UTM Computing Proceedings
Innovations in Computing Technology and Applications
Volume: 6 | Year: 2024 | ISBN No. : 2976-2278

ASZ Frozen Food Ordering System

Nor Farahziba Binti Hamadun and Lizawati binti Mi Yusuf
Faculty of Computing
Universiti Teknologi Malaysia
81310 Skudai, Johor Bahru, Malaysia
norfarahziba@graduate.utm.my, lizawati@utm.my

***Abstract*— During the Covid-19 pandemic, ASZ Frozen Food, a small frozen food business, faced challenges due to its manual approach. The business relied on WhatsApp and Facebook for communication and commercial platforms, leading to inefficiencies and time-consuming issues. Thus, this project aims to introduce an effective frozen food ordering system that simplifies order collection and improves customer service. The system uses hashing, access control, and the Waterfall model, and has undergone three types of testing, including white-box, black-box, and user acceptance testing (UAT). The project benefits ASZ Frozen Food by optimizing its ordering process, reducing the risk of losing potential customers, and storing order records in a database to prevent damage or loss.**

Keywords — UAT, ASZ Frozen Food, order, delivery

I. INTRODUCTION

Frozen foods are stored in the freezer until ready to be used, preserving vitamins, minerals, carbohydrates, protein, and fat contents. They reduce food waste and save time and money for sellers by preventing wastage of materials and money. Frozen food gained popularity during the Covid-19 pandemic due to its simplicity and affordability. Despite the pandemic, the need for frozen food is expected to continue, as buyers spend more time with their families and enjoy meals at home. Frozen food offers meal planning, cooking innovation, time-saving, and low-cost qualities, attracting customers and increasing sales. Additionally, remaining frozen food can be refrozen and sold again, saving time and money for sellers.

Many people started frozen food businesses, same as ASZ Frozen Food, which began with a small operation selling 'pau' in seven flavors. The business expanded to include pizza, curry puffs, and buns, hiring more employees and selling their own products. ASZ Frozen Food uses WhatsApp and Facebook as communication and commercial platforms, but faces challenges from both the seller's and customer's perspectives. The seller faces difficulties in checking order limits, collecting customer data, and notifying customers about order declines or insufficient stock. They also face issues with miscalculations of order quantities, recalculating orders, and ensuring accurate business performance analysis. Additionally, the seller may not have enough workers to produce the food, and customers may not be aware of expiration dates or stock availability. Blacklisted customers can still chat with the seller and place orders. On the other hand, the customer is unsure of food availability, must manually notify the seller about cancellations and additions, and may not receive enough food due to calculation errors. Customers also need to provide addresses and request total payment, including additional costs.

The project aims to introduce a reliable frozen food ordering system, simplifying order collection, and improving customer service. It focuses on identifying user requirements, designing, and developing the system, and testing its functionality. The system will only serve frozen food, and delivery will be handled by ASZ Frozen Food workers. This system uses a simulation of an online transaction rather than actual online transactions that require a payment gateway.

The project can improve ASZ Frozen Food's ordering process by reducing customer loss risk and improving customer satisfaction. The system stores order records in a database, preventing damage or loss. It differs from food ordering services like FoodPanda, GrabFood, and ShopeeFood, that deliver frozen food right away. The system is customized for ASZ Frozen Food's unique needs, as online shopping platforms like Shopee and Lazada often sell non-frozen items.

II. LITERATURE REVIEW

The current operation of the ASZ Frozen Food involves customers choosing their preferred food and placing an order. They notify the owner via WhatsApp or Facebook, and can cancel or change the order. If cancellation occurs, the owner adds the item to the stock list. If no cancellation occurs, the order is picked up. If the food is unavailable, the owner informs the customer. Next, if the total order exceeds the stock limit, the owner closes the order. The chef prepares the food, replaces expired items, and confirms enough stock. If there is a shortage, the owner informs the remaining customers and rejects or postpones orders. The owner negotiates delivery mode, day, and time for the customer, and the customer must wait for delivery. The total price of the food, including extra charges, is

then passed to the runner. Payment options include cash or online transactions.

This manual approach to business expansion presents challenges, including inefficiency, unproductiveness, and time-consuming tasks. The owner must collect customer order details from platforms like Facebook and WhatsApp, double-check information, and announce order declines. Miscalculations of order quantities and reminders of total costs can be time-consuming. The chef may overlook expiry dates and stock availability, leading to potential bans on orders. Consumers may be confused about food availability and need to inform the owner of cancellations or changes. Insufficient food received due to calculation errors can lead to confusion and additional costs.

However, it is solved with the help of proposed system. The proposed system solution utilizes a checker to prevent orders from being executed when they are out of stock, allowing sellers to choose early customers. The system displays food and remaining stock, allowing customers to know the remaining amount for each food. It also calculates the total payment, including additional costs, eliminating the need for manual calculations. The system also generates payment receipts for sellers and customers, reducing manual labor. There are two cancel buttons for sellers and customers: one for the seller, which generates a message informing the customer about the cancellation, and another for the customer, which allows them to notify the seller without texting. The system also provides two notifications: one for the expiration date of the food (1 to 3 days before the expiration date) and another for the food being almost out of stock (3 food left). For delivery, the system displays the customer's address and phone number, eliminating the need for texting. The system also advertises job requests on the homepage, and has a banning feature to prevent blacklisted customers from placing orders until they are unblocked. The system allows administrators to access food sales records, track business performance, and analyze reports for better decision-making.

Next, the proposed system was compared to existing applications like GrabFood[1], Foodpanda[2], Shopee[3], and Lazada[4] to understand their capabilities and originality. These apps serve the same purpose of allowing customers to easily explore restaurants and order their preferred dishes via website or mobile. However, they are not suitable for ASZ Frozen Food, as they send food immediately after the order is completed and processed. Most of these apps provide restaurant-style meals, but some features are identical to the proposed application.

Lazada and Shopee are popular e-commerce shopping apps in Southeast Asia in 2019, but they offer more features like product availability, stock alerts, and business performance tracking. However, they can sell a wide variety of products, making them unsuitable for the proposed application. Additionally, none of the four existing applications can broadcast a seller's request to recruit a worker, as they are used for buying goods and services.

The ASZ Frozen Food Ordering System is a web-based application that requires web hosting to access it via the internet. A web database, such as MySQL [5], is used for storing information about customers, owners, employees, and sales. A framework Laravel [6] was chosen for this project to speed up the development process. The system must validate user passwords using uppercase, lowercase, numeric, and special characters. Hashing functions [7] are used to maintain data integrity, with the system code hashing the password and saving it into the database. Access control [8] is crucial for data security, ensuring authorized access to business information. The One-Time Password (OTP) [9] security method is used to verify a user's phone number, as people are more likely to check their phone messages than their emails. This method helps to prevent users from falsifying their contact information and ensures the security of the system.

III. METHODOLOGY

The waterfall model [10] is a linear sequential SDLC methodology used in software development, ensuring that each step is completed before moving on to the next. This approach is easy to understand and maintain, as it follows a uniform sequence of phases. The project begins with requirements analysis, which examines specifications, feasibility, and any constraints. System design specifies hardware and system needs, and source code is developed according to requirements. The system is built in units, tested, and tested to ensure it meets all requirements. Quality assurance is provided through corrections to weaknesses and problems discovered during testing. The system is deployed into a client's server for end-user access, and maintenance services are offered to ensure proper operation. The waterfall model involves six phases: requirements analysis, system design, implementation, testing, deployment, and maintenance.

A. Requirements Analysis

The first phase involves gathering stakeholder data through interviews, such as with the owner of ASZ Frozen Food. This data is used to establish user requirements and understand the proposed system. It focused on gathering stakeholder data, identifying background problems, and analyzing existing systems. The proposed system is compared to similar applications to highlight its advantages and uniqueness. The system must satisfy user requirements and minimize unnecessary features.

B. System Design

The system requirements are defined based on analysis, and the next phase involves constructing the system. Identifying hardware and software requirements is crucial. Diagrams like use case, sequence, and activity diagrams describe the system's behavior, functionality, and scope. Additional diagrams like UML, ERD, and data dictionary diagrams help understand database table relationships.

C. Implementation

The system design phase involves developing inputs into units, which are tested and implemented in the implementation phase. Unit testing ensures the system design becomes source code with functional program modules.

D. Testing

Testing ensures a balanced and useful system, preventing disruptions or faults during production. It can be manual or automated. After code completion, the product is tested, merged, and continuous examination is conducted. User involvement ensures satisfaction, and vulnerabilities and faults are corrected for quality assurance.

E. Deployment

Deployment phase completes product testing, deployment to client server for performance testing, end-user access, and education. System is installed, migrated, and supported in user environment, making it available to market.

F. Maintenance

The waterfall model's final phase involves addressing user issues and improving the system. This phase focuses on resolving issues raised during deployment and testing, such as faults or flaws discovered during live system usage or user requests.

The proposed system utilizes Laravel framework, Hostinger[11], and Visual Studio Code [12] technologies. Laravel is an open-source PHP framework that simplifies tasks like authentication, routing, sessions, and caching, making development easier. Hostinger is a reliable and affordable web hosting service suitable for startups. The Premium Plan of Hostinger provides free SSL for each domain, enhancing security and data transmission between browsers and servers. Visual Studio Code is a simplified code editor that supports development tasks, such as building, debugging, and executing websites and cloud apps. However, understanding the basic requirements for the system is crucial for its proper development and functionality. Tables I and Table II outline the minimum hardware and software requirements for the proposed web application.

TABLE I. THE MINIMUM HARDWARE REQUIREMENTS

Hardware	Specification
Processor	Intel Core i5, 2.30GHz or above
Memory RAM	At least 8GB
Hard drive	At least 1TB storage HDD or 256GB SSD
I/O Devices	Mouse and keyboard
Internet connection	WiFi or mobile data

TABLE II. THE MINIMUM *SOFTWARE* REQUIREMENTS

Software	Specification
Operating System	Windows 9 and above
Web Browser	Google Chrome, Firefox
IDE	Visual Studio Code
Front-end Tools	Laravel
Back-end Tools	MySQL

IV. REQUIREMENT ANALYSIS AND DESIGN

The requirement analysis includes use case diagrams, sequence diagrams, activity diagrams, and class diagrams to summarize information about a system and its users.

A. Use Case Diagram

A use case diagram represents each actor's activity, while a sequence diagram represents the system's events. The use case diagram for the entire ASZ Frozen Food Ordering System is shown in Fig 1.

Figure 1. The use case diagram for the proposed system

B. Sequence Diagram

A sequence diagram illustrates the ASZ Frozen Food Ordering System's interaction with objects, including user interactions, information acquisition, and food order fulfillment. The ASZ Food Ordering system requires users to input their contact number and password, which are then confirmed by the login controller. Successful logins direct users to the home page, while unsuccessful logins return them to the login page. Fig. 2 displays the sequence diagram for user login.

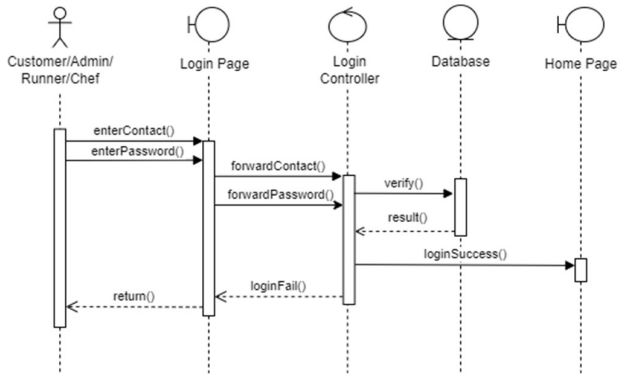

Figure 2. The sequence diagram for user login

However, users who have not yet registered must register before logging in to the system. They input personal information, including contact number and password, on the register page. The system sends a confirmation code (OTP), confirming registration and storing information in the database. Invalid codes return users to the registration page. Fig.3 shows the sequence diagram for register.

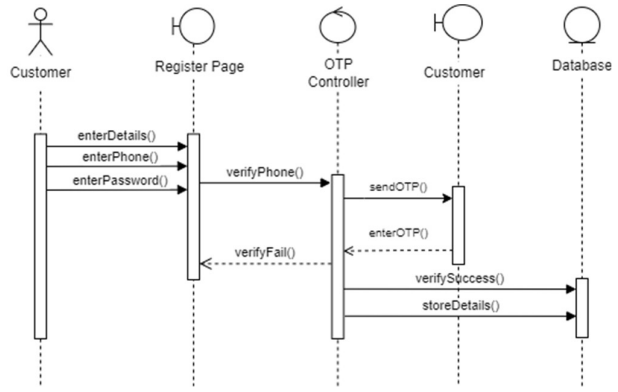

Figure 3. The sequence diagram for register

Then, the food ordering sequence diagram begins with the customer logging in and navigating to the home page. They can order food by viewing available options, selecting desired food, flavor, and quantity, and adding it to their cart. Once confirmed, the order is saved in the database and removed from the cart. The order is sent to the admin for approval, and if approved, the chef prepares the food. If rejected, the system updates the order as a cancelled order and notify the customer

.

C. *Activity Diagram*

The ASZ Frozen Food Ordering System uses an activity diagram to illustrate its flow of control. The diagram is divided into three sections: customer, system, and staff. The customer logs in, and the system validates their contact information. They can view their cart, select food, or view their order. The admin accepts the order, and the chef prepares it before the runner delivers it safely. The activity diagram is similar to a flowchart and helps visualize the system's flow of control.

D. *Class Diagram*

The class diagram represents the static view of ASZ Frozen Food Ordering System, describing its characteristics, actions, interfaces, associations, collaborations, and limitations, analyzing and designing the system's responsibility.

E. *Project Design*

The website requires user login verification and password encryption using a hash function. Only authorized users can access the system, and admin, chef, runner, and customer have access to their functions. The system restricts user access by requiring a contact number for login and a password verification and hashing procedure. If a user forgets their password, an OTP code is sent for verification. The web page is displayed when data requests are delivered to the customer, and all customer information and orders are saved in the database. Data is extracted to display relevant information for the ordering, confirmation, and fulfillment pages. The system allows runner, admin, owner, and chef to access the system via their mobile devices to fulfill orders, allowing them to access customer and order information.

F. *Database Design*

Databases are crucial for storing, managing, and retrieving data in a proposed system. They store information about users, orders, and food, allowing for examination and evaluation. Database design determines data storage and interaction. An Entity-Relationship Diagram (ERD) illustrates the relationships between entities, including one-to-one and one-to-many relationships, using visual aspects like Order, Admin, Customer, Chef, and Runner.

G. *Interface Design*

UI design, also known as user interface design, is a crucial aspect of the proposed system. It focuses on the user's interaction with the system, ensuring simplicity and ease of use. The system begins with a login or registration interface, where customers register and create a new account. Once logged in, they can access their cart, select their preferred flavor, and place an order. The total price of selected foods is displayed, and customers can remove or pick only the required food before checking out. Customers can also search for food by clicking the magnifying glass button, select their preferred flavor, and place an order. The chosen food is then placed in the cart, and the total price of all selected foods is displayed. The customer can also remove or pick only the required food before checking out. Before placing an order, customers must input their delivery address, select a shipping method, and submit a proof of payment on the checkout interface. A payment receipt for the customer's purchases is generated.

After placing an order, the order is submitted to the admin for approval. Besides, the user can change their profile by clicking the person symbol on the home page. Chefs have authority over the food product, adding or deleting products,

and updating product information. Each approved order is displayed on the "Order" page.

Chefs prepare the food based on the order displayed, collect it from the freezer, pack it, and hand it to the runner. Once the preparation is complete, the chef clicks the "Done" button on the "Order" page. The "Delivery Order" option on the main page displays all orders given to the runner. When the runner is ready to deliver the product, they click the "Accept" button. The "On The Way" page displays the accepted order delivery, and the proof of delivery form is completed before the "Submit" button is clicked. The system also generates a sale report for the owner to monitor the business's performance weekly and monthly. The admin has additional authority to view customers, view staff, and manage staff.

V. IMPLEMENTATION AND TESTING

This section discussed the system's main function, including the security elements used, food ordering function, order status function, food management function, user management function, sales report function, and notification function.

A. Security Elements

There are four security elements implemented in the system which are password strength verification [12], One-Time Password (OTP), password hashing, user authentication and verification and access control.

1) Password Strength Verification

Customers must register before accessing the ASZ Food Ordering System, and their information, including their password, must be entered. A simple password can be easily guessed by unauthorized users, and brute-force methods can be used. Complex passwords are more difficult to crack, so it is crucial to verify their complexity. A password must be longer than 7 characters, uncompromised, and a combination of letters, numbers, and symbols. Completion of the signup process ensures the protection of personal information and accounts from unauthorized access.

2) One-Time Password (OTP)

The system uses a contact number to verify users, as people are more likely to check their phone messages than emails. However, some customers may fake their contact number to meet registration requirements. The One-Time Password (OTP) security method is used to verify the customer's contact number, sending a six-digit code via SMS to confirm its validity. The customer can complete registration only after the code is verified. The OTP is also used to reset users' passwords, requiring their mobile number to authenticate as an authorized user. If the contact number exists in the database, the user can request an OTP and reset their password by providing the correct OTP code. The OTP is performed using Google Firebase Authentication service.

3) Password Hashing

Hashing is a process where a user's password is hashed before being stored in a database. This encryption algorithm converts the password into a short string of letters and numbers, protecting it from being exposed to attackers. In case of a system hack, the attacker only obtains the hashed password, not the original password.

4) User Authentication and Verification

Users must log into the system after registration to access its contents, ensuring authorized access. They must input their contact number and password to verify their identity. The system checks the contact number in the database before comparing the password. If the hashed password matches, the user is authenticated and directed to the main page.

5) Access Control

The system has four user types: customer, chef, runner, and admin. Users are assigned roles and access data differently based on their type. Customers can place food orders, check order status, and edit their profiles. Chefs manage food, view orders, and edit their profiles. Runners accept deliveries and change their profiles. Administrators can approve orders, manage staff, view customer details, and change profiles. Access control regulates who can access resources and data.

B. Food Ordering Function

The Frozen Food Ordering System focuses on food ordering, allowing customers to add desired food, variations, and quantities to their cart. They can manage their cart by editing quantities, deleting unwanted items, or selecting items. To place an order, customers must verify their address and select a payment method. After payment, they wait for admin approval before placing their order. Fig. 4 shows the UI when customer select food to order.

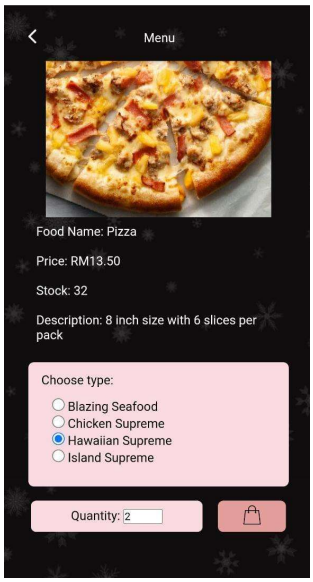

Figure 4. The UI for select food

C. Food Management Function

The system relies on food, so food is the most crucial component, and chefs can add, modify, or delete food. Chef input information about the food, including its name, description, variety, price, and stock. The added food is displayed on the customer's home page, and the chef must restock and update the system to keep the customer's interface updated on stock. Fig. 5 displays the UI for chef to manage food.

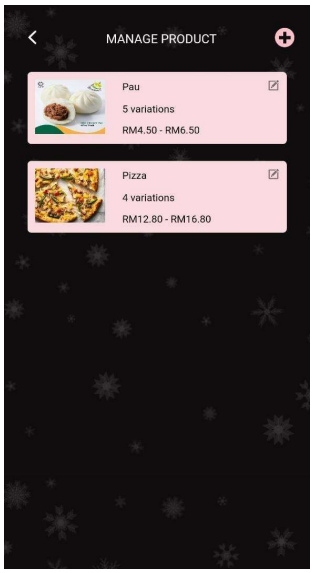

Figure 5. The UI for manage food

D. User Management Function

Manage staff is an essential part of the system, as staff handle food and customer orders. It is crucial to add new staff, modify staff information, and remove existing ones. Only the administrator has authority to manage staff, but certain information, like contact numbers and identity cards, cannot be changed once registered. Fig 6 shows The UI for admin to manage staff.

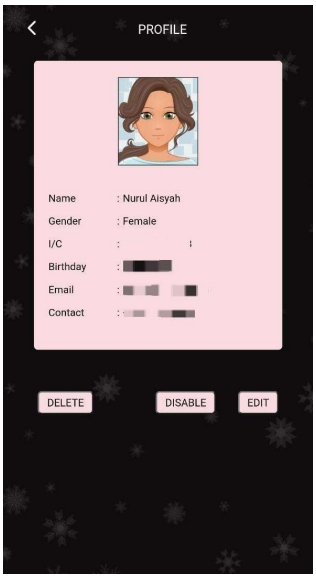

Figure 6. The UI for manage staff

E. Sales Report Function

ASZ Frozen Food owner uses Google Charts to create a sales report for tracking orders and developing a successful strategy. The report includes a line chart of food sales and a pie chart of food sold by variation and category.

F. Testing

The ASZ Food Ordering System undergoes testing after the development phase to ensure its functionality and accuracy before public release. The testing are black-box testing, white-box testing, and user acceptance testing. White-box testing involves understanding the internal system's functions and identifying coding mistakes, implementation issues, and logical issues. It confirms code behavior and functionality of each system component. Black-box testing, on the other hand, is accessible to anyone without programming knowledge or internal system implementation. It involves entering input and observing the output, ensuring the system performs as intended and evaluating external behavior and functionality from the user's perspective. User Acceptance Testing (UAT) is a final check that requires the end user to validate the system before it is officially used. UAT ensures that the system's functionality is performed as expected and meets the ASZ Frozen Food business requirement.

Participants are given a briefing and instructions on how to use the ASZ Frozen Food Ordering system, and are free to explore all its features related to their role. After exploring and examining the system, participants must complete Google Forms to provide feedback on the system. The form gathers feedback, summarizes it in a chart, and makes it ready for analysis. From the information collected, fifteen participants performed the UAT: eight customers, two admins, three chefs, and two runners. The system evaluation was split into system functionality feedback, overall feedback, and rating on the system.

The data shows that customers are most interested in the food selection function (six votes), followed by the security

element. The contact number verification function has the second-highest percentage of respondents, with five respondents voting for it. The announcement function has the lowest rate, with only one respondent voting for it. The staff management function receives a full score percentage of 100%, followed by order approval and sales report functions. The admin is most interested in staff management features, such as adding, modifying, deleting, and disabling staff. The chef's feedback shows that two out of three respondents are interested in the food management function, followed by food preparation and notification functions. The delivery acceptance and proof of delivery functions are most appealing to respondents (runners).

Based on overall feedback on system functionality, it is reasonable to conclude that each user is more interested in the system's primary function relevant to their role than the additional function. Security elements are also a vital component of the system. In conclusion, the ASZ Frozen Food Ordering System is highly satisfied by eleven respondents, with nine stating it meets expectations and is simple to use. No respondents rate the system's user interface or attractiveness as "Somewhat Satisfied." Users in all roles agree that the system is attractive and user-friendly. Nine respondents rated the system five stars, while six rated it four stars. No respondent rated below four, indicating satisfaction with the system.

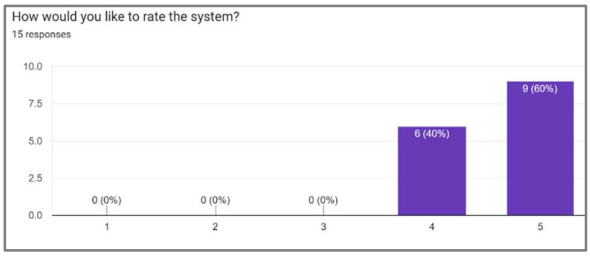

Figure 7. Overall rating of the system

VI. CONCLUSION

In conclusion, the ASZ Frozen Food ordering system is an effective solution for managing orders and providing customers and staff with a convenient ordering experience. The system's advantages and disadvantages are determined through literature review, organizational structure, and technology comparison. The project follows a waterfall model, using Laravel framework, Hostinger, and Visual Studio Code technologies. Then, the system's interface was designed, and the system was tested and deployed. User feedback is collected and examined for system functionality from the UAT.

To enhance user satisfaction and business outcomes, future improvements are needed for the ASZ Frozen Food Ordering System. These include improving the user experience, responsiveness to different devices, and security. The system should have filter buttons for approval, order, and delivery history, allowing users to sort by date, status, and address. Additionally, the interface dimension should be upgraded to accommodate different screen sizes, ensuring efficient access and interaction. Security measures should be implemented to prevent hackers from accessing data. Secure coding, such as input validation, output encoding, and parameterized queries, can be used to prevent SQL injection and cross-site scripting. OTP can be used for two-way authentication, and URL encryption is crucial for preventing URL poisoning attacks.

REFERENCES

[1] GrabFood Food Delivery Service - Order Food Online. (2022). Grabfood MY. Retrieved June 23, 2022, from https://food.grab.com/my/en/

[2] Order online ~ Express Food Delivery in Malaysia | foodpanda. (2019). Foodpanda.my. https://www.foodpanda.my/

[3] Shopee. (2019). Shopee.com.my. https://shopee.com.my/

[4] Lazada.com.my: Best Online Shopping in Malaysia. (2019). Lazada.com.my. https://www.lazada.com.my/

[5] What is MySQL? A Beginner-Friendly Explanation. (2022, April 11). Kinsta. https://kinsta.com/knowledgebase/what-is-mysql/

[6] Otwell, T. (2019). Introduction - Laravel - The PHP Framework For Web Artisans. Laravel.com. https://laravel.com/docs/4.2/introduction

[7] Arias, D. (2018, April 25). Hashing Passwords: One-Way Road to Security. Auth0 - Blog. https://auth0.com/blog/hashing-passwords-one-way-road-to-security/

[8] Piscitello, D. (2015, December 2). What is Authorization and Access Control? Www.icann.org. https://www.icann.org/en/blogs/details/what-is-authorization-and-access-control-2-12-2015-en

[9] Richards, K., & Wigmore, I. (2021, September). *What is a one-time password (OTP)?* SearchSecurity. https://www.techtarget.com/searchsecurity/definition/one-time-password-OTP

[10] What is Waterfall Model? Pros and Cons. (2019, May 20). Testbytes. https://www.testbytes.net/blog/waterfall-model/

[11] *Go Online With Hostinger*. (2011). Hostinger. https://www.hostinger.my/

[12] Visual Studio Code. (2016, April 14). Visualstudio.com. https://code.visualstudio.com/docs/supporting/faq

[13] Mombrea, M. (2013, June 11). How to validate password strength using a regular expression. Computerworld. https://www.computerworld.com/article/2833081/how-to-validate-password-strength-using-a-regular-expression.html