ABE'S BURGER SHOP

CIS 435- Project 4

Ibrahim Ammar
Professor Baugh
CIS 435 – Web Technology
November 29, 2023

Introduction

Welcome to my project report for Abe's Burger Shop's new web application. In this report, I'm excited to share all about my journey in creating a website that I believe will significantly enhance the experience for both the burger shop and its customers.

My goal was to build something both useful and creative, more than just a minor adjustment of previous work. I aimed to create a standout, substantial project that would genuinely contribute to the restaurant's operations and customer satisfaction.

For this full-stack application, I delved into both front-end and back-end development. The front end consists of HTML, CSS, and JavaScript, the essential elements of any dynamic website. I chose to incorporate React in parts of the project to make the user interface more interactive and engaging.

On the back end, I worked with Node and PHP combined with a MySQL dabtabase. This setup allowed me to efficiently manage and store data, such as menu items and customer orders, crucial for the restaurant's daily operations.

Taking on a full-stack application was a significant challenge, but one I was eager to tackle. I drew inspiration and knowledge from numerous resources but ensured my project was distinctly my own. This report is more than a summary; it's a comprehensive overview of the hard work, creativity, and learning that went into creating the Abe's Burger Shop web application.

In the following pages, I will guide you through the design decisions, development process, and unique features of the application. I will also share the challenges I encountered and the solutions I implemented. By the end of this report, you will have a clear understanding of the effort and thought put into developing a functional, reliable, and engaging website for Abe's Burger Shop.

Design Considerations

When embarking on the development of the Abe's Burger Shop web application, several critical design considerations were at the forefront of my decision-making process. The primary objective was to create a user-friendly, efficient, and aesthetically pleasing digital platform that mirrors the quality and ethos of Abe's Burger Shop.

Understanding User Needs: The first step in the design process was to understand the needs and behaviors of the restaurant's customers. I conducted informal surveys and observed that most customers wanted a quick and easy way to view the menu, place orders, and get information about the restaurant. It was essential that the application catered to these needs while maintaining simplicity and ease of use.

Choice of Technologies: For the front end, I chose HTML, CSS, and JavaScript due to their versatility and widespread use in web development. This trio allowed me to create a responsive and interactive user interface. React was selected to build a dynamic single-page application, enhancing user experience with its fast rendering and state management capabilities.

The back-end technology choice was driven by the need for robustness and reliability. I opted for PHP, a server-side scripting language known for its ease of use and strong community support. PHP's compatibility with various databases and servers made it an ideal choice for this project. For the database, MySQL was selected due to its proven performance in handling complex queries and large volumes of data, which was vital for managing the restaurant's dynamic menu and customer orders.

Responsive Design: With a significant portion of internet traffic coming from mobile devices, creating a responsive design was non-negotiable. I ensured that the website's layout, images, and navigation elements adapt seamlessly to different screen sizes, providing an optimal viewing experience across all devices.

Aesthetic Appeal: The visual design aimed to reflect the warmth and welcoming atmosphere of Abe's Burger Shop. I used a color palette that resonates with the restaurant's theme, incorporated high-quality images, and ensured that the text was

readable against various backgrounds. Careful attention was paid to the layout to make sure that it was not only appealing but also intuitive for the users.

Security and Performance: Security and performance were also critical considerations. I implemented best practices to protect user data and ensure the application's smooth operation, even under heavy load. Techniques like data validation, secure API endpoints, and optimized database queries were used to enhance security and performance.

System Components and Functions

Architecture Overview:

The architecture of Abe's Burger Shop web application is bifurcated into a front-end and a back-end, each responsible for specific functionalities within the system.

- **Front-end:** Built using React, the front-end is designed as a single-page application (SPA) that provides a dynamic and responsive user experience. React's component-based architecture allows for efficient updates and rendering of UI components. The front-end interacts with the back-end via API calls to fetch and display data, as well as to send user inputs to the server.
- **Back-end:** The back-end of the application is powered by PHP, a server-side scripting language known for its versatility and ease of integration with databases. It handles all the business logic, processes client requests, and performs CRUD (Create, Read, Update, Delete) operations on the MySQL database.
- **MySQL Database**: The MySQL database stores all the application data, including burger menu items, prices, descriptions, and customer orders. It's structured to support efficient data retrieval and manipulation by the PHP back-end.

Component Description:

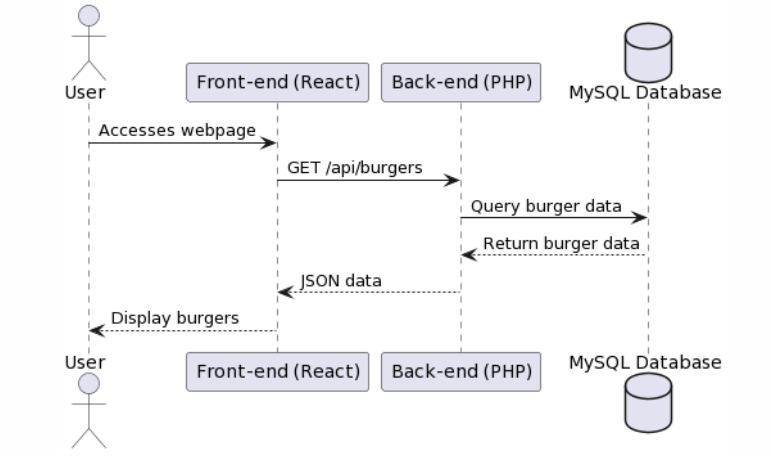
- **API Endpoints:** The back-end exposes various API endpoints for interaction with the front-end. These include:
    - /api/burgers for retrieving the list of burgers.
    - /api/burgers/{id} for fetching, updating, or deleting a specific burger.
    - /api/order for submitting a new order.
- **Database Schema:** The MySQL database includes tables like Burgers (with fields ID, Name, Description, Price) and Orders (with fields like OrderID, CustomerName, BurgerID, Quantity, etc.). These tables are designed to optimize data storage and retrieval for the application's needs.

Interaction of Components:

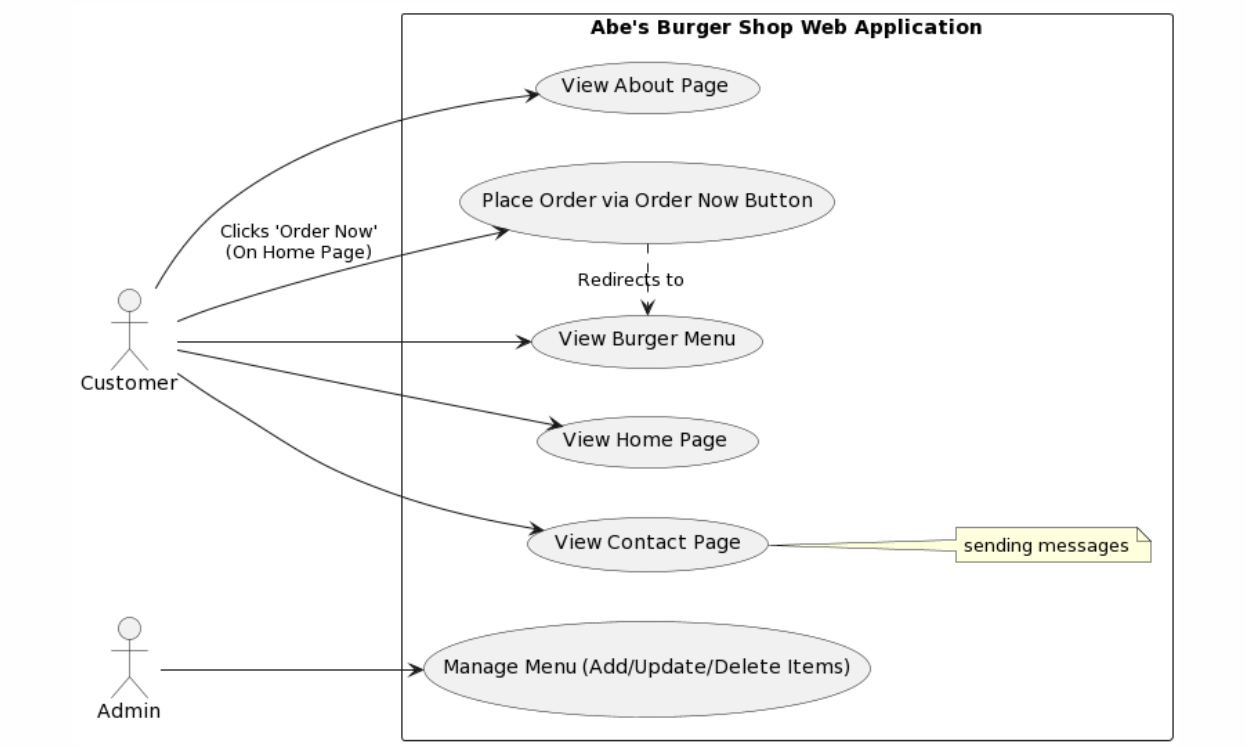
- Data flow begins at the front-end, where user actions trigger API requests to the back-end.
- The PHP back-end processes these requests, interacts with the MySQL database, and returns the appropriate data or confirmation.
- The front-end then processes and displays this data, ensuring a dynamic and interactive experience for the users.

UML Diagrams:

- **Sequence Diagram**
  - **User Accesses Webpage:** The user starts by accessing the webpage where the burger menu will be displayed.
  - **Front-end (React) Sends GET Request:** The React front-end sends a GET request to the back-end. The request is specifically for the /api/burgers endpoint, which is designed to return data about the burgers.
  - **Back-end (PHP) Queries Database:** Upon receiving the GET request, the PHP back-end queries the MySQL database for burger data. This involves executing a SQL query that fetches information about all the burgers available in the menu.
  - **MySQL Database Returns Data:** The MySQL database processes the query and returns the burger data to the PHP back-end. This data typically includes details like the burger names, descriptions, prices, etc.
  - **Back-end (PHP) Sends JSON Data:** The PHP back-end takes the data received from the MySQL database and formats it as JSON, which is a lightweight data-interchange format that's easy for humans to read and write and easy for machines to parse and generate.
  - **Front-end (React) Displays Burgers:** Finally, the React front-end takes the JSON data received from the back-end and renders it in the browser for the user to view. This is where the list of burgers is displayed on the webpage.



- **Use Case Diagram**
  - **Customer** can view the Home, Contact, About, and Menu pages.
  - The "Place Order via Order Now Button" use case illustrates the specific interaction on the Home page, where clicking the "Order Now" button leads the user directly to the Menu page.
  - **Admin** has the capability to manage the menu, including adding, updating, and deleting items.



Summary and Conclusion

In this project, I embarked on the journey of designing and implementing a web-based burger ordering system. It was a solo endeavor that challenged me to expand my technical skills and take complete ownership of the project from inception to execution.

The primary objective of this project was to create a user-friendly platform that streamlines the process of ordering delicious burgers. Inspired by the convenience offered by modern food delivery apps, I set out to design a system that would cater to both customers and restaurant owners.

My journey began with thorough planning. I carefully considered the needs of potential users, envisioning a system that offers an easy browsing experience for customers while providing robust management tools for restaurant administrators.

The choice of technologies played a pivotal role in shaping the project. I opted for React for the front-end, allowing me to craft a dynamic and responsive user interface. Meanwhile, PHP served as the backbone of the back-end, enabling the creation of efficient API endpoints. The MySQL database ensured structured data storage, facilitating easy retrieval and management of menu items and orders.

One of the key challenges I encountered was orchestrating the interaction between the front-end and back-end. Designing RESTful API endpoints was a critical task, ensuring that data could flow seamlessly between the user interface and the database. Additionally, I implemented error handling mechanisms to provide users with informative feedback when issues arose.

The database structure, a fundamental component of the system, was carefully designed to accommodate menu items (burgers), customer orders, and order items. This schema allowed for organized storage of data, essential for maintaining menu integrity and tracking orders accurately.

On the server side, PHP scripts managed the logic of the application, handling tasks such as order placement, menu updates, and user authentication. Implementing authentication mechanisms ensured that only authorized users, including administrators, could access certain features.

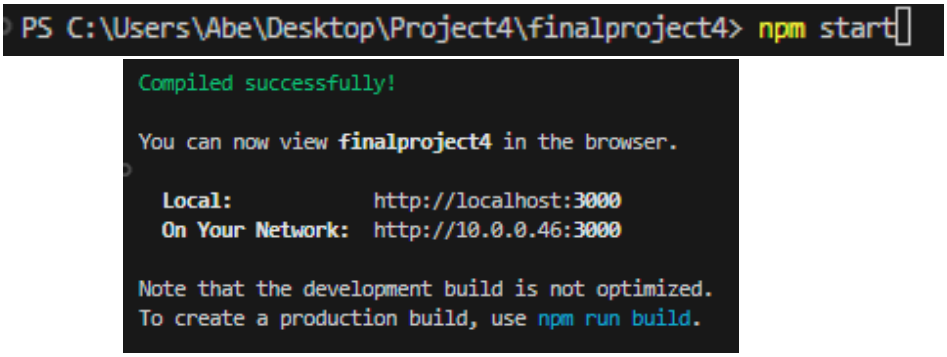
The project's success is evident in the final product. Customers can effortlessly navigate the menu, customize their burger orders, and receive real-time confirmation. The order management system provides administrators with a centralized platform to track and fulfill orders efficiently.

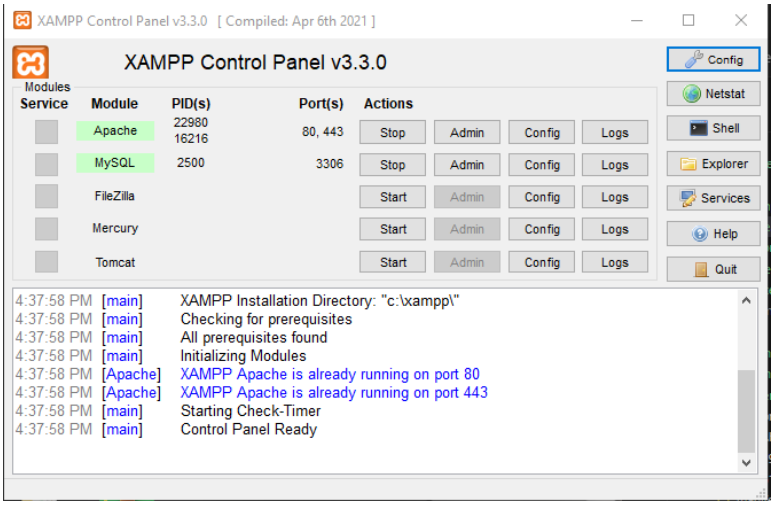
In conclusion, this project has been a transformative experience. It has deepened my understanding of web development and allowed me to witness the impact of technology on the food industry. Beyond the technical aspects, I've honed problem-solving skills and learned to balance user-centric design with efficient back-end architecture.

Tutorial

In the tutorial section of the project, I have incorporated visual aids to enhance the learning experience. Two illustrative screenshots of the home page are included in the code to provide step-by-step guidance for users.

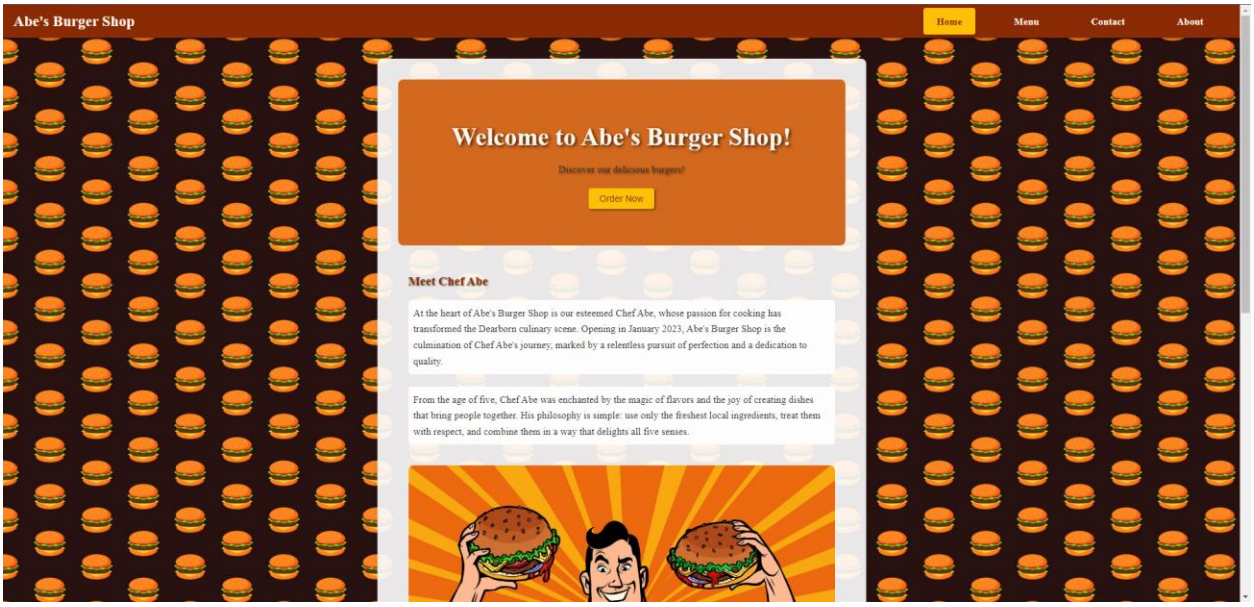
- Before we show the website, in order to run the program, we have to go to the directory that the project is in, in this case \finalproject4. Then, we run the program using 'npm start' and it should run successfully. We also need XAMPP Control Panel open in order to show ONLY the Menu content on Website by starting Apache and MySQL because it includes a database.

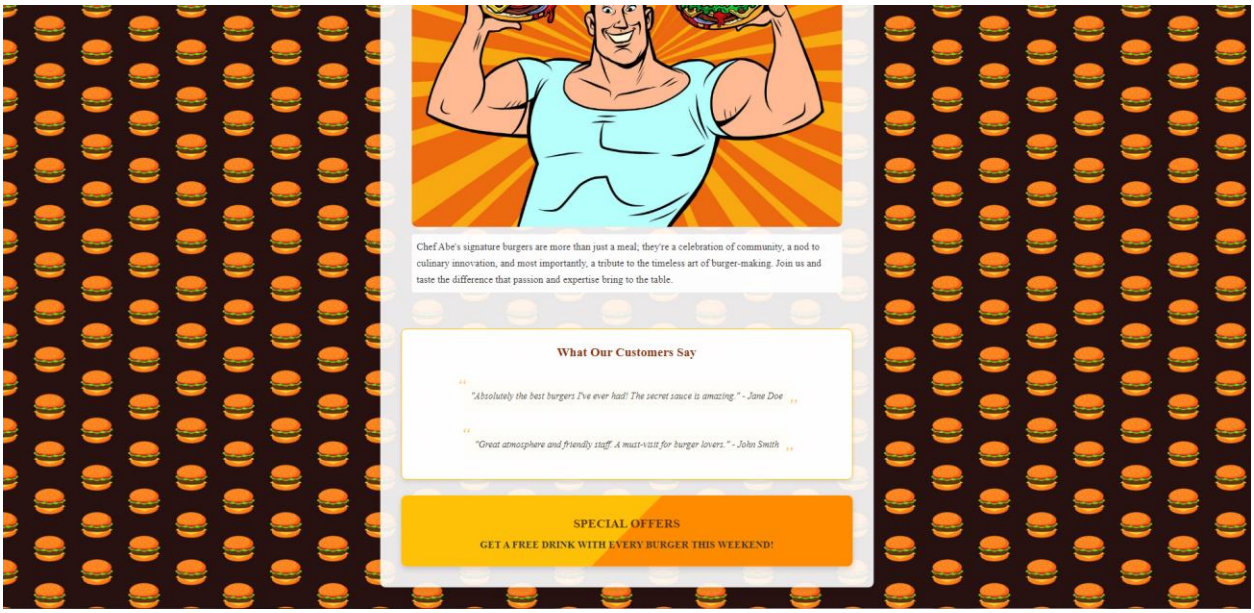




Home Page:

- The Home Page serves as the initial landing point for users visiting the website. It offers a clean and inviting interface that provides a brief introduction to the application's purpose.
- Users are welcomed with an eye-catching banner and a concise description of the services offered.
- Navigation is intuitive, with a prominent "Order Now" button that directs users to the Menu Page.



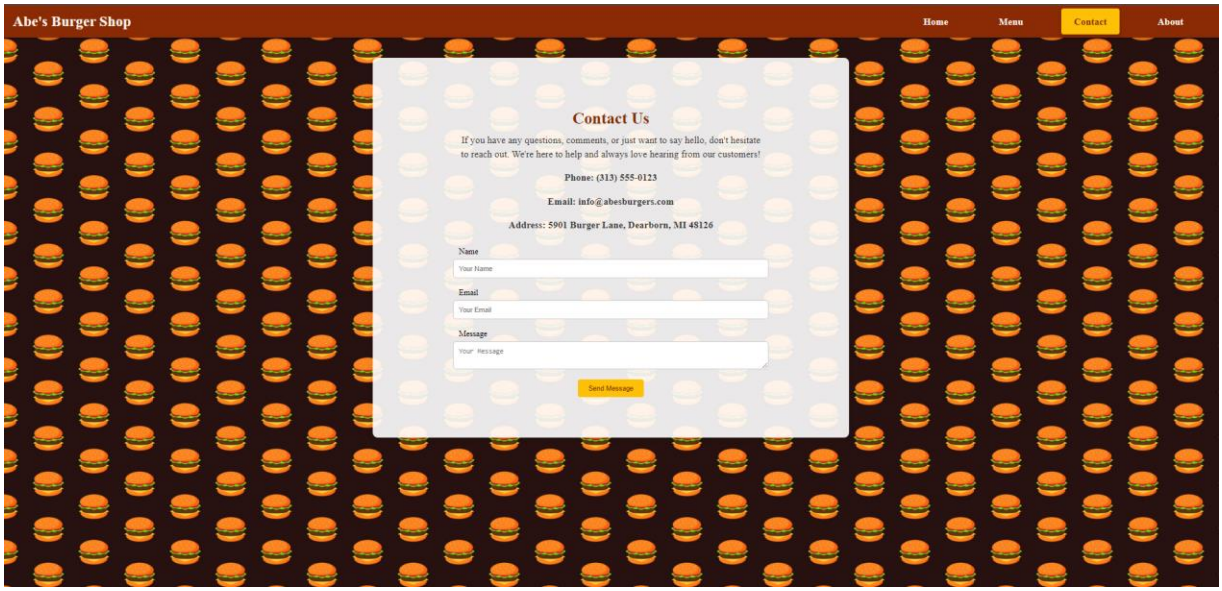


Menu Page:

- The Menu Page is the heart of the application, presenting an enticing array of food options.
- It features high-quality images of menu items, accompanied by detailed descriptions and pricing.

Contact Page:

- The Contact Page is designed to facilitate communication between users and the business. It provides essential contact information, including phone numbers and email addresses.
- A user-friendly contact form allows visitors to send inquiries or feedback directly from the page.

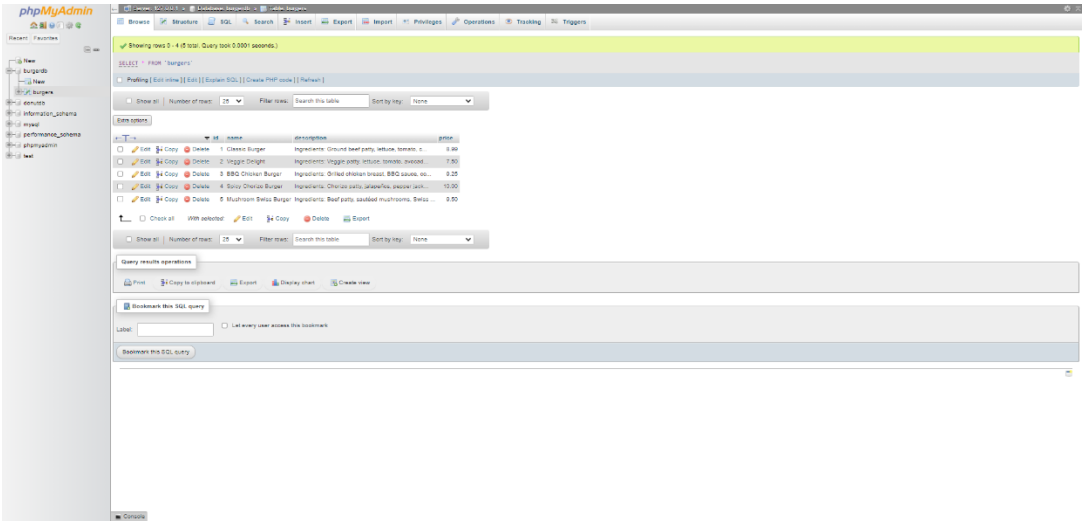


About Page:

- The About Page offers insights into the background and mission of the business.
- It includes engaging content, such as the story behind the establishment, the team's dedication, and the values that drive the business.

phpMyAdmin Page:

- The phpMyAdmin Page is a powerful database management tool used for the backend of the application.
- It allows administrators to interact directly with the database, making updates, running queries, and ensuring data integrity.
- While not visible to users, it plays a crucial role in maintaining the application's functionality.



Admin Page (admin.html):

- The Admin Page is a secure and private section of the application accessible only to authorized personnel.
- It provides administrative tools for managing orders, updating the menu, and overseeing user interactions.
- Robust security measures are in place to protect sensitive data and ensure seamless management.