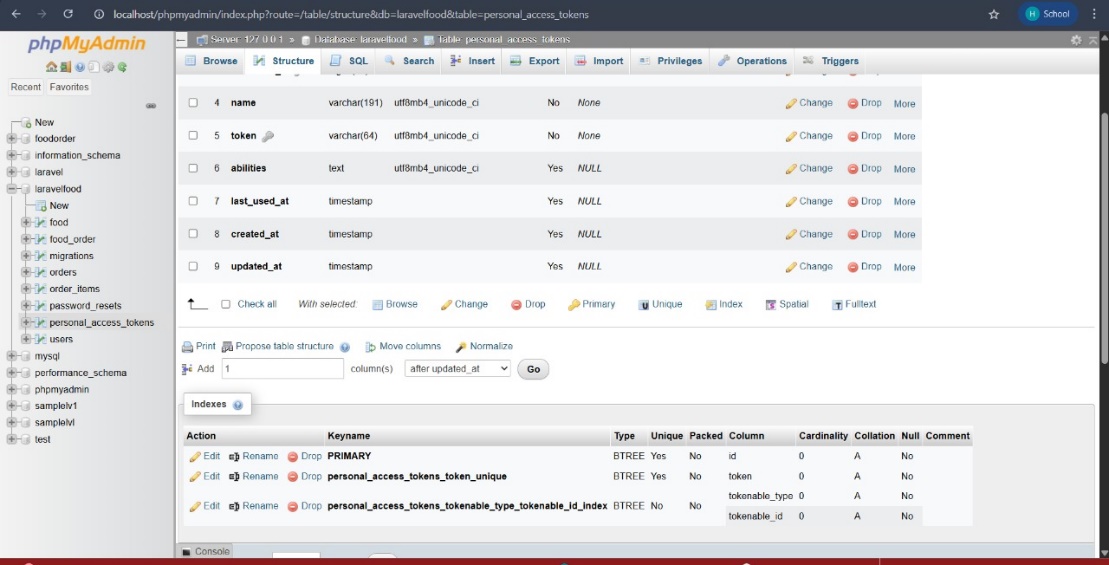
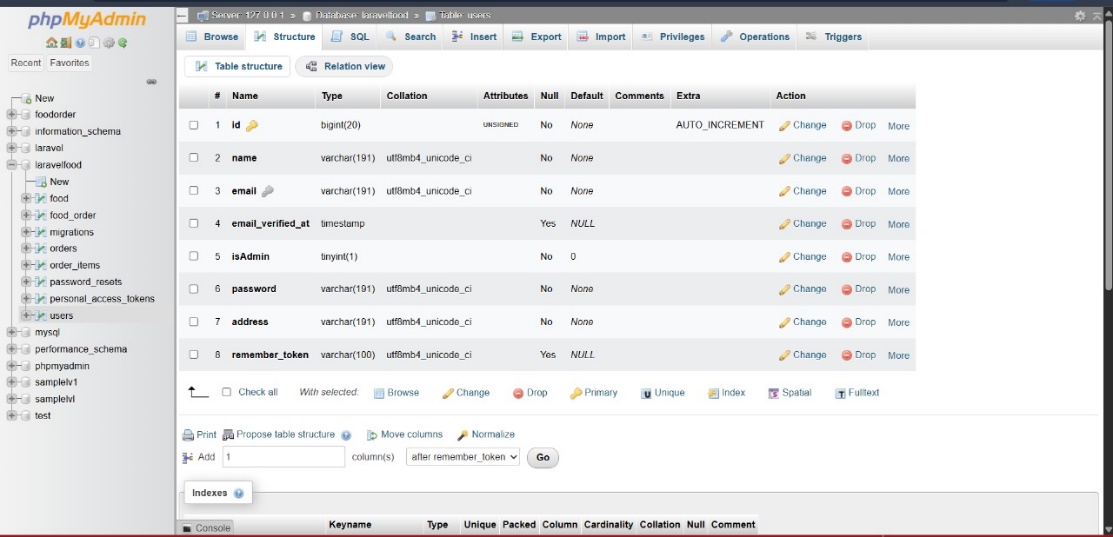


Table for users:



Queries for table food:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `food`

Queries for table food\_order:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `food\_order`

Queries for table migrations:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `migrations`

Queries for table orders:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `orders`

Queries for table order\_items:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `order\_items`

Queries for table password\_resets:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `password\_resets`

Queries for table personal\_access\_tokens:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `personal\_access\_tokens`

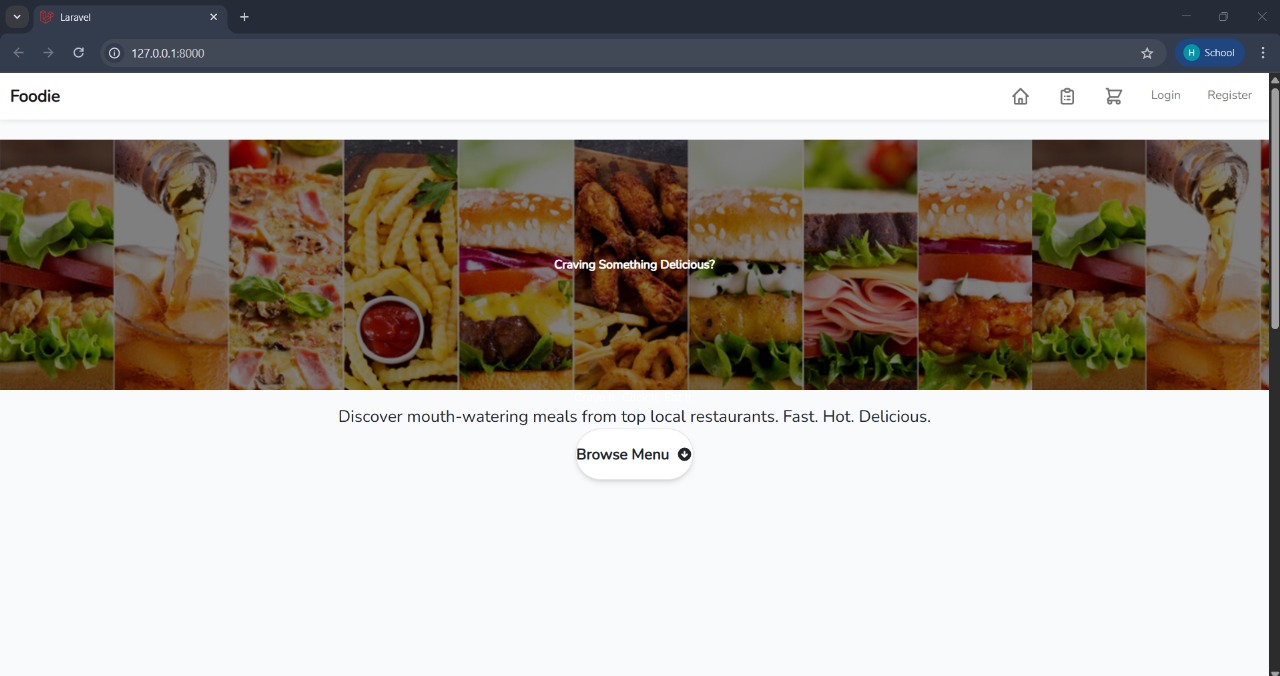
Queries for table users:

[SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `users`

1.5 COMPREHENSIVE IMPLEMENTATION DETAILS (LARAVEL):

Home Page:

Displays a welcome message and basic navigation options for users to explore the menu, log in, or sign up.



Food Menu Page:

Shows a list of available food items with names, description, prices, quantity, and "Add to Cart" buttons for customers to select their desired meals.

A screenshot of a menu

AI-generated content may be incorrect.

Login Page:

Allows both customers and managers to securely log in using their registered credentials to access their respective dashboards.

A screenshot of a computer

AI-generated content may be incorrect.

Shopping cart Page:

Displays the items selected by the customer along with quantities, total price, and an option to place the order or remove the order form the list.

A screenshot of a food menu

AI-generated content may be incorrect.

Placing Order:

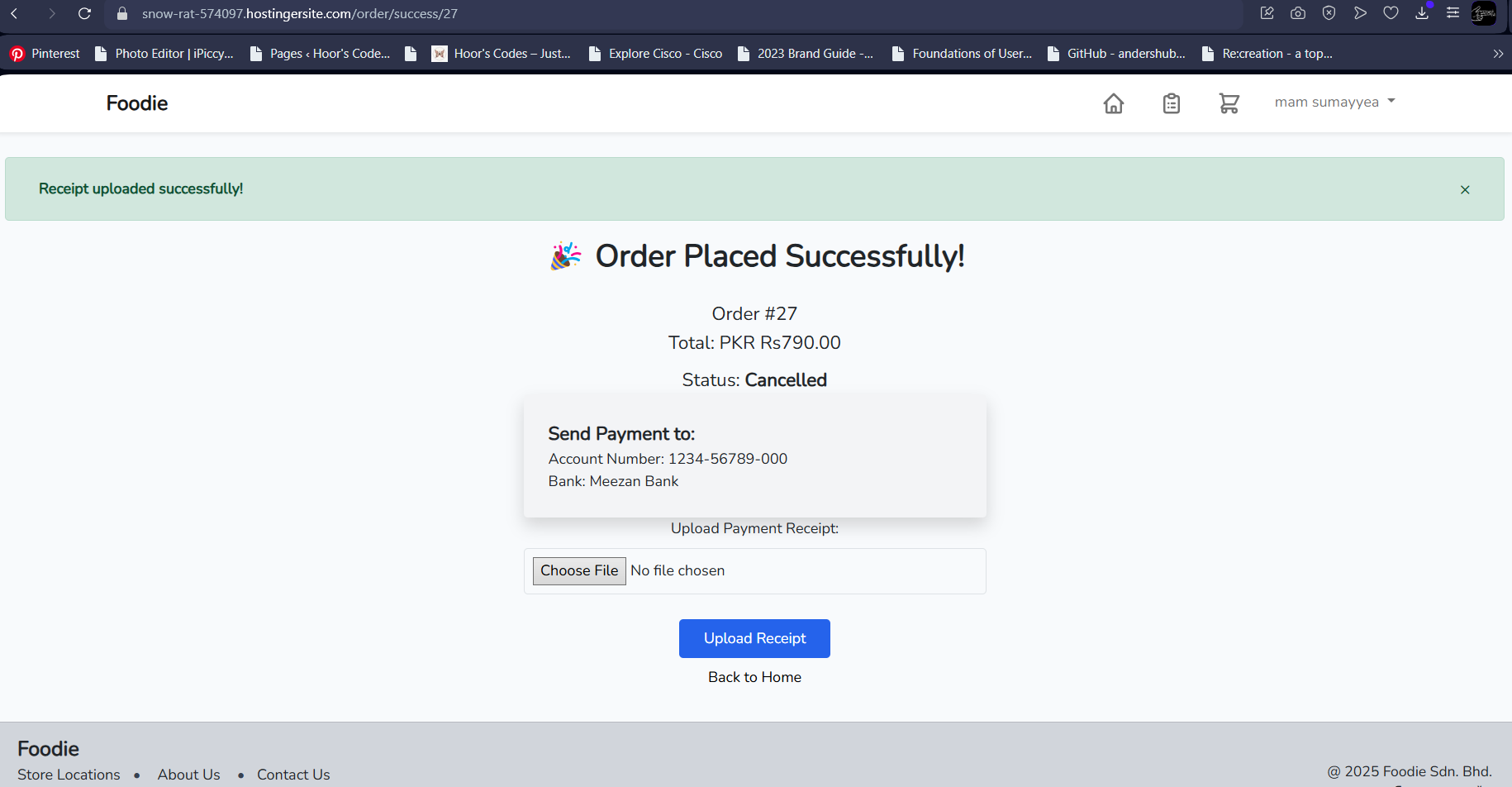
Allows the customer to choose the order type, either a pickup or a delivery and customers can include their delivery address if option “Delivery” is selected.

A screenshot of a computer

AI-generated content may be incorrect.

Order placed successfully:

After confirmation of the order, the order is processed on the manager side. The page shows the order number, delivery status, total amount and a “Back to Home” button. Also allows users to check the real-time delivery status of their order using their unique order number.

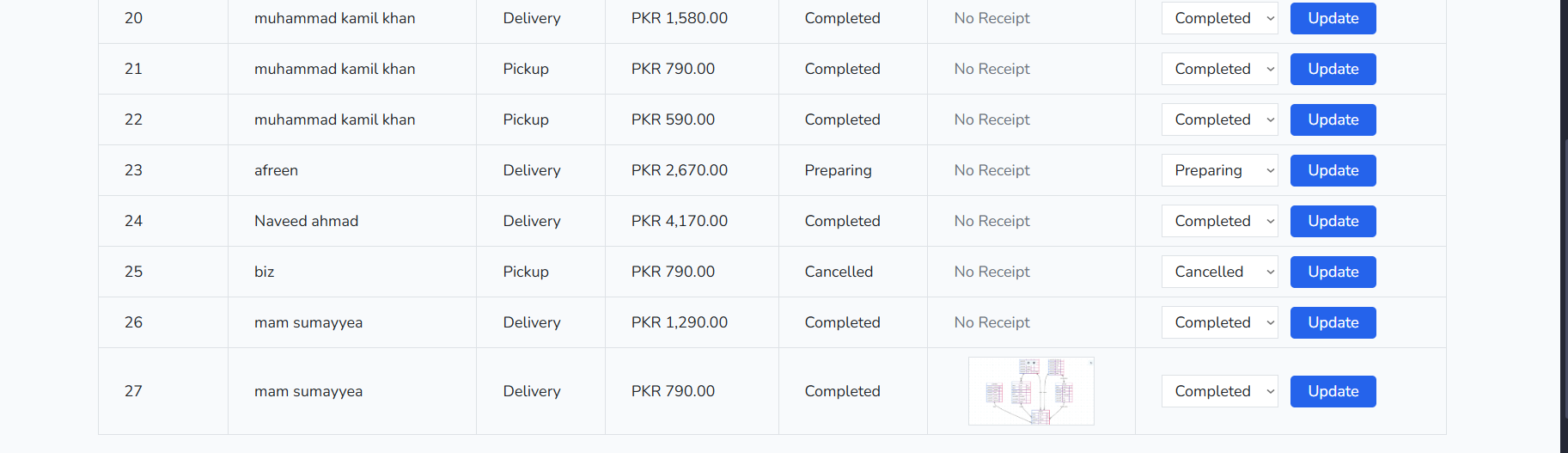


Admin Order status Control Page:

Displays a list of all orders, categorized as Pending or Completed.  
The manager can review receipts uploaded, and update orders as Completed once they are processed.

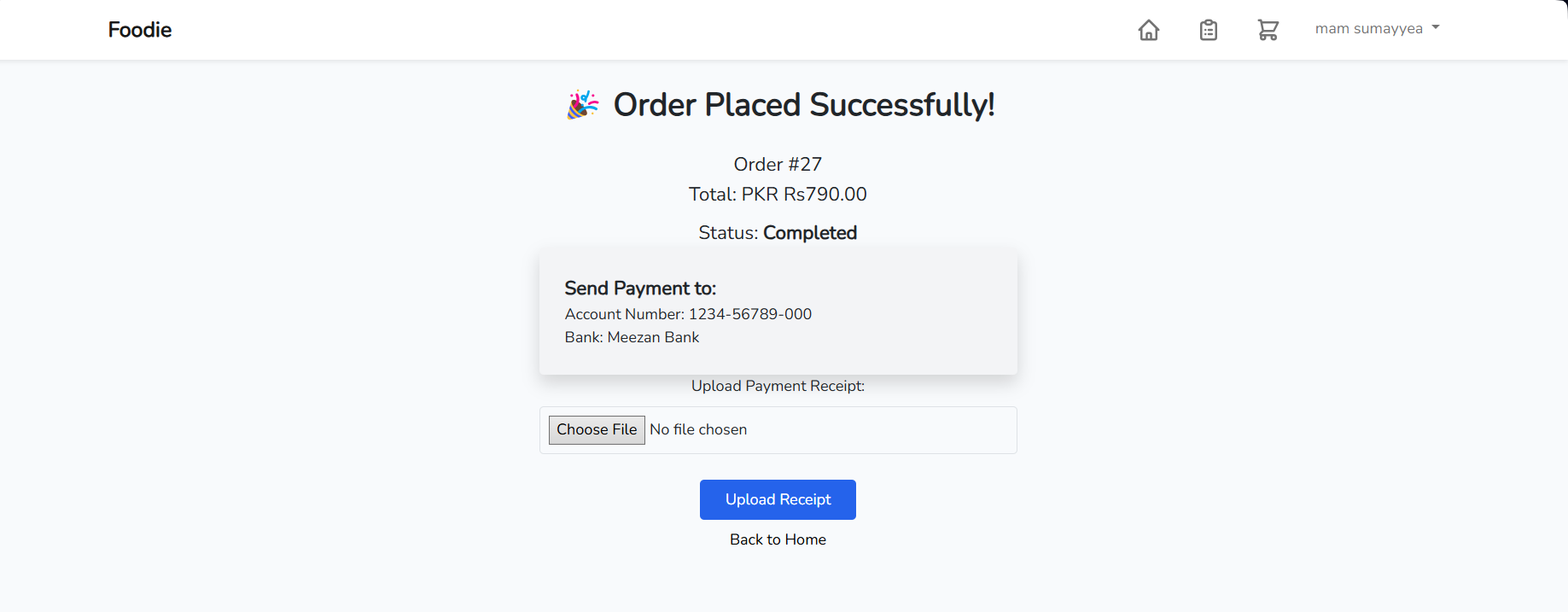
A screenshot of a computer

AI-generated content may be incorrect.



Order completed successfully:

Once the manager marks an order as completed, the delivery status is updated in real time and reflected on the customer’s delivery status page.



1.6 FUTURE ENHANCEMENTS:

* User Ratings and Reviews: Allow customers to rate food items and leave feedback for restaurants.
* Live Order Tracking: Integrate a map-based tracking system to show the real-time location of the delivery rider.
* Responsive Mobile App: Develop a mobile version of the system for easier access and improved user experience.
* Advanced Search and Filters: Enable search by food type, price range, or dietary preferences.
* Email/SMS Notifications: Notify customers about order status, promotions, or delays through automated messages.
* Enhanced Security: Implement two-factor authentication and better session management for both users and admins.

1.7 CONCLUSION:

Developing the Online Food Delivery System using Laravel has provided in-depth experience with a modern PHP framework and real-world web application development. Through this project, we learned how to structure applications using the MVC pattern, implement secure user authentication, handle dynamic data with Eloquent ORM, and create responsive user interfaces using Blade templates.

The Laravel framework helped us build a scalable, maintainable, and secure application with clean code practices and reusable components. This project enhanced our understanding of full-stack development, Laravel routing, database management, and user role handling, making it a valuable learning experience in building real-time, feature-rich web applications.

1.8 REFERENCES:

Laravel Official Documentation  
<https://laravel.com/docs>

Used for understanding Laravel's routing, controllers, Eloquent ORM, authentication, and Blade templating.

W3Schools – PHP, HTML, CSS, and MySQL Tutorials  
<https://www.w3schools.com/>

Used for refreshing foundational web development concepts and SQL queries.

ChatGPT by OpenAI  
<https://chat.openai.com/>

Used for project guidance, explanation of concepts, and code structure suggestions.

Deployed Website:

<https://snow-rat-574097.hostingersite.com>

Database Credentials:

DB\_DATABASE=u773711032\_laravelfood

DB\_USERNAME=u773711032\_foodadmin

DB\_PASSWORD="Pizza$499"