Mach Lagbe:
An Online Fish Marketplace

by

Azizur Rahaman
ID: CSE2203027169

Morium Hossain Chaity
ID: CSE2203027071

Marufa Akhter Meem

ID: CSE2203027174

MD. Shihab Islam Patwary

ID: CSE2001019057

Supervised by
Prof. Bulbul Ahamed

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2026

Mach Lagbe:
An Online Fish Marketplace

by

Azizur Rahaman
ID: CSE2203027169

Morium Hossain Chaity
ID: CSE2203027071

Marufa Akhter Meem

ID: CSE2203027174

MD. Shihab Islam Patwary

ID: CSE2001019057

Supervised by
Prof. Bulbul Ahamed

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2026

APPROVAL

The project titled "**Mach Lagbe: An Online Fish Marketplace**" submitted by Azizur Rahaman (CSE2203027169), Morium Hossain Chaity (CSE2203027071), Marufa Akhter Meem (CSE2203027174) and Md. Shihab Islam Patwary (CSE2001019057) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and content.

Board of Examiners

Bulbul Ahamed — **Supervisor**
Professor and Head
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 1**
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 2**
Department of Computer Science and Engineering
Sonargaon University (SU)

(Examiner Name and Signature) — **Examiner 3**
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Bulbul Ahamed,** Professor and Head Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Bulbul Ahamed)
Professor and Head

Signature

Azizur Rahaman
ID: CSE2203027169

Morium Hossain Chaity
ID: CSE2203027067

Md. Shihab Islam Patwary
ID: CSE2001019057

Marufa Akhter Meem
ID: CSE2203027174

ABSTRACT

The growth of digital commerce has created opportunities to improve how fresh food products are presented, ordered, and managed. Traditional fish buying often requires customers to visit physical markets, compare products manually, and place orders through informal communication, while sellers may struggle to manage product information, stock availability, and customer orders efficiently. To address these challenges, this project introduces “Mach Lagbe: An Online Fish Marketplace,” a web-based platform that provides a convenient digital environment for browsing fish products and placing orders. The system connects customers to a centralized marketplace where fish information, prices, availability, and orders can be managed in an organized manner. The platform is developed using React.js for the frontend, Node.js and Express.js for backend services, and MongoDB for database management. It includes user registration and login, fish/product browsing, search and filtering, cart and ordering functionality, order history, and an administrator panel for managing products and orders. Authentication and basic validation are implemented to control access to protected operations. Overall, the project demonstrates how modern web-based e-commerce architecture can be applied to a fish marketplace. By centralizing product information and order management, Mach Lagbe aims to make fish shopping more organized, accessible, and user-friendly while reducing the administrative effort required to manage products and customer orders.

ACKNOWLEDGEMENT

First and foremost, we express our deepest gratitude to the Almighty for granting us the strength, patience, and perseverance to complete this project within the scheduled time.

We are profoundly indebted to our supervisor, **Bulbul Ahamed, Professor and Head**, Department of Computer Science and Engineering, Sonargaon University, for his thoughtful guidance, constructive feedback, and unwavering support throughout the development of Class Matrix. His insights at every stage from problem framing to architectural decisions have shaped this work in ways that go far beyond what these pages can capture.

We extend our sincere thanks to the faculty members, and the staff of the Department of Computer Science and Engineering at Sonargaon University. Their teaching across the four years of our undergraduate study laid the foundation for this project, and their continued encouragement made it possible to attempt something at this scale.

We also thank our peers, classmates, and the wider open-source community whose tools, libraries, and freely shared knowledge made the technical work tractable within an academic timeline.

Finally, our heartfelt gratitude goes to our parents and families for their patience, sacrifices, and endless love. None of what follows would exist without them.

LIST OF ABBREVIATIONS

API	Application Programming Interface
CSS	Cascading Style Sheets
DBMS	Database Management System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JWT	JSON Web Token
REST	Representational State Transfer
UI	User Interface
UX	User Experience

TABLE OF CONTENTS

Title		Page No.
DECLARATION		iii
ABSTRACT		iv
ACKNOWLEDGEMENT		v
LIST OF ABBREVIATIONS		vi
CHAPTER 1		1 – 6
INTRODUCTION		
1.1	Introduction.....	1
1.2	Objectives……..	2
1.3	Problem Statement.....	3
1.4	Why Mach Lagbe	4
1.5	Scope of the Project.....	5
1.6	Limitations.....	6
CHAPTER 2		7 – 14
FEATURES AND FUNCTIONS OF THE PROJECT		
2.1	Features of the Project.....	7
2.2	User Registration and Login	8
2.3	Fish/Product Browsing.....	9
2.4	Search and Filtering.....	10
2.5	Cart and Ordering	11

Title Page No.

2.6 Order History..... 12

2.7 Admin Panel..... 13

2.8 Product and Order Management..... 14

CHAPTER 3 15 – 17

TOOLS AND TECHNOLOGY

3.1 Frontend Technologies..... 15

3.2 Backend Technologies..... 16

3.3 Database Management System..... 16

3.4 Authentication and Security..... 17

3.5 Development Tools..... 17

CHAPTER 4 18 – 21

REQUIREMENT ANALYSIS, DESIGN AND IMPLEMENTATION

4.1 Requirement Analysis..... 18

4.2 System Architecture..... 19

4.3 Use Case Description..... 20

4.4 Database Design 20

4.5 Implementation..... 21

CHAPTER 5 22 – 31

PROJECT DESCRIPTION AND USER INTERFACE

5.1 Home Page 22

5.2 Registration and Login 23

5.3 Fish Marketplace..... 24

5.4 Fish Details and Search..... 25

5.5 Shopping Cart..... 26

5.6 Order Placement..... 27

Title Page No.

5.7 Order History 28

5.8 Admin Dashboard 29

5.9 Add/Edit Fish 30

5.10 Order Management 31

CHAPTER 6 32 – 32

CONCLUSION AND FUTURE WORK

6.1 Conclusion..... 32

6.2 Future Work..... 32

REFERENCES..... 33

LIST OF FIGURES

Figure No.	Title	Page No.
Fig. 1.1	Overview of the Mach Lagbe Marketplace	2
Fig. 2.1	Registration and Login	8
Fig. 2.2	Fish Browsing	9
Fig. 2.3	Search	10
Fig. 2.4	Cart and Ordering	11
Fig. 2.5	Order History	12
Fig. 2.6	Admin Panel	13
Fig. 2.7	Product and Order Managements	14
Fig. 5.1	Home Page	22
Fig. 5.2.1	Registration	23
Fig. 5.2.2	Login	24
Fig. 5.3	Product List	24
Fig. 5.4	Product Details	25
Fig. 5.5	Shopping Cart	26
Fig. 5.6	Order Placement	27
Fig. 5.7	Order History	28
Fig. 5.8	Admin Dashboard	29
Fig. 5.9	Add/Edit Fish	30
Fig. 5.10	Order Management	31

CHAPTER 1

INTRODUCTION

1.1 Introduction

Mach Lagbe is a web-based fish marketplace developed to make the process of browsing, selecting, and ordering fish products more convenient through a digital platform. The main idea of the project is to provide customers with a simple online environment where they can explore available fish products without depending completely on physical visits to traditional markets. Customers can view product names, prices, descriptions, availability, stock information, and images before deciding which products they want to purchase. They can also add selected products to a shopping cart and place an order through the website. In addition to customer facilities, the system provides an administrative interface through which administrators can manage fish products and customer orders.

Traditional fish markets are generally based on direct communication between customers and sellers. Although this method is familiar and widely used, customers may need to physically visit different shops to compare fish, prices, and availability. The process can become inconvenient when customers have limited time or when they want to know product information before visiting a market. A digital marketplace can reduce this dependency by presenting product information in one centralized system. Mach Lagbe is designed with this purpose in mind.

The system follows a client-server architecture. The frontend is developed using React.js and provides the user interface through which customers and administrators interact with the application. The backend is developed using Node.js and Express.js and is responsible for handling API requests, authentication, application operations, validation, and communication with the database. MongoDB is used to store application data such as user information, fish/product information, and order-related data. This separation between the frontend, backend, and database provides a clear structure for developing and maintaining the application.

The marketplace is intended to provide a straightforward user experience. A customer can create an account, log in, browse available fish, search for products, add desired items to a cart, and submit an order. After placing an order, the customer can access the order history to review previously submitted orders. On the other hand, an administrator can use the administrative section to add new fish products, edit existing product information, remove products when necessary, and review customer orders.

Another important purpose of Mach Lagbe is to demonstrate the practical implementation of a full-stack web application. The project combines frontend development, backend API

development, database management, authentication, and user interface design into a single application. Therefore, the project is not only focused on online fish shopping but also demonstrates how different web technologies can work together to solve a practical problem.

The system also provides a foundation that can be extended in the future. Although the current version focuses on basic marketplace operations, additional features such as online payment, delivery tracking, mobile applications, product reviews, advanced recommendations, analytics, and improved inventory management can be added later. The current project therefore represents a functional base for a more complete online fish marketplace.

Overall, Mach Lagbe attempts to make fish product information and ordering more organized by moving important marketplace activities into a centralized web-based environment. The system provides separate facilities for customers and administrators while maintaining application data through a centralized MongoDB database. [1], [2]

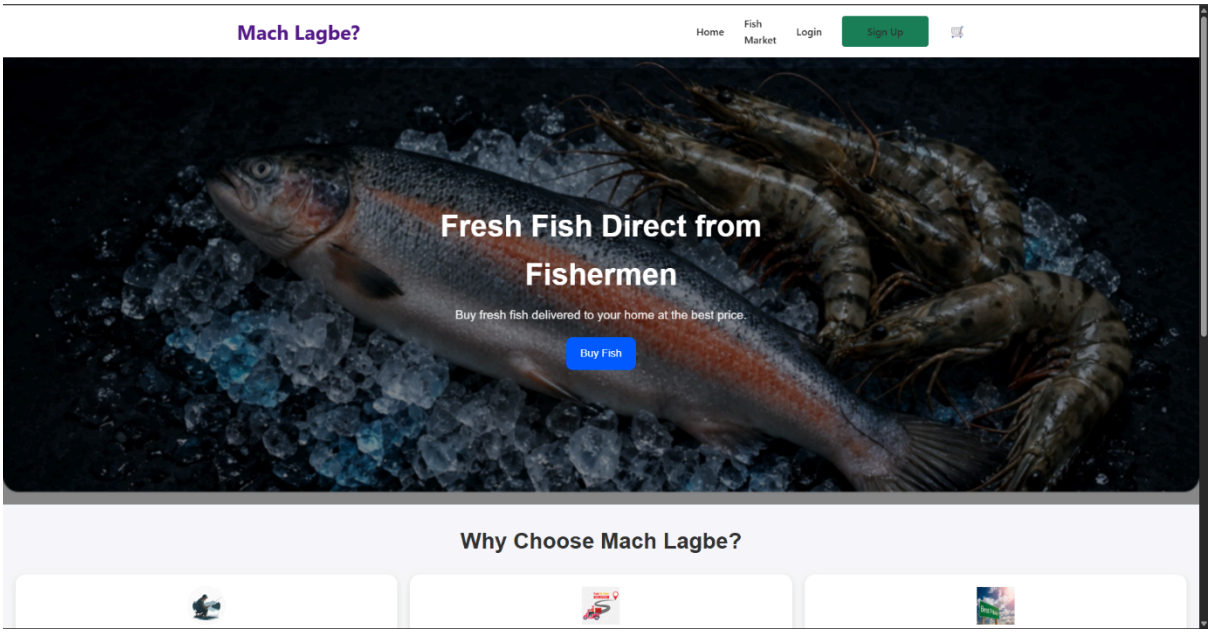


Figure 1.1: Overview of the Mach Lagbe Marketplace

1.2 Objectives

- Provide customers with an online platform for browsing fish products.
- Allow users to register and log in to the system.
- Display fish names, prices, descriptions, stock/availability, and images.
- Allow customers to add fish products to a shopping cart and place orders.
- Provide customers with access to their order history.
- Provide an administrator panel for managing fish products and orders.
- Maintain application data in a centralized MongoDB database.

1.3 Problem Statement

Traditional fish markets generally require customers to visit physical shops to find available products, check prices, compare different fish items, and communicate directly with sellers. This process can be inconvenient when customers want to compare several products or when they do not have enough time to visit a physical market. Product availability may also change frequently, which means that information obtained during one visit may not remain accurate later.

Another challenge is that product and order information may be maintained manually. Manual record keeping can make it more difficult to organize customer orders, review previous transactions, and maintain updated product information. As the number of customers and products increases, managing information manually may become more difficult.

Customers may also have limited access to product information before visiting a market. In a traditional environment, they may need to physically inspect products to know their availability and price. This can make product comparison less convenient. A centralized online platform can address part of this problem by presenting product information in a single location.

Order communication is another area where organization can be improved. When orders are communicated manually through phone calls or informal messages, maintaining a consistent record can become challenging. A web-based ordering system can store submitted orders in a structured database and make those records available to authorized users.

From the administrative perspective, sellers or administrators need a way to manage product information and customer orders. Adding new products, changing prices, updating stock information, and reviewing submitted orders are common marketplace activities. Without a structured system, these activities can require additional manual effort.

Mach Lagbe addresses these problems by providing a centralized online fish marketplace. Customers can browse products, view product information, search for products, add items to a shopping cart, and place orders through the application. The system also provides order history so that customers can review previous orders.

For administrators, the system provides facilities for managing fish products and customer orders. Product information can be added, edited, or deleted, while submitted orders can be reviewed and their status can be managed where supported. Therefore, the project attempts to organize both customer-side and administrative activities within a single application.

The problem addressed by Mach Lagbe can therefore be summarized as the lack of a simple centralized digital environment for browsing fish products, managing customer orders, and maintaining marketplace information. The proposed system does not attempt to completely

replace traditional fish markets; rather, it provides a digital alternative that can make selected marketplace activities more organized and accessible. [1], [2]

1.4 Why Mach Lagbe

The main motivation behind Mach Lagbe is to provide a simple digital alternative for fish shopping. Fish is an important food product, and customers often need information about price, availability, product type, and quantity before making a purchase. A web-based marketplace can make this information easier to access by presenting it through a single platform.

One reason for developing Mach Lagbe is convenience. Customers can browse available fish products from the website instead of depending entirely on physical visits to different shops. They can inspect product information and select products according to their requirements. The shopping cart also gives them an opportunity to review selected products before submitting an order.

Another reason is better organization of marketplace information. Product details, customer accounts, and orders can be maintained digitally. This can make it easier for the system to retrieve and display information when required. The use of a centralized database also provides a structured method for storing application data.

Mach Lagbe is also motivated by the need for a simple administrative workflow. Administrators need to keep marketplace information updated. Through the administrative panel, they can add fish products, edit existing product information, delete products when necessary, and review customer orders. These facilities provide basic control over the marketplace.

The project is also important from an academic perspective. It provides practical experience in designing and implementing a full-stack web application. The development process involves frontend design, backend programming, database management, authentication, API communication, testing, and user interface development.

The project demonstrates how different technologies can be integrated to create a practical application. React.js is used for the frontend, Node.js and Express.js are used for backend operations, and MongoDB is used for persistent data storage. This combination allows the project team to understand how separate components communicate with each other in a modern web application.

Mach Lagbe was therefore selected as a project idea because it represents a familiar real-world activity that can be improved through software. The project focuses on a manageable set of marketplace operations while leaving opportunities for future development. [1], [2]

1.5 Scope of the project

The scope of Mach Lagbe defines the functions that are included in the current version of the application. The project mainly focuses on customer-side fish browsing and ordering along with basic administrative management.

Customer-side Scope

The customer-side functions include:

- Customer account registration.
- Customer login and authentication.
- Browsing available fish products.
- Viewing fish/product details.
- Checking product price and availability.
- Searching for products.
- Using basic filtering options.
- Adding products to the shopping cart.
- Reviewing selected cart items.
- Placing customer orders.
- Viewing previously submitted orders through order history.

The customer interface is designed to provide a simple flow from product discovery to ordering. A user can first browse the marketplace, inspect a product, add it to the cart, and then continue to the order placement process.

Administrator-side Scope

The administrator functions include:

- Administrator login.
- Access to the administrator dashboard.
- Adding new fish products.
- Editing existing fish products.
- Deleting fish products.
- Viewing customer orders.
- Managing order status where supported.

These features allow the administrator to maintain the marketplace and keep product and order information organized.

Technical Scope

The technical scope includes:

- React.js-based frontend development.
- HTML and CSS for structure and presentation.
- JavaScript for application interaction.
- Node.js runtime environment.
- Express.js backend framework.
- MongoDB database.
- Mongoose for database modeling and interaction.
- Authentication and protected routes.
- Password hashing.
- Token-based authentication.
- Basic CORS configuration.
- Server-side validation.
- REST-style backend API communication.

The project does not currently cover advanced logistics, online payment processing, real-time delivery tracking, or large-scale warehouse management. These areas are considered future development opportunities.

1.6 Limitations

- The current project is a basic web marketplace and does not implement advanced delivery tracking.
- Online payment gateway integration is not included in the current version.
- The system depends on network availability when deployed online.
- Advanced recommendation or AI-based personalization is outside the current scope.
- Large-scale logistics, warehouse automation, and real-time driver tracking are not implemented.

CHAPTER 2
FEATURES AND FUNCTIONS OF THE PROJECT

2.1 Features of The Project

Mach Lagbe includes several features designed to support both customers and administrators. The main features are user registration and login, fish/product browsing, search and filtering, shopping cart, order placement, order history, admin dashboard, product management, and order management.

The features are connected to create a complete basic marketplace workflow. Customers can begin by creating an account or logging into an existing account. After authentication, they can browse products and inspect available information. Search and filtering make it easier to locate products. Selected products can then be placed into the cart before an order is submitted.

The system also maintains a record of customer orders. This provides continuity after an order has been submitted because customers can return to the order history page to review previous orders.

The administrator side provides another level of functionality. Administrators can manage product information and customer orders. This separation between customer and administrator functions helps protect management operations from normal customer access.

The major features of the project are:

- User Registration and Login
- Fish/Product Browsing
- Search and Filtering
- Shopping Cart
- Order Placement
- Order History
- Admin Dashboard
- Product Management
- Order Management

Together, these features form the core functionality of Mach Lagbe.

2.2 User Registration and Login

The authentication module allows customers to create accounts and log into the system. Registration provides the basic account information required to identify a customer, while login allows an existing user to access the application.

The authentication process is handled through the backend. Protected routes require valid authentication so that restricted operations are not freely accessible to unauthenticated users. Administrative functions are also protected and are intended to be accessed only by authorized administrators.

The registration page provides a structured form where a new customer can enter the required information. After registration, the user can use the login page to access the marketplace.

Authentication is important because different users have different permissions. A normal customer should be able to browse products and manage their own orders, while an administrator needs additional permissions for product and order management.

The login process also provides a consistent starting point for protected customer activities. After successful authentication, the user can continue to use features such as the cart and order history.

The authentication design therefore supports both usability and basic access control within the application.

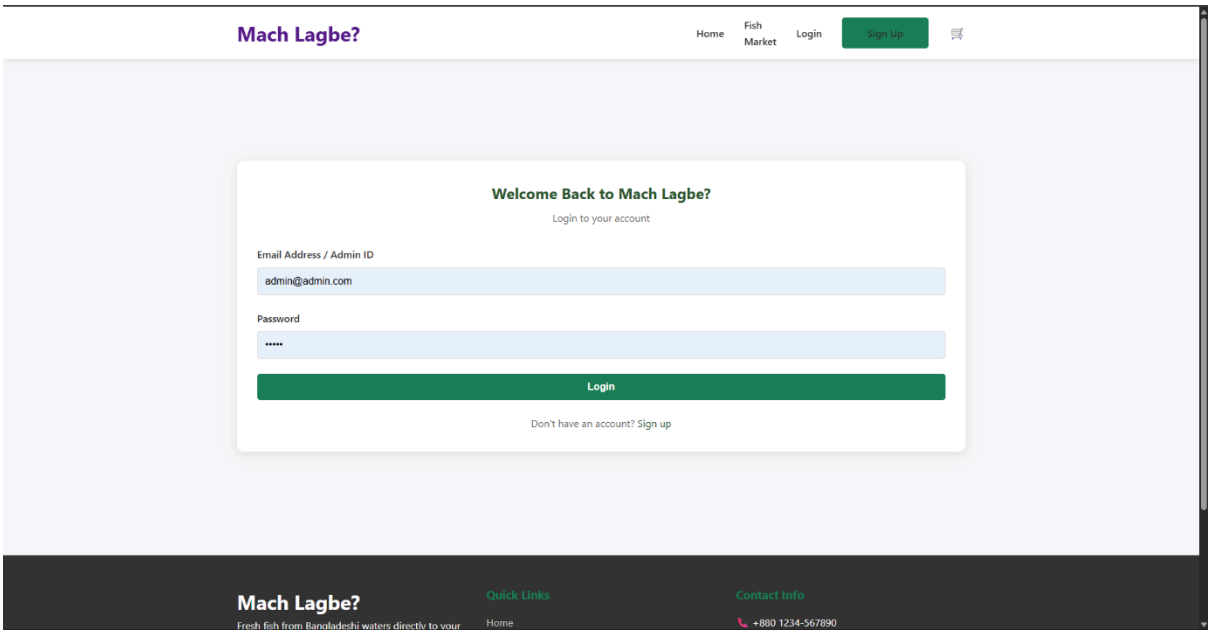


Figure 2.1: Registration and Login

2.3 Fish/Product Browsing

Fish/product browsing is one of the central features of Mach Lagbe. The marketplace displays available fish products with useful information such as product name, price, description, availability, stock information, and product image.

The purpose of the product browsing page is to give customers a clear overview of available items. Instead of requiring users to contact an administrator for every product detail, the information is presented directly through the interface.

Product images provide a visual representation of the item, while names and descriptions help users understand what is being offered. Price and availability information allow customers to make decisions before adding products to the cart.

The product browsing feature also acts as the starting point for the shopping process. After viewing available products, a customer can select an item and continue toward the product details and cart functions.

From the administrative perspective, the information displayed on the marketplace is based on product data managed through the administrator functions.

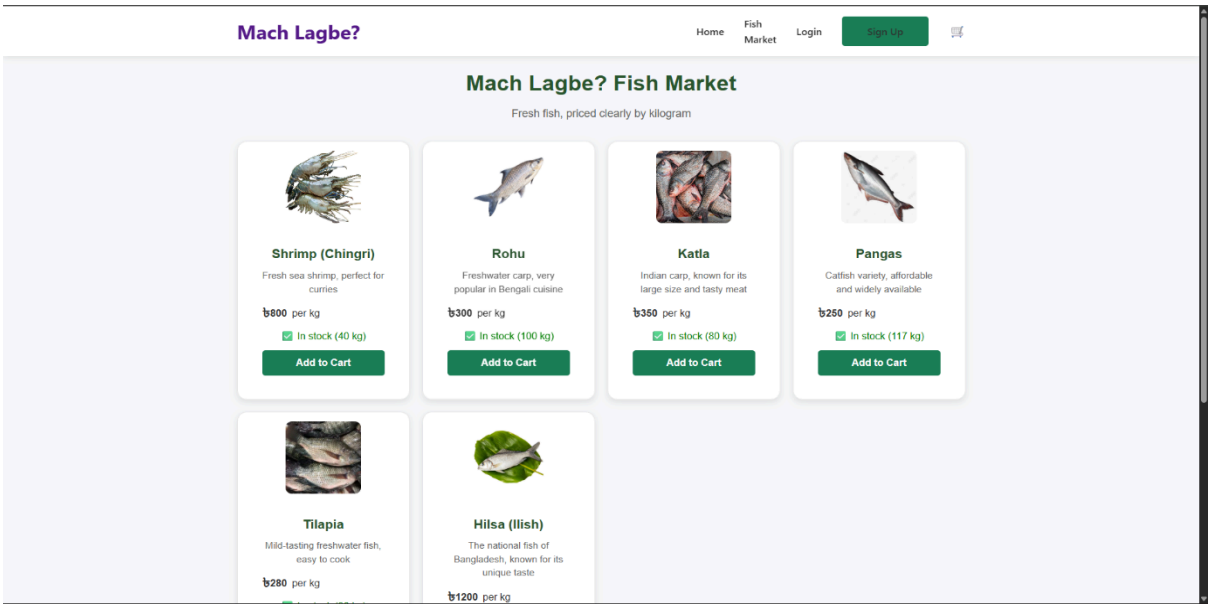


Figure 2.2: Fish Browsing

2.4 Search and Filtering

The project includes basic search and filtering facilities to help customers locate fish products more efficiently.

Without search functionality, customers may need to manually inspect a large list of products. Search reduces this effort by allowing customers to look for products based on relevant information.

Filtering also provides a basic way to narrow the displayed product results. As the number of available fish products grows, such facilities become increasingly useful.

The search and filtering feature is intentionally kept simple in the current version. It is designed to support the basic marketplace requirements rather than implementing a complex recommendation or intelligent search engine.

In the future, this feature could be expanded with more advanced filtering criteria and personalized recommendations.

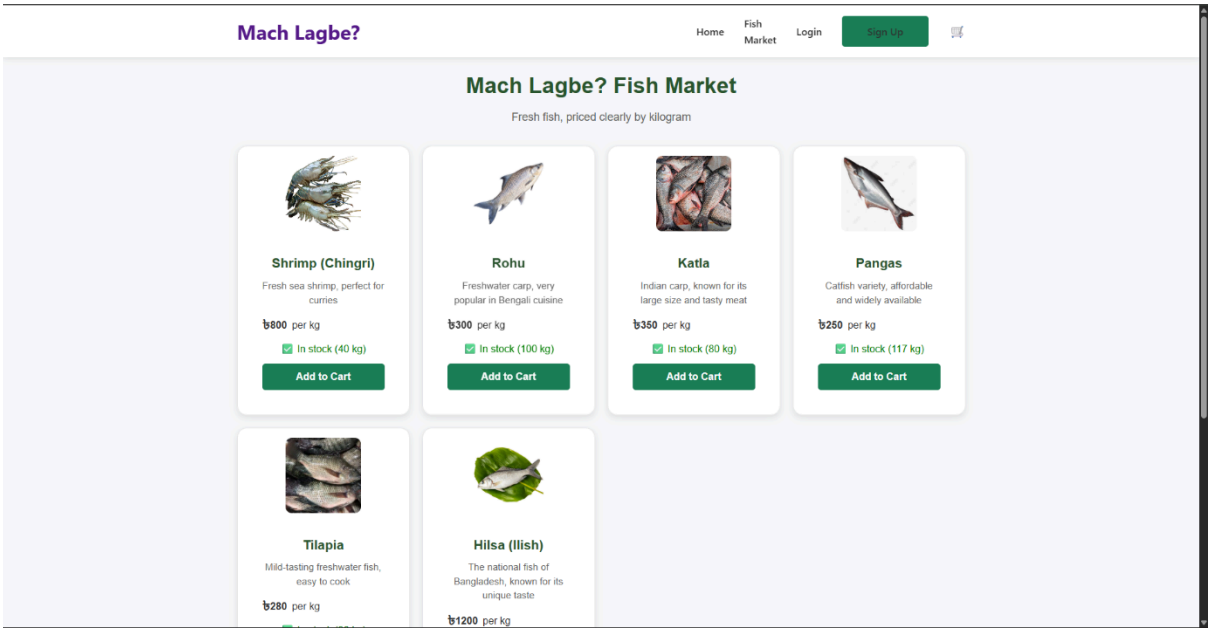


Figure 2.3: Search

2.5 Cart and Ordering

The shopping cart allows customers to collect their desired fish products before submitting an order. Customers can add products to the cart, review the selected items, and proceed to the order placement process.

The cart is important because customers may want to select more than one product before making a final decision. It creates a temporary collection of selected products and separates product selection from final order submission.

After reviewing the cart, the customer can proceed with the ordering process. The selected products are submitted to the backend, where the order information can be processed and stored.

This feature creates a simple shopping workflow:

Browse → Select Product → Add to Cart → Review Cart → Place Order

The cart therefore acts as an important connection between the product marketplace and order management system.

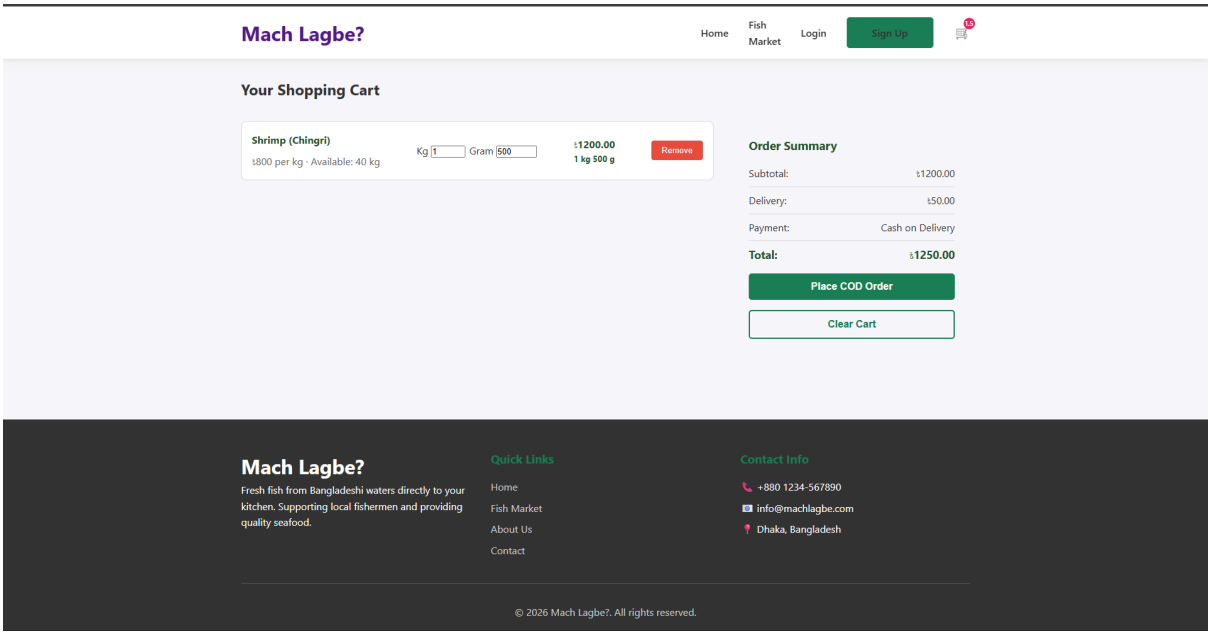


Figure 2.4: Cart and Ordering

2.6 Order History

The order history feature allows authenticated customers to view their previously submitted orders and relevant status information.

Order history is useful because customers may need to check what they ordered previously. Instead of relying on memory or informal communication, customers can access their order records through the application.

The feature also demonstrates the use of persistent database storage. Once an order is submitted, its information can be stored in MongoDB and later retrieved for display to the authenticated customer.

Order history provides a basic form of transaction record management within the marketplace. In a future version, additional information such as detailed delivery progress, payment status, invoices, and delivery dates could be included.

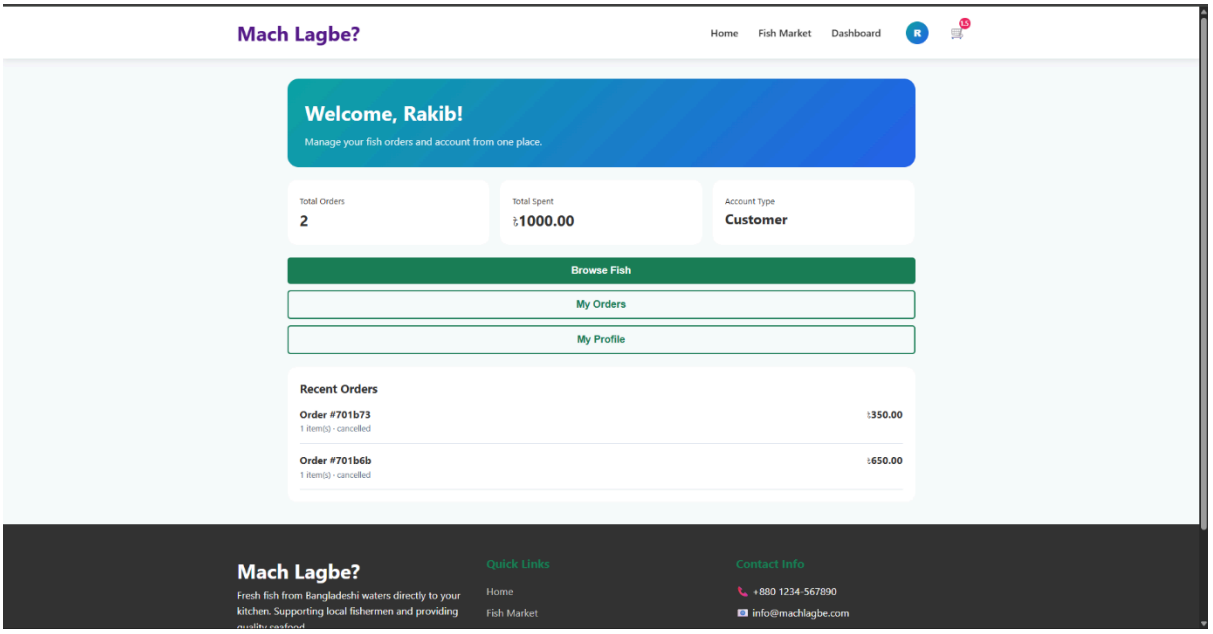


Figure 2.5: Order History

2.7 Admin Panel

The administrator panel provides basic control over marketplace operations, including product and order management.

The admin panel is separated from the normal customer interface because administrative operations require additional permissions. An administrator can access functions that are not intended for regular customers.

The dashboard provides a central location from which the administrator can access marketplace management functions. This makes it easier to maintain product information and review submitted customer orders.

A structured admin panel is especially useful when the number of products and orders increases. Instead of modifying database records manually, administrators can use the application interface to perform common management activities.

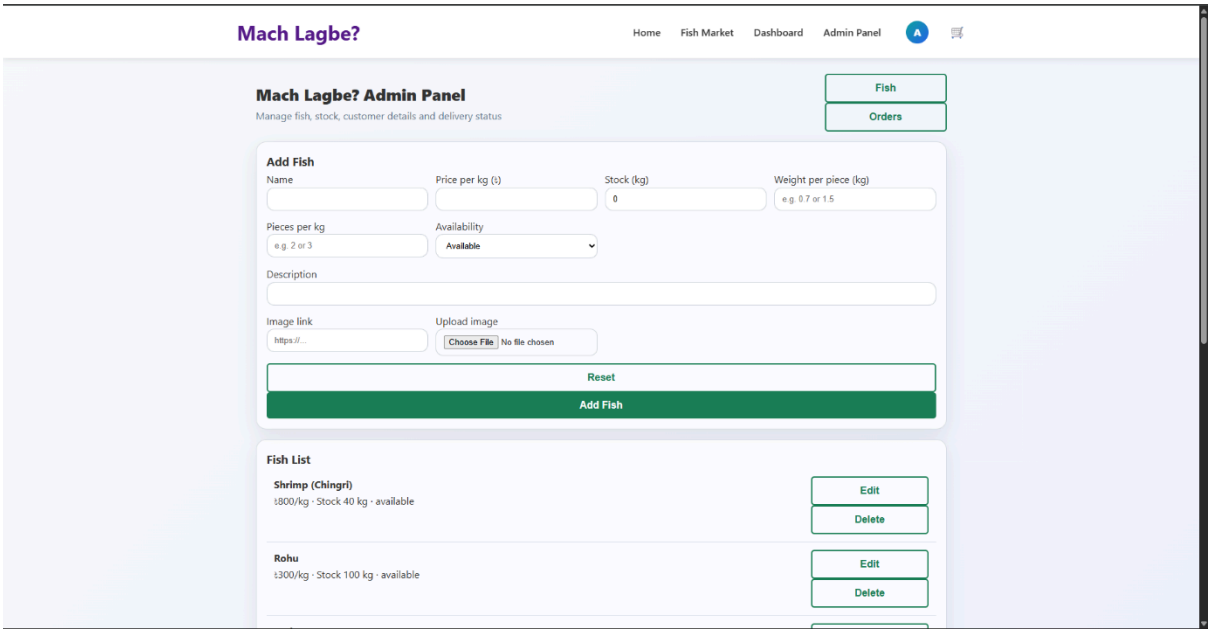


Figure 2.6: Admin Panel

2.8 Product and Order Management

Product management allows administrators to add, edit, and delete fish products. Administrators can update product information when required so that the marketplace can display current information.

For example, when a new fish product becomes available, an administrator can add the product to the marketplace. If the price, description, stock, or availability changes, the administrator can edit the existing product information.

Order management allows administrators to review submitted customer orders and update order status where supported. This gives the administrator a basic workflow for handling customer orders.

The combination of product management and order management makes the administrative section an important part of the system. Without these functions, customers would not receive updated marketplace information and submitted orders would be more difficult to manage.

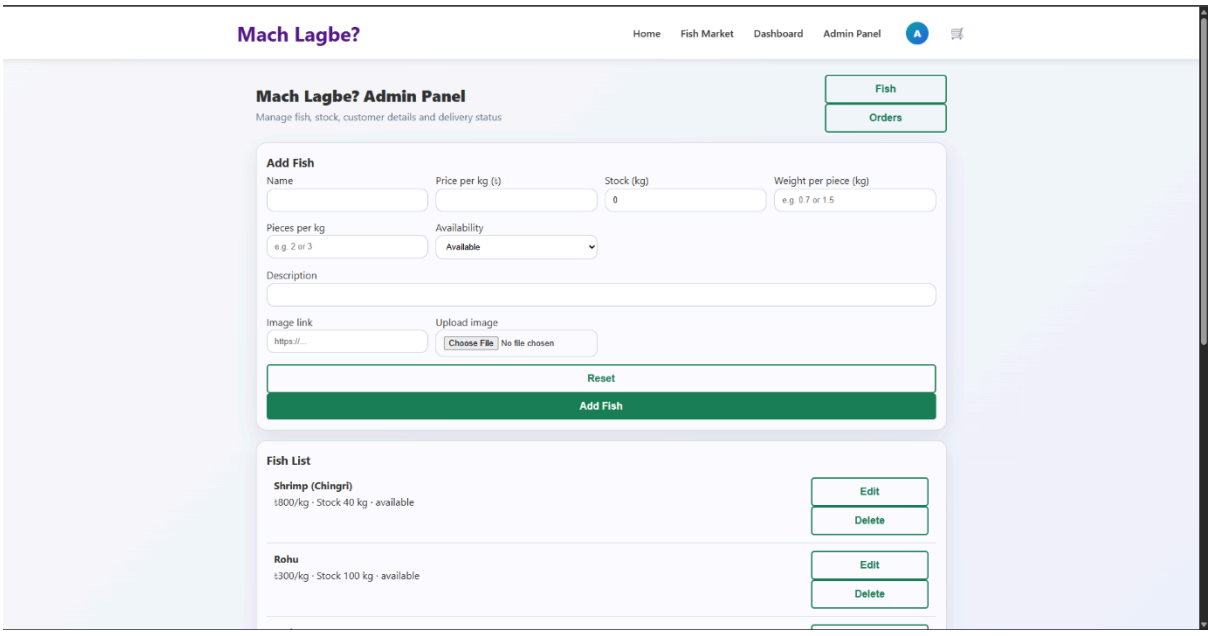


Figure 2.7: Product and Order Managements

CHAPTER 3
TOOLS AND TECHNOLOGY

3.1 Frontend Technologies

HTML and CSS

HTML is used to provide the basic structure of web content. It defines elements such as forms, headings, sections, buttons, input fields, and other interface components.

In Mach Lagbe, HTML concepts are used as part of the frontend structure to organize different marketplace screens. Forms such as registration and login require structured input elements, while product pages require organized content areas.

CSS is used for layout, styling, spacing, and presentation. It helps transform the basic HTML structure into a usable interface.

In Mach Lagbe, CSS contributes to the visual presentation of product listings, forms, navigation areas, buttons and other interface components. Consistent styling helps users move between different pages more easily. [3], [8], [9]

JavaScript

JavaScript provides client-side interactivity, data handling, form interactions and communication with backend APIs.

JavaScript is important because the application needs to respond to user actions. For example, adding a product to the cart, submitting forms, retrieving data, and interacting with the backend require application logic. [4], [9]

React.js

React.js is used to build the frontend using reusable components and dynamic pages.

A component-based approach allows different parts of the interface to be developed and maintained separately. This is useful for an application such as Mach Lagbe because several pages share common interface elements.

React also allows the frontend to communicate with backend APIs and update displayed information based on application data. [5], [6]

3.2 Backend Technologies

Node.js

Node.js provides the server-side JavaScript runtime used by the backend.

Using JavaScript on the server allows the project to maintain a common programming language across frontend and backend development.

The backend is responsible for processing requests from the frontend and performing operations related to authentication, products, orders, and database communication. [6], [8]

Express.js

Express.js is used to create the backend server, API routes, middleware, authentication flow, and request handling.

The Express framework provides a structured way to define API endpoints. The frontend can send requests to these endpoints to retrieve or submit application data.

Express also supports middleware, which can be used for authentication, validation, and other request-processing activities.

3.3 Database Management System

MongoDB is used as the database management system for Mach Lagbe. The application stores user, fish/product and order-related information in MongoDB collections.

The main categories of stored information include:

- User information and authentication data.
- Fish/product information.
- Order information and order status.

MongoDB provides persistent storage, meaning that application information can remain available after a user leaves the website or a server request is completed.

Mongoose is used to work with MongoDB through schemas and models. The database design is centered around the main entities required by the marketplace: User, Fish/Product, and Order.

The database layer is important because both customer and administrator functions depend on stored information. Product browsing requires product data, login requires user data, and order history requires previously stored order data.

3.4 Authentication and Security

Security is an important part of an online marketplace because user accounts and administrative functions need protection. The backend uses password hashing and token-based authentication to protect user accounts and restricted operations. Protected administrative operations require authenticated access. Basic CORS configuration and server-side validation are also used.

Password hashing helps avoid storing user passwords directly in plain text. Token-based authentication provides a mechanism for identifying authenticated users when they access protected resources.

Server-side validation is also important because data received from the client should not automatically be trusted. The backend can validate incoming information before processing it.

CORS configuration helps control communication between the frontend and backend when they operate in different origins or development environments.

The current project provides basic security mechanisms appropriate for the scope of an academic web application. More advanced security measures can be added in future versions.

3.5 Development Tools

Several tools were used during the development and testing of Mach Lagbe.

Visual Studio Code

Visual Studio Code is used as the primary source-code development environment. It provides support for writing and organizing frontend and backend code.

Git and GitHub

Git and GitHub are used for version control and repository management. Version control helps the development team maintain different versions of the project and track changes.

Web Browser

A web browser is used for testing the frontend interface. Different pages and user interactions can be checked through the browser during development.

CHAPTER 4
REQUIREMENT ANALYSIS, DESIGN AND IMPLEMENTATION

4.1 Requirement Analysis

Requirement analysis identifies the functions and quality characteristics required for Mach Lagbe. It helps define what the system should do and how the system should behave.

The requirements are divided into functional and non-functional requirements.

Functional Requirements

The main functional requirements are:

- Users can register and log in.
- Users can browse available fish products.
- Users can search and filter fish products.
- Users can add products to a cart.
- Users can place orders.
- Users can view order history.
- Administrators can manage fish products.
- Administrators can manage customer orders.

These requirements describe the main operations that the application must support.

Non-Functional Requirements

The main non-functional requirements include:

- User-friendly interface.
- Responsive web presentation.
- Reasonable response time for normal operations.
- Data consistency.
- Basic authentication and access control.
- Maintainable frontend and backend structure.

Non-functional requirements are important because a system is not useful only because it provides functions. It should also be reasonably easy to use, maintain, and operate.

A user-friendly interface helps customers understand how to browse products and place orders. Data consistency is important because incorrect product or order information can negatively affect the marketplace.

4.2 System Architecture

Mach Lagbe follows a client-server architecture. The React frontend communicates with the Node.js and Express.js backend through API requests. The backend performs application logic, authentication, validation, and database operations, while MongoDB stores persistent application data.

The architecture can be divided into three primary layers:

Presentation Layer

The presentation layer is the React-based frontend. It provides the interface through which customers and administrators interact with the application.

Application Layer

The application layer is the Node.js and Express.js backend. It receives requests from the frontend, applies application logic, performs authentication and validation, and communicates with the database.

Data Layer

The data layer is MongoDB. It stores persistent information related to users, fish products, and orders.

The overall communication flow can be represented as:

Customer/Admin → React Frontend → Node.js/Express Backend → MongoDB

The response then travels back through the backend to the frontend, where the information is displayed to the user.

This layered architecture provides separation of responsibilities. Changes to the frontend can be made without completely redesigning the database, while backend operations can be updated independently of the user interface. [1], [2], [5]

4.3 Use Case Description

The primary actors are Customer and Administrator. [1], [7]

Customer use cases:

- Register
- Login
- Browse Fish
- Search/Filter Products
- View Product Details
- Manage Cart
- Place Order
- View Order History

Administrator use cases:

- Login
- View Dashboard
- Add Fish
- Edit Fish
- Delete Fish
- View Orders
- Update Order Status

4.4 Database Design

The database is organized around the core entities required for the marketplace. The exact schema is implemented through MongoDB and Mongoose models. The main entities are User, Fish/Product, and Order.

User Collection

The User entity stores account and authentication-related information. It supports registration and login and allows the system to identify authenticated customers.

Fish/Product Collection

The Fish/Product entity stores information such as:

- Product name.
- Price.
- Category.
- Stock.
- Description.
- Image.

- Availability.

This information is used by the marketplace page and product details interface.

Order Collection

The Order entity stores customer order information, selected products, totals, and order status.

The database design therefore supports the major operations of the marketplace. Product browsing depends on product records, authentication depends on user records, and order history depends on order records.

Mongoose models provide a structured method for interacting with the MongoDB database from the Node.js backend.

4.5 Implementation

The frontend is implemented as a React application with pages and components for marketplace browsing, authentication, cart management, order history, and administration. The backend is implemented using Node.js and Express.js with route modules for authentication, fish/product operations, and orders. MongoDB provides persistent storage.

The implementation process can be viewed as several connected activities.

First, the frontend provides the screens through which users interact with the application. These include registration, login, product browsing, product details, cart, order placement, order history, and administrative screens.

Second, the backend provides APIs that connect the frontend with application logic. Requests received from the frontend are processed by appropriate backend routes.

Third, the database stores information that must remain available after individual requests are completed. This includes customer accounts, product records, and orders.

Finally, authentication and validation mechanisms are used to protect restricted operations and maintain basic application security.

The implementation therefore connects the user interface, backend logic, and database into one complete system. [5], [8]

CHAPTER 5
PROJECT DESCRIPTION AND USER INTERFACE

This chapter presents the main user interfaces and functional areas of Mach Lagbe. The blank spaces are intentionally reserved for screenshots from the final running project.

5.1 Home Page

The home page is the main entry point of the Mach Lagbe marketplace. It introduces the marketplace to users and provides access to the main sections of the application.

A good home page is important because it creates the first impression of the application. Users should be able to understand the purpose of the website and navigate toward product browsing or account-related functions.

The home page of Mach Lagbe is designed around the fish marketplace concept. It provides a starting point from which customers can continue to browse available products.

The page also contributes to the overall visual identity of the project. Consistent navigation, layout, images, and interface elements help create a connected experience across different pages.

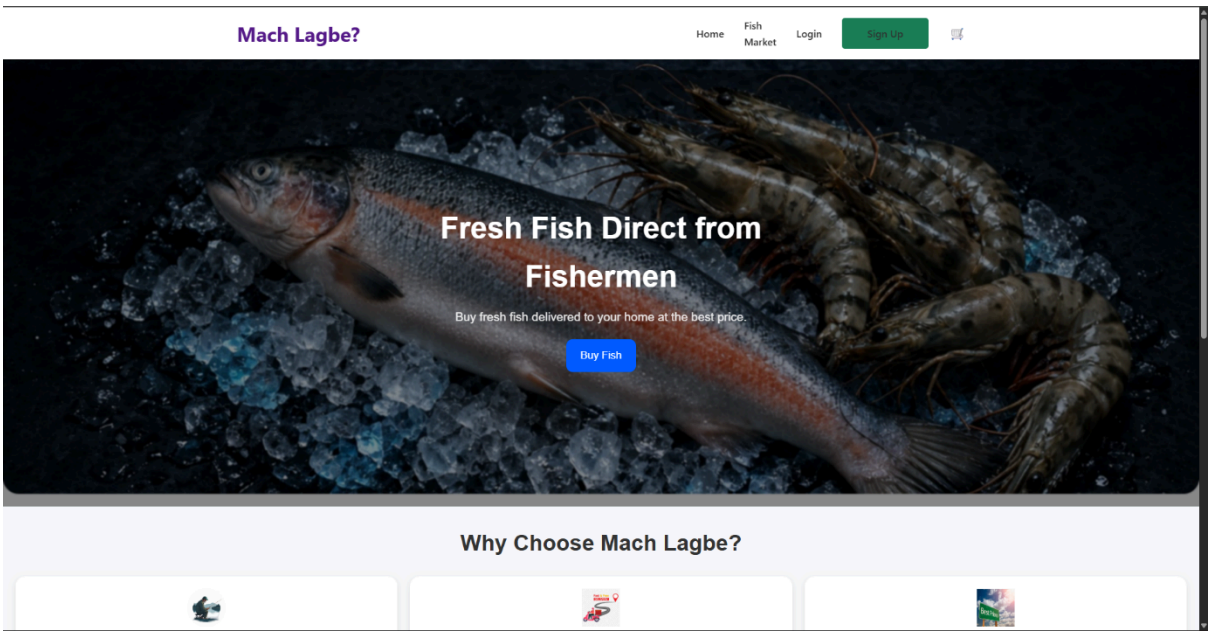


Figure 5.1: Home Page

5.2 Registration and Login

Registration and login are important parts of the customer access process. New users can register for an account, while existing users can log into the system.

The registration page provides a form for entering the required account information. The purpose of registration is to create an account that can later be used for authentication.

The login page allows an existing user to submit their credentials and access the application. Authentication is handled by the backend, and protected operations require valid authentication.

The registration and login interfaces should remain simple so that users can understand what information is required. Clear input fields and buttons reduce confusion during the account creation process.

The administrator also uses authenticated access to reach protected administrative functionality.

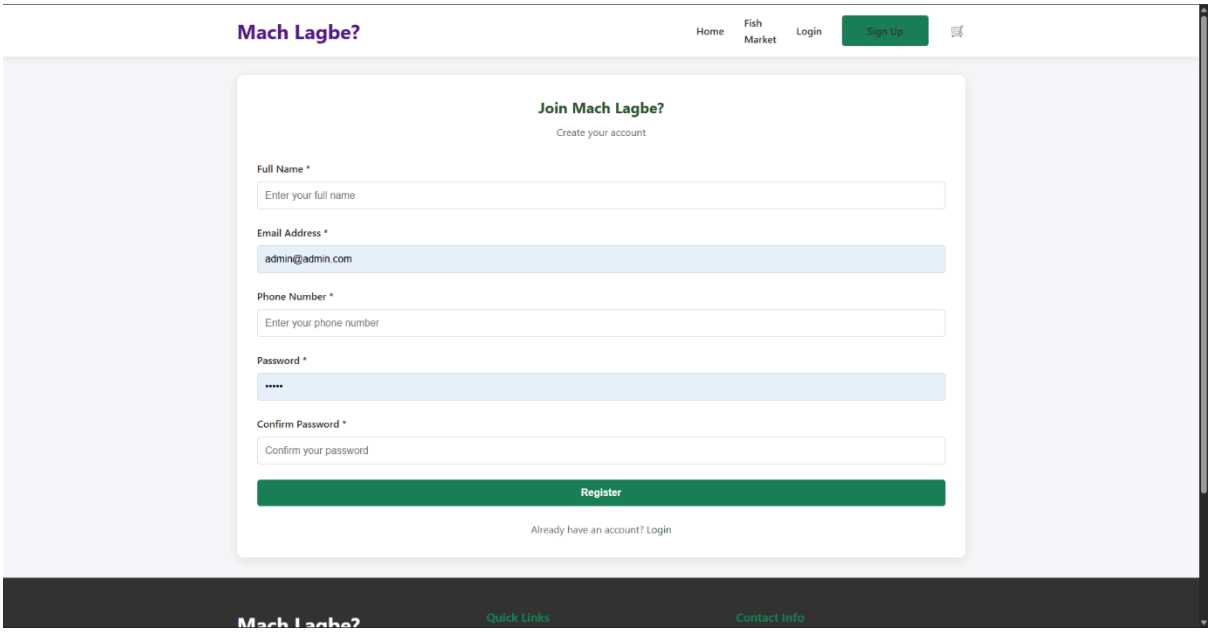


Figure 5.2.1: Registration

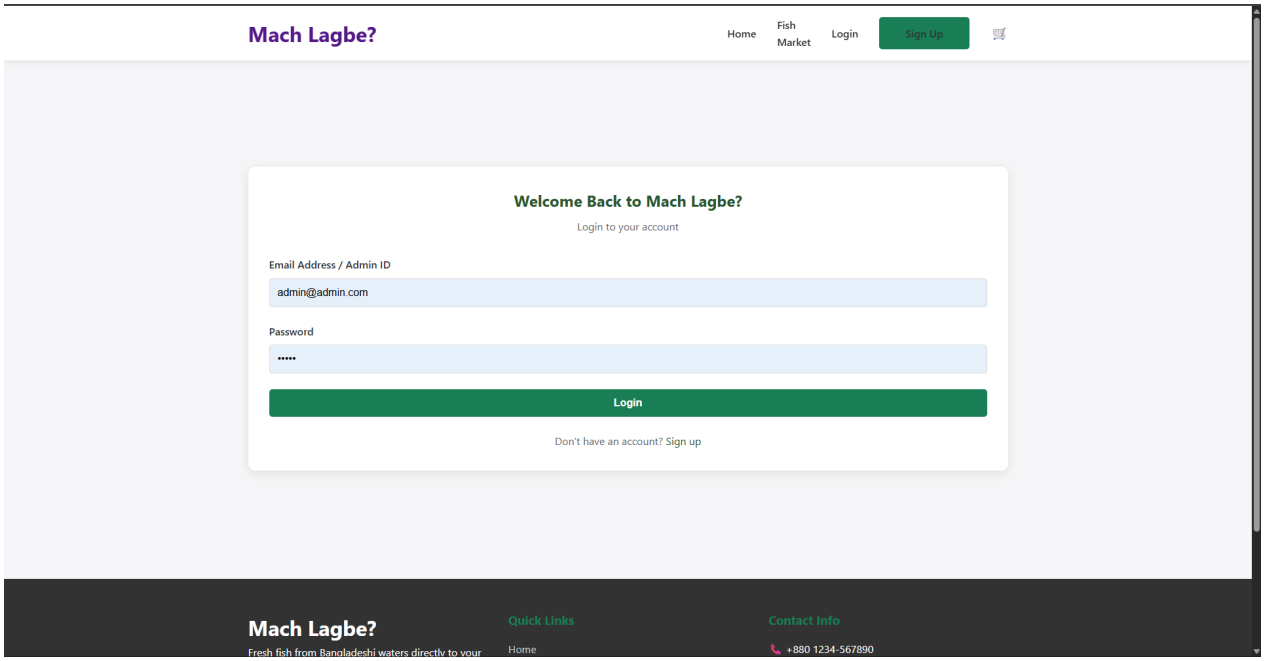


Figure 5.2.2: Login

5.3 Fish Marketplace

The fish marketplace page presents the available fish products to customers.

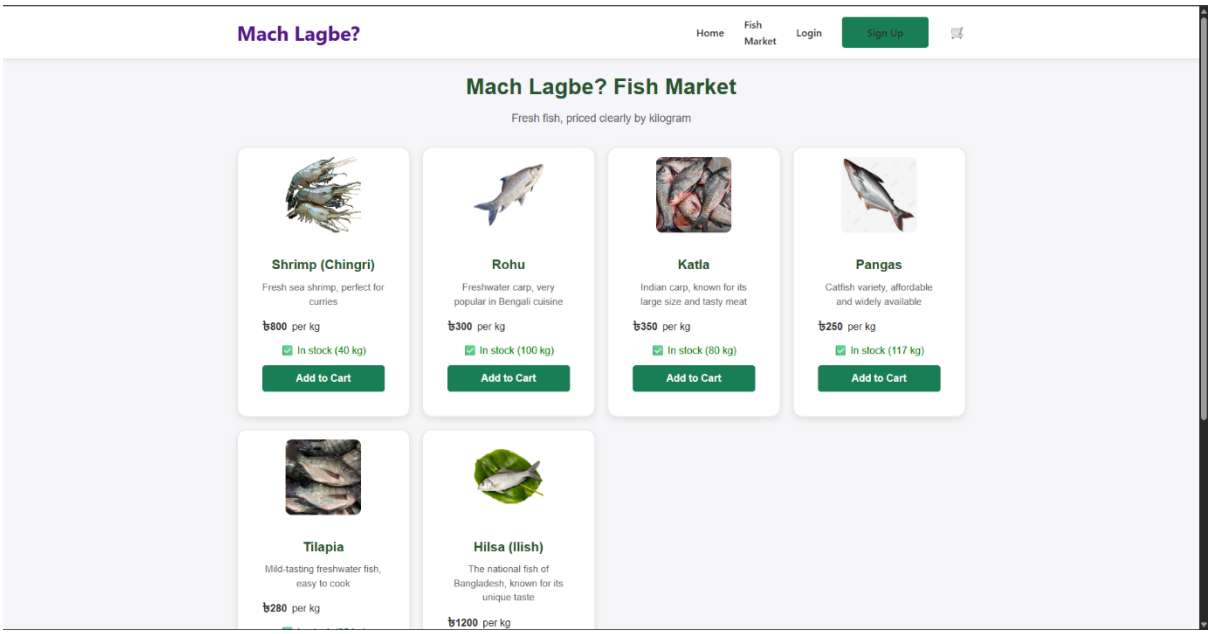


Figure 5.3: Product List

5.4 Fish Details and Search

The fish details section allows customers to inspect product information more closely. Customers can use search and filter controls to locate products and can then inspect the information associated with a selected item.

Product details are important because customers may need more information than what is visible on the main marketplace page. A detailed view can help users understand the product before adding it to the shopping cart.

The search facility makes the marketplace easier to use when several products are available. Instead of manually checking the entire product list, customers can search for relevant items.

Basic filtering further improves product discovery by narrowing the displayed products according to available criteria.

The current implementation focuses on basic search and filtering. More advanced search features can be considered as part of future development.

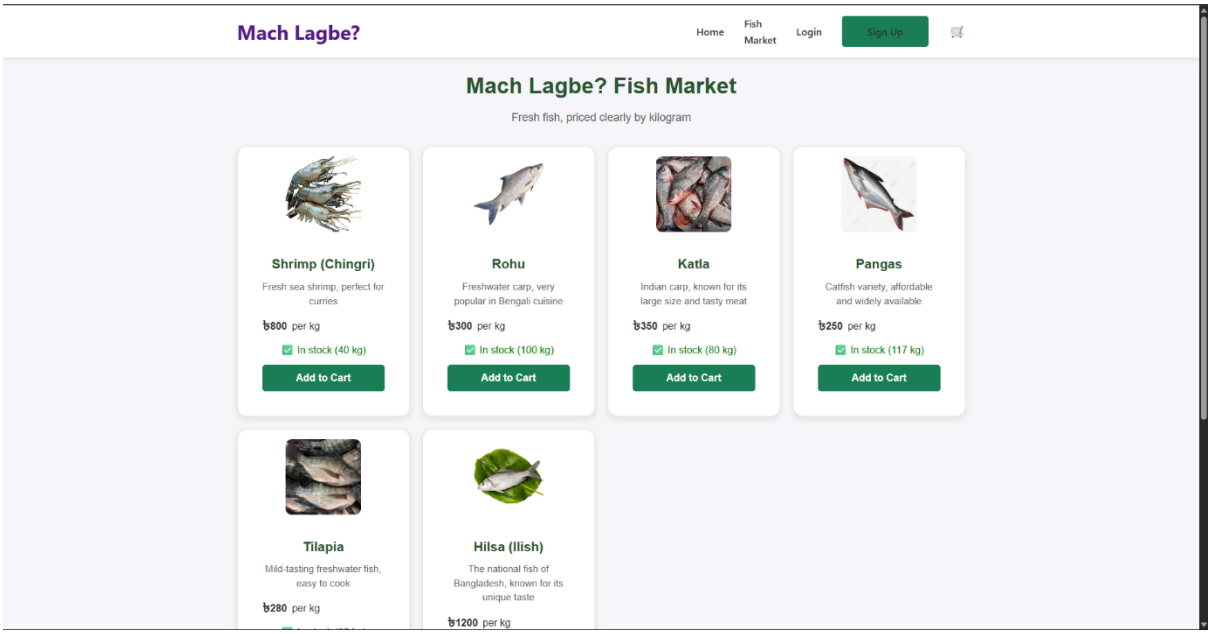


Figure 5.4: Product Details

5.5 Shopping Cart

The shopping cart allows customers to review the products they have selected.

The cart works as a temporary collection of products before an order is submitted. It is useful because customers may want to select multiple products before making the final ordering decision.

The customer can review the selected products before continuing to order placement. This provides an additional opportunity to verify the selection.

The shopping cart is also an important part of the overall marketplace flow because it connects product browsing with order submission.

The basic customer flow can therefore be represented as:

Product Browsing → Product Selection → Shopping Cart → Order Placement

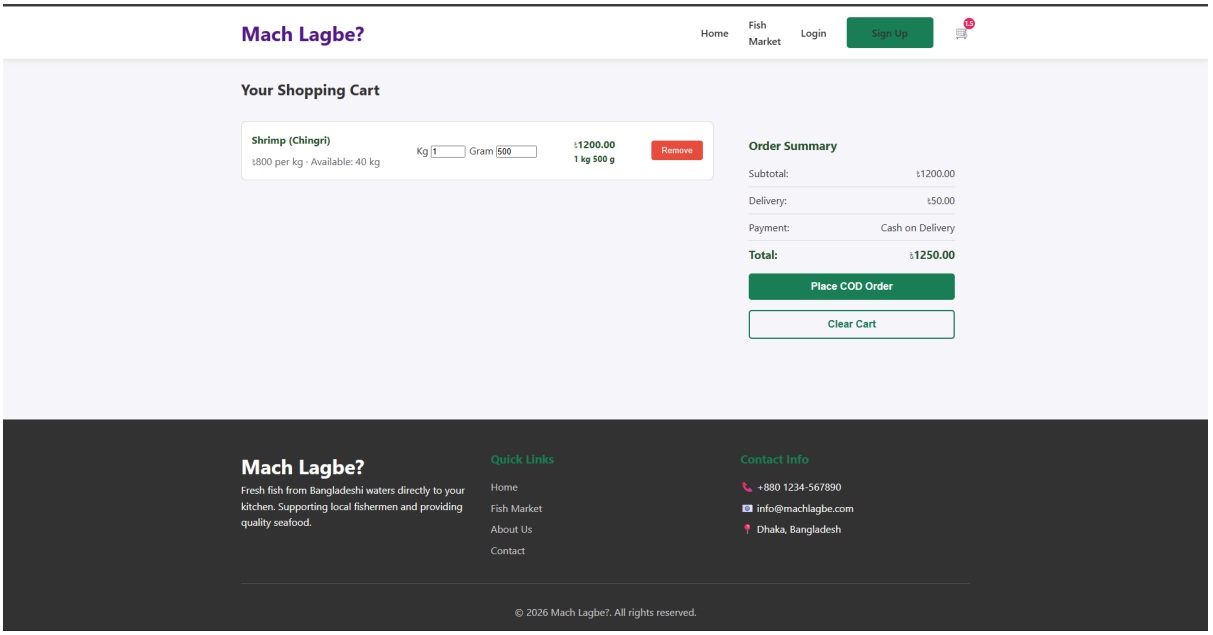


Figure 5.5: Cart

5.6 Order Placement

The order placement interface submits the selected products to the backend.

After reviewing the shopping cart, the customer can proceed with the order placement process. The selected product information is sent to the backend for processing.

The backend handles the request and stores the order information in the database. This creates a persistent record that can later be accessed through order history.

Order placement is therefore the point where a customer's temporary product selection becomes a recorded marketplace order.

A successful ordering workflow requires communication between the frontend, backend, and database. The frontend collects the customer's action, the backend processes the request, and MongoDB stores the resulting order information.

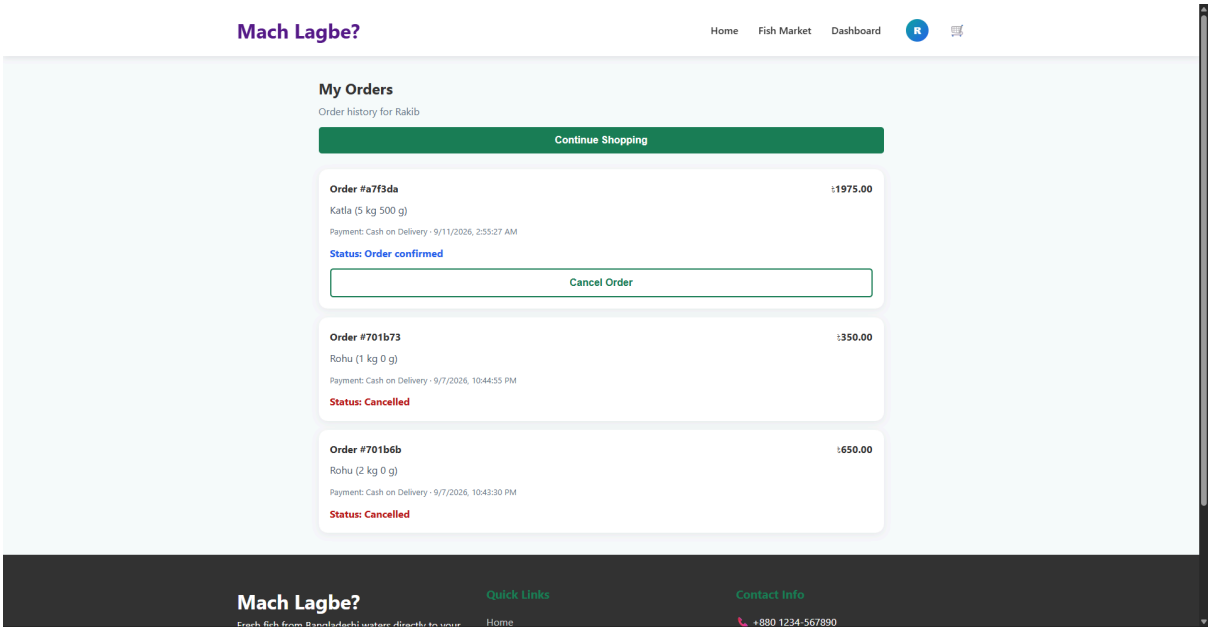


Figure 5.6: Order Placement

5.7 Order History

The order history page displays previously submitted customer orders.

This page gives customers access to their earlier order records. It is particularly useful for reviewing previous purchases and checking relevant order information.

The order history feature depends on persistent database storage. When an order is placed, its information is stored and can later be retrieved for display.

The feature also demonstrates the importance of authenticated access. Customers should only access the order information associated with their own account.

In a future version, order history could be expanded with detailed order tracking, delivery information, payment status, and downloadable invoices.

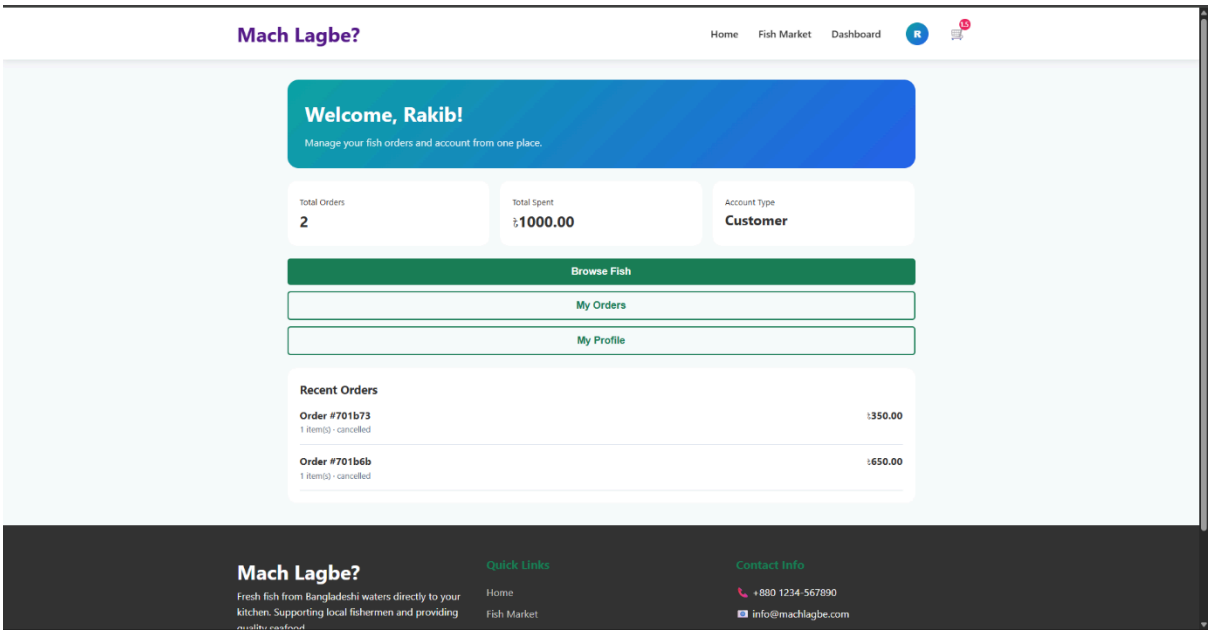


Figure 5.7: Order History

5.8 Admin Dashboard

The admin dashboard provides access to marketplace management functions.

Unlike the customer interface, the dashboard is intended for authorized administrators. It acts as a central location for accessing product and order management functions.

The dashboard helps organize administrative tasks by bringing important controls into one interface. An administrator can use it to manage product information and review customer orders.

A dashboard is useful because administrators do not need to interact directly with database records for routine management activities. Instead, they can use the graphical interface provided by the application.

The administrative interface also demonstrates the separation of user roles within the system.

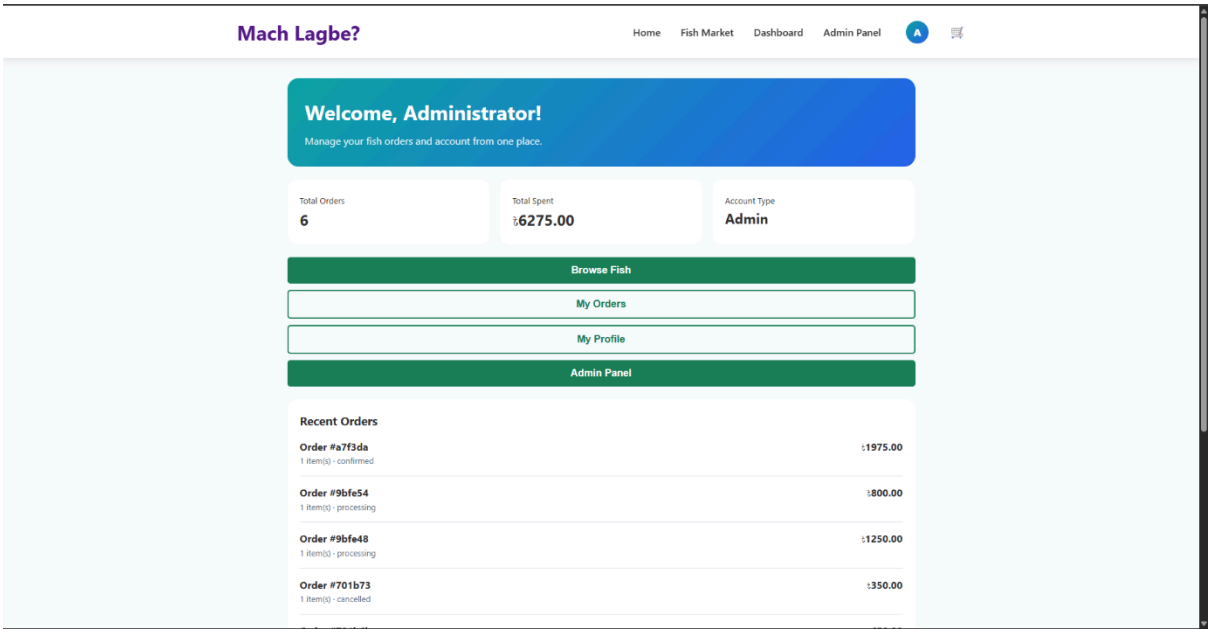


Figure 5.8: Admin Dashboard

5.9 Add/Edit Fish

The Add/Edit Fish interface allows administrators to add new fish products and update existing information.

Product information can change over time. New fish may become available, prices may change, stock information may need updating, or product descriptions may require modification.

The add function allows administrators to introduce new products into the marketplace. The edit function allows existing information to be updated.

This feature is directly connected to the MongoDB database because changes made by the administrator need to be stored and later displayed on the marketplace page.

The ability to maintain product information is important for keeping the customer-facing marketplace useful and organized.

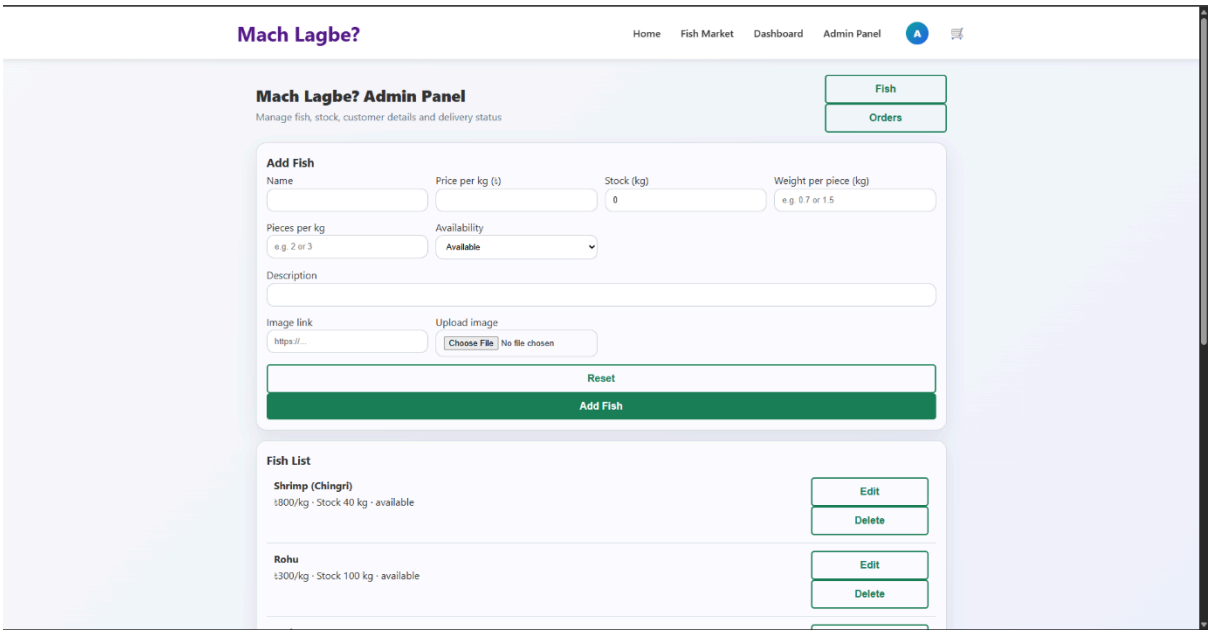


Figure 5.9: Add/Edit Fish

5.10 Order Management

The Order Management interface allows administrators to review submitted customer orders and update their status where supported.

Order management is necessary because customer orders should not simply remain as database records without administrative attention. The administrator needs to review submitted orders and manage their progress within the supported system.

The interface provides a structured view of submitted orders and allows the administrator to perform the available management actions.

In future versions, order management could be extended with delivery assignment, payment confirmation, detailed status history, customer notifications, and real-time delivery tracking.

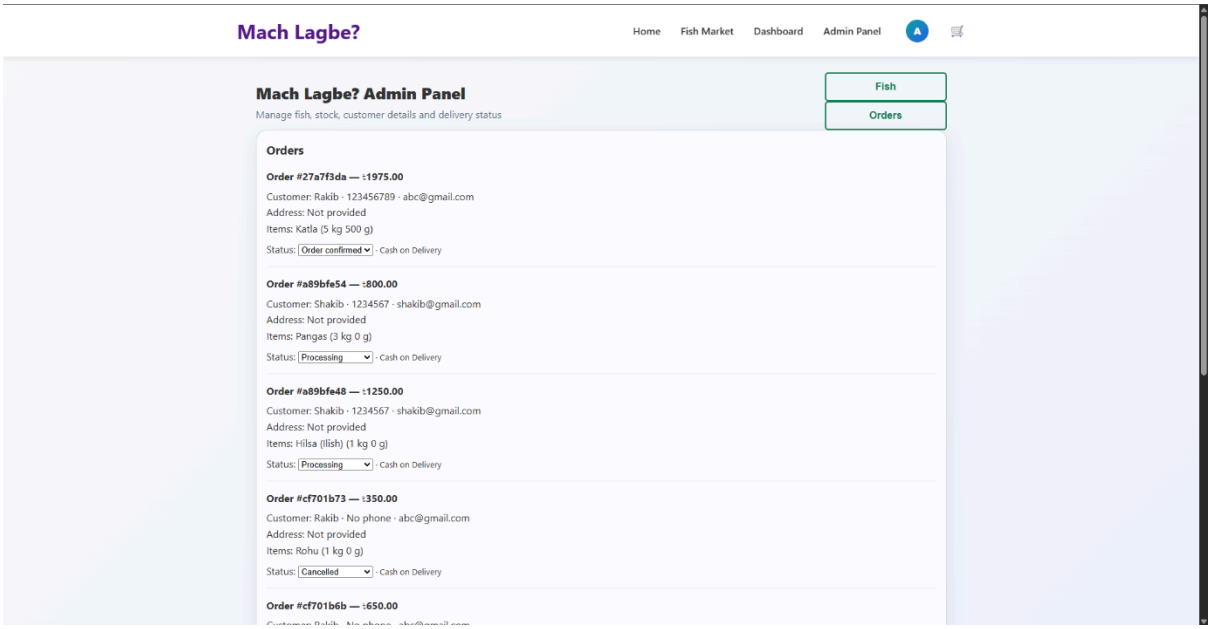


Figure 5.10: Order Management

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Mach Lagbe successfully demonstrates the concept of a web-based fish marketplace that combines product browsing, customer ordering, and basic administrative management within a single application. Customers can explore available fish products, review product information, manage a shopping cart, place orders, and view their previous order history.

The project addresses a practical problem by providing a digital alternative to traditional fish shopping. Instead of depending entirely on physical markets and manual communication, customers can use a centralized web application to access product information and submit orders.

One of the important strengths of the system is the separation between the frontend, backend, and database layers. This separation provides a clear structure for development and makes it easier to maintain or extend different parts of the application.

The project also provides practical experience in authentication, API communication, database operations, user interface design, and administrative management. Therefore, Mach Lagbe is not only a marketplace application but also a demonstration of full-stack software development concepts.

The current implementation is intentionally limited to the basic marketplace workflow. It does not attempt to implement every feature required by a commercial e-commerce platform. However, the existing functionality provides a foundation upon which additional features can be developed.

Overall, Mach Lagbe demonstrates that a web-based system can organize important fish marketplace activities within a centralized platform. The project achieves its primary objectives and provides a suitable foundation for future improvement.

6.2 Future Scope

- Integration of online payment gateways.
- Real-time delivery tracking and delivery-partner management.
- Mobile application development.
- Customer reviews and product ratings.
- Advanced product recommendation features.
- Cloud deployment and scalable infrastructure.
- Improved analytics and sales reporting.
- More advanced inventory and stock management.

REFERENCES

[1] Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.

[2] Pressman, R. S., & Maxim, B. R. (2024). Software Engineering: A Practitioner's Approach (9th ed.). McGraw Hill.

[3] Duckett, J. (2014). HTML and CSS: Design and Build Websites. John Wiley & Sons.

[4] Flanagan, D. (2020). JavaScript: The Definitive Guide (7th ed.). O'Reilly Media.

[5] React Documentation. React – Learn and Reference Documentation. Meta/Open-Source Contributors.

[6] Mozilla. (n.d.). MDN Web Docs: Web Technology for Developers. Mozilla Developer Network. https://developer.mozilla.org/en-US/

[7] Meta Open Source Contributors. (n.d.). React Documentation: Learn and Reference Documentation. React. https://react.dev/

[8] OpenJS Foundation. (n.d.). Node.js Documentation and Learn Resources. Node.js. https://nodejs.org/docs/latest/api/

[9] OpenJS Foundation. (n.d.). Express.js: Node.js Web Application Framework Documentation. Express.js. https://expressjs.com/

[10] MongoDB, Inc. (n.d.). MongoDB Manual: CRUD Operations and Data Modeling. MongoDB Documentation. https://www.mongodb.com/docs/manual/

[11] Mongoose. (n.d.). Mongoose Documentation: Schemas, Models, Queries, and Validation. Mongoose Documentation. https://mongoosejs.com/docs/

[12] Mozilla. (n.d.). MDN Web Docs: Web Technology for Developers. Mozilla Developer Network. https://developer.mozilla.org/en-US/