

AQUA-HUB

Project Report

Submitted by

Jeswin George

Reg. No.: AJC23MCA-2034

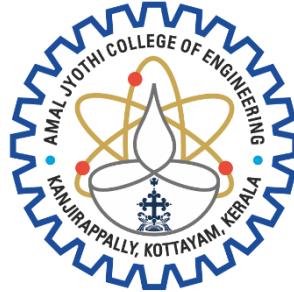
In Partial fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

(MCA TWO YEAR)

(Accredited by NBA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

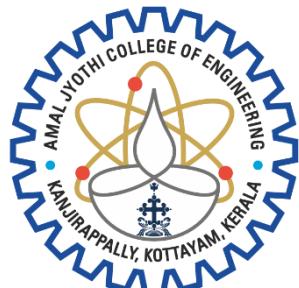


**AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,
Accredited by NAAC. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2023-2025

**DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS
KANJIRAPPALLY**



CERTIFICATE

This is to certify that the Project report titled “**AQUA-HUB**” is the bonafide work of **JESWIN GEORGE (Regno: AJC23MCA-2034)** carried out in partial fulfillment of the requirements for the award of the **Degree of Master of Computer Applications** at **Amal Jyothi College of Engineering Autonomous, Kanjirappally**, Affiliated to **APJ Abdul Kalam Technological University**. The project was undertaken during the period from **December 10, 2024, to March 27, 2025**.

**Ms.Merin Chacko
Internal Guide**

**Ms.Meera Rose Mathew
Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department**

External Examiner

DECLARATION

I hereby declare that the project report “**AQUA-HUB**” is a bonafide work done at **Amal Jyothi College of Engineering Autonomous, Kanjirappally**, Affiliated to **APJ Abdul Kalam Technological University**, towards the partial fulfilment of the requirements for the award of the **Master of Computer Applications (MCA)** during the period from **December 10, 2024 to March 27, 2025**.

Date:
KANJIRAPPALLY

JESWIN GEORGE
Reg: AJC23MCA-2034

ACKNOWLEDGEMENT

First and foremost, I thank God Almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Director (Administration) **Rev. Fr. Dr. Roy Abraham Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms.Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms.Merin Chacko** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

JESWIN GEORGE

ABSTRACT

The Aqua Hub platform is an innovative solution tailored to meet the needs of aquarium enthusiasts, sellers, and administrators by offering a seamless digital marketplace for ornamental fish and related products. Initially focusing on fish sales, Aqua Hub has evolved to encompass a wide range of aquatic products and features, ensuring a holistic user experience. The platform now includes fish food and aquatic plants, providing users with a one-stop destination for all aquarium-related necessities. A wishlist feature allows users to save their favorite products for future purchases, while AI-powered personalized recommendations offer tailored suggestions based on user preferences .

Aqua Hub also integrates AI-driven disease detection, enabling users to upload images of their fish for health analysis and diagnostic suggestions, fostering proactive fish care. The fish recommendation system suggests suitable fish species based on water pH and temperature, helping users maintain a healthy aquarium environment. Customer reviews and ratings allow buyers to make informed decisions based on community feedback, while comprehensive fish care guides provide essential information about fish maintenance, feeding habits, and water quality management. Additionally, the platform features chatbot support, offering 24/7 assistance to address user queries and provide guidance.

One of the standout features, "Find a Fish," enables users to search for information about various fish species by entering a fish name. This feature retrieves details such as a brief description, an image, and a link to more information from Wikipedia, enhancing knowledge and aiding in better fish care. The platform's user-friendly design ensures a consistent and visually appealing interface across all features, maintaining the green tone and stylistic uniformity that Aqua Hub is known for. By integrating these functionalities, Aqua Hub establishes itself as a comprehensive and reliable ecosystem for aquarium enthusiasts, promoting convenience, education, and sustainability in the ornamental fish industry.

CONTENT

SL. NO	TOPIC	PAGE NO
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	3
2.1	INTRODUCTION	4
2.2	EXISTING SYSTEM	4
2.2.1	DRAWBACKS OF EXISTING SYSTEM	5
2.3	PROPOSED SYSTEM	5
2.3.1	ADVANTAGES OF PROPOSED SYSTEM	6
3	REQUIREMENT ANALYSIS	7
3.1	FEASIBILITY STUDY	8
3.1.1	ECONOMICAL FEASIBILITY	8
3.1.2	TECHNICAL FEASIBILITY	8
3.1.3	BEHAVIORAL FEASIBILITY	8
3.1.4	FEASIBILITY STUDY QUESTIONNAIRE	9
3.1.5	GEOTAGGED PHOTOGRAPH	11
3.2	SYSTEM SPECIFICATION	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	12
3.3.1	PHP	12
3.3.2	MYSQL	12
4	SYSTEM DESIGN	13
4.1	INTRODUCTION	14
4.2	UML DIAGRAM	14
4.2.1	USE CASE DIAGRAM	15
4.2.2	SEQUENCE DIAGRAM	16
4.2.3	STATE CHART DIAGRAM	17
4.2.4	ACTIVITY DIAGRAM	18
4.2.5	CLASS DIAGRAM	19
4.2.6	OBJECT DIAGRAM	20

4.2.7	COMPONENT DIAGRAM	21
4.2.8	DEPLOYMENT DIAGRAM	22
4.2.9	COLLABORATION DIAGRAM	23
4.3	USER INTERFACE DESIGN USING FIGMA	24
4.4	DATABASE DESIGN	28
5	SYSTEM TESTING	37
5.1	INTRODUCTION	38
5.2	TEST PLAN	38
5.2.1	UNIT TESTING	38
5.2.2	INTEGRATION TESTING	38
5.2.3	VALIDATION TESTING	39
5.2.4	USER ACCEPTANCE TESTING	39
5.2.5	AUTOMATION TESTING	39
5.2.6	SELENIUM TESTING	39
6	IMPLEMENTATION	57
6.1	INTRODUCTION	58
6.2	IMPLEMENTATION PROCEDURE	58
6.2.1	USER TRAINING	59
6.2.2	TRAINING ON APPLICATION SOFTWARE	58
6.2.3	SYSTEM MAINTENANCE	59
6.2.4	HOSTING	60
7	CONCLUSION & FUTURE SCOPE	63
7.1	CONCLUSION	64
7.2	FUTURE SCOPE	64
8	BIBLIOGRAPHY	66
9	APPENDIX	68
9.1	SAMPLE CODE	69
9.2	SCREEN SHOTS	78
9.3	GIT LOG	83
9.4	CERTIFICATES	88

List of Abbreviations

- API - Application Programming Interface
- HTML - Hypertext Markup Language
- CSS - Cascading Style Sheets
- JS - JavaScript
- URL - Uniform Resource Locator

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Aqua Hub is an online platform designed for buying and selling fish, offering a comprehensive solution for both customers and sellers. Initially focused on the sale of fish, the platform will expand to include aquariums and fish food. Aqua Hub caters to three primary roles: users (customers), sellers, and admin.

Customers can browse a variety of fish offered by approved sellers, view detailed care information, and make purchases through an intuitive e-commerce interface. The platform also features a cart system and order functionality. Sellers can register and list products, while the admin manages the platform, including seller approvals and user blocking.

1.2 PROJECT SPECIFICATION

The Aqua Hub project is designed as an e-commerce platform focused on the sale and delivery of fish, catering to users, sellers, and administrators. In its initial phase, Aqua Hub provides an interactive space where users can browse, purchase, and learn about various types of fish, while sellers can register, list products, and manage inventory. The platform includes features such as an open-box delivery option, a notification system for efficient pickups, a virtual aquarium builder, and a comprehensive complaint registration system. As the project evolves, Aqua Hub plans to expand its offerings to include aquariums, fish food, and care information, fostering a well-rounded, reliable marketplace for aquatic enthusiasts.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

The system design of Aqua Hub focuses on creating an efficient, scalable, and user-friendly platform that connects fish sellers and buyers. It involves defining the architecture, data flow, and interactions between various system components. The design ensures a smooth user experience for buyers and sellers, with features such as product browsing, order placement, seller management, and secure payments. Aqua Hub's system is built on a modular architecture using Django for the backend, MySQL for data management, and a structured design approach for handling product listings, user accounts, and transactions. The goal is to deliver a reliable and secure platform with a focus on performance, maintainability, and scalability.

2.2 EXISTING SYSTEM

NATURAL SYSTEM STUDIED

The natural system of fish ecology, behavior, and health management forms the backbone of the existing system studied in Aqua Hub. This system encompasses understanding the specific environmental conditions, dietary needs, and social behaviors essential for the well-being of different fish species. Key factors such as water quality, temperature control, tank size, and appropriate filtration are crucial elements in supporting healthy fish habitats. Knowledge of these natural requirements informs Aqua Hub's approach to product care instructions, ensuring customers receive detailed guidelines on maintaining optimal conditions for their fish post-purchase.

DESIGNED SYSTEM STUDIED

The designed system for Aqua Hub is a structured solution aimed at streamlining the process of buying and selling fish online, ensuring the proper care of live products, and enhancing the overall customer experience. The designed system is focused on addressing key challenges in managing live fish sales while providing a user-friendly and efficient platform for customers, sellers, and administrators. Key components of the designed system include:

2.2.1 DRAWBACKS OF EXISTING SYSTEM

- Limited Accessibility:
- Customers are restricted by the physical location of stores.
- Limited online presence of dedicated fish shops.
- Inconsistent Quality and Pricing:
 - No standardization leads to varied quality and prices.
 - Difficult to ensure the health and quality of fish.
- Lack of Information:
 - Customers often lack detailed information about fish species, care, and compatibility.

- Sellers may not always provide comprehensive guidance.
- Inadequate Customer Support:
- Limited support for customers post-purchase.
- Issues with returns, refunds, or replacements are common.
- Limited Variety:
- Physical stores have limited space and inventory.
- Customers may not find the specific fish or products they are looking for.

2.3 PROPOSED SYSTEM

Aqua Hub is a dedicated online platform designed to address the shortcomings of the existing system by providing a comprehensive and user-friendly solution for buying fish and related products.

Features of the Proposed System

1. User-Friendly Interface:

- Intuitive and easy-to-navigate website and mobile app.
- Clear categorization of products for easy browsing.

2. Wide Variety of Products:

- Extensive range of fish species, aquariums, and fish food.
- Regular updates and additions to the inventory.

3. Detailed Product Information:

- Comprehensive descriptions, care instructions, and compatibility information for each fish species.
- High-quality images and videos to showcase products.

4. Standardized Quality and Pricing:

- Strict quality control measures to ensure the health and well-being of fish.
- Transparent and competitive pricing.

5. Multiple Roles and Permissions:

- Different user roles (user, seller, admin) with specific permissions and functionalities.
- Sellers can manage their listings, track sales, and communicate with customers.
- Admins can oversee the platform, manage users, and ensure smooth operations.

6. Customer Support and Assurance:

- Dedicated customer support team to handle inquiries and issues, replacement policies.
- User reviews and ratings to build trust and reliability.

7. Secure Payment System:

- Multiple payment options, including credit/debit cards, digital wallets, and bank transfers.
- Secure and encrypted transactions.

8. Community and Resources:

- Forums and discussion boards for hobbyists to share knowledge and experiences.
- Articles, tutorials, and guides on fish care and aquarium maintenance.

9. Expansion Potential:

- Initial focus on fish, with plans to expand to aquariums and fish food.
- Continuous improvement and addition of new features based on user feedback.

10. Delivery aspects

- Easy and seamless delivery

Aquarium fish can generally survive without food for a considerable amount of time, depending on their species, size, age, and health. Here are some general guidelines:

- Our team ensures that the fish are carefully packed using a 3:2 ratio of oxygen and water and ammonia pills are used in the water to tackle the problems regarding ammonia from fish waste contaminating the water

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Dedicated Platform for Fish and Aquarium Products
- Detailed Product Information and Fish Care Instructions
- Streamlined Seller Registration and Approval Process
- Open Box Delivery for Live Fish
- Secure Payment Gateway Integration (Razorpay)
- User-Friendly Interface with Consistent Styling
- Efficient Complaint Registration and Handling
- Customizable Virtual Tank Feature (Upcoming)
- Improved Seller Authentication and Management
- Responsive Design for Seamless Mobile and Desktop Use
- Enhanced Customer Trust through Seller Verification
- Real-Time Notifications for Delivery and Complaints
- Cart and Checkout System for Easy Purchases
- Admin Control Over User Blocking and Unblocking

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility study for Aqua Hub aims to assess the practicality and viability of developing a dedicated online platform for the buying and selling of fish and related products. This study evaluates the technical, operational, and financial aspects of the proposed system to ensure its successful implementation. Given the unique nature of Aqua Hub, which deals with live fish and specific product care, the feasibility study is crucial in identifying the potential challenges and opportunities in delivering a reliable, user-friendly, and scalable solution. Through this analysis, the study will provide a clear understanding of the resources, technology, and processes required to achieve the platform's goals, while ensuring it meets customer needs and business objectives.

3.1.1 Economical Feasibility

1. Cost Analysis: Initial costs will include development expenses, hosting fees, marketing, and ongoing maintenance. These costs are moderate given the platform's scope, and can be managed through a phased rollout.
2. Revenue Model: Aqua Hub can generate revenue through sales commissions, seller subscription fees, and advertising. This diversified revenue stream increases the platform's financial viability.
3. Profitability: Given the niche market and potential for high margins on specialized products, Aqua Hub has the potential to be profitable within the first few years of operation, especially if it captures a significant share of the growing aquaculture market.

3.1.2 Technical Feasibility

Platform Development: Aqua Hub will be developed using Django, a robust and scalable framework that allows for the integration of multiple user roles (user, seller, admin) and functionalities like video uploads and real-time notifications.

Features and Functionality: The platform will include key features like real-time notifications for product delivery and a system for handling product replacements based on video proof of damage. These features are technically feasible and align with current e-commerce standards.

Infrastructure Requirements: The project will require a reliable hosting service, a secure payment gateway, and integration with SMS services for text notifications. All of these are readily available and can be integrated with Django.

3.1.3 Behavioral Feasibility

The behavioral feasibility of Aqua Hub assesses the acceptance, adaptability, and ease of use

of the system by the target users—customers, sellers, and administrators. It focuses on how the users will interact with the system and how it meets their expectations.

User-Friendly Interface: Aqua Hub's consistent and clean design, with a green tone across templates, ensures that users will find the system intuitive and easy to navigate, improving user satisfaction.

Seller Adaptability: Sellers can easily manage their products, including detailed fish care instructions, through a simple, structured product listing system. The streamlined registration and approval process ensures that sellers can onboard without difficulty.

Customer Experience: Customers can quickly browse fish and related products, read care information, and complete purchases using a secure payment gateway. The open box delivery system enhances trust and convenience, encouraging user adoption.

Admin Control: Admins have control over user and seller management through a clear and well-organized dashboard, allowing them to handle tasks such as seller approval, complaint management, and user blocking/unblocking efficiently.

Complaint and Feedback System: The easy-to-use complaint registration process, where users can file complaints against sellers, ensures that users feel their issues will be addressed, improving trust and engagement.

3.1.4 Feasibility Study Questionnaire

Questions:

1. Can you describe the different types of ornamental fish you breed and sell at Eden's Fish Farm?
2. What are the most common challenges you face in maintaining the health of your fish stock?
3. How do you currently diagnose and treat fish diseases at your farm?
4. Would an AI-based fish health analysis tool help you in early disease detection? If yes, what features would you find useful?
5. What are the most frequently asked questions from your customers regarding fish care, diet, and compatibility?
6. How do you currently manage customer complaints about fish health or quality, and what improvements would you like to see?
7. Are there any particular fish species that are more prone to diseases? If yes, which ones and what are the common issues?
8. How do you currently educate customers on fish care? Would a digital guide integrated into an online platform be useful?

9. Do you see a demand for online consultations or expert advice for customers facing fish health issues?
10. Would you be open to integrating an AI-powered fish identification feature into your business to assist customers in identifying fish species and care requirements?

Answers (Based on Industry Insights & Owner's Perspective):

1. We specialize in freshwater ornamental fish like guppies, bettas, discus, and tetras. We also have some exotic varieties like Arowanas and Flowerhorns.
2. The biggest challenge is disease outbreaks, especially during seasonal changes. Maintaining water quality and preventing infections is a constant struggle.
3. We rely on experience, visual observation, and sometimes external expert advice. Treatments include antibiotics, salt baths, and herbal remedies.
4. Yes, an AI tool would be very helpful, especially if it can analyze images of fish and suggest possible diseases with solutions. Instant water quality checks would also be great.
5. Customers mostly ask about proper tank setup, water parameters, best food options, and which fish can live together without aggression.
6. Right now, we handle complaints case by case, offering refunds or replacements. A structured system where customers can report issues and get instant guidance would be beneficial.
7. Yes, bettas are prone to fin rot, goldfish often get swim bladder issues, and discus are highly sensitive to water changes, making them vulnerable to stress-related illnesses.
8. We give verbal advice and printed guides, but a digital guide with step-by-step videos and an interactive Q&A section would definitely improve customer understanding.
9. Definitely! Many customers struggle with fish care and don't always have access to a vet or expert. A virtual consultation feature would be useful.
10. Yes, that would be amazing! Many customers come in with fish they don't know much about. If they could upload a picture and get instant care tips, it would help both them and us.

3.1.5 Geotagged Photograph



3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i5

RAM - 4 G B

Hard disk - 256 S S D

3.2.2 Software Specification

Front End - Html,css,Js

Back End - Django Python

Database - Mysql

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS, List APIs, List ML Functionalities

3.3 SOFTWARE DESCRIPTION

3.3.1 Django

Aqua Hub's backend is built using Python and the Django framework, which provides a robust and scalable foundation for the application. Django's model-view-controller (MVC) architecture enables efficient management of data and the separation of business logic from the user interface. The framework offers built-in features such as user authentication, form handling, and an admin panel, which streamline the development process and enhance security. Django's ORM (Object-Relational Mapping) simplifies database interactions, making it easy to define and manipulate data models for sellers, products, and orders. This setup ensures that Aqua Hub can handle user requests efficiently, maintain data integrity, and provide a responsive experience for both customers and sellers.

3.3.2 MySQL

Aqua Hub utilizes MySQL as its relational database management system to store and manage data effectively. MySQL offers reliable data storage, high performance, and scalability, making it suitable for handling the dynamic nature of e-commerce transactions. The database structure includes tables for users, sellers, products, orders, and complaints, allowing for organized data management and quick retrieval of information. With MySQL's robust querying capabilities, Aqua Hub can efficiently handle complex queries, ensuring fast and reliable access to data. This ensures that users can seamlessly browse products, manage their accounts, and track orders while maintaining the integrity and security of the stored information.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

The system design of Aqua Hub focuses on creating a comprehensive and user-friendly platform for buying and selling fish and related products online. By integrating advanced technologies and adhering to best practices in web development, Aqua Hub aims to provide an efficient and secure environment for both customers and sellers.

The design process involves a thorough analysis of user requirements, allowing us to define the core functionalities of the platform. Key aspects include a seamless user experience, robust backend architecture, and an intuitive interface that caters to the needs of all stakeholders. The system architecture is built on Python's Django framework, which provides scalability and security, while MySQL serves as the database management system to ensure reliable data handling.

This document outlines the various components of the Aqua Hub system, including user interfaces, data models, and interactions, as well as the overall architecture that supports the platform's functionalities. The goal of the system design is to create a reliable, efficient, and engaging marketplace for fish enthusiasts, enabling them to connect easily while ensuring the health and well-being of live products. Through this structured design approach, Aqua Hub seeks to set a standard in the aquatic e-commerce industry.

4.2 UML DIAGRAM

A UML (Unified Modeling Language) Diagram is a visual representation that describes the architecture, design, and functionality of a software system. In the context of Aqua Hub, UML diagrams, such as use case diagrams, class diagrams, and state chart diagrams, illustrate the interactions between users and the system, the structure of classes and their relationships, and the dynamic behavior of objects. These diagrams help stakeholders understand the system's requirements and design, clarify workflows, and facilitate communication among team members. By providing a clear visual framework, UML diagrams play a vital role in guiding the development process and ensuring that all aspects of the system are well-defined and aligned with project goals.

4.2.1 USE CASE DIAGRAM

Illustrates the interactions between users (customers, sellers, and admins) and the system, highlighting the primary functionalities such as registration, product management, order processing, and complaint registration.

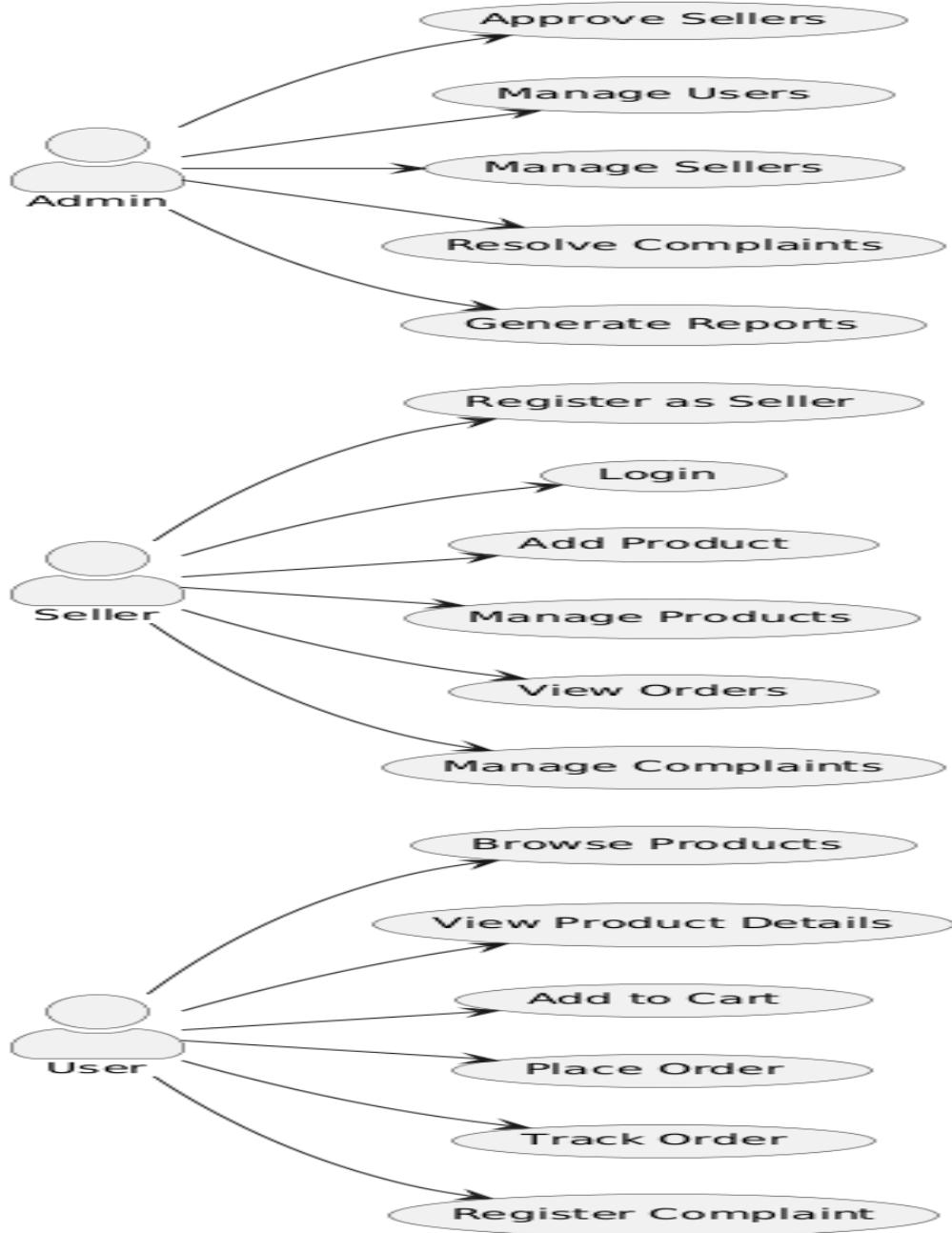


Fig 4.2.1

4.2.2 SEQUENCE DIAGRAM

Demonstrates the flow of messages and interactions between objects in specific scenarios, such as the process of adding a product to the cart or completing a purchase, outlining the step-by-step operations involved.

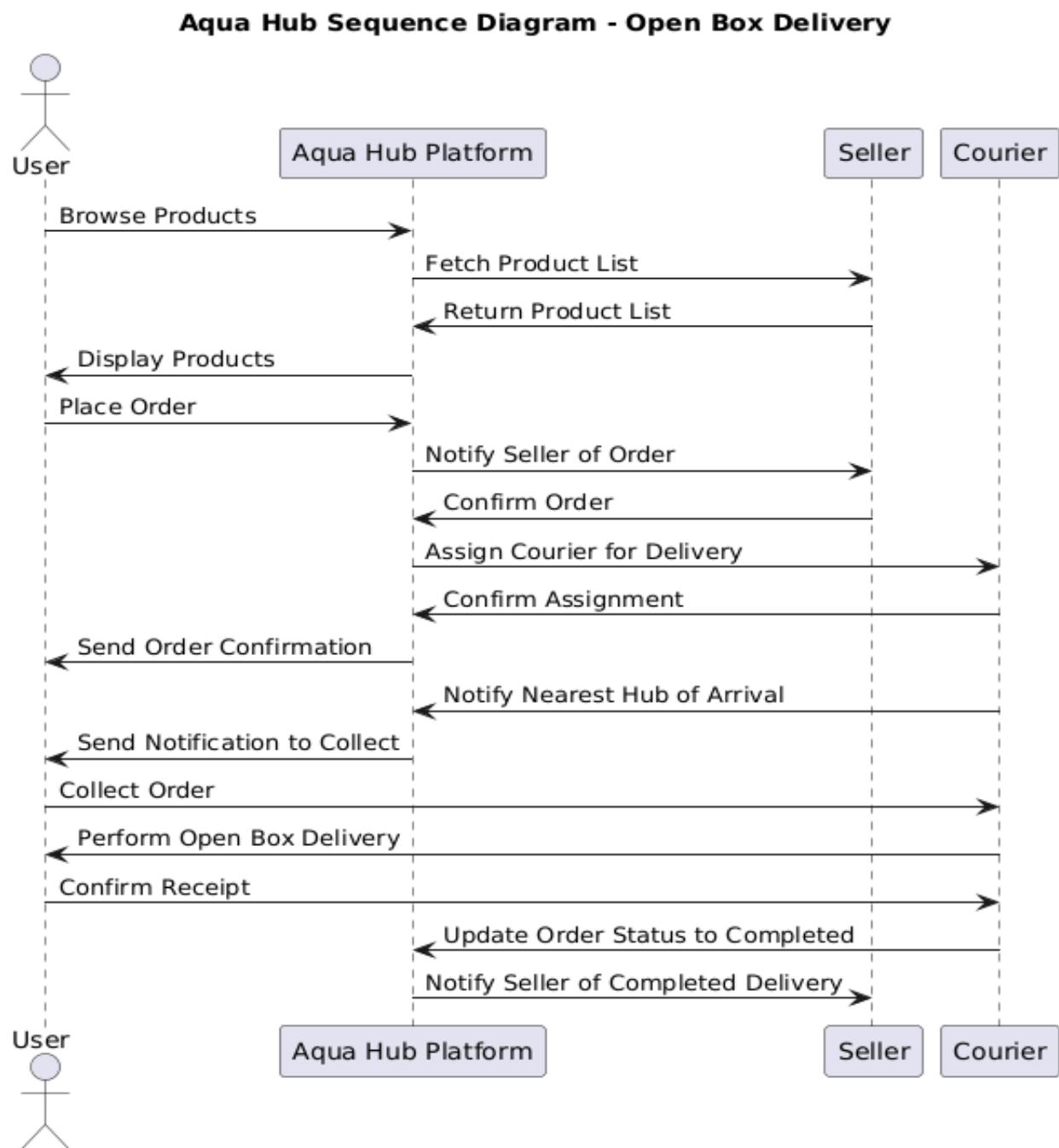


Fig 4.2.2

4.2.3 State Chart Diagram

State Chart Diagram: Represents the states of key entities, such as the order lifecycle, from creation to delivery or cancellation, capturing the transitions between these states.

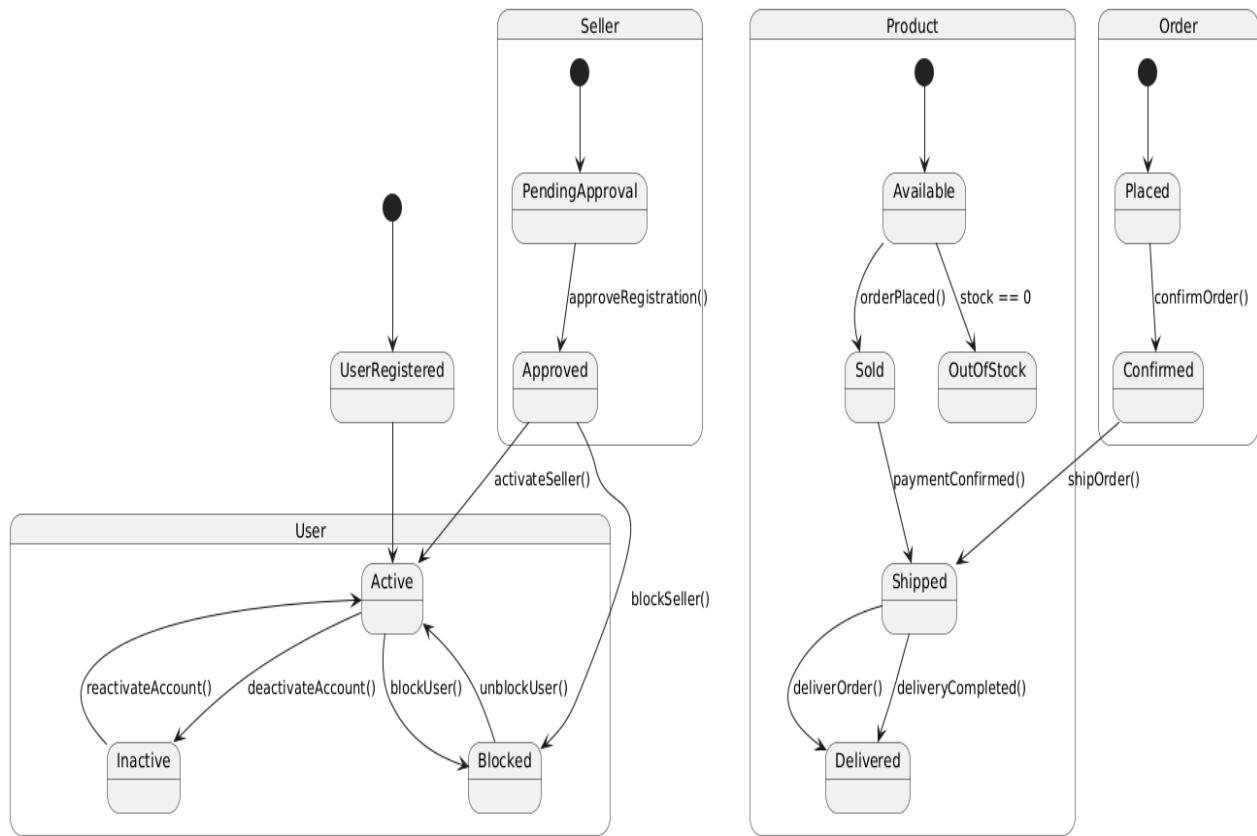


Fig 4.2.3

4.2.4 Activity Diagram

The Activity Diagram for Aqua Hub outlines the step-by-step flow of activities involved in placing an order on the platform. It represents the user's interaction with the system, the system's responses, and the decision points during the order lifecycle.

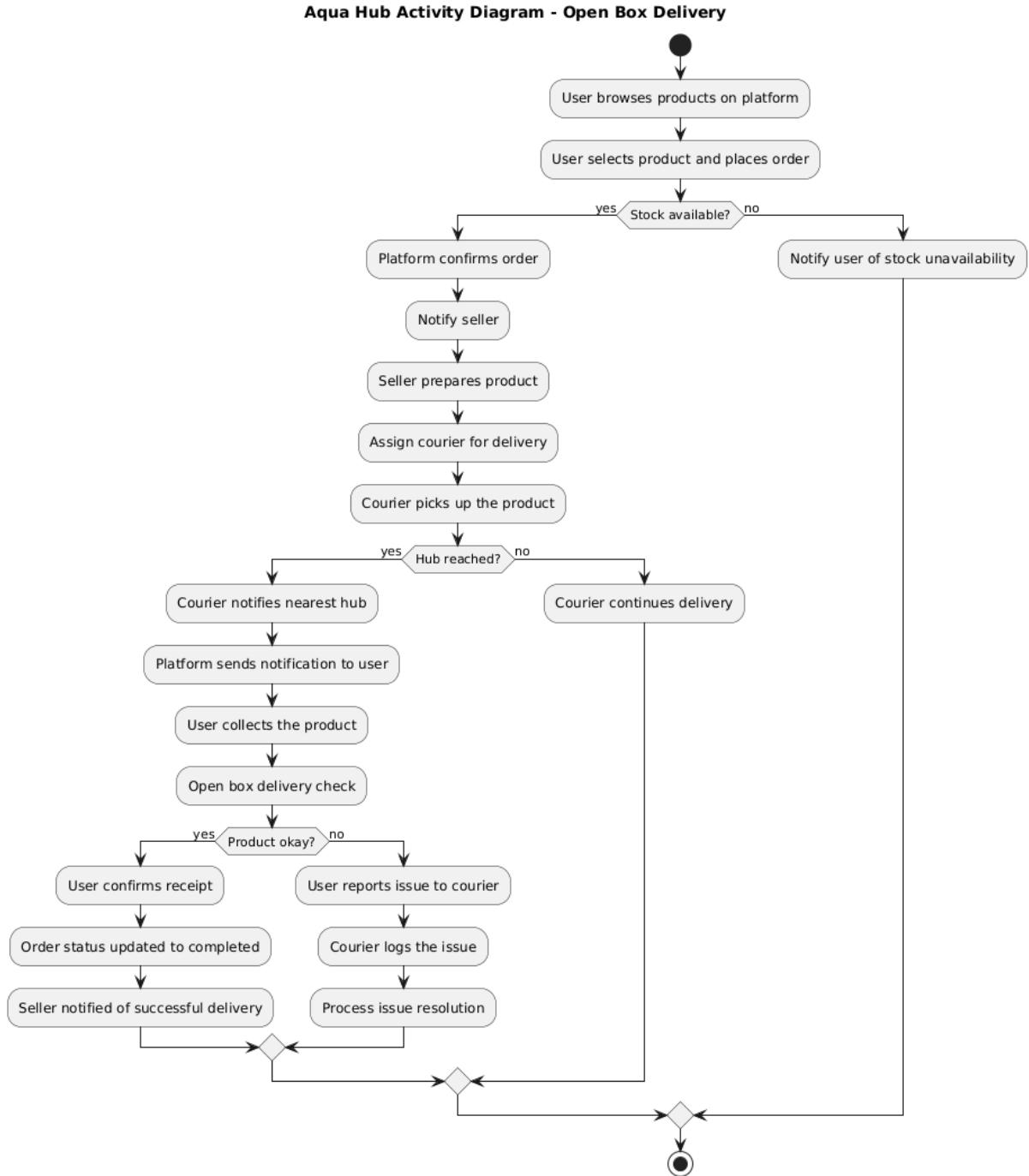


Fig 4.2.4

4.2.5 Class Diagram

The Class Diagram for Aqua Hub illustrates the relationships between various entities within the system, including users, sellers, products, and orders. It defines the structure of the system by showing the classes, their attributes, methods, and the associations between them.

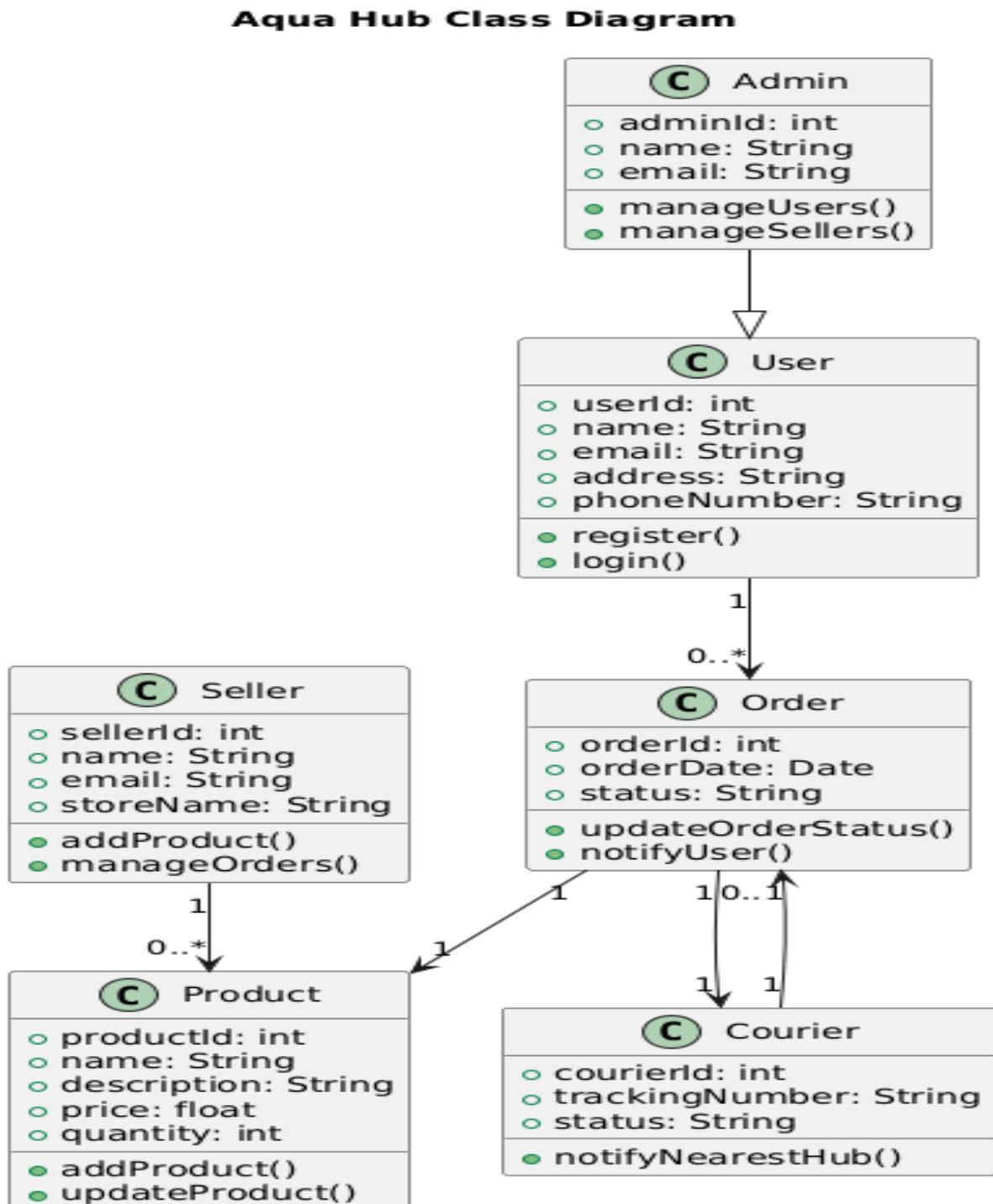


Fig 4.2. 5

4.2.6 Object Diagram

The Object Diagram in Aqua Hub showcases specific instances of system entities like customers, sellers, products, orders, and the admin. It reflects relationships at a particular point in time

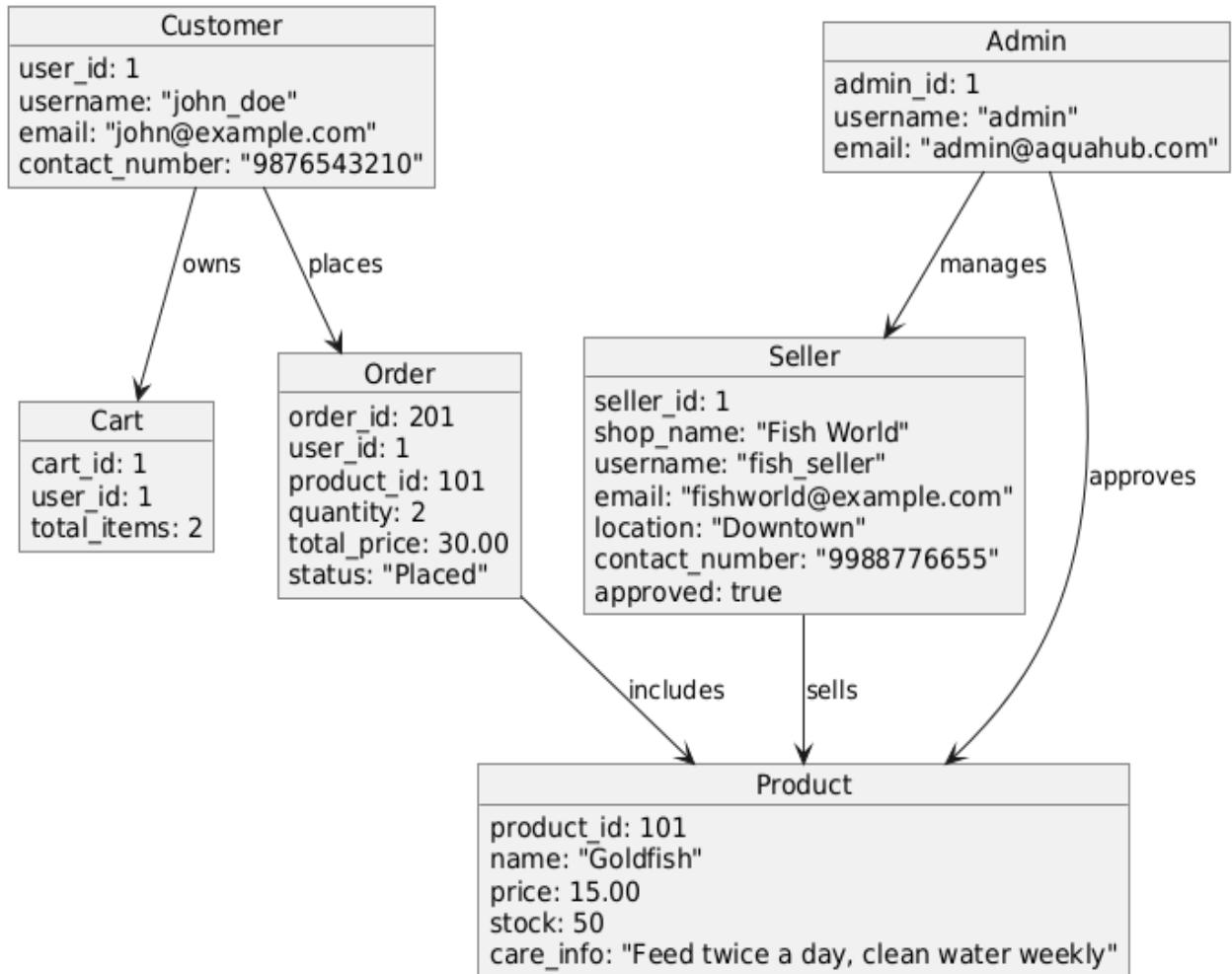


Fig 4.2.6

4.2.7 Component Diagram

The Component Diagram for Aqua Hub depicts the structural organization of the system, showcasing the interaction between essential components like the web server, database server, payment gateway, and client devices. This diagram provides a clear overview of how these components collaborate to deliver a seamless e-commerce experience, facilitating user

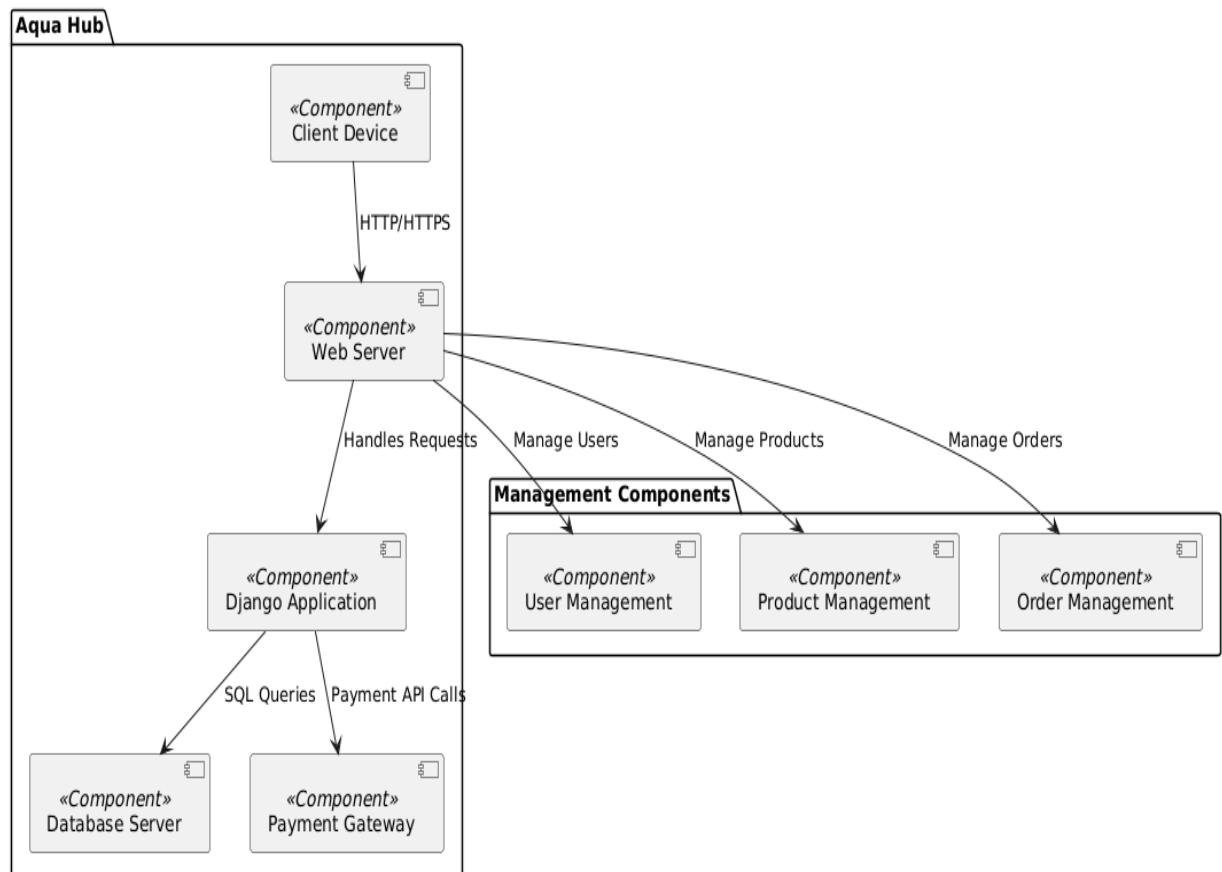


Fig 4.2.7

4.2.8 Deployment Diagram

The Deployment Diagram shows the physical arrangement of hardware and software components involved in the Aqua Hub platform. It demonstrates how the system is deployed across various nodes (such as servers and clients) and how components interact within the infrastructure.

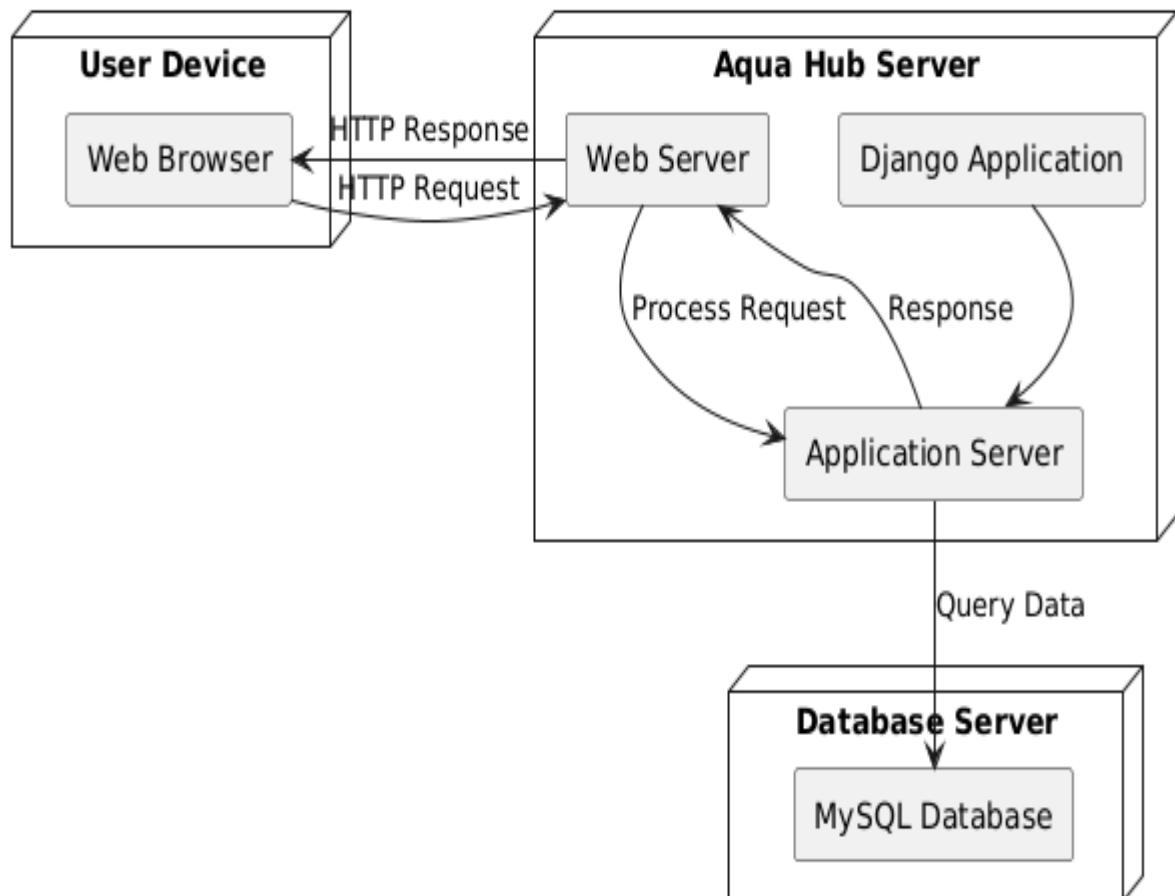


Figure 4.2.8

4.2.9 Collaboration Diagram

The Collaboration Diagram for Aqua Hub illustrates how key objects interact to complete specific tasks. It shows the Customer adding a Product to their Cart, which retrieves product details and updates the total. When the customer places an order, the Order object notifies the Seller and initiates payment processing through the Payment object. This diagram emphasizes the relationships and communication between objects during critical operations, providing a clear view of the system's dynamic behavior.

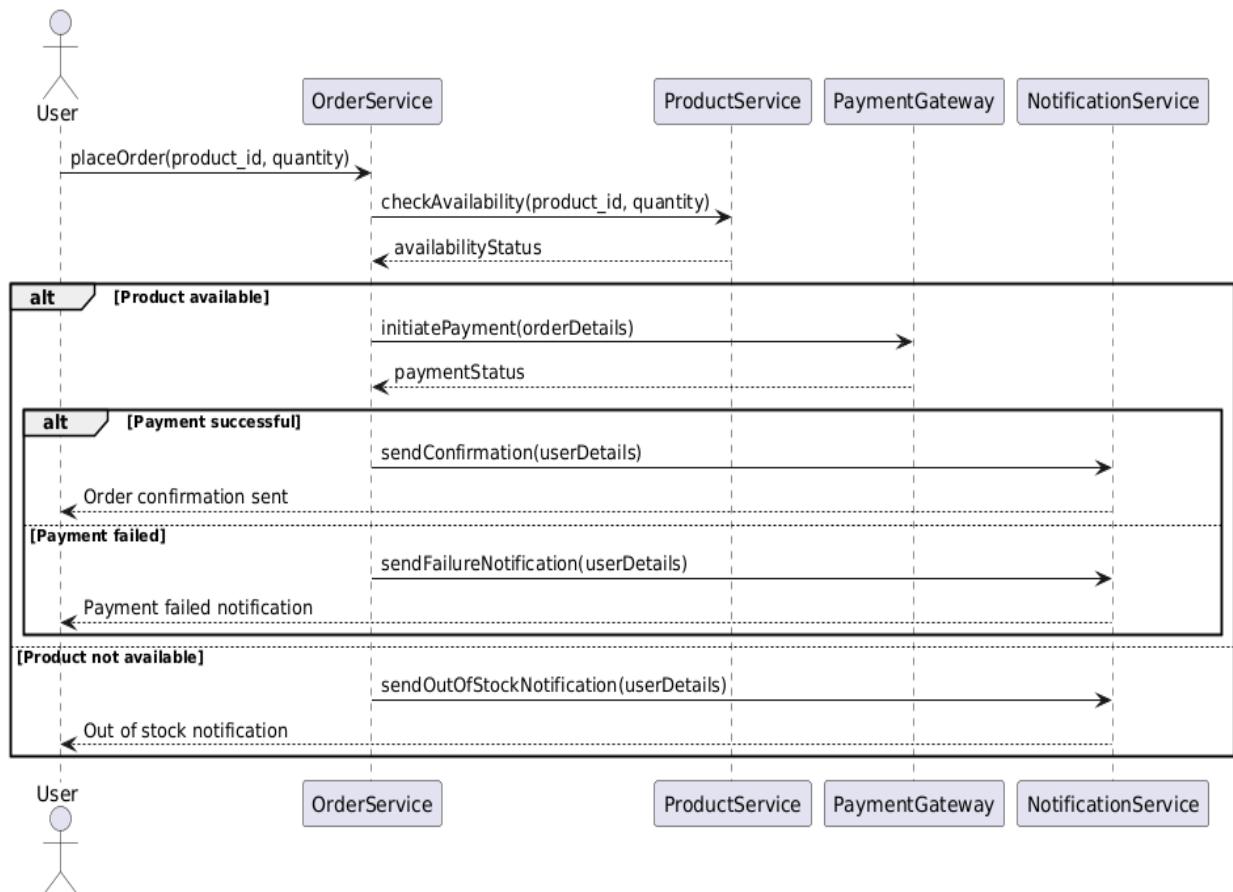
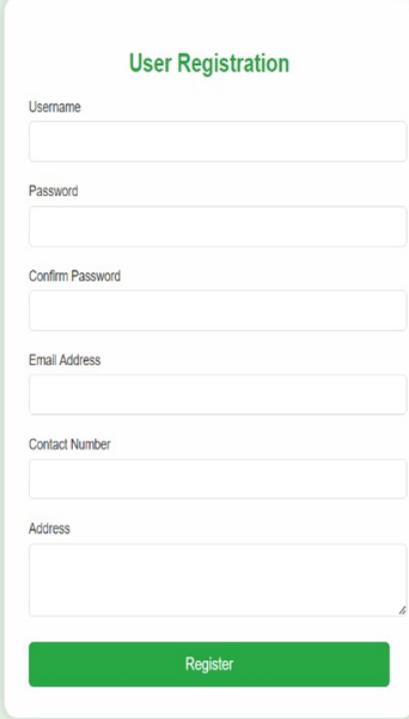


Figure 4.2. I

4.3 USER INTERFACE DESIGN USING FIGMA

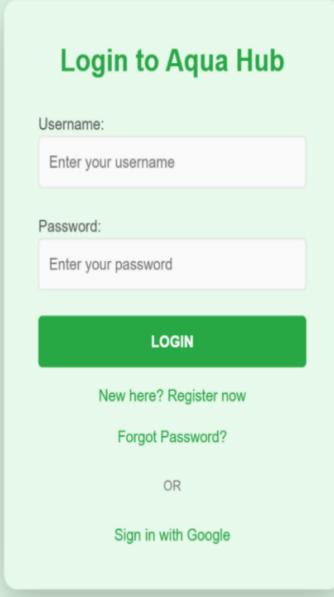
Form Name: User Reg



The form is titled "User Registration". It contains six input fields with labels: "Username", "Password", "Confirm Password", "Email Address", "Contact Number", and "Address". A large green "Register" button is at the bottom.

Figure 4.3. 1

Form Name: User Login



The form is titled "Login to Aqua Hub". It has two input fields: "Username" and "Password", both with placeholder text "Enter your username" and "Enter your password". Below the inputs is a large green "LOGIN" button. Below the button are links for "New here? Register now" and "Forgot Password?". A horizontal line labeled "OR" separates this from the "Sign in with Google" link below it.

Figure 4.3. 2

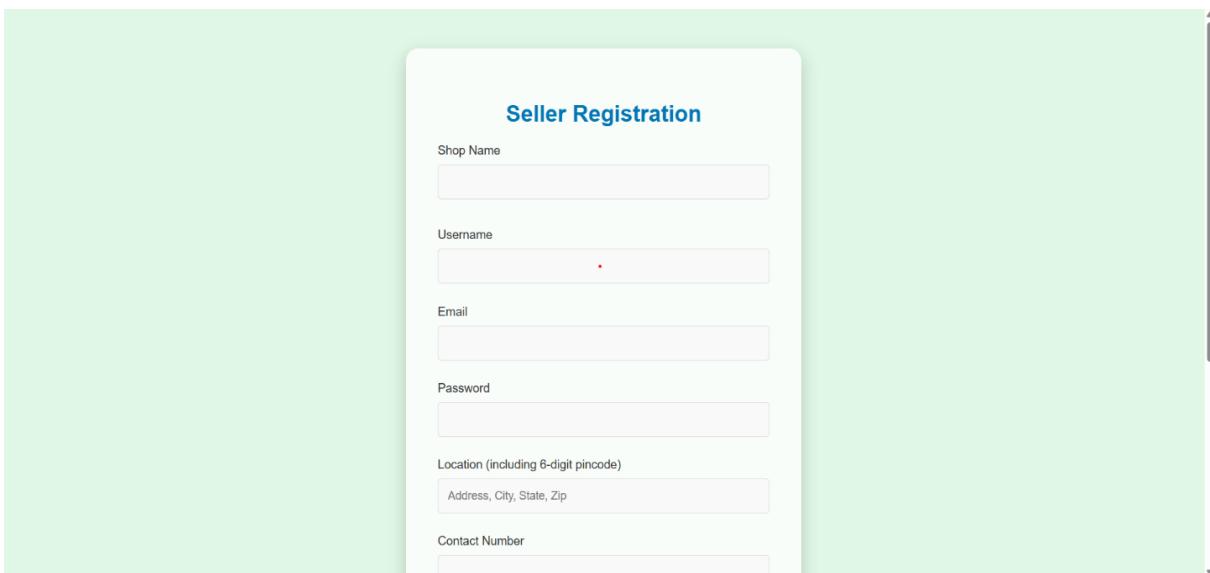
Form Name: Seller Reg

Figure 4.3. 3

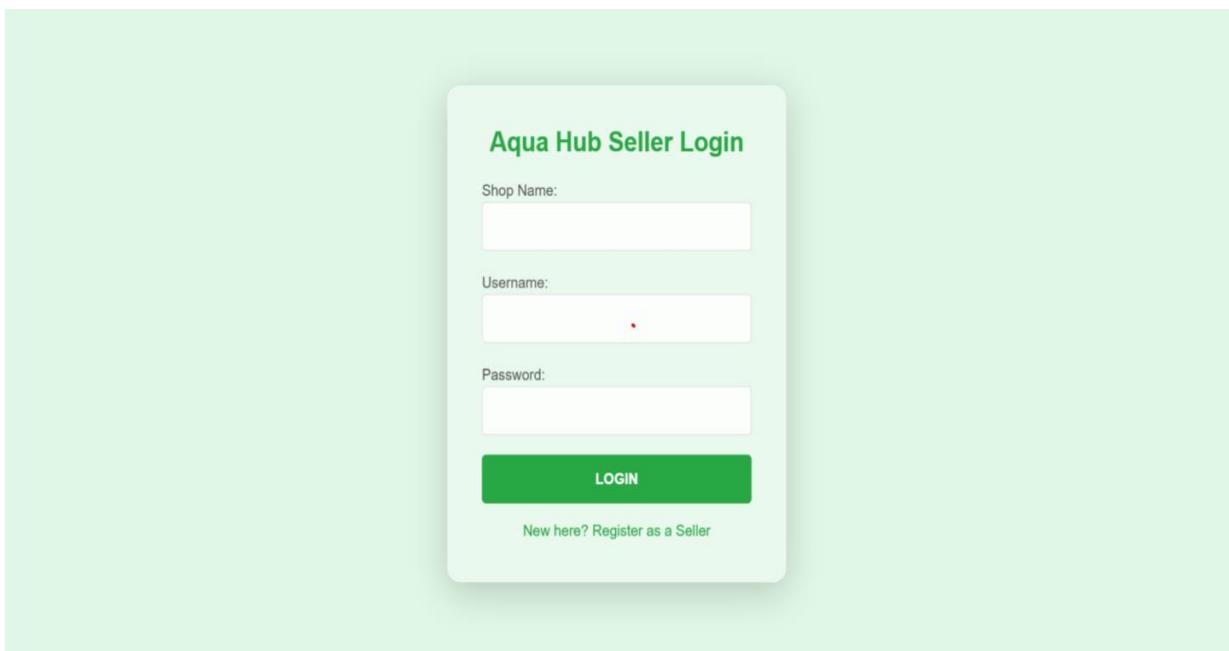
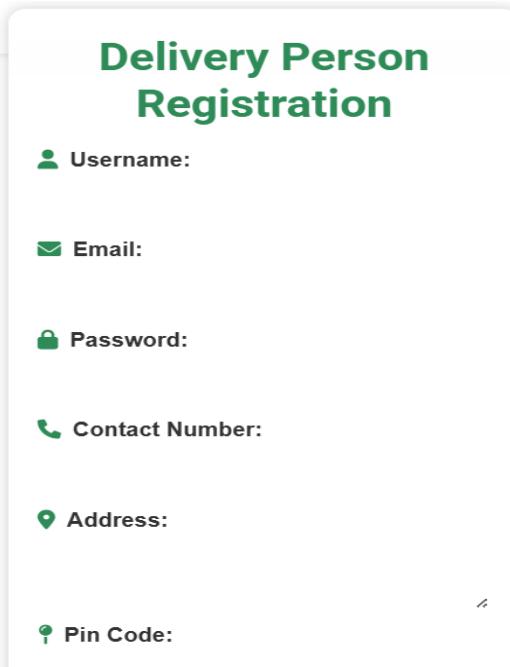
Form Name: Seller Login

Figure 4.3. 4

Form Name: Delivery Registration

The form is titled "Delivery Person Registration". It contains six input fields with icons: "Username" (user icon), "Email" (envelope icon), "Password" (key icon), "Contact Number" (phone receiver icon), "Address" (location pin icon), and "Pin Code" (pin icon). There is also a small "Forgot Password?" link at the bottom right.

Delivery Person Registration

Username:

Email:

Password:

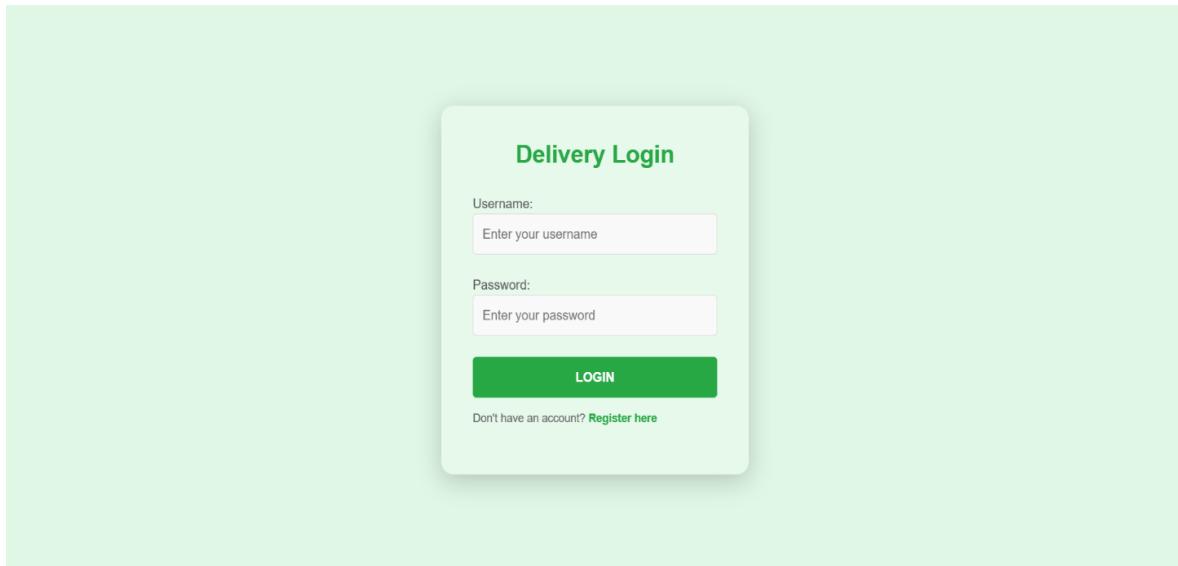
Contact Number:

Address:

Pin Code:

[Forgot Password?](#)

Figure 4.3. 5

Form Name: Delivery Login

The form is titled "Delivery Login". It has two input fields: "Username" and "Password", each with a placeholder "Enter your username" and "Enter your password" respectively. Below the fields is a green "LOGIN" button. At the bottom, it says "Don't have an account? [Register here](#)".

Delivery Login

Username:
Enter your username

Password:
Enter your password

LOGIN

Don't have an account? [Register here](#)

Figure 4.3. 6

Form Name: User Complaint**Register a Complaint**

Select Seller

Order ID

If your complaint is regarding an order, please enter the Order ID.

Subject

Description

SUBMIT COMPLAINT

Figure 4.3. 7

4.4 DATABASE DESIGN

4.4.1 Relational Database Management System (RDBMS)

Aqua Hub uses MySQL as the Relational Database Management System (RDBMS) to store and manage its data. RDBMS ensures data integrity, supports SQL queries, and provides efficient data manipulation through tables and relationships. Each entity in the Aqua Hub system, such as users, sellers, products, and orders, is stored in a structured table format with defined relationships (e.g., foreign keys), allowing seamless querying and updating of data. MySQL's reliability, scalability, and support for ACID (Atomicity, Consistency, Isolation, Durability) properties make it an ideal choice for Aqua Hub's e-commerce platform.

4.4.2 Normalization

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. Aqua Hub's database follows standard normalization practices:

First Normal Form (1NF): All columns in the tables contain atomic values, ensuring no repeating groups.

Second Normal Form (2NF): All non-key attributes are fully dependent on the primary key, eliminating partial dependency.

Third Normal Form (3NF): There are no transitive dependencies; non-key attributes depend solely on the primary key.

These normalization steps ensure that the database remains efficient, reduces data duplication, and makes it easier to manage updates and queries.

4.4.3 Sanitization

Sanitization is the process of cleaning user input to prevent malicious data from being stored or executed in the database. Aqua Hub employs data sanitization techniques to protect against common security threats, such as SQL injection. User inputs, especially in forms like login, registration, product details, and complaints, are validated and sanitized before being stored in the MySQL database. This ensures that only safe, valid data is entered into the system, thereby enhancing security.

4.4.4 Indexing

Indexing is used to improve query performance by allowing the database to locate and retrieve records more efficiently. In Aqua Hub, indexing is applied to frequently queried columns, such as primary keys, foreign keys, and searchable fields like username, email, product_id, and shop_name. Proper indexing ensures that queries for fetching product details, user profiles, and order information are optimized for speed, reducing the time it takes to retrieve data and improving the overall performance of the platform.

4.5 TABLE DESIGN

1. Users

Primary key: **loginid**

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User_Id	Int(11)	PK	Id of the users login
2	Password	Varchar(50)	Not Null	Password for the logged inn users
3	Last_login	Datetime(6)	Not Null	Date for the last login
4	Is_superuser	Tinyint(1)	Not Null	Checks wheather the user is super user
5	Username	Varchar(50)	Not Null	Username for the user
6	First name	Varchar(50)	Not Null	First name of the user
7	Last name	Varchar(50)	Not Null	Last name of the user
8	Is_active	Tinyint(1)	Not Null	Checks if user is active
9	Date_joined	Datetime(6)	Null	The joined date of the user

2.Table Sellers

Primary key: **id**

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Id	Int(11)	PK	Seller id
2	Shop_name	Varchar(100)	Not Null	Name of the shop
3	Email	Varchar(100)	Not Null	Email of the shop
4	Location	Varchar(100)	Not Null	Location of the shop
5	Contact_num	Int(50)	Not Null	Contact number of the seller
6	Product_type	Varchar(50)	Not Null	Selection of product category

3. Product Table

Primary Key: id

Foreignkey:seller

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Seller	Int	Foreignkey	Seller id (Foreign key)
2	Product name	Varchar(50)	Not Null	Name of the seller
3	Price	Int(11)	Not Null	Product Price
4	Description	Varchar(50)	Not Null	Product Description
5	Image	Models.imagefiled	Not Null	Product Image
6	Stock	Boolean Filed	Not Null	Product Stock
7	Water_quality	Varchar(255)	Not Null	Details about water quality

8	Tank Size	Varchar(255)	Not Null	Details about Tank size
9	Feeding	Vrachar(255)	Not Null	Details about Feeding
9	Behaviour	Varchar(255)	Not Null	Details about Fish Behaviour
10	Helath Issues	Varchar(255)	Not Null	Details about Health Issues

4.Blog Table

PrimaryKey: blog_id

Foreginkey: user

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User	Int(11)	ForeignKey	Foreign key connecting this table with user table
2	Title	Varchar(25)	Not Null	Blog Title
3	image	Models.Imagefiled	Not Null	Blog image
4	Description	Varchar(250)	Not Null	Blog Details
5	Created_at	Datetime field	Not Null	Date and time of creation

5.Cart Table

Primarykey=id

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User	Int(11)	ForeignKey	ForeignKey Field
2	Created at	Datetime	Not Null	Date and time of creation

6. Cart Item

PrimaryKey: cart_id

ForeignKey: Cart,Product

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Cart	Int	Foreignkey	Foreignkey connecting cart
2	Product	Varchar(50)	ForeignKey	ForeignKey connecting product table to cart
3	Quanity	Int(11)	Not Null	Quanity which is added to the cart
4	Added_at	DateTimeField	DateTimeField	DateTimeField

7. Comment Table

PrimaryKey: comment_id

ForeignKey: Blog,User

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Blog	Int(11)	ForeignKey	Key connecting Blog table to comment table
2	User	Int(11)	ForeignKey	Key Connecting User to comments
3	Content	Vrachar()25	Not Null	Comment Content

8. Complaint Table

PrimaryKey: complaint_id

ForeignKey: User,Seller

No :	Field name	Datatype (Size)	Key Constraints	Description of the field

1	User	Varchar(100)	Foreign Key	Foreign Key Connecting User to Complaints
2	Seller	Varchar(100)	Foreign Key	Key connecting seller
3	Subject	Varchar(100)	Not Null	Subject Of complaint
4	Description	Varchar(100)	Not Null	Description of Complaint
5	Payment_id	Varchar(100)	Null	Proof of payment
6	Created_at	Datetime	Not Null	DateTime Field

9.User Address Table

PrimaryKey: address_id

ForeignKey: User

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User	Varchar(100)	Foreign Key	Foreign Key Connecting user with address
2	Full_name	Varchar(100)	Not Null	Full name of the user
3	Contact1	Int	Not Null	Contact number of the user
4	Contact2	Int	Not Null	Contact Number of user
5	Locality	Varchar(100)	Not Null	Locality of the user
6	Address	Varchar(100)	Not Null	Full Address
7	Landmark	Varchar(100)	Not Null	Landmark
8	Pincode	Int	Not Null	Pincode of the user
9	State	Varchar(100)	Not Null	State of the user

10.Order Table

PrimaryKey: Order_id

ForeignKey: User,Product

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User	Varchar	ForiegnKey	Foreign Key Connecting user with order
2	Product	Varchar(100)	ForeignKey	Product which is being ordered
3	Quanity	Int(50)	Not Null	Quantity Of Order
4	Total_price	Int(50)	Not Null	Total price of order
5	Address	Varchar(100)	Not Null	Adress os the user from UserAddress Table
6	Payment_id	Varchar(100)	Not Null	Payment_id Generated by API
7	Payment_status	Varchar(100)	Not Null	Payment Update
8	Order_status	Varchar(100)	Not Null	Status of order
9	Created_at	Date time	Not Null	Order of date and time

11.Rating Table

Primarykey : product_id

ForeignKey: product,user

No :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	product	Varchar	Foreignkey	Foreignkey connecting product to review
2	User	Varchar	Foreignkey	Foreignkey
3	Rating	Int	Not Null	For enterting rating
4	Review_text	Varchar	Not Null	Review text
5	Created at	Date	Not Null	Date and time

12.Wishlist Table

Primarykey: id

Foreignkey: User,Product

No : :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User	Varchar	Foreignkey	Foreignkey fields connecting the user to wishlist
2	Product	Varchar	Foreignkey	Foreignkey fields connecting the product to wishlist
3	Added_on	Date	Datetime field	Datetime field

13.Fishwater Preference Table

Primarykey: Fish_id

No : :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Fish_name	Varchar	Not Null	Fishname
2	Min_ph	Float	Not Null	Minimun Ph
3	Max_ph	Float	Not Null	Maximum Ph
4	Min_ph	Float	Not Null	Minimum Ph
5	Max_ph	Float	Not Null	Maximum Ph
6	Hardness	Char	Not Null	Water Hardness

14.Fish careguide Table

Primarykey: id

No : :	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Fish_name	Char	Not Null	Fish Name
2	Description	Text	Not Null	Care description

3	Feeding_habits	Text	Not Null	Fish feeding Habits
4	Water_quality	Text	Not Null	Water quality of fish
5	Fish_image	ImageField	Not Null	Fish Image

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

System testing is a critical phase in the software development lifecycle that aims to evaluate the complete and integrated software system to ensure it meets specified requirements. This phase verifies that all components function as expected in various scenarios and identifies any defects before the product is deployed. System testing involves a comprehensive assessment of the software's functionality, performance, security, and usability. By conducting thorough testing, stakeholders can ensure the software is reliable, user-friendly, and ready for deployment, ultimately leading to a successful product launch.

5.2 TEST PLAN

A test plan outlines the strategy and approach for testing the Aqua Hub system. It defines the scope of testing, objectives, resources, timelines, and deliverables. The test plan serves as a roadmap for the testing process, detailing the types of tests to be performed, such as unit, integration, system, user acceptance, and automation testing. It ensures all stakeholders understand the testing activities and helps coordinate efforts among the development and testing teams. A well-structured test plan enhances the likelihood of identifying defects early and ensuring the software meets quality standards.

5.2.1 Unit Testing

Unit testing focuses on validating individual components or modules of the Aqua Hub system in isolation. Each unit is tested independently to ensure it behaves as intended and meets design specifications. This type of testing helps identify issues at an early stage, facilitating easier debugging and ensuring that each component functions correctly before integration. Automated testing frameworks are often used for unit testing, allowing for efficient and repeatable tests, ultimately leading to higher code quality and reliability.

5.2.2 Integration Testing

Integration testing evaluates the interaction between different modules or components of the Aqua Hub system to ensure they work together seamlessly. This phase aims to identify interface defects, data flow issues, and inconsistencies between integrated units. By testing the integrated components as a group, integration testing helps verify that combined functionality meets the system requirements and behaves as expected in a cohesive manner. This stage is essential for ensuring that the individual components, when integrated, do not introduce new defects or alter the expected behavior.

5.2.3 Validation Testing or System Testing

Validation testing, also known as system testing, is conducted to verify that the entire Aqua Hub system meets the specified requirements and functions correctly as a whole. This comprehensive testing phase assesses the system's functionality, performance, security, and usability under various scenarios, including edge cases. Validation testing ensures that all components work together as intended and that the system meets user expectations. By simulating real-world usage, this testing phase helps identify any defects that may impact the end-user experience.

5.2.4 Output Testing or User Acceptance Testing

Output testing, often referred to as user acceptance testing (UAT), is the final phase of testing before the Aqua Hub system goes live. This phase involves end-users testing the system to ensure it meets their needs and requirements. UAT focuses on validating that the software functions correctly in a real-world environment and provides the expected outcomes. Feedback from end-users is crucial during this phase, as it helps identify any remaining issues, enhances user satisfaction, and ensures the system is ready for deployment.

5.2.5 Automation Testing

Automation testing involves using automated tools and scripts to execute tests on the Aqua Hub system, improving efficiency and accuracy in the testing process. This approach is particularly beneficial for repetitive tasks, regression testing, and scenarios requiring high precision. Automation testing helps save time and resources while increasing test coverage. By integrating automation into the testing process, teams can identify defects faster and ensure that the software remains stable throughout development, ultimately leading to a more reliable product.

5.2.6 Selenium Testing

Selenium testing is a widely used automation testing framework for web applications, including the Aqua Hub system. Selenium allows testers to write scripts in various programming languages to simulate user interactions with the web application, validating that it behaves as expected across different browsers and devices. By automating functional and regression tests, Selenium testing enhances testing efficiency, reduces manual effort, and helps maintain consistent quality in the software. It is particularly useful for continuous integration and delivery processes, ensuring that updates do not compromise existing functionality.

Test Case 1: Login Testing

Code

package definitions;

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver; // Import ChromeDriver
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class stepdefanition {
    WebDriver driver = null;
    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.chrome.driver",
workspace\\aquahub\\src\\test\\resources\\driver\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }
    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:8000/login");
        Thread.sleep(2000);
    }
    @When("user enters username and password")
    public void user_enters_username_and_password() throws Throwable {
        driver.findElement(By.id("username")).sendKeys("ko");
        driver.findElement(By.id("password")).sendKeys("Ko@12345");
    }
    @And("user clicks on login")
    public void user_clicks_on_login() {
        driver.findElement(By.id("login")).click();
    }
    @Then("user is navigated to the home page")
    public void user_is_navigated_to_the_home_page() {
        System.out.println("Login successful and user is on the home page");
    }
}
```

```

        driver.close();
        driver.quit();
    }
}

```

```

Oct 22, 2024 3:50:27 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main

@tag @tag1
Scenario: Title of your scenario      # src/test/resources/feature/login.feature:24
Inside step-Browser is open

Oct 22, 2024 3:50:30 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 130, returning the closest version; found: 127; Please update to a Selenium version t
Given browser is open      # definitions.stepdefanition.browser_is_open()
And user is on login page      # definitions.stepdefanition.user_is_on_login_page()
When user enters username and password # definitions.stepdefanition.user_enters_username_and_password()
And user clicks on login      # definitions.stepdefanition.user_clicks_on_login()
Login successful and user is on the home page

```

Test Report

Project Name: Aqua Hub	
Login Test Case	
Test Case ID: Test_1	Test Designed By: Jeswin George
Test Priority(Low/Medium/High):	Test Designed Date: 26-10-2024
Module Name: login Testing	Test Executed By : Jeswin George
Test Title : Login Test	Test Execution Date: 28-10-2024
Description:	Text Executed Succesfully

Pre-Condition :User has valid username and password

Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page	N/A	Login page loads successfully	Login page loads successfully	Pass
2	Fill details	Username “KO” Password ‘Ko@12345’	Username and password fields are successfully filled in	Username and password fields are successfully filled in	Pass
3	Submit Login form	N/A	User is logged in	User is successfully logged in	Pass

4	Close the browser window	N/A			Pass
Post-Condition: User has valid username and password					

Test Case 2: Fish Recommendation

Code

```

package definitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver; // Import ChromeDriver
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class stepdefanition {

    WebDriver driver = null;
    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\\\jeswi\\\\eclipse-
workspace\\\\aquahub\\\\src\\\\test\\\\resources\\\\driver\\\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }
    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:8000/login");
        Thread.sleep(2000);
    }
    @When("user enters username and password")
    public void user_enters_shopname_username_and_password() throws Throwable {
        driver.findElement(By.id("username")).sendKeys("don");
    }
}

```

```

        driver.findElement(By.id("password")).sendKeys("Don@1234");
    }

    @And("user clicks on login")
    public void user_clicks_on_login() {
        driver.findElement(By.id("login")).click();
    }

    @And("user navigate to feature page")
    public void user_on_feature_page() throws Exception{
        driver.navigate().to("http://localhost:8000/water-quality-analyzer");
        Thread.sleep(2000);
    }

    @Then("user enters data")
    public void user_enters_data() {
        driver.findElement(By.id("ph")).sendKeys("6.5");
        driver.findElement(By.id("temp")).sendKeys("21");
    }

    @Then("user click analyze button")
    public void user_click_disable_button() {
        driver.findElement(By.id("analyze")).click();
        System.out.println("Successfully analyzed");
    }

}

```

Screenshot

```

@tag @tag1
Scenario: Title of your scenario      # src/test/resources/feature/login.feature:24
Inside step:Browser is open
Mar 17, 2025 2:20:35 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 134 *
Mar 17, 2025 2:20:35 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for 134.0.6998.89. You may need to include a dependency on a specific version of the CDP using
Given browser is open      # definitions.stepdefanition.browser_is_open()
And user is on login page      # definitions.stepdefanition.user_is_on_login_page()
When user enters username and password # definitions.stepdefanition.user_enters_shopname_username_and_password()
And user clicks on login      # definitions.stepdefanition.user_clicks_on_login()
And user navigate to feature page      # definitions.stepdefanition.user_on_feature_page()
Then user enters data      # definitions.stepdefanition.user_enters_data()
Successfully analyzed
Then user click analyze button      # definitions.stepdefanition.user_click_disable_button()

1 Scenarios (1 passed)
7 Steps (7 passed)
0m7.652s|

```

Test report

Project Name:Aqua Hub					
Recommendation Test Case					
Test Case ID: Test_1	Test Designed By: Jeswin George				
Test Priority(Low/Medium/High):	Test Designed Date: 17-03-2025				
Module Name: Recommendation Testing	Test Executed By : Jeswin George				
Test Title : Recommendation Test	Test Execution Date: 17-03-2025				
Description:	Text Executed Successfully				
Pre-Condition : User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page	N/A	Login page loads successfully	Login page loads successfully	Pass
2	Fill details	Username “don” Password ‘Don@1234’	Username and password fields are successfull y filled in	Username and password fields are successfully filled in	Pass
3	Submit Login form	N/A	User is logged in	User is successfully logged in	Pass
4	Clicks on the feature button	N/A	User is in the feature page	User is in the recommendation page	Pass
Post-Condition: User has valid username and password					

Test Case 3 : Compatibility Checking

Code

```
package definitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver; // Import ChromeDriver
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
```

```
public class comparison {
    WebDriver driver = null;
    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\\\jeswi\\\\eclipse-
workspace\\\\aquahub\\\\src\\\\test\\\\resources\\\\driver\\\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }
    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:8000/login");
        Thread.sleep(2000);
    }
    @When("user enters username and password")
    public void user_enters_shopname_username_and_password() throws Throwable {
        driver.findElement(By.id("username")).sendKeys("don");
        driver.findElement(By.id("password")).sendKeys("Don@1234");
    }
    @And("user clicks on login")
    public void user_clicks_on_login() {
        driver.findElement(By.id("login")).click();
    }
    @And("user navigate to comparison page")
    public void user_on_feature_page() throws Exception{
        driver.navigate().to("http://127.0.0.1:8000/fish-compatibility/");
        Thread.sleep(2000);
    }
    @Then("user enters data")
    public void user_enters_data() {
```

```

        driver.findElement(By.id("fish1")).sendKeys("Guppy");
        driver.findElement(By.id("fish2")).sendKeys("Molly");
    }

    @Then("user click check button")
    public void user_click_disable_button() {
        driver.findElement(By.id("check")).click();
        System.out.println("Successfully analyzed");
    }

}

```

Screenshot

```

@tag @tag1
Scenario: Title of your scenario      # src/test/resources/feature/comparison.feature:22
Inside step-Browser is open
Mar 17, 2025 2:39:45 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 134
Mar 17, 2025 2:39:45 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for 134.0.6998.89. You may need to include a dependency on a specific version of the CDP using
Given browser is open          # definitions.comparison.browser_is_open()
And user is on login page      # definitions.comparison.user_is_on_login_page()
When user enters username and password # definitions.comparison.user_enters_shopname_username_and_password()
And user clicks on login       # definitions.comparison.user_clicks_on_login()
And user navigate to comparison page # definitions.comparison.user_navigates_to_comparison_page()
Then user enters data          # definitions.comparison.user_enters_data()
Successfully analyzed
    Then user click check button      # definitions.comparison.user_click_disable_button()

1 Scenarios (1 passed)
7 Steps (7 passed)
0m7.879s

```

Test report

Project Name:Aqua Hub					
Compatibility Test Case					
Test Case ID: Test_1	Test Designed By: Jeswin George				
Test Priority(Low/Medium/High):	Test Designed Date: 17-03-2025				
Module Name: Combatibility Testing	Test Executed By : Jeswin George				
Test Title : Login Test	Test Execution Date: 17-03-2025				
Description:	Text Executed Succesfully				
Pre-Condition : User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)

1	Navigate to login page	N/A	Login page loads successfully	Login page loads successfully	Pass
2	Fill details	Username “don” Password ‘Don@1234’	Username and password fields are successfully filled in	Username and password fields are successfully filled in	Pass
3	Submit Login form	N/A	User is logged in	User is successfully logged in	Pass
4	Clicks on the Compatibility button	N/A	User is in the Checking page	User is in the checking page	Pass
5	User enters the fish1 and fish2	Fish1”Molly”, Fish2”Guppy”	User gets results according to input	User gets Compatibility	Pass

Test Case 4: Add To Cart

Code

```

package definitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver; // Import ChromeDriver
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class stepdefanition {

    WebDriver driver = null;
    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\\\jeswi\\\\eclipse-
workspace\\\\aquahub\\\\src\\\\test\\\\resources\\\\driver\\\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }
}

```

```
}

@And("user is on login page")
public void user_is_on_login_page() throws Exception {
    driver.navigate().to("http://localhost:8000/login");
    Thread.sleep(2000);
}

@When("user enters username and password")
public void user_enters_username_and_password() throws Throwable {
    driver.findElement(By.id("username")).sendKeys("ko");
    driver.findElement(By.id("password")).sendKeys("Ko@12345");
}

@And("user clicks on login")
public void user_clicks_on_login() {
    driver.findElement(By.id("login")).click();
}

@Then("user is navigated to the home page")
public void user_is_navigated_to_the_home_page() {
    driver.findElement(By.id("viewb")).click();
}

@Then("user click add to cart")
public void user_clickaddto_cart() {
    driver.findElement(By.id("addToCartBtn")).click();
}

}
```

Screenshot

```

Given browser is open          # definitions.stepdefanition.browser_is_open()
And user is on login page      # definitions.stepdefanition.user_is_on_login_page()
When user enters username and password # definitions.stepdefanition.user_enters_username_and_password()
And user clicks on login        # definitions.stepdefanition.user_clicks_on_login()
Then user is navigated to the home page # definitions.stepdefanition.user_is_navigated_to_the_home_page()
Then user click add to cart     # definitions.stepdefanition.user_clickaddto_cart()

```

1 Scenarios (1 passed)

6 Steps (6 passed)

0m7.180s

Test Report

Project Name:Aqua Hub					
Add to cart test case					
Test Case ID: Test_2		Test Designed By: Jeswin George			
Test Priority(Low/Medium/High):		Test Designed Date: 14-03-2024			
Module Name: Add to cart		Test Executed By : Jeswin George			
Test Title : Add to cart		Test Execution Date: 14-03-2024			
Description:		Text Executed Successfully			
Pre-Condition :User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page	N/A	Login page loads successfully	Login page loads successfully	Pass
2	Fill details	Username “KO” Password ‘Ko@12345’	Username and password fields are successfully filled in	Username and password fields are successfully filled in	Pass
3	User is on product page	N/A	User clicks on View details Enters product details page	User clicks on View details Enters product details page	Pass

4	User select quantity	N/A	User selects quantity and clicks add to cart	Quanity is selected and its is added to cart	Pass
---	----------------------	-----	--	--	------

Post-Condition: User has valid username and password

Test Case 5: Disabling Product

Code

```

package definitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver; // Import ChromeDriver
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class stepdefanition {

    WebDriver driver = null;
    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\\\jeswi\\\\eclipse-
workspace\\\\aquahub\\\\src\\\\test\\\\resources\\\\driver\\\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }
    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:8000/slogin");
        Thread.sleep(2000);
    }
    @When("user enters shop name username and password")
    public void user_enters_shopname_username_and_password() throws Throwable {

```

```
driver.findElement(By.id("shop-name")).sendKeys("fly fish");
driver.findElement(By.id("username")).sendKeys("fly");
driver.findElement(By.id("password")).sendKeys("Fly@1234");
}

@And("user clicks on login")
public void user_clicks_on_login() {
    driver.findElement(By.id("sclick")).click();
}

@Then("user click disable button")
public void user_click_disable_button() {
    driver.findElement(By.id("disable")).click();
}

}
```

Screenshot

```
Given browser is open          # definitions.stepdefanition.browser_is_open()
And user is on login page      # definitions.stepdefanition.user_is_on_login_page()
When user enters shop name username and password # definitions.stepdefanition.user_enters_shopname_username_and_password()
And user clicks on login        # definitions.stepdefanition.user_clicks_on_login()
Then user click disable button # definitions.stepdefanition.user_click_disable_button()
```

1 Scenarios (1 passed)

5 Steps (5 passed)

3m5.664s

Test Report

Project Name:Aqua Hub					
Disable product Test Case					
Test Case ID: Test_3	Test Designed By: Jeswin George				
Test Priority(Low/Medium/High):	Test Designed Date: 3-15-2025				
Module Name: Disable Product	Test Executed By : Jeswin George				
Test Title : Disable Product	Test Execution Date: 3-16-2024				
Description:	Text Executed Succesfuly				
Pre-Condition : User has valid shop name username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page	N/A	Login page loads successfully	Login page loads successfully	Pass
2	Fill details	Shopname”tk aquatics” Username “tk” Password ‘tk@1234’	shopname Username and password fields are successfully filled in	shopnameUser name and password fields are successfully filled in	Pass
3	Submit Login form	N/A	User is logged in	User is successfully logged in	Pass
4	Disable Button is clicked on seller home	N/A	User clicks and disable product	User clicks and disable product	Pass
Post-Condition: User has valid username and password					

Test case 6: Download File

Code

```
package definitions;
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.support.ui.ExpectedConditions;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver; // Import ChromeDriver
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class newtest {
    WebDriver driver = null;

    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\jeswi\\eclipse-
workspace\\aquahub\\src\\test\\resources\\driver\\chromedriver.exe");
        driver = new ChromeDriver();
        driver.manage().window().maximize();
    }

    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:8000/login");
        Thread.sleep(2000);
    }

    @When("user enters username and password")
    public void user_enters_username_and_password() throws Throwable {
        driver.findElement(By.id("username")).sendKeys("don");
        driver.findElement(By.id("password")).sendKeys("Don@1234");
    }

    @And("user clicks on login")
    public void user_clicks_on_login() {
```

```

        driver.findElement(By.id("login")).click();
    }

    @And("user navigate to careguide page")
    public void user_navigates_to_careguide_page() throws Exception{
        driver.navigate().to("http://127.0.0.1:8000/create/");
        Thread.sleep(3000);
    }

    @Then("user clicks on view care guides")
    public void user_clicks_on_view_care_guides() throws Exception {
        driver.navigate().to("http://127.0.0.1:8000/list/");
        Thread.sleep(2000);
    }

    @Then("user click download button")
    public void user_click_download_button() {
        driver.findElement(By.id("wassss")).click();
        System.out.println("Successfully Downloaded");
    }

}

```

Screenshot

```

@tag @tag1
Scenario: testing for downloding      # src/test/resources/feature/newtest.feature:22
Inside step-Browser is open
Mar 18, 2025 12:35:17 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 134
Mar 18, 2025 12:35:17 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for 134.0.6998.89. You may need to include a dependency on a specific version of the
Given browser is open          # definitions.newtest.browser_is_open()
And user is on login page      # definitions.newtest.user_is_on_login_page()
When user enters username and password # definitions.newtest.user_enters_username_and_password()
And user clicks on login       # definitions.newtest.user_clicks_on_login()
And user navigate to careguide page # definitions.newtest.user_navigates_to_careguide_page()
    Then user clicks on view care guides # definitions.newtest.user_clicks_on_view_care_guides()
Successfully Downloaded
    Then user click download button     # definitions.newtest.user_click_download_button()

1 Scenarios (1 passed)
7 Steps (7 passed)
0m10.040s

```

Test Report

Project Name:Aqua Hub					
Download Test Case					
Test Case ID: Test_1			Test Designed By: Jeswin George		
Test Priority(Low/Medium/High):			Test Designed Date: 17-03-2025		
Module Name: Downloading Testing			Test Executed By : Jeswin George		
Test Title : Recommenation Test			Test Execution Date: 17-03-2025		
Description:			Text Executed Successfully		
Pre-Condition : User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page	N/A	Login page loads successfully	Login page loads successfully	Pass
2	Fill details	Username “don” Password ‘Don@1234’	Username and password fields are successfully filled in	Username and password fields are successfully filled in	Pass
3	Submit Login form	N/A	User is logged in	User is successfully logged in	Pass
4	Clicks on the care guide button	N/A	User is in the Care page page	User is in the care page	Pass
5	Clicks on Download Button	N/A	File is downloaded	File is Downloaded successfully	Pass
Post-Condition: User has valid username and password					

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The implementation of Aqua Hub involves creating a user-friendly e-commerce platform for buying fish and related products. Utilizing Django for the backend and MySQL for data management, the project supports essential features such as user and seller registration, product listings, and a cart system for seamless purchasing. Security measures, including user authentication and seller approval, ensure a safe shopping environment.

The platform introduces unique functionalities like a **Find a Fish** feature, allowing users to search for fish based on specific criteria. Additionally, a **Fish Recommendation System** suggests suitable fish based on user preferences, and a **Fish Compatibility Checker** helps users determine which species can coexist harmoniously.

Users can browse through a diverse range of fish, access detailed product descriptions, and utilize a cart system for convenient purchasing. To enhance customer satisfaction, Aqua Hub integrates an **open-box delivery option** and a **notification system** to keep users informed about their orders. These features contribute to an interactive and informative shopping experience, making Aqua Hub an efficient and scalable marketplace for aquatic enthusiasts.

6.2 IMPLEMENTATION PROCEDURES

The implementation process for Aqua Hub involves several key phases to ensure a successful development and deployment of the e-commerce platform. This process can be broken down into the following steps:

1. **Requirements Gathering:** The first step involves identifying and documenting the functional and non-functional requirements of the platform. This includes defining user roles (customers, sellers, and admin), key features such as registration, product management, cart functionality, and complaint registration. Understanding user needs and expectations is critical for guiding the development process.
2. **System Design:** In this phase, the overall architecture of the platform is designed. This includes creating a detailed database schema using MySQL, designing user interfaces with a focus on usability and aesthetics, and outlining the workflow of the application. UML diagrams, including use case diagrams and class diagrams, can be utilized to visualize the system's structure.
3. **Development:** The development phase involves coding the platform using Django as the web framework. This includes setting up the backend to handle user authentication, seller registration, product listings, and cart management. The frontend development focuses on creating responsive and attractive user interfaces that provide a seamless

experience across devices. Key functionalities, such as the virtual tank builder and complaint registration system, are implemented during this stage.

4. **Testing:** Once development is complete, thorough testing is conducted to identify and fix any bugs or issues. This includes unit testing for individual components, integration testing to ensure that different parts of the application work together, and user acceptance testing (UAT) to validate that the platform meets user expectations. Feedback from testing helps refine the application before launch.
5. **Deployment:** After successful testing, the platform is deployed to a production environment. This involves configuring the server, ensuring database connectivity, and setting up necessary security measures. The deployment process also includes making the platform accessible to users, with proper DNS configuration and hosting.
6. **Maintenance and Support:** Post-deployment, the Aqua Hub team provides ongoing maintenance and support to address any issues that arise, implement updates, and introduce new features based on user feedback. This phase ensures the platform remains functional, secure, and aligned with evolving user needs.
7. **Future Enhancements:** As Aqua Hub grows, future enhancements may include expanding the product offerings to include aquariums and fish food, implementing advanced analytics for better decision-making, and exploring partnerships with logistics providers for improved delivery services.

By following this structured implementation process, Aqua Hub aims to deliver a reliable, efficient, and engaging platform that meets the needs of aquatic enthusiasts while ensuring scalability for future growth.

6.2.1 User Training

User training is a critical component of any software development project. It involves providing end-users with the necessary knowledge and skills to effectively use the software product. The goal of user training is to ensure that end-users can use the software product efficiently and effectively to meet their business needs.

Below are some common steps involved in user training:

- **Identify training needs:** Before training can begin, it is essential to identify the training needs of the end-users. This can be done by conducting a needs analysis, which involves assessing the current skills and knowledge of the end-users and identifying any gaps that need to be addressed.
- **Develop training materials:** Once the training needs have been identified, training materials

can be developed. This can include user manuals, online tutorials, and training videos. The training materials should be designed to be easy to understand and follow, and should cover all the necessary features and functionalities of the software product.

- **Conduct training sessions:** Training sessions can be conducted in-person or online, depending on the needs of the end-users. The training sessions should be interactive and engaging, and should provide opportunities for end-users to ask questions and practice using the software product.
- **Assess learning:** After the training sessions, it is important to assess whether the end-users have learned the necessary skills and knowledge to effectively use the software product. This can be done through quizzes, assessments, or practical exercises.
- **Provide ongoing support:** Even after the initial training, it is important to provide ongoing support to end-users. This can include a helpdesk or support team that can provide assistance when needed, as well as updates to the training materials to reflect any changes or updates to the software product.

Effective user training can help to ensure that end-users are able to use the software product efficiently and effectively, which can lead to increased productivity and improved business outcomes.

6.2.2 Training on the Application Software

Training on application software is an essential component of software implementation. It is the process of providing end-users with the knowledge and skills necessary to operate and use the application software effectively. The goal of training is to ensure that end-users are able to utilize all the features and functionalities of the software to perform their tasks and achieve their goals.

Below are some best practices for providing effective training on application software:

- **Identify the target audience:** Before developing training materials, it is essential to identify the target audience and their specific needs. Different groups of end-users may require different levels of training or different types of training materials.
- **Develop training materials:** Training materials can include user manuals, online tutorials, videos, and live training sessions. The training materials should be designed to be easy to understand and follow, and should cover all the necessary features and functionalities of the software product.
- **Use interactive training methods:** Interactive training methods such as demonstrations, simulations, and hands-on exercises can help end-users to better understand and retain the

training materials.

- **Provide ongoing support:** After the initial training, it is important to provide ongoing support to end-users. This can include a helpdesk or support team that can provide assistance when needed, as well as updates to the training materials to reflect any changes or updates to the software product.
- **Evaluate training effectiveness:** It is important to evaluate the effectiveness of the training by assessing the end-users' knowledge and skills before and after the training. This can help identify areas that may require additional training or support.
- **Provide refresher training:** As the software product evolves, it is important to provide refresher training to end-users to ensure that they are up-to-date with the latest features and functionalities.

Effective training on application software can help to ensure that end-users are able to use the software product effectively, which can lead to increased productivity and improved business outcomes.

6.2.3 System Maintenance

System maintenance is the process of keeping a software system up-to-date and functioning properly after it has been deployed. It involves a variety of activities that are designed to ensure that the system remains stable, secure, and efficient over time. Some of the key activities involved in system maintenance include:

- **Updates and patches:** One of the most important aspects of system maintenance is keeping the system up-to-date with the latest software updates and security patches. These updates and patches are released periodically by software vendors to fix bugs, add new features, and address security vulnerabilities. Applying these updates in a timely manner is essential to ensure that the system remains secure and reliable.
- **Performance monitoring:** Regular performance monitoring is another key aspect of system maintenance. This involves tracking system performance metrics such as CPU usage, memory utilization, and network traffic to identify potential issues and optimize system performance.
- **Backup and recovery:** Backing up system data and ensuring that it can be recovered in the event of a system failure or data loss is an important part of system maintenance. This involves creating regular backups of system data and testing the recovery process to ensure that it works as expected.
- **Security management:** Maintaining system security is critical to protecting sensitive data and preventing unauthorized access to the system. This involves implementing security controls

such as firewalls, antivirus software, and access controls, and regularly monitoring the system for potential security breaches.

- **Hardware maintenance:** Ensuring that hardware components such as servers, storage devices, and network devices are functioning properly is another key aspect of system maintenance. This involves performing regular maintenance tasks such as cleaning, repairs, and upgrades to ensure that the hardware components remain in good working condition.

Effective system maintenance is critical to ensuring that a software system remains reliable, secure, and efficient over time. By implementing a proactive system maintenance strategy, organizations can minimize downtime, prevent system failures, and optimize system performance.

6.2.4 Hosting

Hosting refers to the process of storing and serving website files on a remote server, which can be accessed by visitors over the internet. When you create a website, you need to have it hosted on a server so that it can be available for others to view. There are various types of hosting options available such as shared hosting, dedicated hosting, VPS hosting, cloud hosting, etc. The choice of hosting depends on the size of the website, its traffic volume, and the level of control and flexibility required by the owner.

AWS

AWS (Amazon Web Services) is a cloud hosting service that provides a streamlined platform for deploying web applications, APIs, static sites, and databases. It simplifies the process of hosting applications by automating tasks such as server setup, scaling, and load balancing, which allows developers to focus on building their applications rather than managing infrastructure. AWS supports various languages and frameworks, including Django, and offers seamless integration with Git for continuous deployment. Procedure for hosting a website on AWS:

Deploying on AWS

Step 1: Log in to your AWS account.

Step 2: Create a new project using AWS services like EC2, Elastic Beanstalk, or AWS Lambda, depending on your hosting needs.

Step 3: Create a requirements.txt file that includes all your dependencies to be installed.

Step 4: Upload your project to GitHub or an S3 bucket for deployment.

Step 5: Configure the necessary settings in your AWS service (e.g., security groups, environment variables, and domain settings) and then deploy.

Hosted Website:AWS

Hosted Link: <http://13.60.59.249:8000/>

Hosted Link QR Code



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In conclusion, the Aqua Hub project represents a significant advancement in the e-commerce space for aquatic products, aiming to create a comprehensive and user-friendly platform for fish enthusiasts, sellers, and administrators. Through careful planning, systematic implementation, and a focus on user experience, Aqua Hub is poised to address the specific needs of its users while fostering a vibrant community centered around aquatic life. The integration of innovative features, such as a virtual tank builder and complaint registration system, not only enhances user engagement but also promotes transparency and trust within the marketplace.

As the platform evolves, Aqua Hub will continue to expand its offerings and improve its functionalities based on user feedback and market trends. The commitment to ongoing maintenance and support ensures that Aqua Hub will remain a reliable and efficient resource for aquatic enthusiasts. Ultimately, Aqua Hub aims to become a leading destination in the aquatic market, providing a seamless and enjoyable shopping experience while supporting sellers in effectively reaching their target audience.

7.2 FUTURE SCOPE

The future scope of Aqua Hub encompasses a range of exciting possibilities aimed at enhancing the platform's offerings and expanding its reach within the aquatic market. Key areas for development include:

- **Expanded Product Categories:** In addition to fish, Aqua Hub plans to diversify its inventory by incorporating aquariums, fish food, and aquatic accessories. This expansion will cater to a broader audience, including both novice and experienced aquarium enthusiasts.
- **Enhanced User Experience:** Future updates will focus on improving the user interface and user experience, including personalized recommendations based on user preferences and browsing history. Advanced search and filtering options will be implemented to make product discovery more efficient.
- **Mobile Application Development:** To meet the growing demand for mobile shopping, Aqua Hub aims to develop a mobile application, providing users with a convenient platform to browse products, manage their accounts, and complete purchases on the go.
- **Community Engagement Features:** Introducing community-driven features such as forums, user reviews, and blogging capabilities will encourage interaction among users, allowing them to share experiences, tips, and advice related to fish care and

aquarium maintenance.

- **Integration of Advanced Technologies:** Aqua Hub will explore the use of augmented reality (AR) to allow users to visualize how fish and aquariums would look in their own space. Additionally, machine learning algorithms may be utilized to improve inventory management and enhance personalized user experiences.
- **Partnerships with Suppliers and Logistics Providers:** Establishing partnerships with reputable suppliers and logistics services will help ensure a reliable supply chain, improve delivery efficiency, and enhance the overall customer experience.
- **Sustainability Initiatives:** Aqua Hub will consider integrating sustainability practices, such as promoting eco-friendly products and responsible fish sourcing, to align with growing consumer preferences for environmentally conscious shopping.
- **International Expansion:** As Aqua Hub gains traction in its initial market, plans for international expansion may be pursued, adapting the platform to accommodate different regions, languages, and product preferences.

By focusing on these future developments, Aqua Hub aims to strengthen its position as a leading e-commerce platform for aquatic products, continually enhancing its value to users while fostering a vibrant community around the love of fishkeeping.

CHAPTER 8

BIBLIOGRAPHY

BIBLIOGRAPHY

- Django Documentation: Django Software Foundation. (n.d.). Django Documentation. Retrieved from <https://docs.djangoproject.com/>
- MySQL Documentation: Oracle Corporation. (n.d.). MySQL Documentation. Retrieved from <https://dev.mysql.com/doc/>
- E-commerce Trends: Statista. (2024). E-commerce Worldwide - Statistics & Facts. Retrieved from <https://www.statista.com/topics/871/online-shopping/>
- Digital Marketing Strategies: Chaffey, D. (2023). Digital Marketing: Strategy, Implementation, and Practice. Pearson Education.
- Web Development Best Practices: W3C. (2024). Web Development Best Practices. Retrieved from <https://www.w3.org/standards/webdesign/>
- E-commerce Security: Sweeney, D. (2022). E-commerce Security: Protecting Customer Data and Privacy. Security Journal, 35(2), 123-136.
- Community Engagement in Online Marketplaces: Redd, T. (2023). Building Community in Online Marketplaces: Strategies and Best Practices. Journal of Business Research, 145, 482-490.

WEBSITES

- <https://getbootstrap.com/>
- <https://fontawesome.com/>
- <https://www.w3schools.com/>
- <https://getcssscan.com/>

CHAPTER 9

APPENDIX

9.1 Sample Code

ML Implementation

```
# views.py
from django.shortcuts import render
import joblib
import pandas as pd
from sklearn.neighbors import NearestNeighbors
# Load the pre-trained model
MODEL_PATH = "ml_models/fish_recommendation.pkl"
model = joblib.load(MODEL_PATH)
df = pd.read_csv("fish_dataset.csv")
print("CSV file loaded successfully!")
print(df.head())
# Update this path as needed
def recommend_fish(min_ph, max_ph, min_temp, max_temp, hardness):
    input_data = pd.DataFrame([[min_ph, max_ph, min_temp, max_temp, hardness]],
                               columns=['Min pH', 'Max pH', 'Min Temp', 'Max Temp', 'Hardness'])
    # Check if input_data is correctly formatted
    print(f"Input data for prediction: {input_data}")
    # Use the model to predict the closest fish species
    distances, indices = model.kneighbors(input_data)
    # Check if the model is returning valid indices and distances
    print(f"Distances: {distances}")
    print(f"Indices: {indices}")
    # Get the recommended fish based on the closest matches
    recommended_fish = df.iloc[indices[0]] # Get the rows corresponding to the indices
    # Rename columns to remove spaces and make them template-friendly
    recommended_fish = recommended_fish.rename(columns={
        'Fish Name': 'fish_name',
        'Min pH': 'min_ph',
        'Max pH': 'max_ph',
        'Min Temp': 'min_temp',
        'Max Temp': 'max_temp',
        'Hardness': 'hardness',
```

```
'Description': 'description'  
})  
  
# Debugging: Print the recommended fish after renaming columns  
print(f'Recommended fish after renaming columns: {recommended_fish}')  
  
return recommended_fish
```

Template

```
{% extends 'newindex.html' % }  
  
{% block body % }  
<div class="container mt-5">  
    <h2 class="text-center">Smart Water Quality Analyzer</h2>  
  
    <!-- Explanation Section -->  
    <div class="alert alert-info mt-4">  
        <h4>How It Works:</h4>  
        <p>  
            This tool helps you analyze your aquarium's water quality to find out which fish would  
            thrive in it.  
        <br>  
            Simply input the pH level and temperature of your tank. The system will match your water  
            parameters with the ideal fish species that can live in those conditions.  
        <br>  
    </p>  
</div>  
  
<!-- Form for User Input -->  
<div class="mt-4 p-4 shadow-lg rounded bg-light">  
    <form method="post">  
        {% csrf_token %}  
        <label>Enter pH Level (5-8):</label>  
        <input type="number" name="ph" step="0.1" min="5" max="8" id="ph" class="form-
```

```
control"

    value="{{ request.POST.ph }}" required>

<label class="mt-3">Enter Water Temperature (°C) (17-31):</label>
<input type="number" name="temperature" step="0.1" min="17" max="31" id="temp"
class="form-control"
    value="{{ request.POST.temperature }}" required>

<!-- Hidden Water Hardness Input (disabled but still passed to the backend) -->
<input type="hidden" name="hardness" value="medium">

<button type="submit" id="analyze" class="btn btn-success mt-3">Analyze</button>
</form>
</div>

<!-- Display Errors if Any -->
{%
    if error %}
    <div class="alert alert-danger mt-3">
        {{ error }}
    </div>
{%
    endif %}

<!-- Display Recommended Fish -->
{%
    if fish_list %}
    <h3 class="mt-5">Recommended Fish</h3>
    <div class="table-responsive mt-3">
        <table class="table table-bordered table-striped table-hover">
            <thead class="table-success">
                <tr>
                    <th>Fish Name</th>
                    <th>Min pH</th>
                    <th>Max pH</th>
                    <th>Min Temp (°C)</th>
                    <th>Max Temp (°C)</th>
            </thead>
```

```

<th>Water Hardness</th>
<th>Description</th>
</tr>
</thead>
<tbody>
{ % for fish in fish_list %}
<tr>
<td>{{ fish.fish_name }}</td>
<td>{{ fish.min_ph }}</td>
<td>{{ fish.max_ph }}</td>
<td>{{ fish.min_temp }}</td>
<td>{{ fish.max_temp }}</td>
<td class="text-capitalize">{{ fish.hardness }}</td>
<td>{{ fish.description }}</td>
</tr>
{ % endfor %}
</tbody>
</table>
</div>
{ % else %
<div class="alert alert-danger mt-3">
    No matching fish found. Try adjusting your water parameters.
</div>
{ % endif %}
</div>
{ % endblock %



### Fish Compatibility



def check_compatibility(request):



fish_list = fish_data["Fish Name"].tolist()



result = None



fish1_details = None



fish2_details = None



compatibility_reason = ""



compatibility_score = 0 # Default score


```

```
if request.method == "POST":  
    fish1 = request.POST.get("fish1")  
    fish2 = request.POST.get("fish2")  
  
    if fish1 and fish2:  
        fish1_data = fish_data[fish_data["Fish Name"] == fish1]  
        fish2_data = fish_data[fish_data["Fish Name"] == fish2]  
  
        if not fish1_data.empty and not fish2_data.empty:  
            fish1_details = fish1_data.iloc[0].to_dict()  
            fish2_details = fish2_data.iloc[0].to_dict()  
  
            # Extract key attributes  
            fish1_temp = fish1_details["Temperament"]  
            fish2_temp = fish2_details["Temperament"]  
            fish1_water = fish1_details["Water Type"]  
            fish2_water = fish2_details["Water Type"]  
            fish1_score = fish1_details["Compatibility Score"]  
            fish2_score = fish2_details["Compatibility Score"]  
  
            # Compatibility calculation  
            compatibility_score = (fish1_score + fish2_score) / 2  
  
            # Compatibility Logic  
            if fish1_temp == "Aggressive" and fish2_temp == "Peaceful":  
                result = False  
                compatibility_reason = f"{fish1} is aggressive, while {fish2} is peaceful. They may  
not coexist safely."  
            elif fish1_temp == "Peaceful" and fish2_temp == "Aggressive":  
                result = False  
                compatibility_reason = f"{fish2} is aggressive, while {fish1} is peaceful. They may  
not coexist safely."  
            elif fish1_water != fish2_water:
```

```

        result = False
        compatibility_reason = f"{fish1} lives in {fish1_water}, but {fish2} requires
{fish2_water}. They need different water conditions."
    elif compatibility_score > 0:
        result = True
        compatibility_reason = "These fishes have a good compatibility score and similar
water conditions."
    else:
        result = False
        compatibility_reason = "These fishes have different needs or aggressive behavior,
making them incompatible."

return render(request, "check_compatibility.html", {
    "fish_list": fish_list,
    "result": result,
    "fish1_details": fish1_details,
    "fish2_details": fish2_details,
    "compatibility_reason": compatibility_reason,
    "compatibility_score": round(compatibility_score, 2), # Send score to template
Template
{% extends 'newindex.html' %}

{% block body %}
<div class="container mt-4">
    <h2 class="text-center">🐠 Fish Compatibility Checker 🐠</h2>

    <form method="POST" class="mb-4">
        {% csrf_token %}
        <div class="row">
            <div class="col-md-5">
                <label for="fish1">Select First Fish:</label>
                <select name="fish1" id="fish1" class="form-control" required>
                    <option value="">-- Select Fish --</option>
                    {% for fish in fish_list %}

```

```

        <option value="{{ fish }}>{{ fish }}</option>
    {% endfor %}
</select>
</div>
<div class="col-md-5">
    <label for="fish2">Select Second Fish:</label>
    <select name="fish2" id="fish2" class="form-control" required>
        <option value="">-- Select Fish --</option>
    {% for fish in fish_list %}
        <option value="{{ fish }}>{{ fish }}</option>
    {% endfor %}
    </select>
</div>
<div class="col-md-2 d-flex align-items-end">
    <button type="submit" id="check" class="btn btn-primary w-100">🔍 Check</button>
</div>
</div>
</form>
{% if result is not None %}
<div class="mt-4 text-center">
    {% if result %}
        <h4 class="text-success">✓ The selected fishes are compatible!</h4>
    {% else %}
        <h4 class="text-danger">✗ The selected fishes are not compatible!</h4>
    {% endif %}
    <p><strong>Reason:</strong> {{ compatibility_reason }}</p>
    <p><strong>Compatibility Score:</strong> {{ compatibility_score }}</p>
</div>

<!-- Fish Details Section (Only Temperament & Water Type) -->
<div class="table-responsive mt-4">
    <table class="table table-bordered text-center">
        <thead class="thead-dark">

```

```

<tr>
    <th>Feature</th>
    <th>{ fish1_details.Fish_Name }</th>
    <th>{ fish2_details.Fish_Name }</th>
</tr>
</thead>
<tbody>
<tr>
    <td><strong>Temperament</strong></td>
    <td>{ fish1_details.Temperament }</td>
    <td>{ fish2_details.Temperament }</td>
</tr>

</tbody>
</table>
</div>
{ % endif %}

</div>
{ % endblock %}

@csrf_exempt
def payment_handler(request):
    if request.method == 'POST':
        # Extract the payment information from Razorpay's response
        payment_id = request.POST.get('razorpay_payment_id')
        order_id = request.POST.get('razorpay_order_id')
        signature = request.POST.get('razorpay_signature')

        # Initialize the Razorpay client for payment verification
        client = razorpay.Client(auth=(settings.RAZORPAY_KEY_ID,
                                         settings.RAZORPAY_KEY_SECRET))

        try:
            # Verify the payment signature to ensure the request is valid
            client.utility.verify_payment_signature({
                'razorpay_order_id': order_id,
                'razorpay_payment_id': payment_id,
                'razorpay_signature': signature
            })

            # Fetch the order using the payment ID
            order = Order.objects.get(payment_id=order_id)

```

```
# Update the order payment status to 'Completed'  
order.payment_status = 'Completed'  
order.save()  
  
# Reduce stock for the ordered product  
product = order.product  
if product.stock >= order.quantity: # Check if stock is available  
    product.stock -= order.quantity  
    product.save()  
else:  
    # If stock is insufficient, you may want to handle this scenario  
    order.payment_status = 'Failed'  
    order.save()  
    return render(request, 'payment_failed.html', {'order': order, 'error': 'Insufficient stock to fulfill  
the order'})  
  
# Payment success, return success response  
return render(request, 'payment_success.html', {'order': order})  
  
except razorpay.errors.SignatureVerificationError as e:  
    # Handle payment verification failure  
    order = Order.objects.get(payment_id=order_id)  
    order.payment_status = 'Failed'  
    order.save()  
  
    # Log the error and show a failure page to the user  
    return render(request, 'payment_failed.html', {'order': order, 'error': 'Payment verification failed.  
Please try again.'})  
  
return HttpResponseBadRequest("Invalid request method")
```

9.2 Screen Shots

Home Page

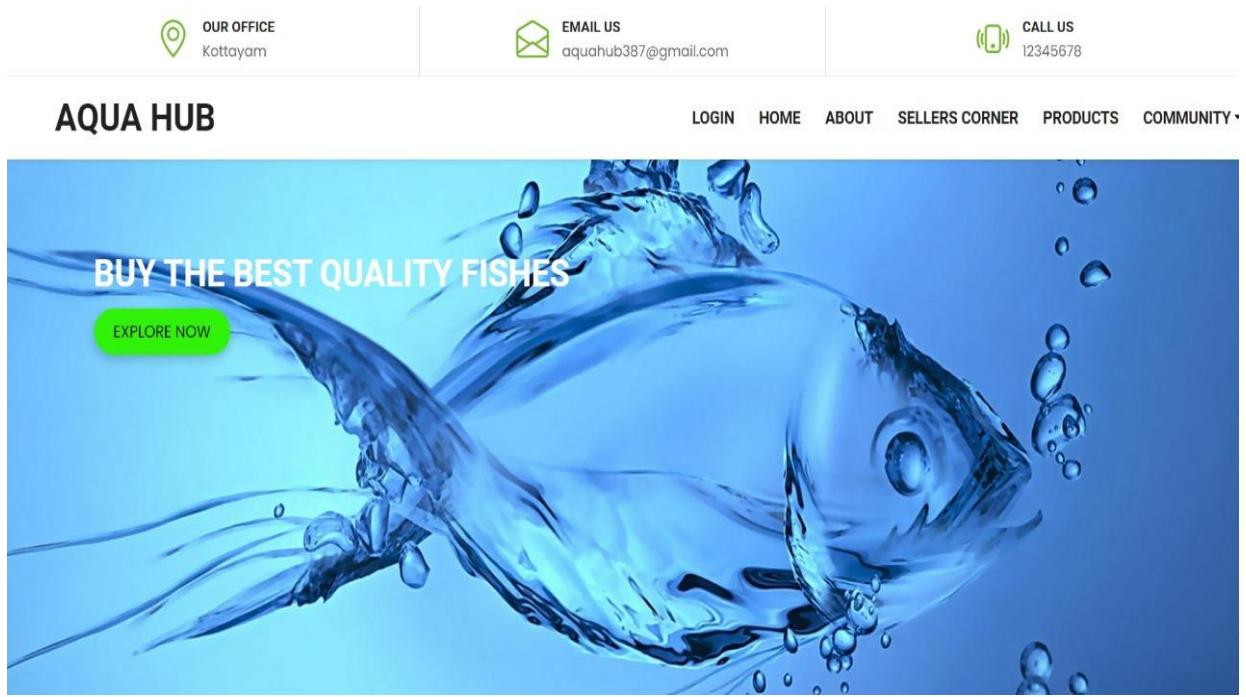


Figure 9.2 1

User Registration

User Registration	
Username	<input type="text"/>
Password	<input type="password"/>
Confirm Password	<input type="password"/>
Email Address	<input type="text"/>
Contact Number	<input type="text"/>
Address	<input type="text"/>
<input type="button" value="Register"/>	

Figure 9.2 2

Seller Dash

 AQUA HUB - SELLER

MY PRODUCTS PROFILE USER COMPLAINTS ORDERS LOGOUT

Products		Search Products...		SEARCH	ADD PRODUCT	
Product Name	Price	Stock	Description	Image	Actions	
Gold Fish	₹30.00	320	Imported red tail goldfish		EDIT	DISABLE
Watermelon Pleco	₹120.00	12	SouthAfrican Water melon Pleco		EDIT	DISABLE

User Dash

 OUR OFFICE
Kottayam

 EMAIL US
aquahub387@gmail.com

 CALL US
8590195869

 AQUA HUB

HOME BLOG ORDERS COMPLAINT CART FEATURE COMPATIBILITY DON ▾

FISH	FISH FOOD	AQUATIC PLANTS
 Gold Fish ₹30.00 Japanese Red Goldfish View Details	 Watermelon Pleco ₹120.00 Yellow Pleco View Details	 Glassfish ₹12.00 Indian GlassFish View Details

Figure 9.2 4

Admin Dash

Username	Email	Status	Action
jeswin	jeswingeorgegr@gmail.com	Active	BLOCK
aquahub837	aquahub837@gmail.com	Active	BLOCK
ko	ko@gmail.com	Active	BLOCK

Figure 9.2 5

User Blog Page

BLOG

Search Blogs by Title...

Common GoldFish Diseases

Figure 9.2 6

Delivery Registration

Figure 9.2 7

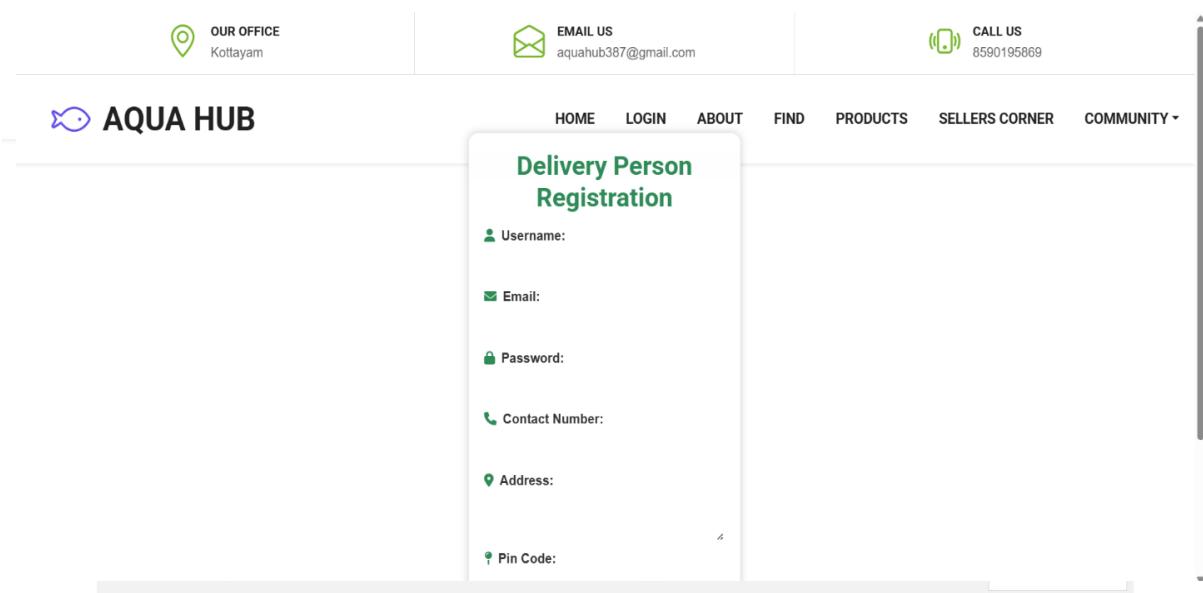


Figure 9.2 8

Delivery Dash

User wishlist

Figure 9.2 9

AQUA HUB

HOME BLOG ORDERS COMPLAINT CART FEATURE COMPATIBILITY DON ▾

Your Wishlist

	₹50.00 Azolla	VIEW DETAILS	ADD TO CART	REMOVE
	₹40.00 Thin, lightweight flakes designed for a variety of fish species, ...	VIEW DETAILS	ADD TO CART	REMOVE
	₹40.00 Tropical water Duck Weed	VIEW DETAILS	ADD TO CART	REMOVE

Glassfish
Price: ₹12.00

- 1 +

Figure 9.2 10

User Cart

Admin Delivery Dash

AQUA HUB-ADMIN

USERS SELLERS ORDERS DELIVERY COMPLAINTS LOGOUT

Pending Delivery Person Approvals

Name	Email	Contact Number	Pin Code	Vehicle Number	Actions
sarun	sarun@gmail.com	7467657654		None	Approve REJECT
anoop	anoop@gmail.com	6854245126	685511	KL676445	Approve REJECT

Approved Delivery Persons

Name	Email	Contact Number	Pin Code	Vehicle Number
huh	Jaq@gmail.com	8413267189		None
mathews	miloxo2160@andinews.com	6425123756		None
Rohit	Rohit@gmail.com	8590195869	685511	None
purith	Rohit69@gmail.com	8590195869		None
nirmal	nirma@gmail.com	9876567898		None

Figure 9.2 11

9.3 Git Log

commit 6356ec9758e666ab23afe36ebb067b8c94eca72a
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Mon Mar 17 11:13:40 2025 +0530

navbar

commit f2e6cb4647c9f5d7e5711c266ade38cc7c945bba
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Mon Mar 17 10:57:25 2025 +0530

new additions

commit 554758fc0f8aca9c6a14e02c306edadef1b16bf
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Feb 13 13:52:21 2025 +0530

new model

commit 1c3cdf2ad8e1f14bf5fbaeb94a6875644a49b
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Tue Feb 11 09:56:52 2025 +0530

wishlist com

commit abf2e132e2a6f1aa052cb6acb1d7fa11e889eedf
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Wed Feb 5 09:18:30 2025 +0530

After

commit 820d5b475833aaf6ae3324f58692654ad01dcb9a
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Sat Nov 9 23:35:24 2024 +0530

v

commit cbc8e860d6dea012d4d12aaeca0eb6bf374f6eeb
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Sat Nov 9 23:30:42 2024 +0530

redeploy2

commit 1cae05a5ffd55d2f6231b567548bec2321c2d22c
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Sat Nov 9 23:26:54 2024 +0530

redeploy

commit c7f7c8fcfcdfc0cd27717abf53709ab2821761c
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Wed Nov 6 23:17:46 2024 +0530

new deploy1

commit e030e6dfe4401ccb25a5d643001fdb95ff493dc9
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Wed Nov 6 22:47:54 2024 +0530

updates

commit 5e279aea5c394bf23beb1baf4769daf1f682247b
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Wed Nov 6 22:22:11 2024 +0530

new

commit 310151bbe48fcacf2443dd197ad7cf28b6afe076e
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Wed Nov 6 20:22:37 2024 +0530

deploy3

commit 1e6965c88ea2eb976b8f98c3142f512c681a9725
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Wed Nov 6 19:35:15 2024 +0530

aa

commit c631d06369e8df50974220b7470f43f2e3e60620
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Wed Nov 6 19:33:59 2024 +0530

h

commit 8a71df636ae49730c019f761ee772681e035a49b
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Wed Nov 6 19:29:15 2024 +0530

new depoly

commit 59740418c667a880df6a32ab4fef1c30ae902f9c
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Tue Nov 5 14:27:47 2024 +0530

new host

commit 0aa936c6c3f6726bf84d43ab207dbd1ac50280bc
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Fri Nov 1 08:40:47 2024 +0530

a

commit 04bf076bd69231132b44bb9f72bf34950a7c1aa2
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Oct 31 23:47:51 2024 +0530

/

commit 105c2238a28052afb35496da65d3800629e1c3a3
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Oct 31 23:32:41 2024 +0530

n

commit 1c5f35f654011eb0AAF4d6b93749fdebe58ce7a6
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Oct 31 23:26:06 2024 +0530

hosting again

commit 649f7d5d720540e2216b070e825fdb90554bef8c
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Oct 31 23:21:12 2024 +0530

hosting

commit b7841555047bb3b57393df64a28f8b0ed9c16369
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Oct 31 23:11:12 2024 +0530

static chane 1

commit 0321288b60fb7b817786a0682d20dc4b8918a5c1
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Oct 31 23:01:55 2024 +0530

static

commit 6c35769bf36e92995068d99229b6b6f59422f4fc
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Oct 31 22:54:02 2024 +0530

host3

commit 4f22286d959431df8bee20e75a0c4c034785899e
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Thu Oct 31 22:34:00 2024 +0530

host2

commit 1994e975ed8252cc6eb370fe61362736d7091dc8

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Thu Oct 31 22:29:47 2024 +0530

hostt

commit c4013f45426f96cb9d33ff4ab7f17838b12610b6

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Thu Oct 31 22:15:35 2024 +0530

payment complete

commit 551ddb161513bfa02b971ce5b5ac4eff2e3e09bd

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Mon Oct 21 15:36:03 2024 +0530

additions payment_in progress

commit 0185b4790084647eb34e0301375ca5b548b47b79

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Thu Oct 17 19:38:06 2024 +0530

booking_progressing

commit 5ac795cba48f28f5d5357546fec1da26d7872936

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Mon Oct 14 11:30:15 2024 +0530

Additions-search,profile validtaions

commit e16df6dcac04db245c7b7a28afe93c940873edde

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Mon Oct 14 07:41:21 2024 +0530

additions 4

commit daea6802f6a35af82a7e8cf0e81a00110f5846d

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Thu Oct 10 10:24:36 2024 +0530

adiitons 1

commit 5ce40709a68d782b4d9ef833ae8761f64e4bec26

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Mon Oct 7 16:54:58 2024 +0530

product listing

commit 6bba97a98faf19475bdfd74638d2b78ec3c6063c

Author: jeswinn <jewsingeorgegr@gmail.com>

Date: Wed Sep 25 00:12:53 2024 +0530

changes

commit 860ad8acd69cc901b3f12507bb2198abf84f9688
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Tue Sep 24 23:41:22 2024 +0530

validations

commit 96d670c8fc703677c06f7bf8211b3d706069221b
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Tue Sep 24 21:14:09 2024 +0530

forgot password

commit ba03b09854ec4c7bec3cb0648528882bf481583f
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Tue Sep 24 19:01:57 2024 +0530

google login

commit 4068cf662a724e327cf4efb3a15ec2eb65abe71
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Tue Sep 24 14:48:14 2024 +0530

productseller_view

commit 2b62d122bd77b103f84684ac4fb2f93a5b422318
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Tue Sep 24 11:39:19 2024 +0530

addproduct

commit e034d06df7de46434865ae65aee19935d027e196
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Tue Sep 24 08:10:50 2024 +0530

ps

commit 5a1495deaf07be1bd5c0ca088a9218073b20156e
Author: jeswinn <jewsingeorgegr@gmail.com>
Date: Mon Sep 23 15:35:20 2024 +0530

2

commit eb00a262658d8439e2eebb5068935da76362d3b
Author: Aqua-hub <jewsingeorgegr@gmail.com>
Date: Mon Sep 23 15:26:52 2024 +0530

commi

