Track A Shop: Web App

Submitted by	**Muhammad Zeeshan Yousaf** **Fatima Yousaf**
Roll. No	**0259-BSCS-19** **0279-R-BSCS-19**
Session	**2019 - 2023**
Supervised by	**Sir Yahya Khurram** **Assistant Professor**

BS(HONS)
IN
COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE
GC UNIVERSITY LAHORE

Track A Shop: Web App

Submitted to GC University Lahore in partial fulfillment of the requirements for the award of degree of

BS(HONS)
IN
COMPUTER SCIENCE

Submitted by	**Muhammad Zeeshan Yousaf** **Fatima Yousaf**
Roll. No	**0259-BSCS-19** **0279-R-BSCS-19**
Session	**2019 - 2023**
Supervised by	**Sir Yahya Khurram** **Assistant Professor**

DEPARTMENT OF COMPUTER SCIENCE

GC UNIVERSITY LAHORE

Declaration

We, Muhammad Zeeshan Yousaf and Fatima Yousaf students of **BS (Hons)** in the subject of **Computer Science**, session **2019-2023**, hereby declare that the material printed in this thesis titled, **Track A Shop: Web App** is our own work and has not been printed, published, or submitted as research work, thesis, or publication in any form in any university, research institution etc in Pakistan or abroad.

Date: _____ **Signatures of Deponent**

Research Completion Certificate

It is certified that the research work contained in this thesis titled **Track A Shop: Web App** has been carried out by **Muhammad Zeeshan Yousaf** Roll. No **0259-BSCS-19** and **Fatima Yousaf** Roll. No **0279-R-BSCS-19** under my supervision.

Sir Yahya Khurram
Assistant Professor

Date: _____

Submitted Through

Prof. Dr. Muhammad Waqas Anwar
Chairperson
Department of Computer Science
GC University Lahore

Controller of Examination
GC University Lahore

Acknowledgements

We are grateful to the Almighty Allah, Who blessed us with health, wisdom, knowledge, thoughts and opportunity to make some contribution in the form of present effort. We offer our humblest thanks from the deepest core of our heart to the **Holy Prophet Muhammad (Peace be upon him)**, the most perfect and excelled among and ever born on the surface of earth. The research work embodied in this dissertation was accomplished under the able guidance and affectionate supervision of **Sir Yahya Khurram**, GC University, Lahore. We will always remember his moral encouragement, skillful guidance, positive criticism and valuable advice throughout the course of our study. We also express our feelings of love and respect for our beloved parents that are the real asset of our life and gave us confidence.

Dedication

This project is dedicated to our family and our mentors, especially to our esteemed guide and teacher, for this project, who encouraged and supported us throughout the development and completion of "Track A Shop: Web App" - a leap in the realm of technology and innovation in computer science.

Abstract

The "Track A Shop: Web App" project aims to address a significant issue prevalent in the digital era, where small businesses such as service providers, mini stores, and repair shops struggle to establish an online presence due to cost constraints. In response to this, we propose the development of a Web Application Software that acts as a centralized platform for local businesses to connect with their local customer base. By leveraging this application, shopkeepers can extend their reach and increase their earnings by reaching more customers online. Customers, in turn, can easily access and purchase products or avail services from nearby local shops within their specified geographical proximity. The project integrates an image recognition service, allowing sellers to effortlessly add products or services by simply uploading images. On the customer side, this service facilitates image-based search, enhancing the user experience and streamlining the purchasing process.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

In an era characterized by rapid technological advancements, businesses are constantly seeking efficient, automated, and precise solutions to streamline their operations. This paradigm shift has led to the creation of numerous software programs tailored to meet the diverse needs of different industries, facilities, and various workplaces. Historically, manual methods were employed for data management, but evolving preferences have spurred a transition to digital solutions. Recognizing the digital divide prevalent among small businesses, particularly those serving local communities, this project, "Track A Shop: Web App," endeavors to empower these enterprises and enable their seamless integration into the online marketplace.

In today's digital landscape, the majority of businesses across sectors have embraced online platforms to market and sell their products and services, swiftly reaching customers through home deliveries. However, smaller businesses such as service providers, mini stores, repair shops, and local vendors, including cobblers, barbers, milk shops, bakeries, and spare parts shops, often find it financially challenging to establish an online presence. The "Track A Shop: Web App" web application sets out to address this disparity by providing local shops a centralized

online platform to connect with their local customer base. Through this application, shopkeepers can significantly broaden their customer reach, attract a larger audience, and consequently, enhance their earnings with minimal effort of input.

The project envisions simplifying and modernizing operations for local businesses by integrating innovative features such as image recognition for effortless addition of products services, and image-based search capabilities for customers.

1.2 Project Overview

The proposed system, "Track A Shop: Web App," is a web-based application designed to facilitate the seamless creation of online shops for sellers. Utilizing image recognition technology, sellers can effortlessly add products or services to their online shop by capturing or uploading images. The system automatically identifies and categorizes the items, making them readily accessible to customers in the vicinity of the shop. Conversely, customers can easily browse and search for products or services using images, names, or proximity filters. They can initiate requests for product delivery or obtain directions to the seller's physical shop, enhancing their overall shopping experience.

1.3 Problem Statement

In the contemporary business landscape, small enterprises encounter a significant barrier in establishing an online platform for their operations due to prohibitive costs associated with web development and maintenance. The high financial investment required often restricts their ability to tap into the potential of the digital marketplace, limiting their growth and outreach.

Furthermore, the traditional method of manually inputting products and services into an online platform has become increasingly burdensome and time-consuming.

This arduous process not only consumes valuable time but also poses a hindrance to the efficient management and scaling of businesses, leading to decreased productivity and suboptimal user experiences.

In addition, both customers and local shops face challenges in connecting efficiently. The conventional approach of physically visiting or relying on word-of-mouth recommendations to discover shops or services is cumbersome and time-intensive. This lack of a seamless and convenient means to connect customers with local shops impedes the potential for increased business transactions and growth for these small enterprises. Therefore, an effective solution is imperative to address these pressing issues and bridge the existing gap between small businesses and their online presence while enhancing accessibility and connectivity for both customers and shops.

1.4 Project Scope

The scope of "Track A Shop: Web App" project encompasses the development of a comprehensive web-based platform targeting small businesses and local shops to establish an online presence for their products and services. The platform will facilitate seamless integration and accessibility for both sellers and customers. It will incorporate image recognition technology, allowing sellers to effortlessly list their products and services by capturing or uploading images. Customers can easily search for and purchase products or avail services from nearby local shops within a specified geographical radius.

Key Features and Functionalities:

- Online platform for small businesses and local shops to list their products and services.

- Image recognition technology [3] for effortless product and service listing by sellers.
- Geolocation-based search and exploration for customers to find nearby shops and offerings.
- Seamless user interface ensuring a smooth user experience for both sellers and customers.
- Easy process of search by text, image or categories.
- An intuitive interface for seamless interactions between sellers and customers.

1.4.1 Study Limitations

The progress of this project was significantly affected by various challenges, primarily related to time and financial constraints. The project incurred several financial obligations, encompassing expenses for software development, server deployment [4], integration of image recognition technology [3], and implementation of geolocation features [5]. Balancing these financial demands while managing development tasks and adhering to project timelines proved to be demanding. However, despite these constraints, we remained dedicated to ensuring the accuracy and effectiveness of the web application. Our commitment was to meet the expectations and needs of both sellers and customers, striving for a successful and impactful outcome.

1.5 Literature Review

"Track A Shop: Web App" embodies the essence of modern business augmentation through technology. In an era where online presence is paramount, this web

application offers a cost-effective solution for small and local businesses to establish their digital footprint. By seamlessly integrating products and services using image recognition and geolocation features, the application enhances accessibility and connectivity between customers and local shops. It paves the way for an intuitive and efficient shopping experience, bridging the gap between traditional brick-and-mortar stores and the vast digital market. The project encapsulates the vision of empowering businesses and customers alike, promoting growth, efficiency, and a seamless convergence of technology and commerce.

Chapter 2

Requirement Specification

2.1 Functional Requirements

Function and features that derives to the end user of the system are following:

2.1.1 Affordable Online Presence:

Develop a cost-effective solution that enables small businesses to establish and maintain an online platform for their products and services without incurring prohibitive expenses, democratizing access to the digital marketplace.

2.1.2 Streamlined Product/Service Integration:

Create an automated system leveraging image recognition technology to simplify and expedite the process of adding products and services to the online platform, reducing manual effort and improving efficiency for business owners.

2.1.3 Enhanced Customer-Shop Connectivity:

Facilitate easy and efficient connection between customers and local shops by providing a user-friendly interface that allows customers to search for products or services based on images, names, or location, thereby fostering a seamless shopping experience.

2.1.4 Geolocation Features:

Implement geolocation features to enable customers to explore nearby shops and their offerings, aiding in quicker decision-making and fostering a stronger connection between customers and local businesses.

2.1.5 Delivery Request Mechanism:

Integrate a system that allows customers to request product delivery from the respective sellers, promoting convenience and encouraging increased sales for the shops while enhancing customer satisfaction.

2.1.6 Navigation Assistance:

Provide navigation features that guide customers to the physical locations of shops, improving accessibility and encouraging more foot traffic to the local businesses, ultimately boosting their visibility and sales.

2.1.7 User Experience Optimization:

Prioritize an intuitive and appealing user interface and experience to ensure that both sellers and customers find the application easy to use and navigate, enhancing overall satisfaction and encouraging continued usage.

2.1.8 Search History Recording:

Record customer searches to provide a history for their convenience and future reference.

2.1.9 Comprehensive Search Results:

Ensure that search results include a wide array of products, services, and corresponding shops for a complete view of available offerings.

2.1.10 Flexible Result Filtering:

Allow customers to filter search results based on various criteria such as price, ratings, shop proximity, or categories, enabling a personalized and efficient shopping experience.

2.1.11 Delivery Requests for Products:

Implement a mechanism for customers to request product delivery, promoting convenience and seamless transactions.

2.1.12 Navigation to Shops:

Enable customers to obtain directions to the physical shops offering the desired product or service, facilitating a smooth transition from online exploration to in-person shopping.

2.1.13 Category-Based Exploration:

Provide an option for customers to explore products and services based on categories, allowing for effortless browsing without initiating a specific search.

2.2 Non-functional Requirements

Here are the non-functional requirements:

2.2.1 Performance

- The web application must have rapid response times and load efficiently.
- API calls for data fetching and processing should complete within milliseconds for a seamless user experience.
- The application should maintain high performance even during peak usage.

2.2.2 User Interface (UI) and User Experience (UX)

- The UI should be intuitive, aesthetically pleasing, and utilize vibrant colors with smooth animations.
- The design should be consistent and compatible across various screen sizes and devices.
- User interactions and transitions within the application should be smooth and visually appealing.

2.2.3 Compatibility

- The web application must be compatible with a wide range of browsers, ensuring consistent functionality and appearance across different platforms.

2.2.4 Scalability

- The architecture should allow for seamless scaling to accommodate a growing user base and increased data load.

- The application should be capable of handling increased traffic without compromising performance.

2.2.5 Accessibility

- The application should adhere to accessibility standards to ensure usability for individuals with disabilities.
- Content and features should be easily navigable and understandable for users with basic English skills.

2.2.6 Image Storage

- Images should be stored securely in AWS S3 buckets [6], ensuring data integrity and accessibility.
- The image storage system should handle image uploads, retrievals, and deletions in an efficient and organized manner.

2.2.7 Database Management

- The database should be maintained automatically, including regular password changes for enhanced security.
- Database operations should be optimized to ensure data retrieval and modification are swift and efficient.

2.2.8 Response Time

- The web application should have minimal loading time, providing near-instantaneous responses to user actions.

2.2.9 Reliability

- The application should operate consistently without frequent crashes or downtimes, ensuring a reliable user experience by instantly report and monitor any bugs or errors [7], enhancing application reliability.

Chapter 3

Project Design

3.1 Methodology

The methodology adopted for designing the "Track A Shop: Web App" is an iterative process. The methodology encompasses various stages from project initiation to implementation, ensuring an organized and systematic approach as each stage is mentioned below:

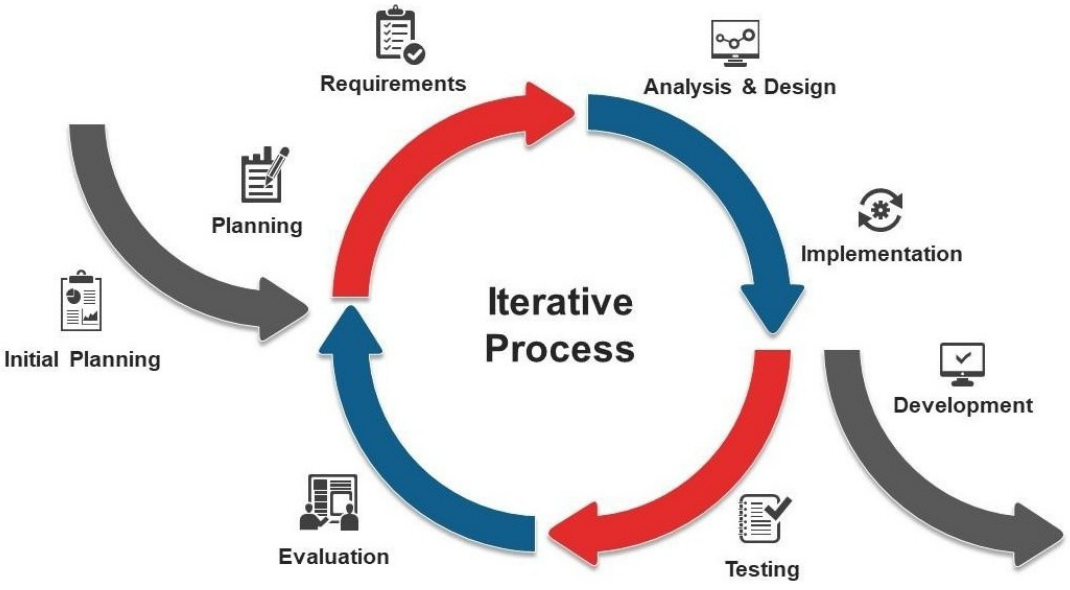

FIGURE 3.1: Iterative Process Model

3.1.1 Initial Planning

In this initial planning stage, the project goals and initial requirements are outlined. Key activities include:

Defining Project Goals: Setting the overall goals and vision for the "Track A Shop: Web App".

Preliminary Requirement Analysis: Performing an initial analysis of high-level requirements and constraints.

3.1.2 Planning

The planning stage involves planning the upcoming iteration and setting specific goals for it. Key activities include:

Setting Iteration Goals: Defining the objectives and outcomes for the current iteration.

Identifying Resources: Allocating the necessary resources, including manpower, technology, and tools [8], for the iteration.

3.1.3 Requirements Analysis and Design

In this stage, detailed requirements are gathered and analyzed, and the system's design is conceptualized. Key activities include:

Requirements Refinement: Analyzing and refining requirements based on feedback from previous iterations and stakeholders.

System Design: Conceptualizing the system's design and architecture based on the refined requirements.

3.1.4 Implementation and Development

The implementation stage involves writing code and developing the iteration. Key activities include:

Coding and Development: Writing the code based on the design and architecture defined in the previous stage.

Integration: Integrating the developed components and modules into a cohesive iteration.

3.1.5 Testing

The testing stage involves validating the functionality and performance of the developed iteration. Key activities include:

Functional Testing: Ensuring that the iteration meets the specified functional requirements.

User Acceptance Testing: Engaging users to validate the iteration against their expectations and requirements.

3.1.6 Evaluation

The evaluation stage involves assessing the outcomes of the iteration. Key activities include:

Evaluation of Goals: Evaluating whether the iteration goals were achieved and identifying areas for improvement.

3.1.7 Development Stage

The development stage of the "Track A Shop: Web App" project involves translating the project design and requirements into functional software components. This stage is characterized by a series of iterative cycles, each aimed at enhancing and expanding the platform's features.

1. **Iteration 1: Initial Prototyping**

   The first iteration focuses on creating a basic prototype of the web application, emphasizing core features such as user registration, product listing, and basic geolocation functionalities. User feedback is collected to inform further development.

2. **Iteration 2: Image Recognition Integration**

   Building upon the initial prototype, the second iteration introduces image recognition technology to simplify product and service listing. Sellers can now capture or upload images, and the system automatically populates product/service details.

3. **Iteration 3: Geolocation and User Interface**

   This iteration enhances geolocation features, allowing customers to search for nearby shops and products/services within a specified radius. Simultaneously, the user interface is refined, focusing on aesthetics and usability.

4. **Iteration 4: User Feedback and Usability**

   The fourth iteration revolves around collecting user feedback and making usability improvements. User testing and feedback surveys are conducted to identify pain points and areas for enhancement.

5. **Iteration 5: Scalability and Performance**

   With a more refined platform, the fifth iteration addresses scalability and performance optimization. Measures such as caching, load balancing, and database query enhancements are implemented.

6. **Iteration 6: Integration and Security**

   In this phase, third-party integrations, enhanced security measures, and multi-language support are introduced. Payment gateways are integrated to facilitate seamless transactions.

7. **Iteration 7: Testing and Quality Assurance**

   The final iteration focuses on comprehensive testing and quality assurance. Unit testing, integration testing, and user acceptance testing are conducted to ensure the platform's reliability and performance.

The iterative development approach ensures that each phase builds upon the previous one, allowing for continuous improvements and a more robust final product.

3.2 Architecture Overview

3.2.1 Class Diagram (UML)

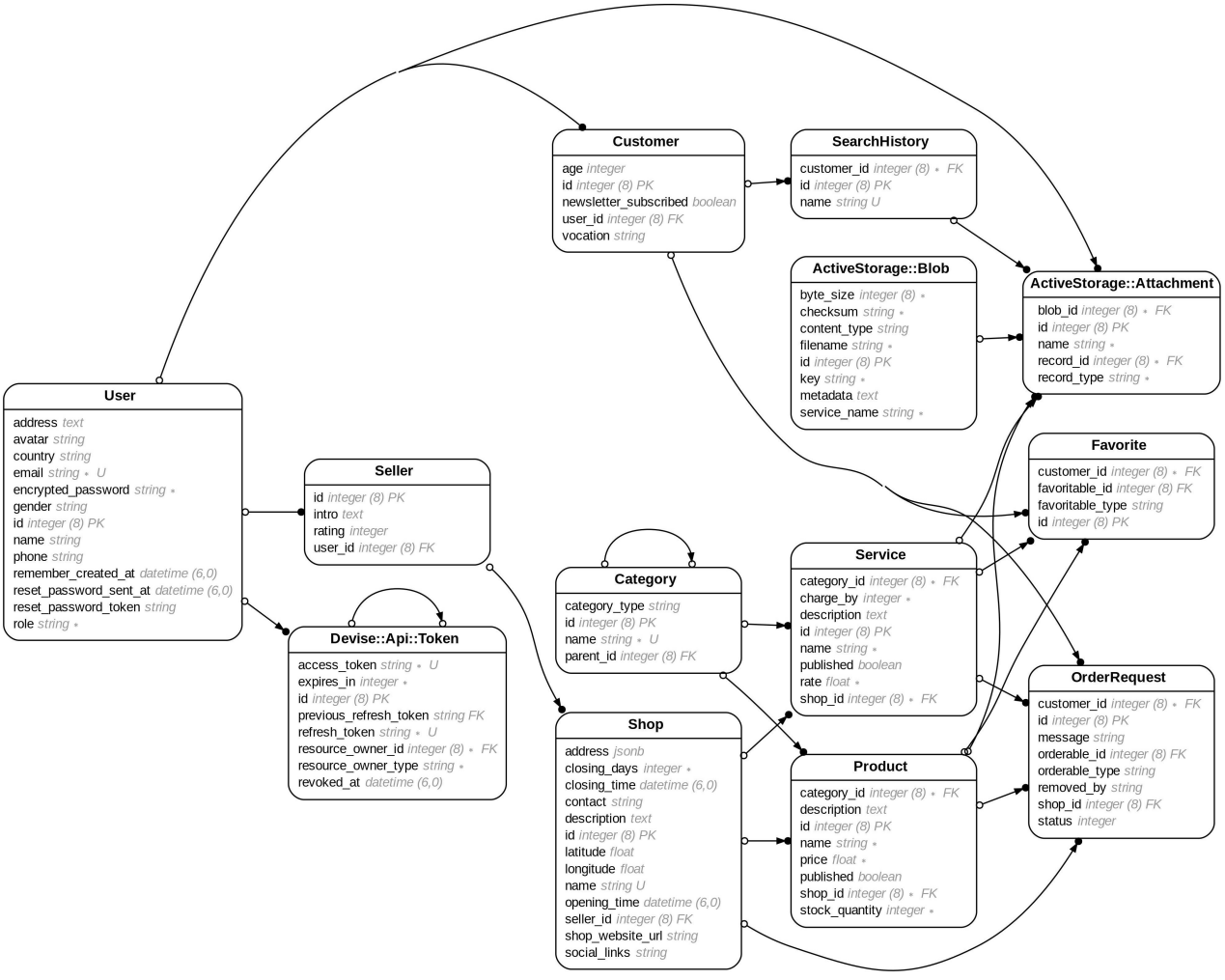

FIGURE 3.2: Track A Shop: Web App - Class Diagram [1]

3.2.2 Customer Sequence Diagram

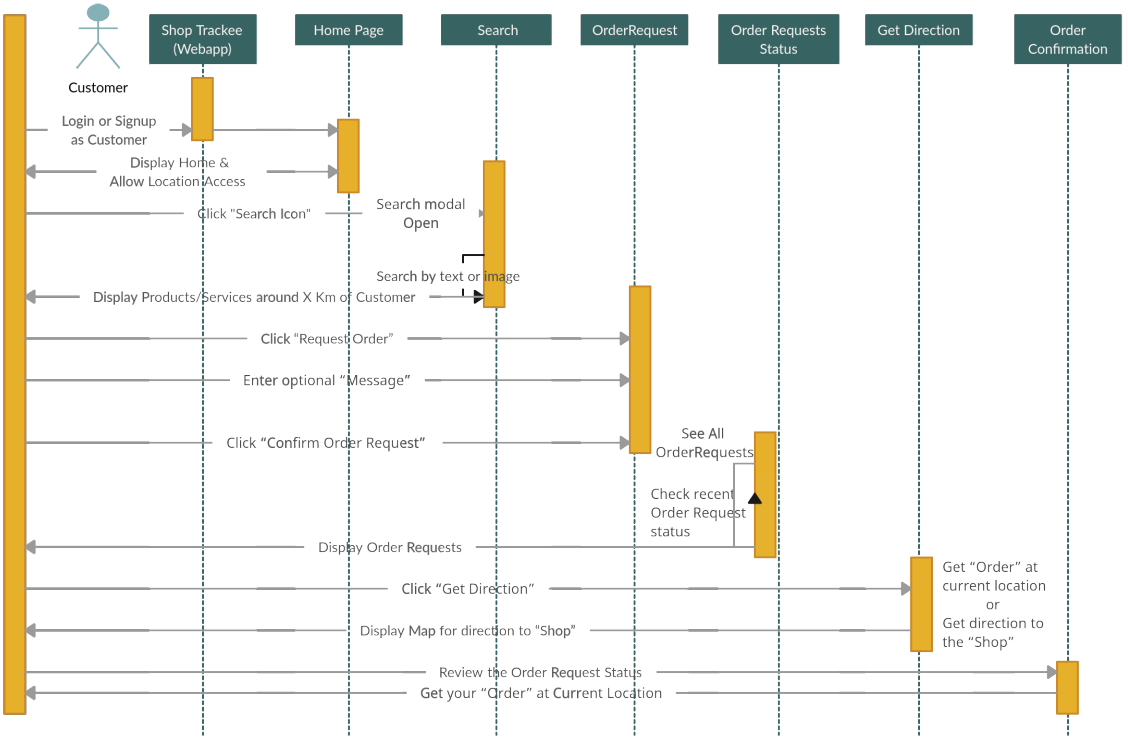

FIGURE 3.3: Track A Shop: Web App - Customer Sequence Diagram [2]

3.2.3 Seller Sequence Diagram

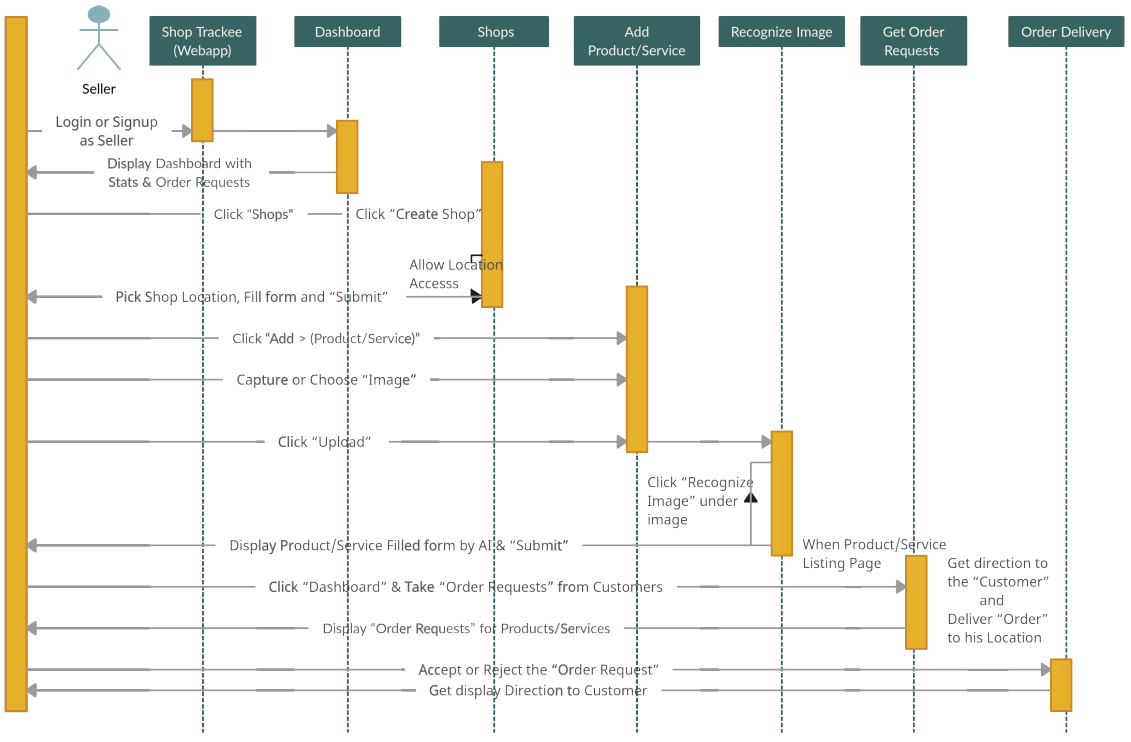

FIGURE 3.4: Track A Shop: Web App - Seller Sequence Diagram [2]

3.3 Design Description

Here is the basic architectural design of our Web Application:

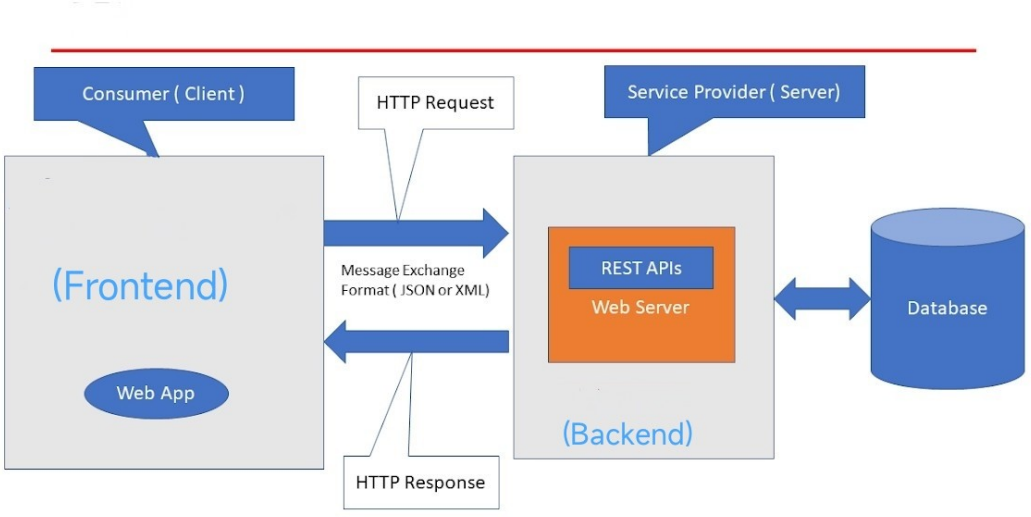

FIGURE 3.5: Track A Shop: Web App - Design Description [2]

Chapter 4

Implementation and Evaluation

4.1 Development Stages

Following are the stages of development:

4.1.1 Strategy Stage

We designed our web application while keeping in mind the requirements of our end sellers and customers. In order to make it a success we planned each and everything beforehand. We have learned our users demand and then planned our project on it. We have designed it in such a way that it can be easy to use and handle. After jotting down our requirements we made diagrams so that it can give an outlook of our system. After this we worked on our database and later on its implementation. We also wrote the project code, then we integrated and tested our it to verify if its working.

4.2 Implementation

Following is described about the implementation level things:

4.2.1 Tools and Technologies

1. Server-Side API Framework **Ruby on Rails** [9]
2. Front-End Frameworks:
    - **Next.js** (framework built on REACT Js library) [10]
    - **Bootstrap5** (CSS framework) [11]
    - **Google Material UI** (for interactive components) [12]
    - **Google Maps API** [5]
3. DBMS **PostgreSQL** [13]
4. Version Control System (VCS): **Git, Github** [14]
5. Integrated Development Environment (IDE): **RubyMine, VS Code**
6. API and Web testing tool: **Postman** [15]
7. Data Caching server: **Redis**
8. Hosting Service for API: **Heroku** [4]
9. Hosting Service for Front-End App: **Amazon Static Hosting S3, Vercel**
10. CI — CD tool: **Github Actions** [16]
11. Media Storage Service: **AWS S3** [6]
12. Image Recognition Service: **AWS Rekognition** [3]
13. Domain Name Service: **AWS Route53**
14. Domain Platform: **Namecheap.me**
15. Bug Report Service: **Sentry** [7]
16. Mailer Service: **Mailjet**
17. Documentation Writing Software: **LaTex, TeXstudio** [8]

4.3 System Integration

System integration in the "Track A Shop: Web App" project is a critical process where all the individual components and modules of the system are combined and tested to ensure they function as a unified, cohesive unit [17]. This integration involves merging the backend functionalities, including database management and server operations, with the frontend user interface to create a seamless and functional application. The integration process also incorporates third-party services, such as image recognition [3], mapping APIs [5] and Deployment on AWS/Heroku [4], ensuring their proper interaction within the application. Rigorous testing and validation are conducted to confirm that the integrated system operates smoothly, with data flowing seamlessly between various components.
Achieving a successful integration is vital for delivering a reliable, efficient, and feature-rich web application that can be run on any environment and on any web browser, meets the needs and expectations of both sellers and customers.

4.4 User Interface

The user interface (UI) of the "Track A Shop: Web App" project is crafted to deliver a seamless and engaging experience for both sellers and customers. Sellers are provided with a comprehensive dashboard offering insights into their shops, order requests, and sales statistics. They can efficiently manage their shop listings, adding new products/services and updating existing ones. On the other hand, customers are greeted with an intuitive interface that allows effortless browsing of nearby shops and their offerings, categorized for easy exploration. The search functionality is versatile, enabling customers to search by text, images, or based on their preferences. Clear product/service details, interactive maps for shop locations [5], easy order request processes, and a user-friendly profile management system further enhance the overall usability. The UI is designed to be accessible and responsive to various type of devices and user needs, ultimately fostering a positive and satisfying interaction for all users [18].

4.4.1 Login

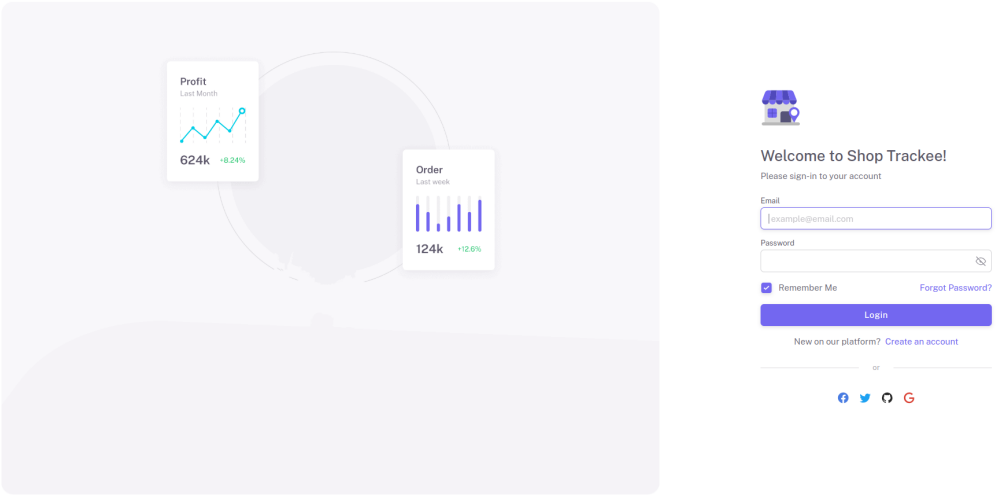

FIGURE 4.1: Track A Shop: Web App - Login

4.4.2 Register

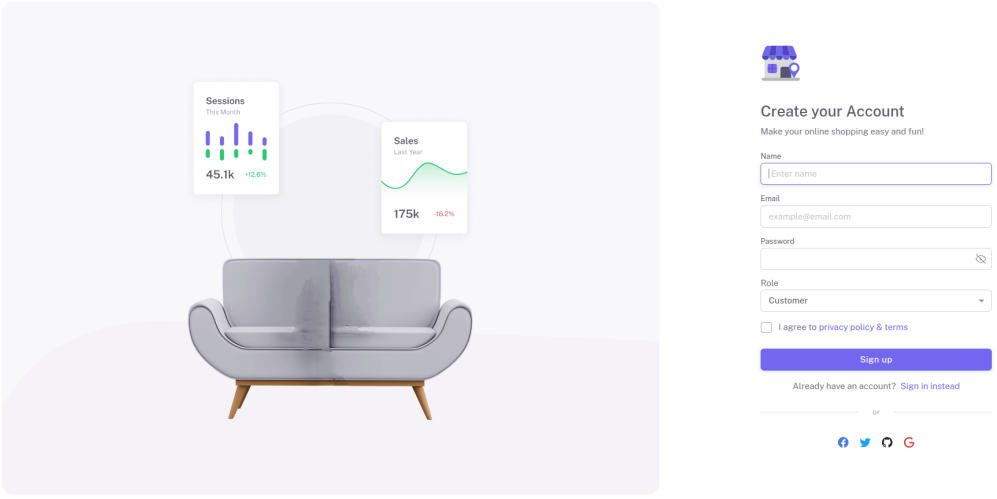

FIGURE 4.2: Track A Shop: Web App - Register

4.4.3 Forgot Password

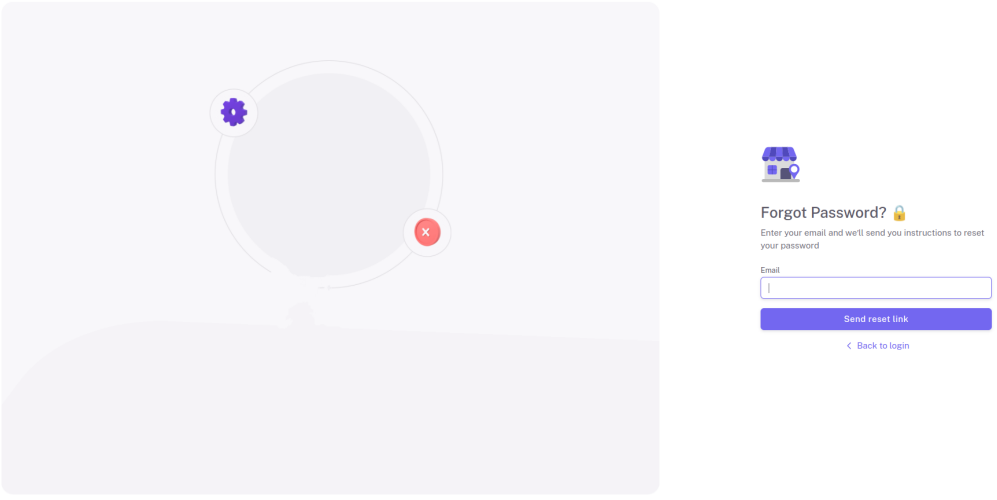

FIGURE 4.3: Track A Shop: Web App - Forgot Password

4.4.4 Profile

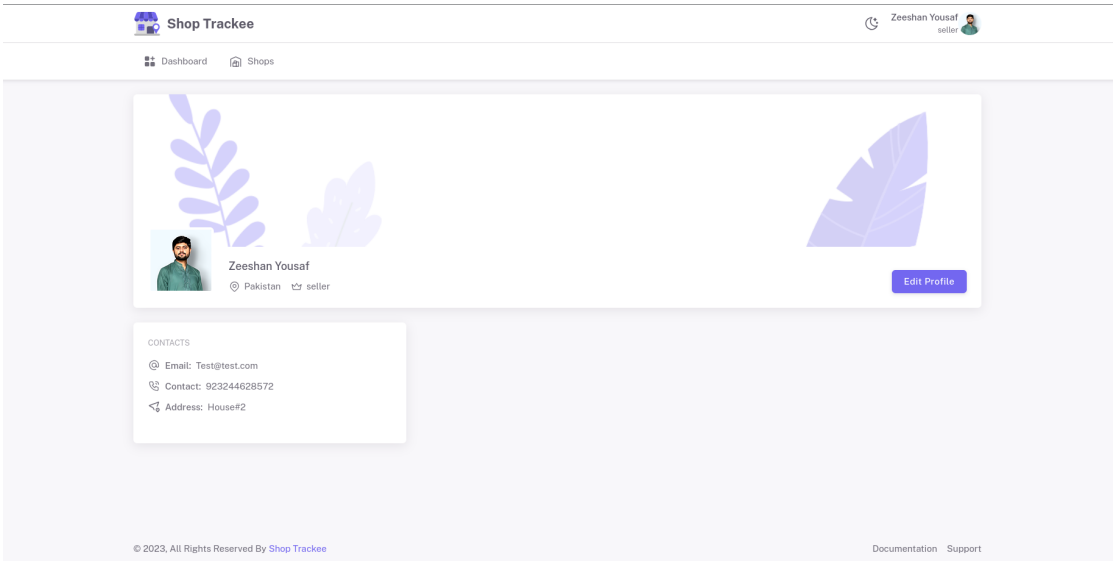

FIGURE 4.4: Track A Shop: Web App - Profile

4.4.5 Profile Edit

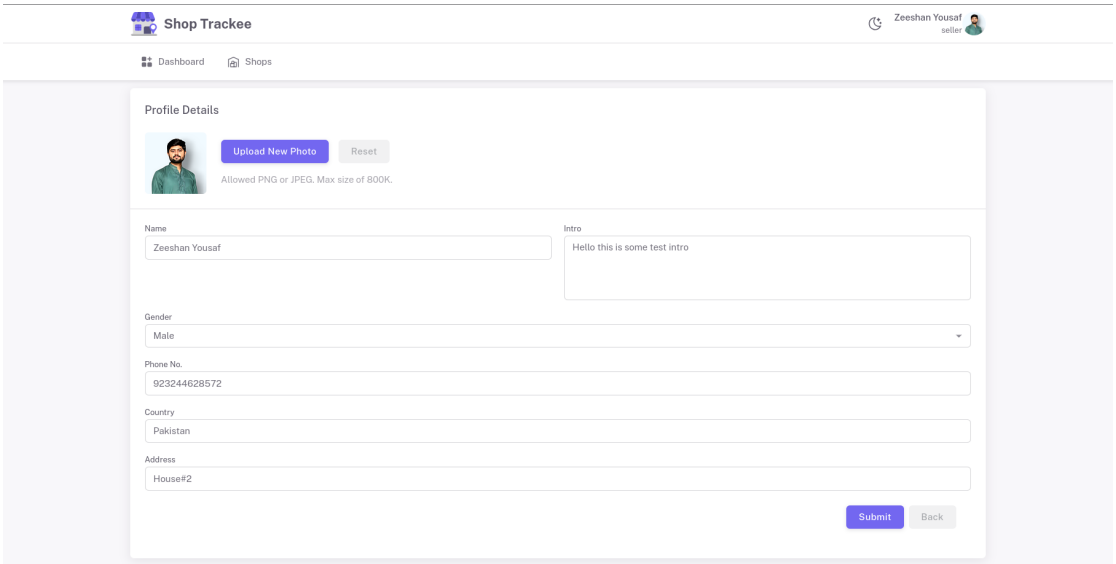

FIGURE 4.5: Track A Shop: Web App - Profile Edit

4.4.6 Seller Dashboard

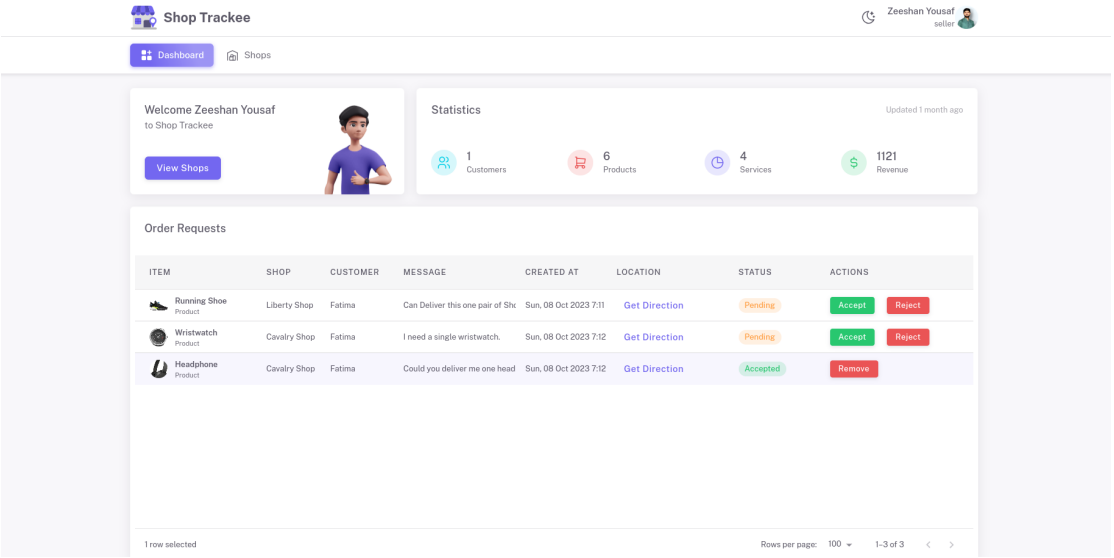

FIGURE 4.6: Track A Shop: Web App - Seller Dashboard

4.4.7 Seller Shops

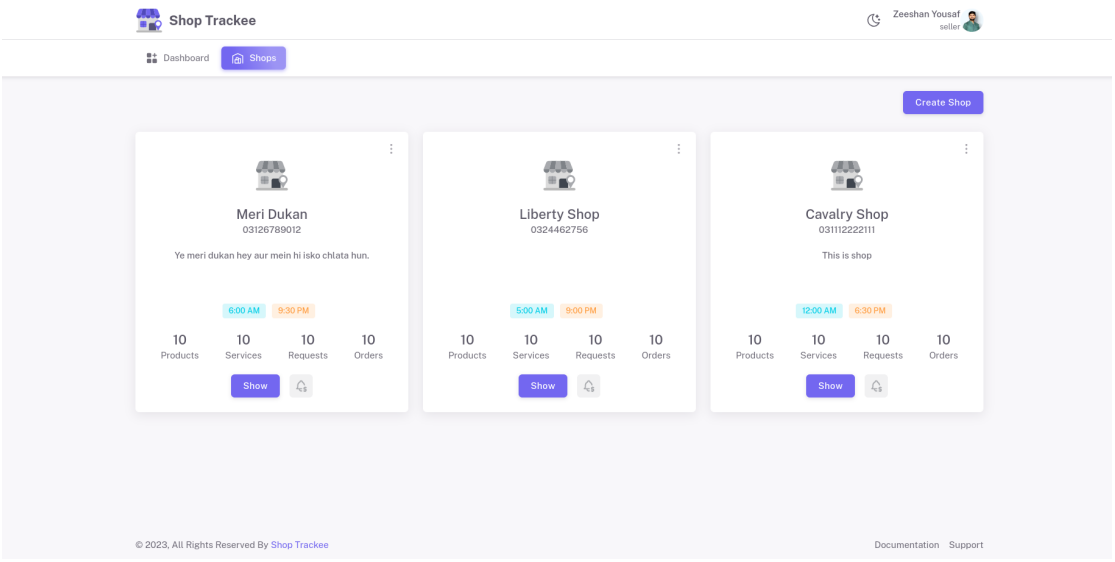

FIGURE 4.7: Track A Shop: Web App - Seller Shops

4.4.8 Seller Create Shop

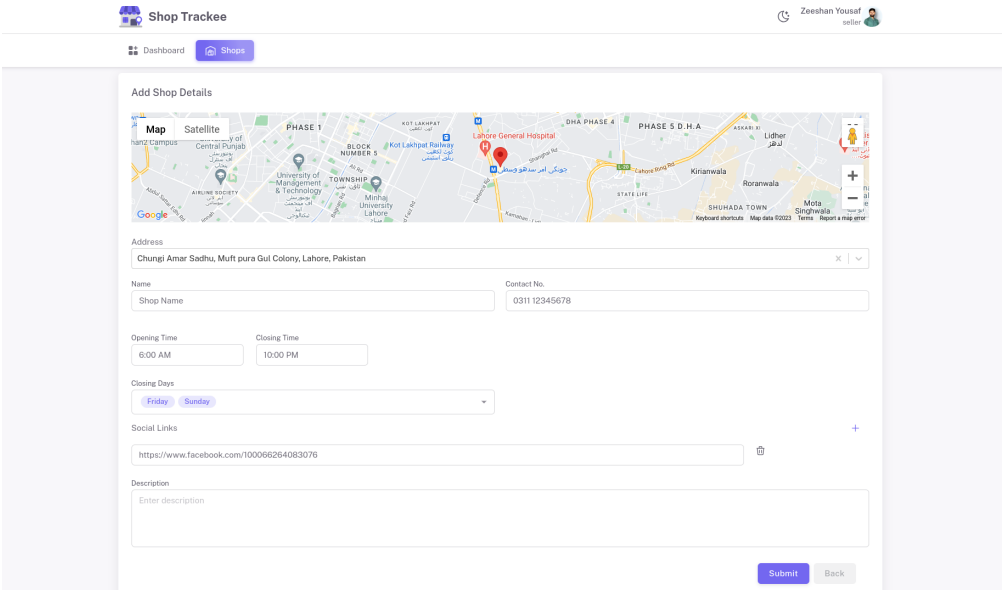

FIGURE 4.8: Track A Shop: Web App - Create Shop

4.4.9 Seller Add Product

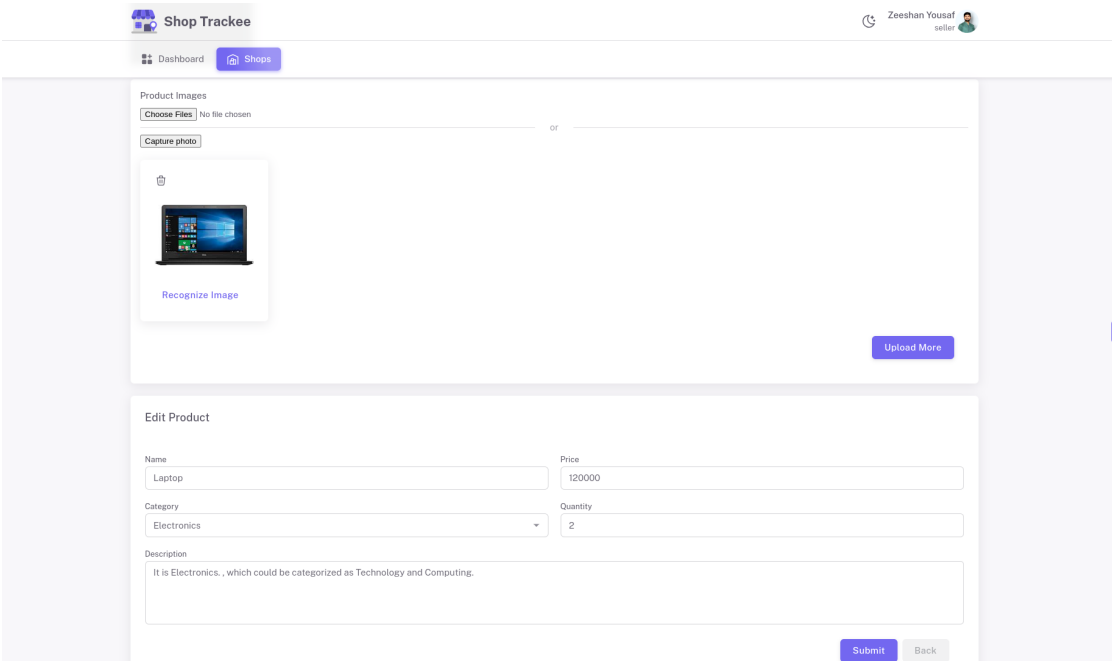

FIGURE 4.9: Track A Shop: Web App - Add Product

4.4.10 Seller Add Service

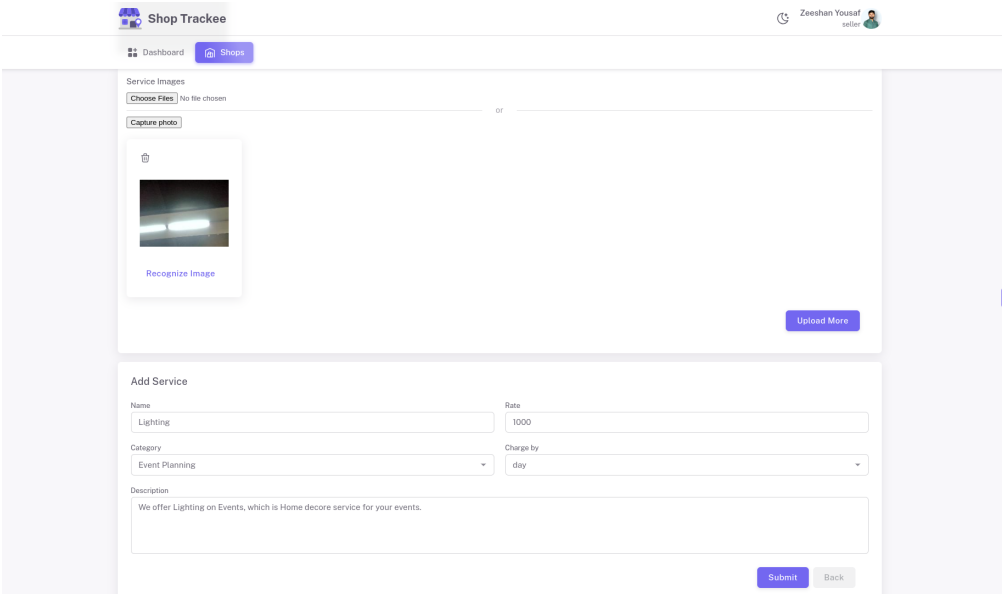

FIGURE 4.10: Track A Shop: Web App - Add Service

4.4.11 Seller Products and Services Listing

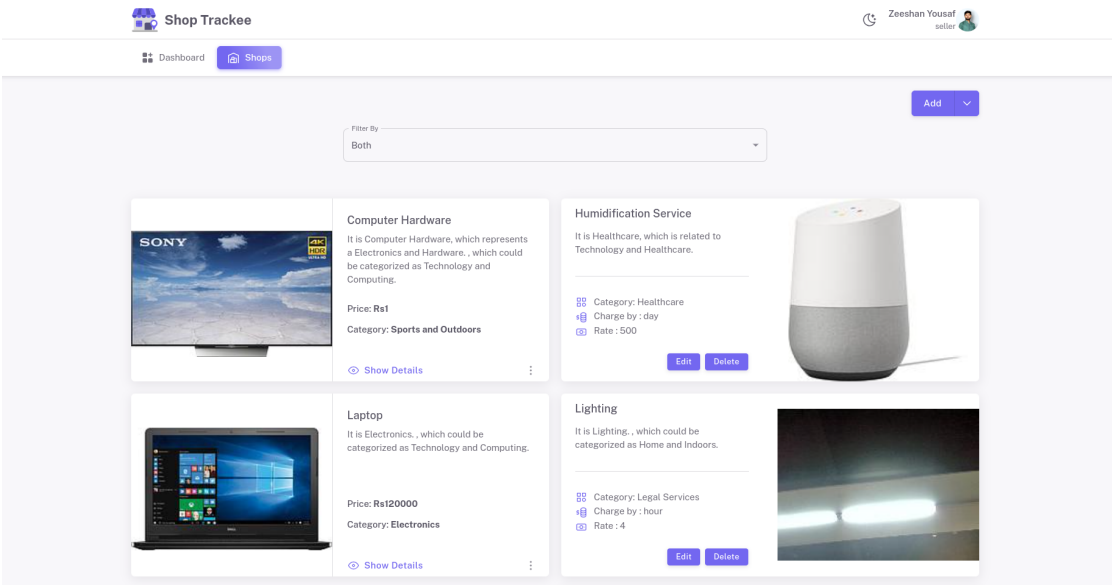

FIGURE 4.11: Track A Shop: Web App - Products and Services Listing

4.4.12 Customer Home

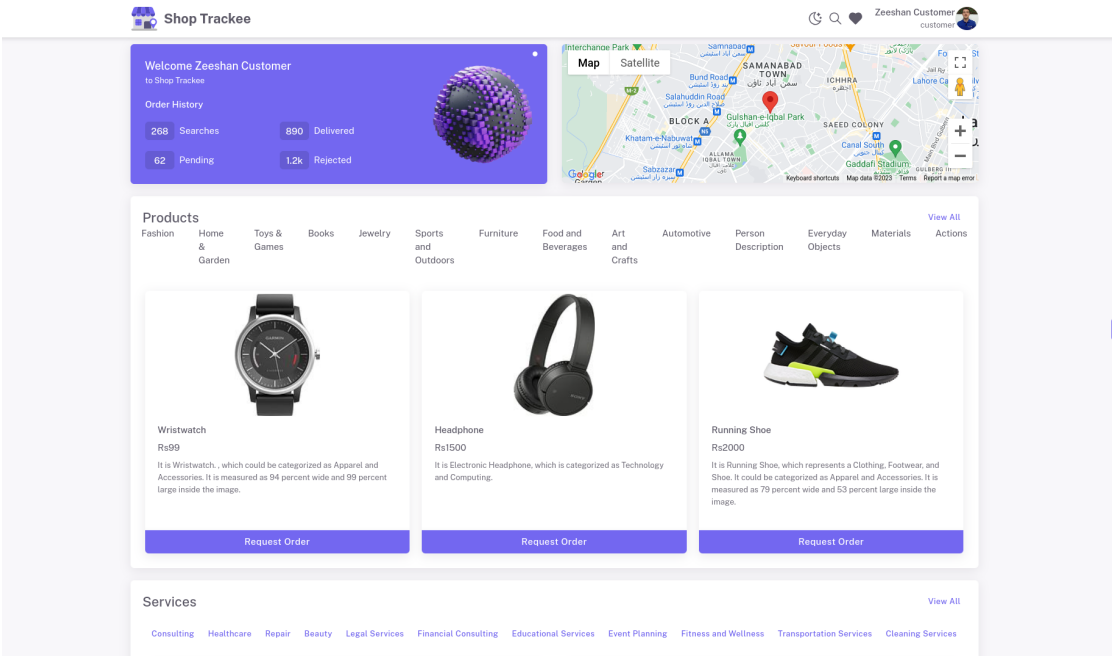

FIGURE 4.12: Track A Shop: Web App - Customer Home

4.4.13 Customer Searching Around

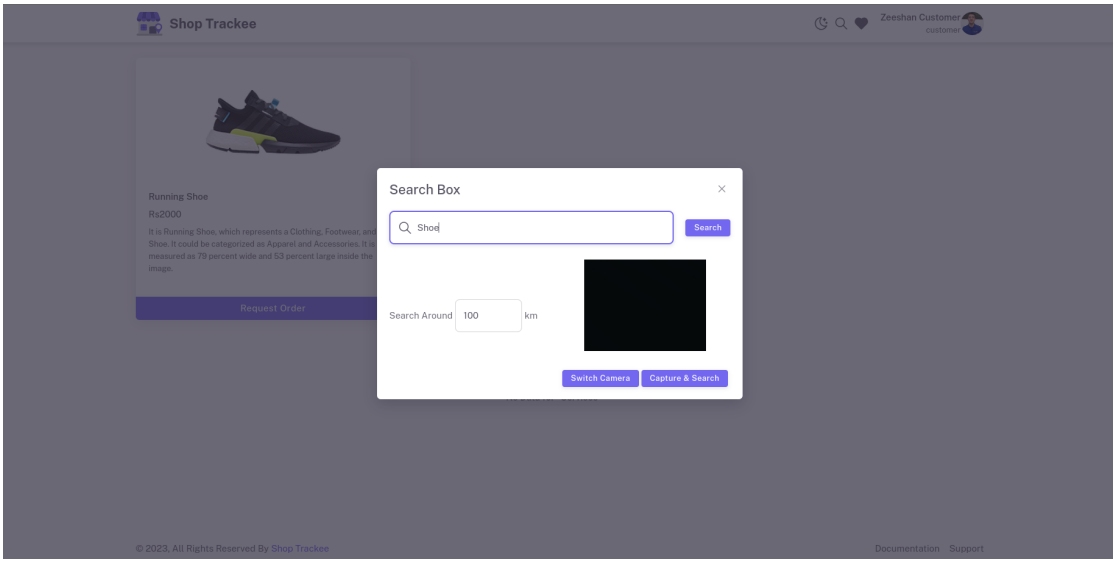

FIGURE 4.13: Track A Shop: Web App - Customer Searching Around

4.4.14 Customer Requesting Order

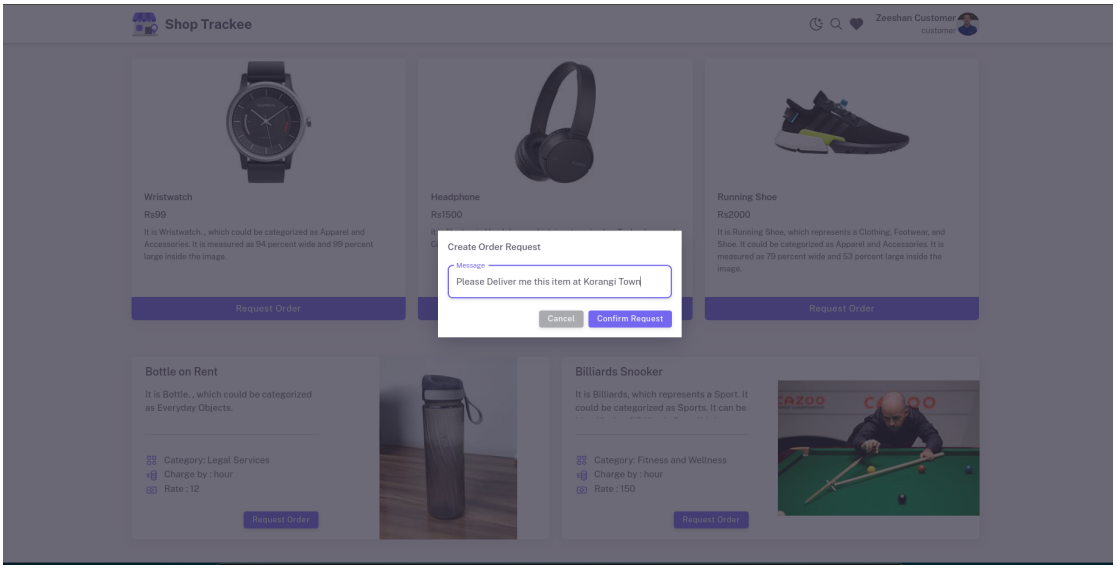

FIGURE 4.14: Track A Shop: Web App - Customer Requesting Order

4.4.15 Customer Order Requests History

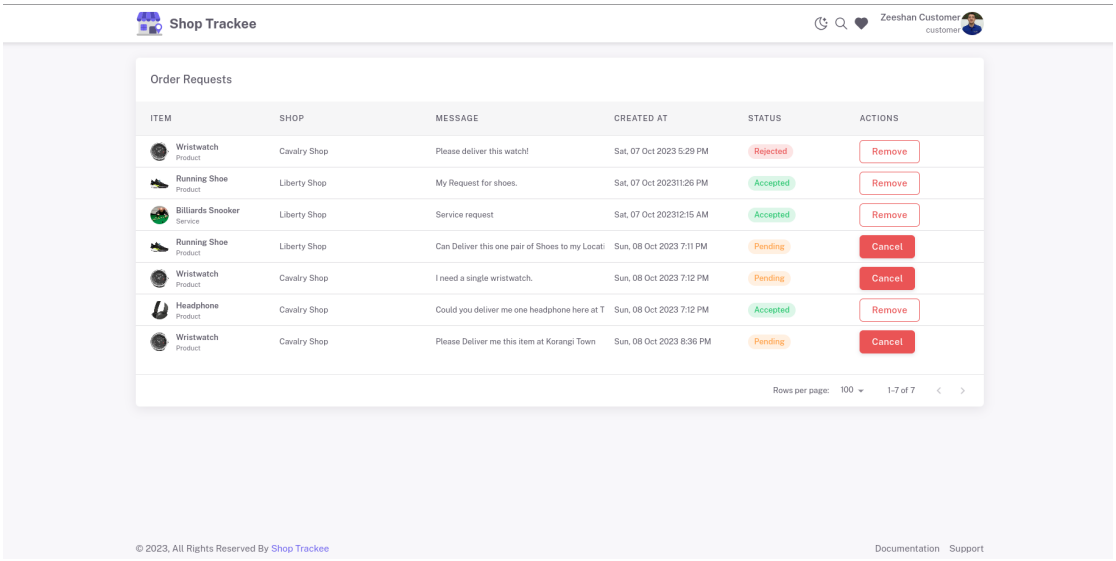

FIGURE 4.15: Track A Shop: Web App - Customer Order Requests History

4.4.16 Customer Favorites

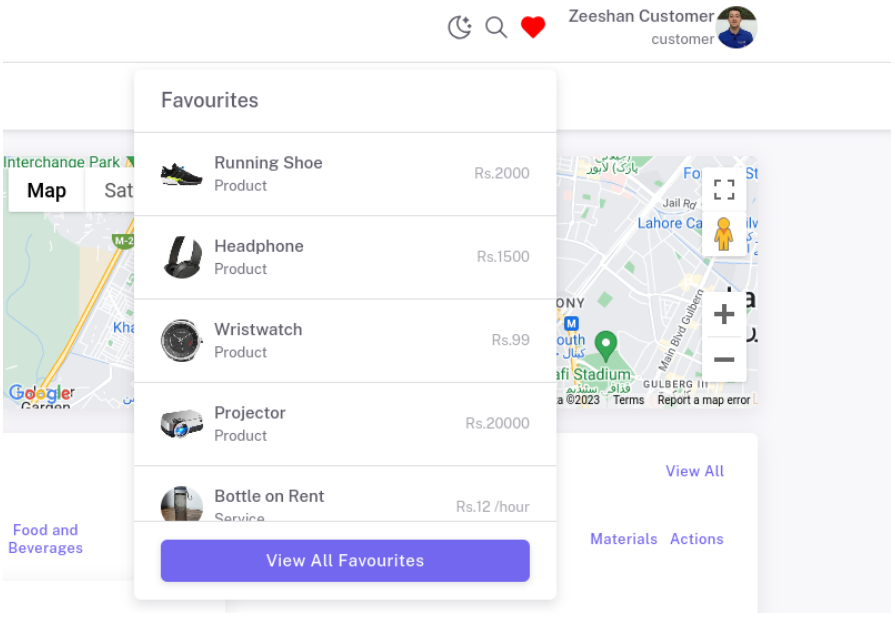

FIGURE 4.16: Track A Shop: Web App - Customer Favorites

4.4.17 Customer All Favorites

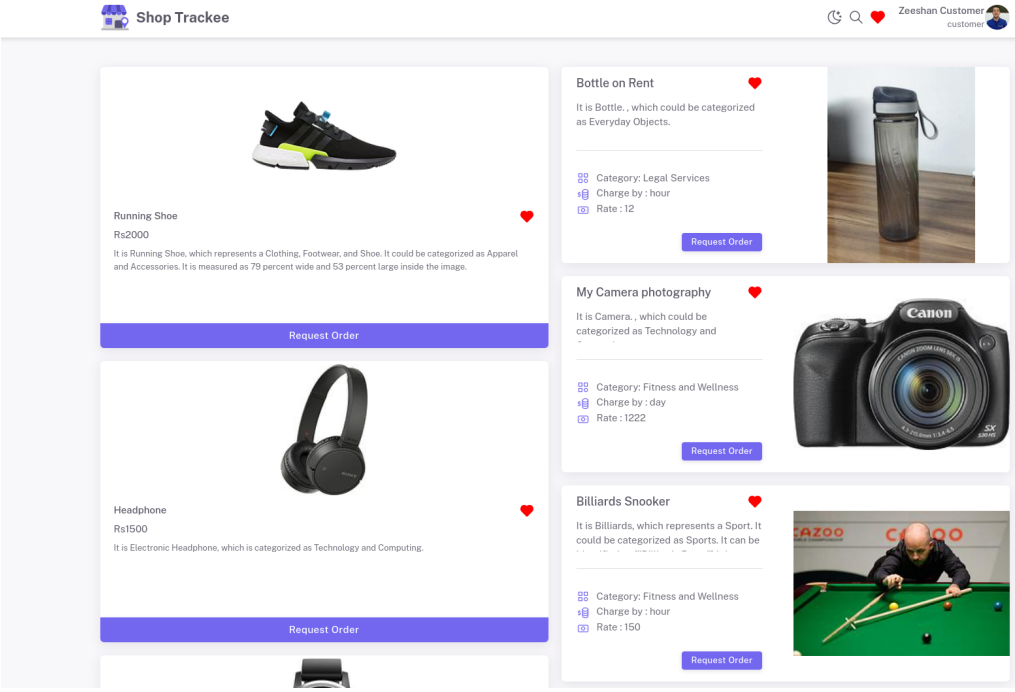

FIGURE 4.17: Track A Shop: Web App - Customer Favorites

4.4.18 Dark Mode

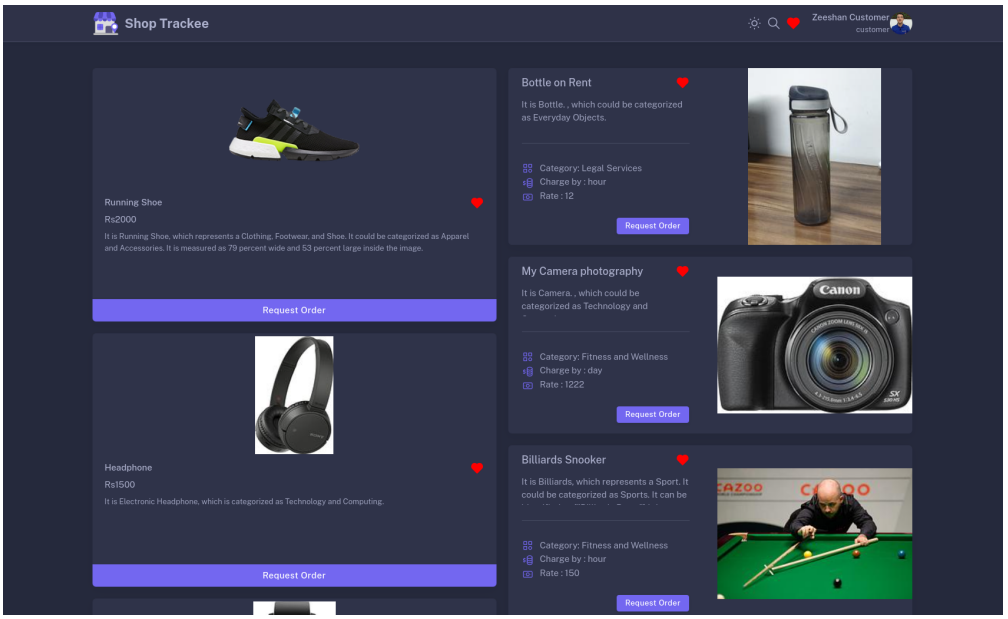

FIGURE 4.18: Track A Shop: Web App - Dark Mode

4.5 Hardware Interface

The Track A Shop web application is primarily a software-based system that relies on internet connectivity and web browsers to function. However, there are a few essential hardware components and interfaces that enable its operation:

- Computer Systems (Desktops, Laptops, Tablets)
- Smartphones and Mobile Devices
- Stable Internet Connection
- Cameras (for Image Capture)
- Global Positioning System (GPS) and Location Services
- Input Devices (Keyboards, Mouse, Touchscreens)

4.5.1 Hardware Requirements

Here are some minimal requirements of hardware to run this Web Application:

- Processor: core i5 - 2.0GHz
- RAM 4 GB+
- Free System Storage (minimum 2 GB)
- Operating System: Windows, Linux, Mac, Andriod, IOS
- Browsers (Chrome 64+, Edge 79+, Firefox 67+, Opera 51+, Safari 12+)
- GPS WGS 84 (World Geodetic System 1984) datum for Google Maps

4.6 Evaluation

Each element of our website is meticulously designed based on the findings of Market Shop Sellers and Customer Research, which takes into account criteria such as technical knowledge, and comprehension.The user is given a seamless experience, allowing them to easily see, update, upload, and manage their data. While designing the website, Human Computer Interaction concepts [18] were taken into account, allowing the user to see where they are presently working and easy to use. Before releasing the website, we make sure that all of the tabs and buttons are visible and double-checked and everything is working properly. Once the website is ready for the user, it will be extensively tested and any necessary changes will be done. Hence the website is user friendly and unambiguous.

4.7 Unit Testing

Unit testing is a crucial component of the development process in "Track A Shop: Web App." It involves testing individual units or components of the application to ensure their accuracy, reliability, and correctness. The primary focus is to verify that each unit of the software performs as designed.

In this project, unit testing will encompass the following aspects:

4.7.1 Frontend Components

Test individual frontend components [19], like UI elements, input fields, buttons, and their respective functionalities to ensure proper rendering and interactivity.

4.7.2 Backend API Endpoints

Test the backend API endpoints using testing frameworks to verify their response, data integrity, error handling, and authentication mechanisms [15].

4.7.3 Database Interactions

Test the interactions with the database, including CRUD operations, to ensure accurate data retrieval, storage, and manipulation.

4.7.4 Integration Testing

Conduct integration tests to validate the interaction and communication between different modules, ensuring the overall system functions seamlessly [20].

4.7.5 Error Handling and Edge Cases

Perform tests to evaluate the system's behavior under various error conditions and edge cases to guarantee robustness and graceful error handling [7].

Adopting a comprehensive unit testing strategy will enhance the project's quality, reduce bugs, and provide a stable and reliable application for users.

4.8 Test Cases

Unit testing is a crucial component of the development process in "Track A Shop: Web App". It involves testing individual units or components of the application to ensure their accuracy, reliability, and correctness. The primary focus is to verify that each unit of the software performs as designed. Here are some test cases for major units of the application:

4.8.1 Frontend Components

Here are the unit test cases for the frontend components:

4.8.1.1 Test Case 1: User Registration Form Validation

Table 4.1: User Registration Form Validation (TC1)

Test Case ID	TC1
Test Case Name	User Registration Form Validation
Component	User Registration Form
Input	Incomplete or incorrect user registration details
Expected Output	Prevent registration and display appropriate validation errors
Test Steps	1. Fill the registration form with incomplete or incorrect details 2. Attempt to submit the registration form 3. Check the interface to verify appropriate validation errors are displayed
Execution Result	Pass

4.8.1.2 Test Case 2: Product/Service Image Uploading & Recognition

Table 4.2: Product Image Uploading (TC2)

Test Case ID	TC2
Test Case Name	Product/Service Image Uploading and Recognition
Component	Product/Service Image Chooser / Image Capturing Camera
Input	Image upload/capture request
Expected Output	Successful image upload and display in the product/service management UI
Test Steps	1. Initiate image upload process 2. Check the Recognize Image button visibility under image 3. Recognize Image by using button 4. Check the UI to verify the image is displayed appropriately 5. Check the product/service form to verify recognized data
Execution Result	Pass

4.8.1.3 Test Case 3: Product/Service Component Rendering

Table 4.3: Unit Test for Rendering Product/Service Display Component (TC3)

Test Case ID	TC3
Test Case Name	Rendering Product/Service Display
Component	Product/Service Display Component
Input	None (Component rendering)
Expected Output	Rendered product/service display
Test Steps	1. Load the application 2. Navigate to the page where the products/services display component is rendered
Execution Result	Pass

4.8.1.4 Test Case 4: Order Request with Message

Table 4.4: Order Request with Message (TC4)

Test Case ID	TC4
Test Case Name	Order Request with Message
Component	Order Management UI
Input	Order request along with an optional message from the customer
Expected Output	Successful submission of the order request with the attached message
Test Steps	1. Select a product and initiate the order request process 2. Provide additional instructions or messages (optional) 3. Complete the order request 4. Check all order requests table to verify the order request was successful
Execution Result	Pass

4.8.1.5 Test Case 5: Search Box Functionality

Table 4.5: Search Bar Functionality (TC5)

Test Case ID	TC5
Test Case Name	Search Box Functionality
Component	Search Bar, Image Capturing Camera, Search Radius Field
Input	Search query, Search Image, Distance radius
Expected Output	Display of relevant search results
Test Steps	1. Enter a search query in the search bar 2. Initiate the search process 3. Check the interface to verify relevant search results are displayed
Execution Result	Pass

4.8.2 Backend API Endpoints

Here are the unit test cases for the backend API endpoints:

4.8.2.1 Test Case 6: Response of Product Listing Endpoint

Table 4.6: Test Case for Response of Product Listing Endpoint (TC6)

Test Case ID	TC6
Test Case Name	Response of Product Listing Endpoint
Endpoint	Product Listing
Input	Send a request to the product listing endpoint
Expected Output	Receive a list of products
Test Steps	1. Send a valid request to the product listing endpoint 2. Receive the response from the endpoint
Execution Result	Pass

4.8.2.2 Test Case 7: Error Handling of Invalid Request

Table 4.7: Test Case for Error Handling of Invalid Request (TC7)

Test Case ID	TC7
Test Case Name	Error Handling of Invalid Request
Endpoint	Any API Endpoint
Input	Send an invalid request
Expected Output	Receive an appropriate error response
Test Steps	1. Send an invalid request to the specified endpoint 2. Receive the error response from the endpoint
Execution Result	Pass

4.8.3 Database Interactions

4.8.3.1 Test Case 8: Product Data Retrieval

Table 4.8: Product Data Retrieval (TC8)

Test Case ID	TC8
Test Case Name	Product Data Retrieval
Component	Database Interaction
Input	Request to retrieve product data
Expected Output	Valid product data from the database
Test Steps	1. Send a request to retrieve product data 2. Check the received product data
Execution Result	Pass

4.8.3.2 Test Case 9: Adding New Product

Table 4.9: Adding New Product (TC9)

Test Case ID	TC9
Test Case Name	Adding New Product
Component	Database Interaction
Input	New product details (name, description, price, etc.)
Expected Output	Product added successfully in the database
Test Steps	1. Provide input for a new product 2. Initiate the process to add the product 3. Check the response to verify successful addition
Execution Result	Pass

4.8.4 Integration Testing

4.8.4.1 Test Case 10: Integration of User Authentication

Table 4.10: Integration of User Authentication (TC10)

Test Case ID	TC10
Test Case Name	Integration of User Authentication
Components	Authentication Module, Database Interaction
Input	User credentials
Expected Output	Successful user authentication
Test Steps	1. Provide user credentials 2. Initiate the authentication process 3. Check the response to verify successful authentication
Execution Result	Pass

4.8.4.2 Test Case 11: Integration of Product Listing

Table 4.11: Integration of Product Listing (TC11)

Test Case ID	TC11
Test Case Name	Integration of Product Listing
Components	Product Listing Module, Database Interaction
Input	Request to list products
Expected Output	Successful retrieval of product listing
Test Steps	1. Send a request to list products 2. Check the response to verify the successful retrieval of product listing
Execution Result	Pass

4.8.5 Error Handling and Edge Cases

4.8.5.1 Test Case 12: Invalid User Authentication

Table 4.12: Invalid User Authentication (TC12)

Test Case ID	TC12
Test Case Name	Invalid User Authentication
Components	Authentication Module, Database Interaction
Input	Incorrect user credentials
Expected Output	Authentication failure with appropriate error message
Test Steps	1. Provide incorrect user credentials 2. Initiate the authentication process 3. Check the response to verify authentication failure and error message
Execution Result	Pass

4.8.5.2 Test Case 13: Product Listing Empty

Table 4.13: Product Listing Empty (TC13)

Test Case ID	TC13
Test Case Name	Product Listing Empty
Components	Product Listing Module, Database Interaction
Input	Request to list products when no products are available
Expected Output	Successful retrieval with an empty product list
Test Steps	1. Send a request to list products 2. Check the response to verify an empty product list
Execution Result	Pass

4.8.5.3 Test Case 14: Invalid Product Addition

Table 4.14: Invalid Product Addition (TC14)

Test Case ID	TC14
Test Case Name	Invalid Product Addition
Components	Product Management Module, Database Interaction
Input	Incomplete or incorrect product details
Expected Output	Failure to add the product with an appropriate error message
Test Steps	1. Provide incomplete or incorrect product details 2. Attempt to add the product 3. Check the response to verify the error message
Execution Result	Pass

4.9 Functional Testing

Functional testing is a critical aspect of ensuring the "Track A Shop" web application meets its intended functional requirements. This type of testing involves validating the software's functions and features against the specified requirements and design. It primarily focuses on what the system does.

Functional testing for this project encompasses the following areas:

4.9.1 User Authentication and Authorization

Validate the authentication process to ensure users can securely create accounts, log in, and access appropriate functionalities based on their roles [21].

4.9.2 Product and Service Listings

Verify that sellers can effectively list their products and services using various methods, including image recognition and manual entry, and customers can view and search these listings.

4.9.3 Order Placement and Management

Test the ordering process to ensure customers can place orders for products and services, sellers receive order requests, and they can manage and respond to these orders appropriately.

4.9.4 Location-Based Features

Validate the location-based functionalities such as finding nearby shops, setting a search radius, and accessing directions to physical shops.

4.9.5 Search and Filtering

Verify the accuracy and efficiency of the search and filtering capabilities, allowing users to search for products, services, or shops based on various criteria.

4.9.6 Error Handling

Test the system's behavior under erroneous conditions, ensuring that appropriate error messages and notifications are displayed to users.

4.9.7 UI/UX Testing

Evaluate the user interface and overall user experience to ensure it is intuitive, consistent, and adheres to design guidelines.

Functional testing ensures that the application performs its intended functions correctly, providing users with a seamless and reliable experience.

4.9.8 Testing Requirements

Functional testing for this web application necessitates a comprehensive approach that covers various functionalities and ensures their correctness and reliability. The testing requirements include:

4.9.8.1 User Authentication:

- Verify that users can create accounts securely with the necessary role and personal information.
- Test login functionality, ensuring users can authenticate themselves with correct credentials [21].

- Validate the behavior of the system in case of incorrect login attempts, including appropriate error messages.

4.9.8.2 Product and Service Listings:

- Ensure sellers can accurately list products and services using various methods like image recognition, capturing image, and choosing image manually or manual entry.
- Validate that sellers can edit, delete, and update product and service information as needed.
- Verify that customers can view and filter product and service listings with ease.

4.9.8.3 Orders Management:

- Test the process of requesting orders, ensuring customers can successfully initiate orders for desired products and services.
- Validate that sellers receive order requests promptly and can manage them effectively, including accepting or rejecting order requests.
- Verify that customers receive timely notifications and updates on their orders.

4.9.8.4 Location-Based Features:

- Ensure users can find nearby shops based on their location and specified search radius.
- Validate the accuracy of location-based search results, providing users with relevant and nearby options.
- Test the functionality of accessing directions to physical shops from the user's current location.

4.9.8.5 Search and Filtering:

- Validate the accuracy and efficiency of the search feature, allowing users to search for products, services, or shops based on keywords.
- Test capabilities to ensure users can search based on image, categories, or other relevant criteria.

4.9.8.6 Error Handling:

- Verify that the system displays appropriate error messages and notifications for incorrect inputs or erroneous actions.
- Test the behavior of the system under unexpected conditions, ensuring graceful handling of errors to prevent system crashes.

4.9.8.7 UI/UX Testing:

- Validate that the user interface is consistent, intuitive, and adheres to design guidelines.
- Test the responsiveness of the application across various devices and screen sizes.

These testing requirements form the basis for comprehensive functional testing, ensuring the application meets its intended functionalities reliably and accurately.

Chapter 5

Conclusion & Future Work

5.1 Conclusion

The "Track A Shop: Web App" project introduces a user-friendly and affordable web application tailored for small businesses and local service providers. By offering a centralized platform accessible through standard web browsers on various devices, the project enables businesses to showcase their products and services online. Leveraging image recognition and geolocation features, the application simplifies the process of product listing and enhances the search experience for customers.

The main focus of the project has been to make digitalization affordable and accessible for small businesses. Through a well-structured codebase and an intuitive user interface, the application ensures a seamless and engaging experience for both sellers and customers. The use of modern technologies like Next.js, Ruby on Rails [22], AWS, and PostgreSQL ensures the application is robust and scalable.

Moving forward, gathering user feedback and staying updated with evolving technologies will be key to refining and improving the application. "Track A Shop: Web App" strives to contribute to a more connected digital economy, facilitating a stronger bond between local businesses and their communities.

5.2 Future Work

In the following, we outline the potential areas for future work and enhancements to further improve the "Track A Shop: Web App" platform.

5.2.1 Enhanced User Experience:

Conduct usability testing and gather extensive user feedback to refine the user interface, making it even more intuitive and appealing to a broader user base.

5.2.2 Machine Learning Integration:

Explore the integration of machine learning algorithms to enhance the image recognition capabilities, allowing for more accurate and efficient product and service identification from uploaded images.

5.2.3 Facial Recognition

Investigate the possibility of integrating facial recognition technology for profile data completion using profile image and enhanced security within the application.

5.2.4 Personalized Recommendations:

Implement recommendation algorithms based on user preferences and behavior, offering personalized product and service recommendations to customers, thereby enhancing user engagement and sales.

5.2.5 Integration of Payment Gateways:

Incorporate secure and widely used payment gateways to facilitate online transactions directly within the application, streamlining the purchase process for customers.

5.2.6 Multi-Language Support:

Introduce multi-language support to make the application accessible to a more diverse audience, promoting inclusivity and user engagement across different regions and languages.

5.2.7 Real-Time Chat and Customer Support:

Integrate a real-time chat feature to enable direct communication between customers and sellers, fostering better customer support and trust.

5.2.8 Inventory Management System:

Develop an integrated inventory management system for sellers to efficiently track, manage, and update their product and service offerings in real-time.

5.2.9 Social Media Integration:

Integrate social media sharing functionalities to enable users to easily share their favorite products and services with their social network, enhancing the application's reach and user engagement.

5.2.10 Third Party Integration for Sign-in:

Integrate popular third-party authentication services (e.g., Google (for Google My Business), Facebook, Instagram etc.) to allow users to sign in using their existing credentials, enhancing user onboarding and convenience.

5.2.11 Integration with Other E-commerce Platforms:

Investigate the potential for integrating with established e-commerce platforms to allow seamless synchronization of product listings and sales, potentially broadening the reach and opportunities for sellers.

5.2.12 Customer Analytics Dashboard:

Create a comprehensive analytics dashboard for sellers, providing insights into customer behavior, purchase patterns, and popular products, aiding in informed business decisions.

5.2.13 Mobile Application Development:

Extend the project by developing dedicated mobile applications (iOS and Android) to cater to the increasing number of users accessing the platform via mobile devices.

5.2.14 Offline Access and Progressive Web App (PWA):

Implement Progressive Web App features to ensure limited functionality and access to the application even in offline mode, enhancing user experience.

5.2.15 Import data from Business Websites:

Explore features that allow sellers to import or list their existing products or services from already established business websites, streamlining the onboarding process and saving time for sellers.

5.2.16 Integration with Local Services:

Collaborate with local delivery services to enable seamless integration for order deliveries, further enhancing convenience for customers and boosting sales for sellers. These future directions offer a glimpse into the potential growth and evolution of the "Track A Shop: Web App" project, ensuring its continual adaptation to emerging technologies and the evolving needs of both businesses and consumers in the dynamic digital landscape.

Bibliography

[1] Graphviz. (n.d.). Graph visualization software - class diagram generator, 2021. URL https://graphviz.org/.

[2] Creately.com. Creately — visual collaboration & diagramming platform - sequence diagram, 2019. URL https://creately.com/.

[3] Inc. Amazon Web Services. Image recognition software - ml image & video analysis - amazon rekognition - aws, 2023. URL https://aws.amazon.com/rekognition/.

[4] Heroku. Heroku documentation, 2023. URL https://devcenter.heroku.com/.

[5] Google Cloud. Google maps platform, 2023. URL https://cloud.google.com/maps-platform.

[6] Inc. Amazon Web Services. Cloud object storage - amazon s3, 2023. URL https://aws.amazon.com/pm/serv-s3.

[7] Sentry. Sentry documentation, 2023. URL https://docs.sentry.io/.

[8] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. The latex companion. 2004.

[9] Ruby on Rails. Ruby on rails guides, 2023. URL https://guides.rubyonrails.org/.

[10] Vercel. Next.js documentation, 2023. URL https://nextjs.org/docs.

[11] Mehul Bhatt Harmeet Singh. *Learning Web Development with React and Bootstrap*. Packt Publishing, 2016. URL https://books.google.com.pk/books?id=sJf1vQAACAAJ&source=gbs_navlinks_s.

[12] Material-UI. Material-ui documentation, 2023. URL https://material-ui.com/.

[13] PostgreSQL Global Development Group. Postgresql documentation, 2023. URL https://www.postgresql.org/docs/.

[14] The Git Community. Git documentation, 2023. URL https://git-scm.com/doc.

[15] Postman. Postman documentation, 2023. URL https://learning.postman.com/docs/getting-started/introduction/.

[16] Towards Data Science. Continuous integration and continuous deployment (ci/cd) - a quick guide, 2023. URL https://towardsdatascience.com/a-comprehensive-guide-to-different-git-workflows-ecosystem-38d3e10fdec3.

[17] John Smith. Microservices architecture: What, when, and how. *InfoQ*, 2017. URL https://www.infoq.com/articles/microservices-architecture-introduction.

[18] Wikipedia Contributors. Human–computer interaction. https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction, 2023. Wikimedia Foundation, October 1, 2023.

[19] ReactJS. React.js documentation, 2023. URL https://reactjs.org/docs/getting-started.html.

[20] Mary Johnson. Understanding restful services, 2021. URL https://www.restapitutorial.com/lessons/whatisrest.html.

[21] IETF. Oauth 2.0 authorization framework, 2012. URL https://tools.ietf.org/html/rfc6749.

[22] Obie Fernandez Lucas Dohmen, Tom Henrik Aadland. *The Rails 7 Way*. Addison-Wesley's Professional Ruby Series. The Lean Publishing; 1st Edition, Published February 9, 2023. URL https://leanpub.com/therails7way.