GROUP PROJECT IN SOFTWARE DEVELOPMENT – CS2993

Development and Implementation Report

Group 07

Computer Store Management System with AI Chat Bot

General Sir John Kotelawala Defense University
Faculty of Computing

Group Member Details

Index No	Name	Degree
D/BCE/23/0008	P H D B Nayanakantha	COMPUTER ENGINEERING
D/BCE/23/0002	B D S D Douglas	COMPUTER ENGINEERING
D/BCE/23/0004	R M M Ranathunga	COMPUTER ENGINEERING
D/BCE/23/0014	R B H P Rathnamalala	COMPUTER ENGINEERING
D/BCS/23/0009	P R M K Herath	COMPUTER SCIENCE

Supervisor: Dr. Kaneeka Vidanage

October 2024

Table of Contents

List of Figures

List of Tables

Chapter 1 Software Development: Testing

1.1 Justification of testing technique

Testing was performed using **unit testing**, **integration testing**, **user testing**, and **system testing** for each module of the Computer Shop Management System. Each testing approach was selected based on the specific characteristics and requirements of the different modules, aiming for a comprehensive validation of the overall functionality.

1. Unit Testing

Unit testing was chosen for the **Customer Web Interface Module**, **AI Chatbot for Customer Module**, and **AI Chatbot for Owner Module** to verify the correctness of individual components. Each module was broken down into smaller units, such as form input validation, API response handling, and intent classification. Testing each of these units independently allowed for early detection of errors, ensuring that each feature worked as expected before integrating them into the system.

For instance, in the **Customer Web Interface Module**, unit testing included validating the form inputs, ensuring that each form (e.g., login, product search) handled both valid and invalid data appropriately. This prevented issues such as SQL injection or incorrect search results. Additionally, **API response handling** was evaluated to ensure that all data received from the server (e.g., product details) were displayed correctly, and any errors were handled gracefully.

In the **AI Chatbot for Customer Module**, unit testing was conducted on the **Natural Language Understanding (NLU)** component, specifically to verify the recognition of customer queries and accurate response generation. Boundary value analysis was used to ensure that extreme values, such as product prices at the high and low ends of the range, were recognized correctly. **Equivalence partitioning** was used to group similar intents, such as browsing and purchasing, ensuring the chatbot responded correctly to different yet similar requests.

For the **AI Chatbot for Owner Module**, unit testing focused on the **secure login functionality** and the **report generation component**. Each of these units was tested separately to ensure only authorized access, preventing unauthorized users from accessing sensitive information. Additionally, report generation was tested to ensure accuracy in querying the underlying sales and inventory database, confirming that data retrieval was performed correctly without discrepancies.

Justification: Unit testing allowed for detailed validation of individual functions within each module, ensuring that they worked as intended before proceeding to more complex integration phases. Testing the boundary values for form inputs and chatbot responses, for instance, ensured robust error handling

and validated the system's ability to handle edge cases smoothly. By thoroughly validating each individual component, potential issues were identified and resolved early, thereby enhancing the reliability of the final integrated system.

2. Integration Testing

Integration testing was performed to ensure that individual components worked together seamlessly. This was especially crucial for the **Customer Web Interface Module**, where integrating different services, such as the AI chatbot, product catalog, and checkout module, required seamless communication between these components. Any issues in the integration could lead to poor customer experience, such as incorrect product information being displayed or failure in chatbot interaction.For the **AI Chatbot for Customer Module**, integration testing ensured that the chatbot could correctly pull data from the product catalog and provide accurate recommendations to users. This also involved verifying the proper functioning of the voice input feature, ensuring smooth transcription of user queries and response generation.

For the **AI Chatbot for Owner Module**, integration testing focused on verifying the **secure login functionality**, using data fetched from the owner's database. It was also essential to confirm that the **report generation feature** was compatible with the sales and inventory database, ensuring that the chatbot could generate accurate and timely reports based on user queries.

The integration testing phase utilized a **top-down integration testing** approach, progressively combining individual components starting from the core module. For example, the integration between the **Customer Web Interface** and the **AI Chatbot** was tested to verify the interaction between the web interface and the chatbot's API, ensuring that users could seamlessly access the chatbot from the customer interface. Additionally, the **AI Chatbot for Owner Module** was tested to confirm that the sales data retrieved from the database was correctly processed and presented in an understandable format.

Justification: Integration testing was critical to ensure that the components interacted smoothly. Testing the integration between the web interface and chatbot guaranteed that the user experience was uninterrupted and seamless. The use of top-down integration testing allowed for early identification of interface-level errors and ensured that any issues related to data passing between the components were resolved efficiently.

3. User Testing

User testing was conducted to evaluate the usability and overall experience of the **AI Chatbot for Customer Module**. Participants interacted with the chatbot, and a Google Form was used to gather

feedback regarding ease of use, clarity of responses, and overall satisfaction. This feedback was essential to determine how well the chatbot met customer expectations and to identify any areas that required improvement.

User testing also included validation of the **voice input feature** for the AI Chatbot for Customer. Participants were asked to use the voice input functionality, and feedback was collected to assess how accurately the chatbot transcribed their queries and generated appropriate responses. This testing ensured that the voice input feature worked effectively across different accents and under varying noise conditions, thereby enhancing the chatbot's overall usability.

For the **Customer Web Interface Module**, user testing focused on assessing the system's **navigational flow** and **ease of use**. Users were asked to perform common tasks such as browsing products, adding items to the cart, and initiating a chatbot interaction. Feedback gathered was used to make improvements, such as refining the product search functionality and optimizing page load times.

The **AI Chatbot for Owner Module** was tested with the shop owner and administrative staff. This testing involved generating sales and inventory reports and querying the chatbot for specific information. User feedback focused on the ease of accessing information, report generation speed, and overall satisfaction with the chatbot's ability to provide relevant data. The results were used to make refinements, such as enhancing the accuracy of the query processing logic and improving the user interface for generating reports.

Justification: User testing was essential to evaluate the system from an end-user's perspective and to ensure that each module met the users' needs effectively. Using a Google Form to gather feedback ensured that insights were systematically collected, which allowed for data-driven improvements. The **voice input feature** was a key area of focus, as the accuracy of transcription directly impacted the user experience. Similarly, user testing of the **Owner Module** was critical to confirm that the chatbot could provide timely and accurate information, which was crucial for making informed business decisions.

4. System Testing

System testing was conducted for the **full system**, including all three modules (Customer Web Interface, AI Chatbot for Customer, and AI Chatbot for Owner). This type of testing was done to ensure that the integrated system met the defined requirements and functioned correctly in its entirety. Key features like the complete customer journey—from browsing products to querying orders via the chatbot—and owner-specific operations like sales report generation were evaluated. This level of testing ensured that the entire system operated cohesively and that all modules worked in harmony, without any functional or performance issues.

1.2 Test Case Design and Justification

The selection of test cases was driven by the specific features of each module, focusing on critical functionalities that define user interaction, system reliability, and security. The examples presented below illustrate how these test cases were designed to verify various scenarios and their respective behaviors.

Customer Web Interface Module

1. **Form Validation Test Cases:**
    - **Objective**: To ensure that all input fields, such as product search and contact information, are validated for correct and incorrect entries.
    - **Test Case Examples**:
        - **Valid Input**:
            - **Data Value**: "MacBook air"
            - **Expected Outcome**: The product search should proceed successfully and display the results.
        - **Invalid Input**:
            - **Data Value**: "###$@"
            - **Expected Outcome**: An error message such as "Invalid characters used, please enter a valid product name" should be displayed.
        - **Empty Input**:
            - **Data Value**: ""
            - **Expected Outcome**: The system should prompt the user with "Please enter a product name" message.
    - **Justification**: This ensures that invalid data is rejected, preventing errors or unexpected behavior on the website and ensuring a robust user experience.
2. **Product Search Test Cases:**
    - **Objective**: To validate the accuracy of the search when customers enter product names or categories.
    - **Test Case Examples**:
        - **Full Product Name**:
            - **Data Value**: "Dell Inspiron 15"
            - **Expected Outcome**: The specific product "Dell Inspiron 15" should be displayed.
        - **Partial Product Name**:
            - **Data Value**: "Dell In"

        - **Expected Outcome**: All relevant Dell Inspiron models should be displayed, helping the customer narrow their choices.
    - **Non-Existent Product**:
        - **Data Value**: "XYZ Laptop"
        - **Expected Outcome**: The system should display "No products found" message and may suggest related options.
  - **Justification**: Testing for both correct and incorrect search terms ensures customers receive accurate results and appropriate feedback if a product is not found, improving overall customer satisfaction.

AI Chatbot for Customer Module

1. **Intent Recognition Test Cases**:
    - **Objective**: To evaluate the chatbot's ability to correctly identify user intents, such as browsing products or tracking orders.
    - **Test Case Examples**:
        - **Clear Command**:
            - **Data Value**: "Show me gaming laptops"
            - **Expected Outcome**: The chatbot should display a list of gaming laptops available in the store.
        - **Ambiguous Phrase**:
            - **Data Value**: "I need something powerful"
            - **Expected Outcome**: The chatbot should ask follow-up questions, such as "Are you looking for a gaming laptop or a workstation?" to clarify user needs.
        - **Request for Help**:
            - **Data Value**: "Can you help me?"
            - **Expected Outcome**: The chatbot should offer a list of options, such as "I can help you browse products, track your order, or provide store information."
    - **Justification**: Ensuring the chatbot accurately detects intents helps deliver relevant responses, especially for ambiguous user requests, thereby enhancing user experience and satisfaction.
2. **Voice Input Processing Test Cases**:
    - **Objective**: To verify that the voice-to-text conversion feature functions as expected across different accents and noise conditions.

- **Test Case Examples**:
  - **Different Accents**:
    - **Data Value**: Query in a strong British accent - "What are the best deals on laptops today?"
    - **Expected Outcome**: Accurate transcription and relevant response listing discounted laptops.
  - **Background Noise**:
    - **Data Value**: Query in a moderate noise environment - "Can I get more details on MacBook Pro?"
    - **Expected Outcome**: The voice input should still be accurately transcribed, and the chatbot should provide information about the MacBook Pro.
  - **Low Voice Input**:
    - **Data Value**: Query in a low voice volume - "Track my order"
    - **Expected Outcome**: The chatbot should accurately recognize and proceed to order tracking.
- **Justification**: This tests the robustness of the voice recognition feature and ensures it functions reliably for a diverse customer base, providing a smooth experience even in real-world noisy environments.

AI Chatbot for Owner Module

1. **Secure Login Test Cases**:
   - **Objective**: To verify that login procedures are secure and only authorized users can access the owner interface.
   - **Test Case Examples**:
     - **Valid Credentials**:
       - **Data Value**: Correct username and password combination.
       - **Expected Outcome**: Successful login and access to the owner dashboard.
     - **Invalid Password**:
       - **Data Value**: Correct username and incorrect password.
       - **Expected Outcome**: "Invalid login credentials. Please try again." message, with no access granted.
     - **SQL Injection Attempt**:
       - **Data Value**: "admin' --"

        - **Expected Outcome**: Login fails, and an error message is displayed, with logs showing that an invalid request was attempted.
    - **Justification**: Ensuring secure login is critical for the protection of sensitive business data. By testing various login scenarios, including security threats like SQL injection, the reliability and safety of the system are maintained.

2. **Report Generation Test Cases**:
    - **Objective**: To test the chatbot's ability to generate sales and stock reports.
    - **Test Case Examples**:
        - **Short Timeframe**:
            - **Data Value**: Sales data for "Last 7 days"
            - **Expected Outcome**: Report generated showing sales for the last week, with accurate sales numbers and inventory status.
        - **Long Timeframe**:
            - **Data Value**: Sales data for "Last 12 months"
            - **Expected Outcome**: A detailed report of sales over the year, including a graphical representation for better understanding.
        - **Varying Sales Volume**:
            - **Data Value**: Query report for a period with both high and low sales activity.
            - **Expected Outcome**: Accurate representation of sales trends, clearly indicating periods of high and low activity.
    - **Justification**: Ensuring the reporting system can handle various data sizes and generate correct reports is essential for informed business decision-making. It allows the shop owner to better manage inventory and plan sales strategies effectively.

1.3 Test logs

Here is the complete table for the test logs, incorporating all relevant details for each test case conducted across the three modules of the system:

Table 1: Test Logs

Test Case ID	Description	Data Values	Module	Function Under Test	Expected Result	Actual Result	Conclusion
TC01	Form Validation - Product Search	"Laptop", "@@@", "xYz123"	Web Interface Module	Input Validation	Only valid product names accepted	Incorrect inputs rejected	Successful
TC02	User Authentication - Password Reset	Valid and invalid email	Owner Chatbot	Password Reset	Password Reset	Failed	Failed
TC03	Voice Input Processing - Product Recommendations	Voice query: "Recommend a laptop"	Customer Chatbot	Voice-to-Text Conversion	Accurate transcription and response	Correct transcription and response	Successful
TC04	Integration - Web and Chatbot Interaction	Search product, chat for details	Web Interface Module	Web-Chatbot Communication	Smooth integration, chatbot response visible	Integration successful	Successful
TC05	Secure Login Verification	Valid and invalid credentials	Owner Chatbot	Secure Access	Correct credentials allowed, unauthorized attempts blocked	Unauthorized access blocked	Successful
TC06	Report Generation - Monthly Sales Data	Request sales report for March	Owner Chatbot	Data Retrieval	Report generated with correct details	Correct report generated	Successful
TC07	User Testing - Voice Input	Google Form feedback	Customer Chatbot	Usability Assessment	Positive responses about accuracy	Voice input rated highly	Successful, with minor improvements required

Test Case ID	Description	Data Values	Module	Function Under Test	Expected Result	Actual Result	Conclusion
TC08	System Testing - Full System Evaluation	Complete customer journey	System (All Modules)	End-to-End Functionality	All components work seamlessly	Smooth operation of all modules	Successful
TC09	Product Search - Invalid Entry	Empty value	Web Interface Module	Product Search	Prompt user for valid input	Prompt displayed correctly	Successful
TC10	Product Search - Partial Match	"HP Pavilion"	Web Interface Module	Product Search	Display related products	Correct related products shown	Successful
TC11	Intent Recognition - Ambiguous	"I need something fast"	Customer Chatbot	Intent Classification	Clarify with user about preferences	Clarification question asked	Successful, additional intents added for handling ambiguity
TC12	Voice Input - Background Noise	Query with background music	Customer Chatbot	Voice-to-Text Conversion	Correct transcription, relevant response	Transcription was mostly accurate	Successful, minor issues in noisy environments
TC14	Inventory Update - Adding Product	New Product Details	Web Interface Module	Inventory Management	Product added to inventory	Product successfully added	Successful
TC15	Integration - Chatbot and Database	Query for product stock	Customer Chatbot	Chatbot-Database Connection	Correct stock level shown	Accurate stock level displayed	Successful
TC16	Checkout Process Validation	Multiple items in cart	Web Interface Module	Checkout Process	Successful checkout, correct total calculation	Checkout completed, correct total	Successful
TC17	User Testing - Navigation	Google Form feedback	Customer Chatbot	Usability Assessment	Ease of navigation feedback gathered	Positive feedback with suggestions for improvement	Successful, suggested improvements logged for future development
TC18	Report Generation - Low Sales Data	Sales report for low activity period	Owner Chatbot	Data Retrieval	Report generated with accurate, low sales data	Correct report generated	Successful

Test Case ID	Description	Data Values	Module	Function Under Test	Expected Result	Actual Result	Conclusion
TC19	System Testing - Database Consistency	Product order, inventory check	System (All Modules)	Database Consistency Check	Inventory adjusted after order	Inventory correctly updated	Successful
TC20	Integration - Owner Chatbot and Sales DB	Request for profit margins report	Owner Chatbot	Report Generation	Report generated correctly	Correct margins report displayed	Successful
TC21	Payment Gateway Integration	Valid and Invalid Payment Details	Web Interface Module	Payment Processing	Accept valid payments, reject invalid ones	Payments processed failed	Failed
TC22	Error Handling - Incorrect Inputs	Negative price entry	Owner Chatbot	Data Validation	Reject negative values	Correctly rejected	Successful
TC23	Cross-browser Testing	Access from different browsers (Chrome, Firefox, Safari)	Web Interface Module	Compatibility	Consistent behavior across browsers	Consistent performance	Successful
TC24	Chatbot Response Time	Various user queries	Customer Chatbot	Response Time Measurement	Response time within acceptable limits	Average response time within limits	Successful, monitoring suggested for ongoing performance

Below is a table summarizing the customer chatbot intent test results:

Table 2: Customer Chatbot Intents Test Results

Intent Name	Precision	Recall	F1-Score	Support	Confused With
inquire_gaming_laptops	1.0	1.0	1.0	9	None
inquire_best_brands	1.0	1.0	1.0	10	None
asking_about_shop	1.0	1.0	1.0	15	None
item_price	1.0	1.0	1.0	9	None
asking_about_discounts	1.0	0.89	0.94	9	asking_about_bulk_discounts (1)
bot_challenge	1.0	1.0	1.0	16	None
inquire_review_submission	1.0	1.0	1.0	10	None
ask_available_one_item	1.0	1.0	1.0	9	None
inquire_laptop_accessories	1.0	1.0	1.0	10	None
inquire_order_delivery_time	1.0	1.0	1.0	10	None
asking_home_delivery	1.0	1.0	1.0	15	None
inquire_refurbished_computers	1.0	1.0	1.0	10	None
ask_available_items	1.0	1.0	1.0	15	None
inquire_gift_cards	1.0	1.0	1.0	10	None
asking_how_to_claim_warranty	1.0	1.0	1.0	15	None
Greet	1.0	1.0	1.0	10	None
ask_discount_on_item	0.9	1.0	0.95	9	None
Goodbye	1.0	1.0	1.0	10	None
asking_about_customer_support	1.0	1.0	1.0	15	None
asking_about_location	1.0	1.0	1.0	14	None
ask_repairs	1.0	1.0	1.0	14	None
ask_item_pic	1.0	0.86	0.92	7	ask_discount_on_item (1)
Deny	1.0	1.0	1.0	7	None
ask_warranty_on_item	1.0	1.0	1.0	9	None
inquire_privacy_policy	1.0	1.0	1.0	10	None
asking_capabilities	1.0	1.0	1.0	15	None
asking_about_exchange_product	1.0	1.0	1.0	15	None
inquire_cctv_price	1.0	1.0	1.0	12	None
inquire_budget_computers	1.0	1.0	1.0	10	None
asking_help	1.0	1.0	1.0	15	None
inquire_desktop_for_graphic_design	1.0	1.0	1.0	10	None
inquire_computer_types	1.0	1.0	1.0	7	None
inquire_best_laptops_for_students	1.0	1.0	1.0	10	None
asking_about_working_hours	1.0	1.0	1.0	15	None
inquire_cctv_operation	1.0	1.0	1.0	5	None

Intent Name	Precision	Recall	F1-Score	Support	Confused With
inquire_shipping_cost	1.0	1.0	1.0	10	None
inquire_antivirus_software	1.0	1.0	1.0	10	None
ask_description_of_item	1.0	1.0	1.0	9	None
inquire_custom_pc_build_time	1.0	1.0	1.0	10	None
inquire_custom_computer	1.0	1.0	1.0	10	None
asking_about_bulk_discounts	0.94	1.0	0.97	15	None
inquire_cctv_installation	1.0	1.0	1.0	5	None
find_components	1.0	1.0	1.0	15	None
inquire_custom_pc_build	1.0	1.0	1.0	10	None
asking_about_return_policy	1.0	1.0	1.0	15	None
Affirm	1.0	1.0	1.0	6	None
asking_about_payments	1.0	1.0	1.0	15	None

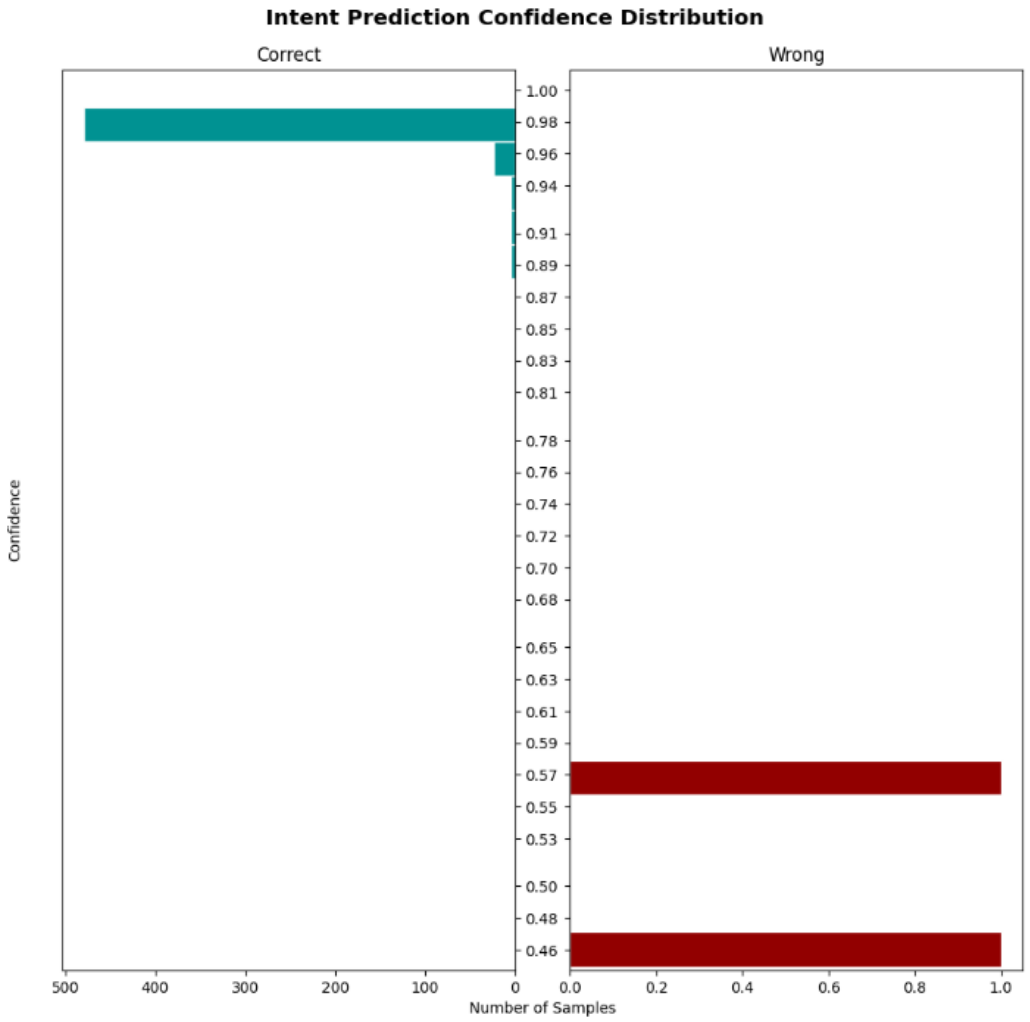


Figure 1: Intent Prediction Confidence Distribution

Key Metrics Explained:

1. **Precision**:
    - Indicates how many predictions of a specific intent were correct.
    - A high precision value (close to 1.0) suggests the model makes very few false-positive errors.
2. **Recall**:
    - Shows how many real instances of an intent were correctly identified.
    - A high recall value (close to 1.0) means the model captured almost all instances, with few false negatives.
3. **F1-Score**:
    - The harmonic mean of Precision and Recall, providing a balanced view.
    - A high F1-Score indicates a good trade-off between precision and recall, meaning the model performs well overall.
4. **Support**:
    - The number of test examples for each intent.
    - It helps determine the amount of data used to calculate metrics for each intent, giving confidence in the reported performance.
5. **Confused With**:
    - Indicates if any intents were incorrectly predicted as others.

Overall Performance Metrics

Table 3: Overall Performance of Intent Test Result

Metric	Value
Accuracy	99.62%
Macro Average	Precision: 0.997
Weighted Average	Precision: 0.997

Key Observations:

- Most intents achieved 100% precision, recall, and F1-score, indicating excellent performance.
- Minor confusion occurred with similar intents: "asking_about_discounts" was confused once with "asking_about_bulk_discounts," and "ask_item_pic" was confused once with "ask_discount_on_item.". Some intents, like "ask_discount_on_item" and "asking_about_bulk_discounts," have slightly lower F1-scores but still exhibit high accuracy.

Below are some of customer chatbot user test results:

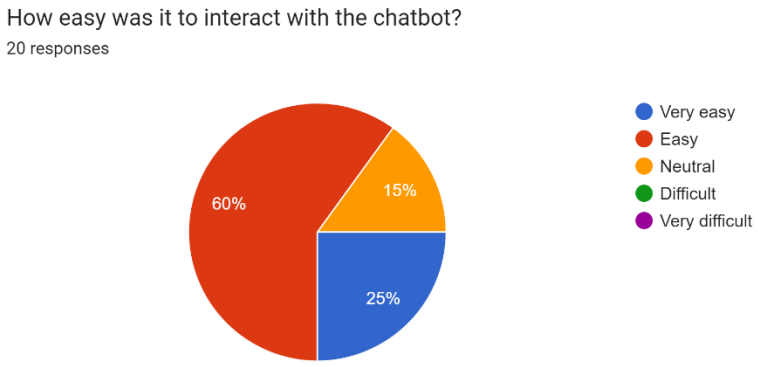


Figure 2: Question 1

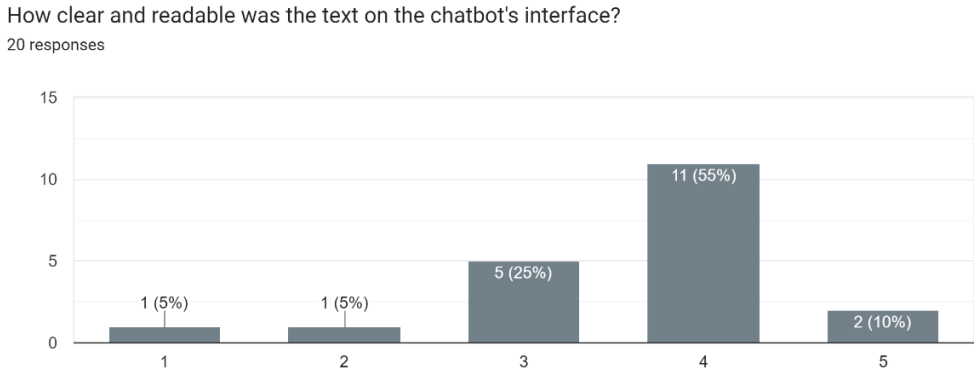


Figure 3 : Question 2

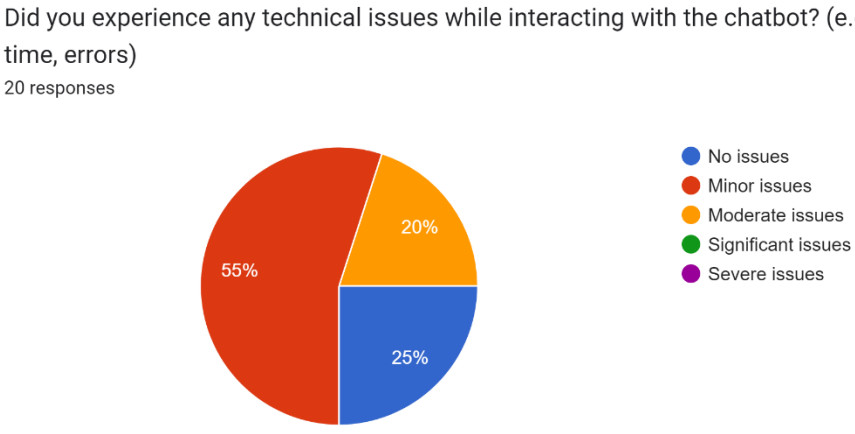


Figure 4 : Question 3

Note:

For a detailed breakdown of each test case, including specific data values, test procedures, and further analysis, please see the appendix.

Chapter 2 Implementation: Conversion and Training Plan

2.1 Justification of conversion technique

The chosen conversion technique is a **Phased Conversion**.

- The system will be implemented in **phases**, starting with the **Customer Web Interface**, followed by the **Customer AI Chatbot**, and finally the **Owner AI Chatbot**.
- This approach allows each component to be tested and integrated sequentially, reducing risks. Phased conversion is also appropriate given the interdependencies between modules, allowing gradual adaptation for both customers and staff.

2.2 Schedule and Planning for Conversion

The **Gantt chart** below outlines the implementation schedule:

Table 4: Implementation Schedule

No	Task	Time							
		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
1	Integration of Customer AI Chatbot and testing								
2	Deployment of Customer Web Interface.								
3	Integration of Owner AI Chatbot.								
4	Training sessions for staff on using the Owner AI Chatbot.								
5	Full system testing and user feedback phase.								

The implementation of the Computer Shop Management System, as outlined in the Gantt chart, involves several key phases distributed across eight weeks. Each phase requires specific resources, including personnel, hardware, software, and designated locations, to ensure a successful and smooth deployment. Below, the narrative is divided into sections that correspond to each phase of the Gantt chart timeline.

Week 1-2: Integration of Customer AI Chatbot and Testing

The initial phase focuses on the integration of the **Customer AI Chatbot** into the web interface and conducting comprehensive testing to ensure seamless performance.

Personnel Involved:

- The **development team** plays a vital role in this phase, leveraging their expertise in AI and software development. Their tasks include setting up the chatbot, connecting it to the web interface via APIs, and optimizing the chatbot's responses for different customer intents and entities.
- **IT staff** assist in monitoring technical aspects and ensuring performance standards are met during testing.

Hardware Resources:

- **Cloud servers** are used to host the chatbot, which is essential for managing customer interactions effectively and processing the natural language understanding (NLU) component.

Software Setup:

- Installation packages and APIs are used to connect the chatbot to the web interface. The software setup also includes configuring the **voice input** feature, offering an alternative for customers preferring voice over text interactions.

Testing Procedures:

- The integration testing is performed using both **manual and automated tests**, simulating various customer interactions to evaluate the chatbot's responses. Testing includes handling FAQs and ensuring the chatbot can recognize voice commands appropriately.
- The IT staff perform performance monitoring, validating that the chatbot responds accurately to different types of inquiries.

Week 3: Deployment of Customer Web Interface

The third week is designated for the **deployment of the Customer Web Interface**, providing customers with access to the system.

Personnel Involved:

- The **development team**, in coordination with **IT staff**, takes responsibility for deploying the web interface on the cloud.
- IT staff conduct follow-up testing to confirm that the system is functioning as intended.

Hardware Resources:

- Deployment utilizes **cloud-based virtual machines** and **scalable storage solutions** to ensure the web interface can handle high user volumes and maintain reliability during peak traffic.

Software Setup:

- The **React framework** is installed to build a responsive user experience for customers. Deployment scripts are executed for configuring the web server, installing dependencies, and ensuring smooth operation.
- **DNS configurations** are managed to provide customers with access to the system, while secure protocols are established to protect user data during transactions.

Deployment Testing:

- After deploying the interface, the IT staff perform **stress testing** to assess how well the system handles large numbers of concurrent users. Any discrepancies are documented and resolved promptly.

Week 4-5: Integration of Owner AI Chatbot

Weeks four and five are reserved for the **integration of the Owner AI Chatbot**, which provides the shop owner with essential business insights.

Personnel Involved:

- The **development team** integrates the chatbot into the administrative interface. This integration includes deploying the AI model and enabling data retrieval for inventory and sales.
- The **administrative staff** and **shop owner** contribute feedback to ensure that the chatbot meets the operational needs of the business.

Hardware Resources:

- **Dedicated servers** host the Owner AI Chatbot, providing high performance and reliable data processing for sales and inventory management.

Software Setup:

- Integration involves using **APIs** to facilitate data retrieval from inventory and sales databases. The software setup ensures that the chatbot can effectively respond to typical business queries, like identifying low stock levels or generating sales summaries.

Testing and Feedback:

- Mock interactions with the **administrative staff** are conducted to verify the chatbot's performance and usability. The feedback gathered is used to improve the system and refine its functionality, ensuring it meets all business requirements.

Week 6: Training Sessions for Staff on Using the Owner AI Chatbot

In the sixth week, training sessions are organized for the **shop owner** and **administrative staff** on using the Owner AI Chatbot.

Personnel Involved:

- **IT staff** and the **development team** conduct training, providing hands-on demonstrations and guided practice sessions.

Hardware Resources:

- Training requires **computers** equipped with the necessary software for accessing and interacting with the chatbot interface.

Training Process:

- Participants are guided through the process of logging into the Owner AI Chatbot using the secure link **computershop-owner.azurewebsites.net**. The training focuses on the different functionalities available, such as querying inventory and sales data.
- To support training, **user manuals** and **tutorial videos** are provided, which the administrative staff can use as reference materials for continued learning.

Support Resources:

- A **helpdesk service** is established by the IT staff to address any questions or technical issues that may arise during or after training, ensuring staff can fully utilize the chatbot.

Week 7-8: Full System Testing and User Feedback Phase

The final phase, extending across weeks seven and eight, includes **full system testing** and collecting **user feedback** to implement final adjustments.

Full System Testing:

- This stage involves a complete end-to-end evaluation of all modules: the **Customer Web Interface**, **Customer AI Chatbot**, and **Owner AI Chatbot**. The **development team** and **IT staff** collaborate on testing the system under real-world conditions, ensuring smooth interaction between all modules.

Personnel Involved:

- The **shop owner** and **administrative staff** actively participate in testing and provide essential **user feedback** on their experiences.

Hardware and Software Setup:

- The **cloud servers** used for hosting the chatbots and web interfaces undergo thorough testing to ensure they meet performance and reliability standards.
- The IT staff also configure **Azure Monitor** to provide continuous alerts in the event of system downtime or other issues, enabling rapid responses to maintain system availability.

User Feedback Collection:

- Feedback from the **shop owner** and **administrative staff** is gathered through structured forms and interviews. The feedback process focuses on evaluating usability, performance, and overall satisfaction.
- Based on the collected feedback, the development team makes necessary refinements, such as optimizing the chatbot's responses or enhancing the user interface for better navigation.

Error Handling Testing:

- Special attention is given to testing **error handling** scenarios, such as incorrect login attempts and insufficient stock during purchases. This step ensures that users receive appropriate notifications and suggestions in case of errors, maintaining a high-quality user experience.

Final Evaluation:

- A **final performance evaluation** is conducted to confirm that all functionalities operate efficiently, data is processed correctly, and any identified issues have been resolved. The system is thoroughly vetted to ensure a stable and secure deployment before going live.

Chapter 3 Implementation: User Guide

3.1 Installation and Setup Instructions

The implementation of the system involves deploying three modules, stored on a portable hard drive and hosted on Azure. The following guide includes the setup for each module and how to deploy them using Azure resources.

Step 1: Prerequisites

1. **Hardware Requirements**:
    - Portable Hard Drive containing the modules.
    - A computer or server for initial setup.
2. **Software Requirements**:
    - **Node.js** for running the React web interface.
    - **Python** for AI Chatbot modules (Rasa and Streamlit).
    - **Azure CLI** for deployment and management of Azure resources.
    - **Docker** (optional) for containerizing the application.
    - **Azure Subscription** for cloud hosting.

Step 2: Setting Up the Modules from Portable Hard Drive

1. **Connect the Portable Hard Drive** to your computer/server where initial setup will be done.
2. **Copy the Modules**:
    - Create a dedicated folder for the project on your system.
    - Copy the following from the portable hard drive:
        - **Customer Web Interface Module** (React).
        - **AI Chatbot for Customer Module** (Rasa, HTML, CSS, JS).
        - **AI Chatbot for Owner Module** (Streamlit and Ollama).
3. Ensure the copied files and directories are properly structured as per module requirements.

Step 3: Setting Up Azure Resources

1. **Login to Azure Portal**:
    - Open the Azure Portal.
    - Sign in using your Azure credentials.
2. **Azure CLI Installation and Configuration**:
    - Download and install Azure CLI.
    - Authenticate by running the following command in your terminal:

```
az login
```

3. **Creating Azure Resources**:
    - **Resource Group**: Create a resource group to contain the resources:

```
az group create --name ComputerShopRG --location eastus
```

    - **Virtual Machine (VM) Setup** (for backend processing and hosting the modules):
        - Create a Virtual Machine:

```
az vm create --resource-group ComputerShopRG --name ShopVM --image UbuntuLTS --admin-username azureuser --generate-ssh-keys
```

        Note the **public IP address** assigned to the VM.

    - **Azure Storage Account**:
        - Create a storage account to store data related to the application:

```
az storage account create --name computeshopstorage --resource-group ComputerShopRG --location eastus --sku Standard_LRS
```

Step 4: Deploying Customer Web Interface Module

1. **Setup Node.js Environment**:
    - Connect to the Azure VM using SSH:

```
ssh azureuser@<public_ip_address>
```

- Navigate to the directory containing the Customer Web Interface files.
- Install dependencies:

```
cd customer-web-interface
npm install
```

2. **Run the React Application**:
   - Start the React server:

```
npm start
```

   - **Configure Azure App Service** (optional) to host the React app permanently.

Step 5: Deploying AI Chatbot for Customer Module (Rasa)

1. **Setup Python Environment**:
   - On the Azure VM, navigate to the directory containing the Rasa chatbot.
   - Create a virtual environment:

```
python3 -m venv chatbot-env
source chatbot-env/bin/activate
```

   - Install required dependencies:

```
pip install -r requirements.txt
```

   - Train the Rasa chatbot using:

```
rasa train
```

2. **Run the Chatbot**:
   - Start the Rasa server:

```
rasa run -m models --enable-api --cors "*"
```

   - Start the Rasa action server for custom actions:

```
rasa run actions
```

Step 6: Deploying AI Chatbot for Owner Module (Streamlit & Ollama)

1. **Setup Python Environment for Streamlit**:
    - On the Azure VM, navigate to the folder containing the owner chatbot.
    - Create a new virtual environment:

```
python3 -m venv owner-chatbot-env
source owner-chatbot-env/bin/activate
```

    - Install Streamlit:

```
pip install streamlit
```

    - Install additional dependencies (including Ollama SDK).
2. **Run Streamlit Application**:
    - Start the Streamlit server:

```
streamlit run app.py
```

Step 7: Setting Up Continuous Deployment Using Azure DevOps

1. **Azure DevOps Configuration**:
    - Create a new project in Azure DevOps.
    - Connect your project repository for continuous deployment.
    - Create a **pipeline** to automatically deploy updates made to the modules:
        - Install Azure CLI and configure scripts to deploy on code changes.
        - Set triggers for automatic builds and deployments.
2. **Deployment Script**:
    - Write a `deploy.sh` script that updates modules on Azure VM whenever changes are pushed:

```
#!/bin/bash
cd /path/to/module
git pull origin main
npm install && npm run build
```

Step 8: Securing Azure Resources

1. **Network Security Group (NSG)**:
    - Set up an NSG to allow only required traffic (such as HTTP, HTTPS, and SSH).
    - Restrict ports to prevent unauthorized access.
2. **SSL/TLS Certificate Setup**:
    - Use **Azure Key Vault** to generate an SSL/TLS certificate.
    - Bind the certificate to your deployed applications to secure connections.

Step 9: Testing and Verification

1. **Test Customer Interface**:
    - Verify if customers can successfully access the React application from a browser.
    - Test all functionalities, including browsing products and accessing the customer chatbot.
2. **Test Owner Chatbot**:
    - Log into the Streamlit application and verify the functionalities, such as report generation and data querying.
3. **User Testing**:
    - Ensure users (customers and shop owner) can access their respective modules without issues.
    - Record feedback for future improvements.

Step 10: Backup and Recovery Plan

1. **Azure Backup Configuration**:
    - Configure **Azure Backup** to back up the virtual machine and storage account regularly.
    - Set up a **recovery services vault** for long-term backup retention.
2. **Local Backup**:
    - Keep an additional backup of the application files on the portable hard drive in case of cloud failure.

Step 11: Final Configuration for Live Usage

1. **Public DNS Setup**:
    - Assign a **DNS name** to the public IP address of the Azure VM for easier access (e.g., `computershop.azurewebsites.net`).

2. **Monitoring and Alerts**:
    - Configure **Azure Monitor** to set up alerts for application downtime or performance issues.
    - Ensure real-time monitoring is enabled for quick response to any incidents.

3.2 Operating Instructions

Customer Web Interface Module:

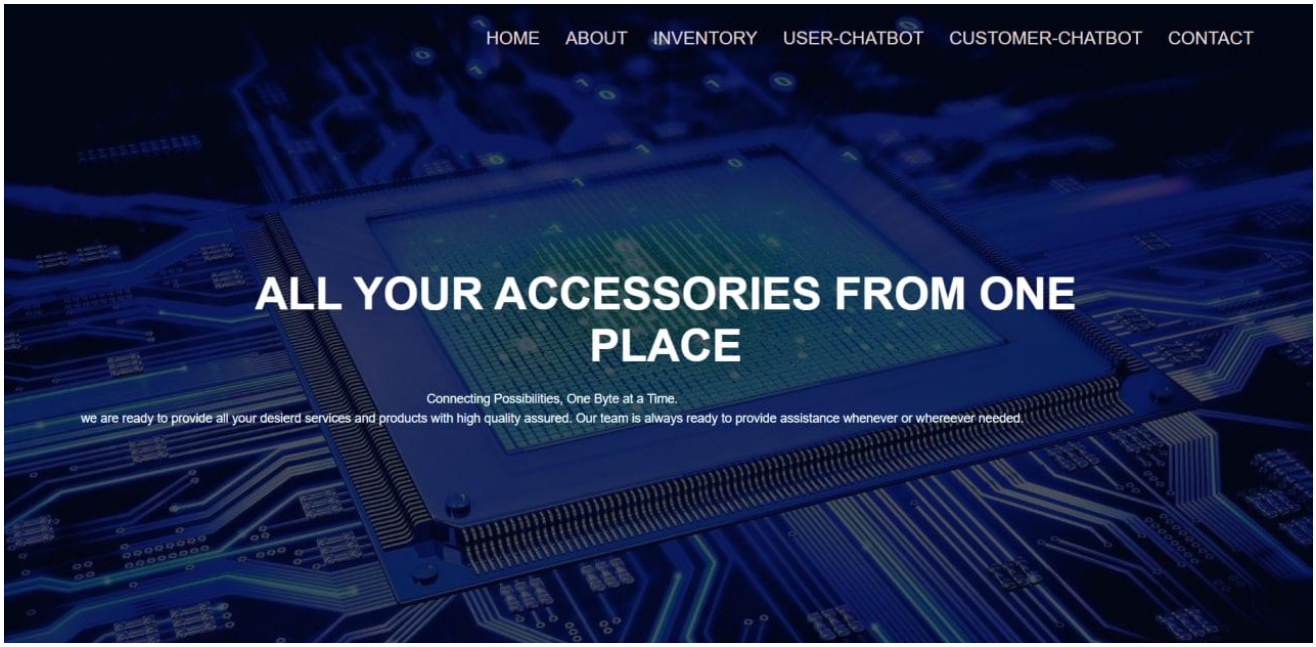


Figure 5: Web Interface UI

- **Access**:
    - Customers can access the computer shop's web interface by entering the assigned **DNS link** in their preferred web browser. The link will direct them to the homepage of the shop, where they can explore all available features.
    - The homepage will present key options like **Browse Products**, **Offers**, **Login/Sign-Up**, and the chatbot feature.
- **Interaction**:
    - Customers can **browse** the available products, use **filter options** to narrow down product categories (such as brand, price, and specifications), and perform **searches** using keywords.
    - Clicking on any product will redirect users to a detailed **product description page**, where they can view the specifications, features, reviews, and ratings of the product.

- To initiate a conversation with the AI Chatbot for assistance, customers can click on the **customer chat link** located in the top-right corner of the screen. This will redirect to customer chatbot interface, and customers can start typing their questions directly to interact with it.

AI Chatbot for Customer Module:

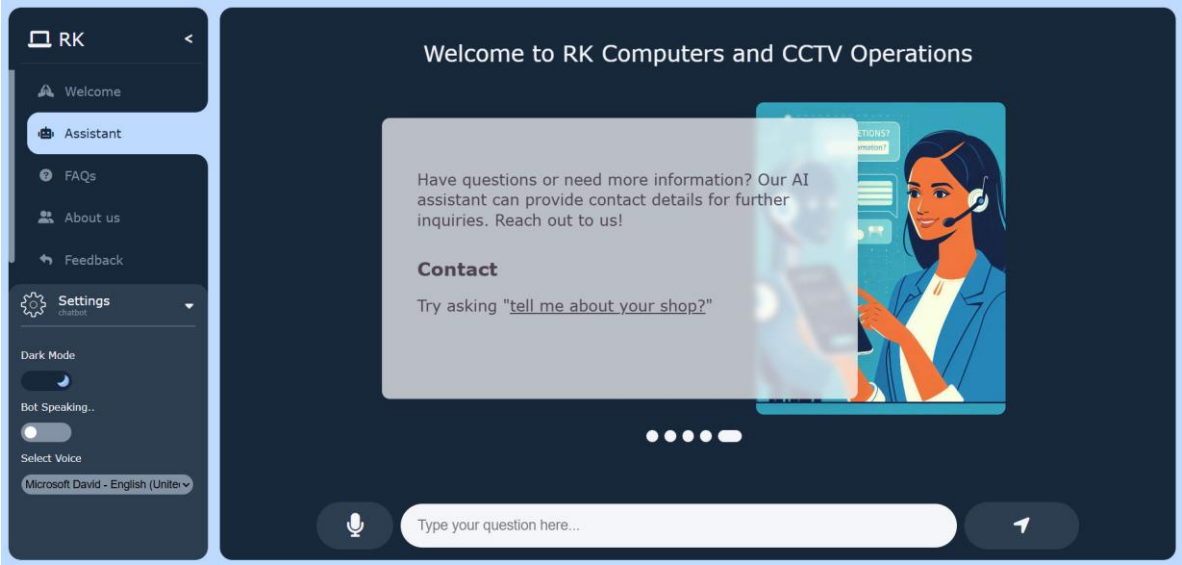


Figure 6: Customer Chatbot User Interface

Customer Care:

- The AI Chatbot can assist customers by answering **product-related questions**. Customers can type inquiries or use the **voice input feature** to ask questions directly. To use voice input, click on the **microphone icon** in the chatbot window and speak your query, such as "What are the features of the latest laptop?" or "What is the price of the intel i3?" The chatbot will process the voice input and respond accordingly.

- **Recommendations**:
  - Customers can interact with the chatbot to receive **personalized recommendations** based on their preferences using either text or **voice input**. For example, they can say, "I'm looking for a laptop for gaming," and the chatbot will process the voice command and provide relevant product options.

AI Chatbot for Owner Module:

Figure 7 : Owner chatbot User Interface

- **Access**:
  - Owners can log in to the **Owner Chatbot Module** by accessing the secure link (`https://computershop-owner.azurewebsites.net`) in a browser.
  - Upon opening the link, the owner will be presented with a **login page**. Owners need to enter their credentials (username and password) to gain access.
- **Data Retrieval**:
  - Once logged in, owners can use the interface to **generate reports** related to sales, such as **daily, weekly, or monthly sales reports**. Owners can select a date range and view comprehensive sales summaries, including product performance and revenue data.
  - Owners can also **query specific information** using a simple question interface. For example, owners may ask, "What is the current stock level for gaming PCs?" or "How many units of Product X have been sold this month?" and the system will provide the required data instantly.
  - The system will allow **exporting data** as spreadsheets for further analysis or record-keeping.

3.3 Description of Main Functions

- **Product Browsing**:

  - Customers visiting the web interface will have the ability to **browse the product catalog** through a user-friendly interface. The catalog is organized into different categories, such as **desktops, laptops, peripherals, and accessories**.
  - Each product page contains **comprehensive product details**, including key specifications, images, customer reviews, and the product's availability status. Customers can use **filtering options** to refine their search based on product features, pricing, or brands.
  - The **search function** allows customers to enter keywords to quickly find specific items in the inventory. Search suggestions will appear as they type, improving ease of access.

AI Assistance:

  - The **Customer Chatbot** helps using either text input or **voice input**. Customers can click on the **microphone icon** in the chatbot window to ask questions by speaking. The voice input feature uses **speech-to-text technology** to convert the spoken words into text, which is then processed using **natural language processing (NLP)**.
  - For **product recommendations** and assistance with orders, customers can choose to either type or speak their requests, making the interaction more convenient and accessible.

- **Owner Operations**:

  - The **Owner Chatbot** enables the shop owner to log in to a secure interface to access key operational features of the computer shop.
  - Owners can **generate detailed reports** on sales performance, including visual representations like **charts and graphs**. These reports are instrumental for **data-driven decision-making** and strategic planning.
  - The owner module also provides the capability to **query specific inventory data**. For instance, owners can check stock levels for specific products or generate **inventory summaries** to assist in replenishment decisions.

3.4 General Error Handling

1. Customer Errors:

- **Insufficient Stock Notification**:
    - If a customer attempts to add a product to their cart that is **out of stock**, a **notification** will be displayed to inform them that the item is not available. The message will provide the expected restock date, if available, and offer to **notify the customer** when the item is back in stock.
    - Example: "The item you selected is currently out of stock. Expected restock date: 05/15/2024. Would you like to receive an alert when it is available?"
- **Incorrect Login Attempts**:
    - If a customer or owner attempts to log in with incorrect credentials, an **error message** will appear specifying that the username or password is incorrect. The error message will include a link to reset the password if needed.
    - After **multiple failed login attempts**, the system will temporarily **lock the account** and send an email to the registered address with information on how to unlock it.

2. Chatbot Fallback:

- **Fallback Responses**:
    - If the chatbot cannot understand a query, it will provide a **fallback response**. The response will include **suggestions for rephrasing the question** or **display common questions and topics** to help guide the customer.
    - Example: "I'm sorry, I didn't quite understand your question. You can try asking about our products or your order status. Here are some examples of questions I can answer: 'What is the price of Laptop X?' '"
- **Escalation to Human Representative**:
    - If the customer continues to face issues or if the chatbot is unable to handle the request, the system will **escalate the query to a human representative**. A message will be displayed to inform the customer that their request is being forwarded for further assistance.

3. System Failures:

- **Error Page Display**:
  - In the event of **system downtime** or **backend errors**, the customer will be directed to a generic **error page** indicating that the system is temporarily unavailable. The page will contain a **friendly message** reassuring the customer and suggesting they try again later.
  - Example: "We're experiencing technical difficulties at the moment. Please try again later. We apologize for any inconvenience."
- **Azure Monitor Alert**:
  - When a system failure occurs, an **alert is sent via Azure Monitor** to notify the development and IT team about the failure. This ensures that the issue is addressed promptly, minimizing downtime.
  - Azure Monitor is configured to detect unusual system behaviors, such as **high server load** or **failure to respond** within a specified time, and sends an alert via email or SMS to the responsible personnel.
- **Automatic Failover (Optional)**:
  - To maintain **high availability**, a failover mechanism can be configured on Azure. If the primary server goes down, the system will automatically switch to a backup server to ensure the website and chatbot remain accessible to customers. This is critical for maintaining **customer satisfaction** and preventing disruption in services.

Chapter 4 Critical Appraisal: combined

The **Computer Shop Management System** has successfully enhanced both customer engagement and administrative efficiency through its integrated modules, comprising the **Web Interface Module**, the **AI Chatbot for Customer Module**, and the **AI Chatbot for Owner Module**. Each module brings specific value to the overall system, delivering distinct functionalities to support both customers and the shop owner effectively.

However, there were a few notable challenges encountered during development and integration that warrant attention for future improvements. One of the primary issues faced was the **integration of the AI chatbot** and ensuring smooth and accurate responses to customer inquiries. The chatbot's ability to handle **ambiguous queries** proved to be challenging, largely due to the complexity involved in natural language processing. Ambiguity in customer questions, such as vague descriptions or multiple meanings, led to instances where the chatbot either provided irrelevant responses or failed to answer effectively.

Another area identified for future enhancement is the system's **information retrieval capabilities**. Currently, the chatbot uses standard database querying mechanisms, which are limited when it comes to understanding complex relationships between various product attributes. A **knowledge graph** could be introduced to improve the chatbot's response accuracy and relevance. Knowledge graphs are known for their ability to establish connections between different data points, enabling the system to provide more nuanced and informative responses. This improvement would ultimately enhance customer satisfaction by enabling more detailed, context-aware answers.

Further, while the current system supports **English** for interactions, expanding to provide **multi-language support** would significantly broaden the system's reach and usability. As a computer shop, a diverse customer base could benefit from interaction in languages other than English. Introducing multi-language support will require training the chatbot using datasets that cover various languages, ensuring accurate translation and response generation. This development would also involve enhancing the **text-to-speech** and **speech-to-text** systems to work effectively across languages, thereby improving the overall customer experience.

Moreover, the system's **scalability** and **maintainability** were critical considerations during implementation. While the current infrastructure, hosted on **Azure**, provides a robust platform for deployment, future scalability might be required as the shop grows or additional features are introduced. To manage increasing user loads, the system may require optimized resource allocation and auto-scaling configurations to ensure performance and reliability.

The **customer data privacy** aspect is another area where continued improvements could enhance system integrity. While basic encryption mechanisms have been implemented, considering more advanced **data protection techniques** such as **end-to-end encryption** and **data anonymization** would provide an added layer of security, reassuring customers that their data is protected.

Overall, the **Computer Shop Management System** marks a significant improvement in managing customer and administrative interactions, laying a solid foundation for the future. Continued development focusing on addressing these challenges will make the system more robust, versatile, and user-friendly.

Chapter 5 Critical Appraisal: individual

5.1 Web Interface Module

The **Web Interface Module** plays a crucial role in bridging the gap between users—both customers and shop staff—and the system itself. This module provides the graphical user interface, ensuring customers have a user-friendly and engaging platform for interaction.

One of the major challenges faced during the development of this module was achieving **responsiveness across different devices**. Given the range of devices available today—from desktops and laptops to tablets and smartphones—ensuring a seamless experience was a significant hurdle. The module initially showed inconsistencies in layout and functionality when accessed from various screen sizes and resolutions, which required additional development time for debugging and testing.

To address this, a **mobile-first approach** was adopted during the later stages of development. However, despite these improvements, there are still areas where **cross-device compatibility** could be further optimized. Future efforts should focus on incorporating more rigorous **UI testing frameworks** to enhance the module's compatibility across devices. Automated tools such as **Selenium** and **Cypress**, combined with manual testing, could help identify issues early in the development cycle, allowing for more efficient resolution.

Another area of improvement for the Web Interface Module lies in **accessibility**. Ensuring that the interface is accessible to individuals with disabilities was considered, but the implementation could be further strengthened by adopting **Web Content Accessibility Guidelines (WCAG)**. This would involve adding features such as **keyboard navigation**, **screen reader compatibility**, and providing **alternative text** for images. Accessibility is key to ensuring that the platform is usable for all customers, including those with visual or motor impairments, which could expand the customer base significantly.

In summary, the Web Interface Module successfully provides a functional and user-friendly interface but requires ongoing enhancements in **responsiveness**, **compatibility**, and **accessibility** to reach its full potential.

5.2 AI Chatbot for Customer Module

The **AI Chatbot for Customer Module** was designed to offer real-time assistance to customers, providing answers to product inquiries, personalized recommendations, and guidance during the purchase process. The chatbot's ability to use both **text** and **voice input** made the interaction versatile and engaging for customers.

A major challenge encountered during the development of this module was the **quality of training data** used to train the chatbot's machine learning models. The initial dataset consisted of predefined questions and responses, which limited the chatbot's effectiveness in handling **unforeseen queries**. As a result, the chatbot struggled to understand customer questions that were phrased in unconventional ways or included colloquial language.

To overcome this challenge, the training dataset was expanded to include more diverse user queries and scenarios. However, there is still room for improvement. Moving forward, efforts should focus on collecting and incorporating **real user interactions** into the training data. This could be done by **logging queries** that the chatbot fails to answer and using these logs to refine the model's understanding. Additionally, implementing a **feedback loop** where customers can rate the responses would provide valuable insights into areas where the chatbot needs improvement.

The **voice input** feature also presented unique challenges, such as accurately converting spoken language into text and managing different accents or speech impediments. Future iterations could benefit from integrating more advanced **speech recognition services**, such as those offered by **Google Cloud Speech-to-Text** or **Microsoft Azure Speech Services**, which are designed to handle a wide range of accents and dialects.

In conclusion, while the AI Chatbot for Customer Module is functional and has significantly improved customer service, enhancements in **training data diversity** and **speech recognition capabilities** are crucial for achieving a higher level of accuracy and reliability.

5.3 AI Chatbot for Owner Module

The **AI Chatbot for Owner Module** is intended to assist the shop owner with administrative tasks, such as **generating sales reports**, **inventory management**, and **data querying**. By providing an easy-to-use, natural language interface, the module helps the owner efficiently access and manage information without the need for extensive technical knowledge.

One of the major concerns in developing this module was ensuring a **secure login** mechanism to protect sensitive business data. Initially, a **basic authentication system** was implemented, which involved username and password verification. However, this approach, while functional, lacked the robust security measures required for handling confidential information such as sales figures and inventory details.

To address this concern, additional security measures were introduced, such as **password hashing** and **HTTPS** for encrypted communication. Despite these improvements, it is recommended that the login system be further upgraded to include **OAuth 2.0** authentication. OAuth 2.0 is an industry-standard protocol for authorization, providing more secure access and the ability to integrate **multi-factor authentication (MFA)**. By implementing MFA, the system can add an extra layer of security, ensuring that unauthorized access is prevented, even in the event of a compromised password.

Another potential enhancement for this module is to introduce **role-based access control (RBAC)**, which would allow for more granular permissions. For example, while the shop owner might require access to all data, other staff members may only need access to inventory details without viewing sensitive sales information. Implementing RBAC would ensure that each user has access only to the data necessary for their role, thereby reducing the risk of data breaches.

In summary, while the **AI Chatbot for Owner Module** has succeeded in simplifying administrative tasks for the shop owner, future improvements in **security**—such as integrating OAuth 2.0 and RBAC—are essential to safeguarding sensitive data and ensuring that the system remains secure as it scales.

Chapter 6 References

[1] “Software Testing Methodologies,” smartbear.com. Accessed: Oct. 02, 2024. [Online]. Available: https://smartbear.com/learn/automated-testing/software-testing-methodologies/

[2] “Software Testing,” Methods. Accessed: Oct. 02, 2024. [Online]. Available: https://www.tutorialspoint.com/software_testing/software_testing_methods.htm

[3] GeeksforGeeks, “Types of Software Testing,” *GeeksforGeeks*, Aug. 01, 2017. Accessed: Oct. 02, 2024. [Online]. Available: https://www.geeksforgeeks.org/types-software-testing/

[4] S. Chatterjee, “What is Software Test Methodology? (With 6 Test Methodologies),” *BrowserStack*, Sep. 27, 2024. Accessed: Oct. 02, 2024. [Online]. Available: https://www.browserstack.com/guide/software-testing-methodologies

[5] “Testing Your Assistant,” Rasa Documentation. Accessed: Oct. 02, 2024. [Online]. Available: https://rasa.com/docs/rasa/testing-your-assistant/

[6] E. Kinsbruner, “The Complete Guide to Different Types of Testing,” Perfecto by Perforce. Accessed: Oct. 02, 2024. [Online]. Available: https://www.perfecto.io/resources/types-of-testing

[7] J. Unadkat, “Website Testing: A Detailed Guide,” *BrowserStack*, Aug. 28, 2023. Accessed: Oct. 02, 2024. [Online]. Available: https://www.browserstack.com/guide/how-to-perform-website-qa-testing

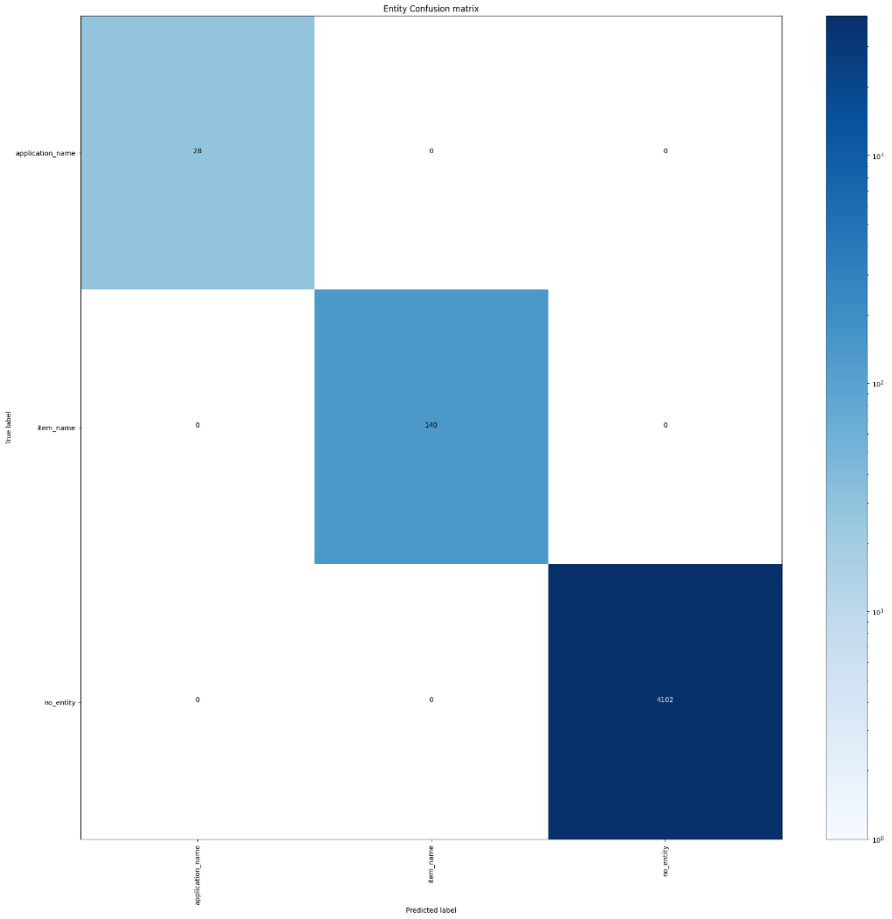
Chapter 7 Appendix

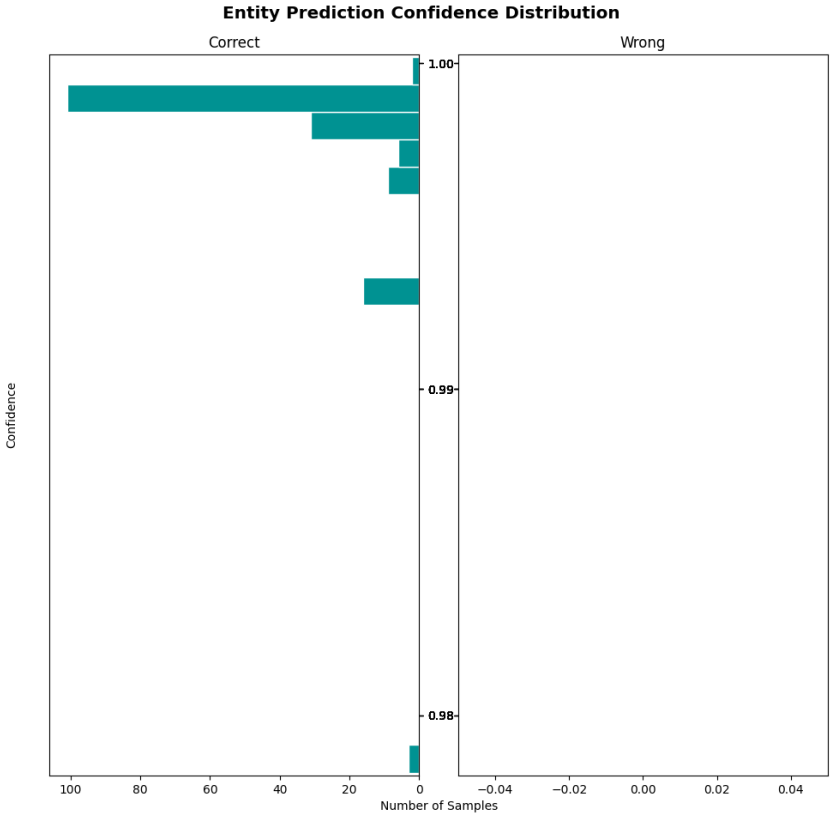
Customer chatbot intent test cases

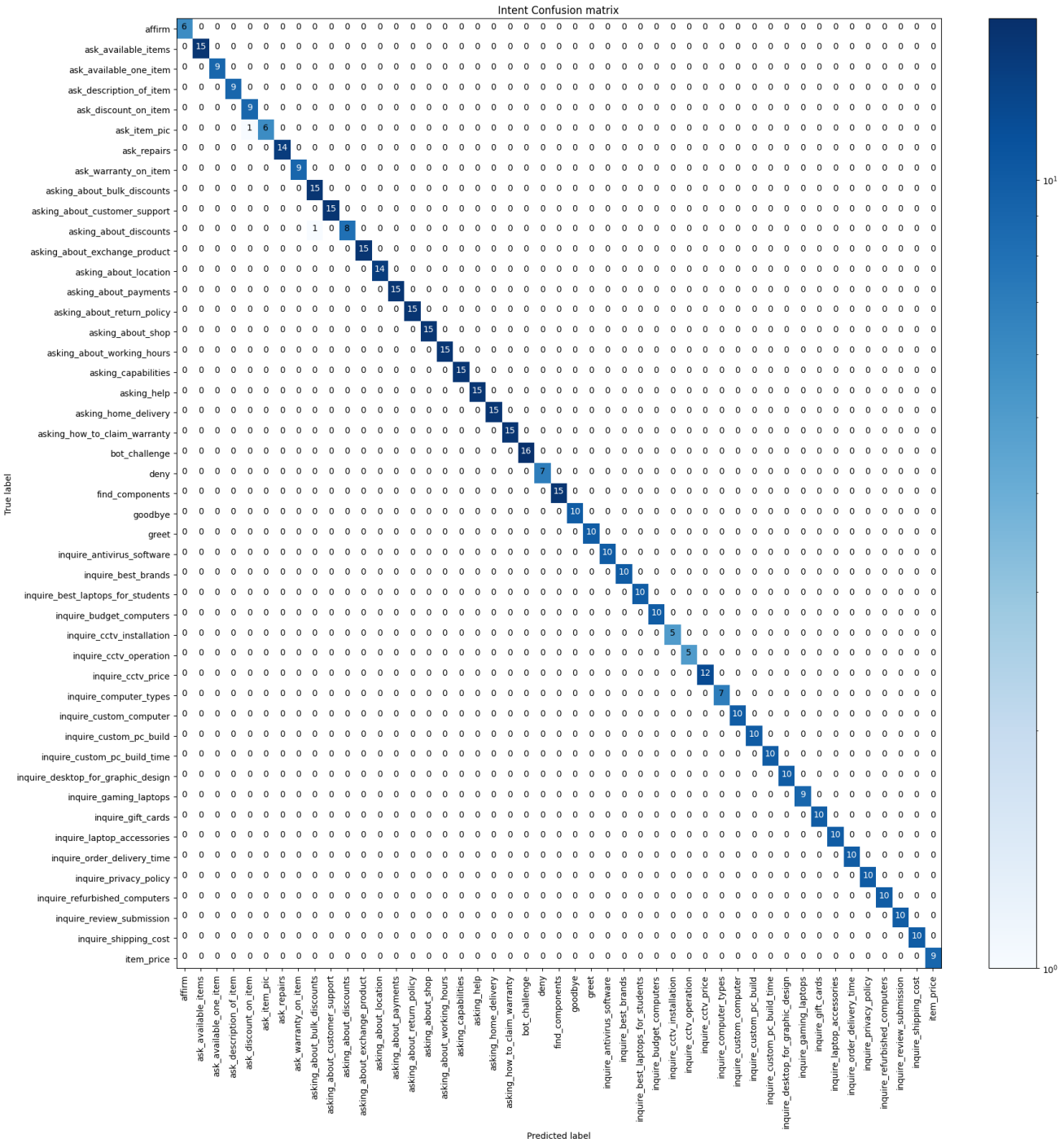
```
stories:
- story: say goodbye
  steps:
  - user: |
      bye-bye!
    intent: goodbye
  - action: utter_goodbye

- story: bot challenge
  steps:
  - user: |
      are you a bot?
    intent: bot_challenge
  - action: utter_iamabot

- story: find_components
  steps:
  - user: |
      What components do I need for playing GTA V
    intent: find_components
  - action: action_find_components

- story: ask for help
  steps:
  - user: |
      I need help with my order.
    intent: asking_help
  - action: utter_offer_help

- story: say hello
  steps:
  - user: |
      Hello!
    intent: greet
  - action: utter_greet


- story: ask about shop
  steps:
  - user: |
      Do you have a physical store?
    intent: asking_about_shop
  - action: utter_about


- story: ask about item price
  steps:
  - user: |
      What is the price of Intel i5?
    intent: item_price
  - action: action_give_price

- story: ask for item picture
  steps:
  - user: |
      show me Lenovo IdeaPad Flex 5
    intent: ask_item_pic
  - action: action_show_item

- story: ask about capabilities
  steps:
```

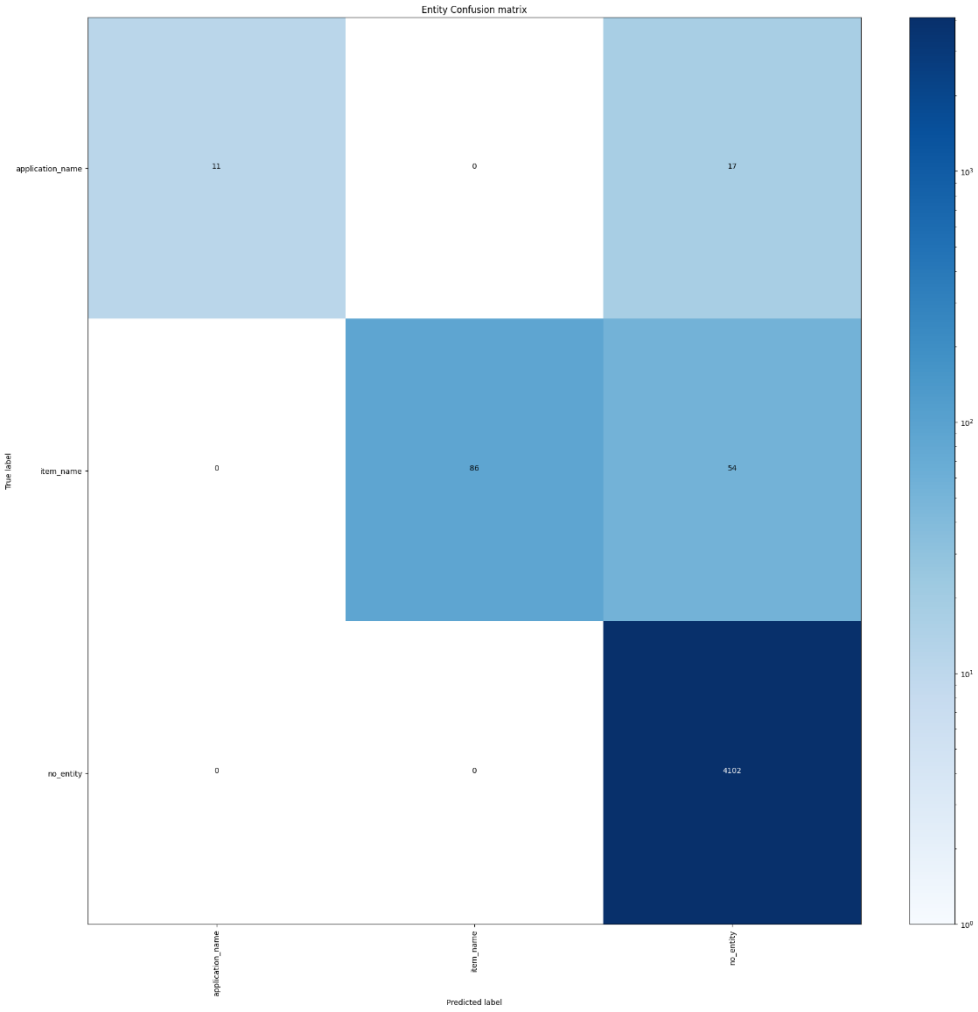
```
  - user: |
      What can you help with?
    intent: asking_capabilities
  - action: utter_give_capabilities

- story: ask about available items
  steps:
  - user: |
      Show what I can buy.
    intent: ask_available_items
  - action: action_list_available_items

- story: ask if specific item is available
  steps:
  - user: |
      Is  Intel Core i9 available?
    intent: ask_available_one_item
  - action: action_check_item_availability

- story: ask about discount on item
  steps:
  - user: |
      Is there a discount on Intel i5?
    intent: ask_discount_on_item
  - action: action_check_discount

- story: ask about warranty on item
  steps:
  - user: |
     Can you tell me the warranty  for Samsung 970 EVO SSD
    intent: ask_warranty_on_item
  - action: action_check_warranty

- story: ask for description of item
  steps:
  - user: |
      Describe the Logitech G502 mouse
    intent: ask_description_of_item
  - action: action_describe_item

- story: ask about working hours
  steps:
  - user: |
      What are your working hours?
    intent: asking_about_working_hours
  - action: utter_ask_working_hours

- story: ask about location
  steps:
  - user: |
      Where  I find you?
    intent: asking_about_location
  - action: utter_location

- story: ask about home delivery
  steps:
  - user: |
      Do you offer home delivery?
    intent: asking_home_delivery
  - action: utter_home_delivery
```

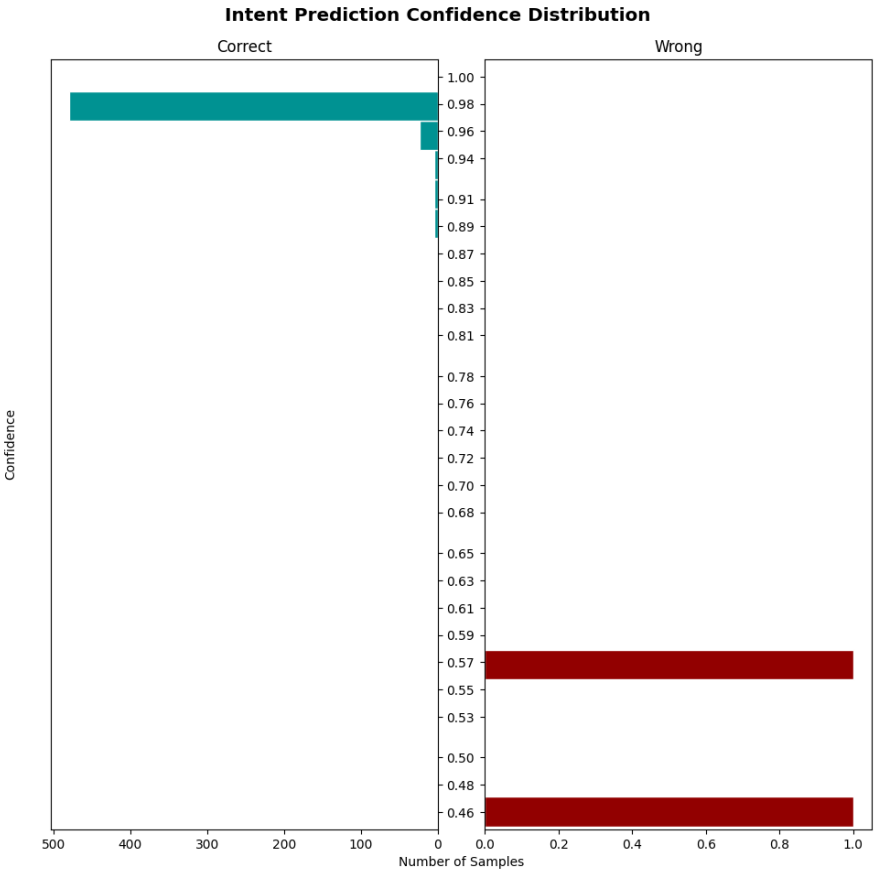
Customer chatbot intent test results:






Intent Confusion matrix
True label
Predicted label
affirm
ask_available_items
ask_available_one_item
ask_description_of_item
ask_discount_on_item
ask_item_pic
ask_repairs
ask_warranty_on_item
asking_about_bulk_discounts
asking_about_customer_support
asking_about_discounts
asking_about_exchange_product
asking_about_location
asking_about_payments
asking_about_return_policy
asking_about_shop
asking_about_working_hours
asking_capabilities
asking_help
asking_home_delivery
asking_how_to_claim_warranty
bot_challenge
deny
find_components
goodbye
greet
inquire_antivirus_software
inquire_best_brands
inquire_best_laptops_for_students
inquire_budget_computers
inquire_cctv_installation
inquire_cctv_operation
inquire_cctv_price
inquire_computer_types
inquire_custom_computer
inquire_custom_pc_build
inquire_custom_pc_build_time
inquire_desktop_for_graphic_design
inquire_gaming_laptops
inquire_gift_cards
inquire_laptop_accessories
inquire_order_delivery_time
inquire_privacy_policy
inquire_refurbished_computers
inquire_review_submission
inquire_shipping_cost
item_price
10^1
10^0

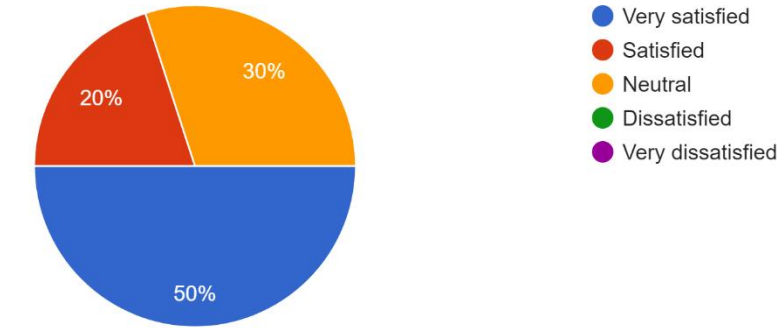

Entity Confusion matrix
application_name
item_name
no_entity
True label
11
0
17
0
86
54
0
0
4102
application_name
item_name
no_entity
Predicted label


Intent Prediction Confidence Distribution
Correct
Wrong
Confidence
1.00
0.98
0.96
0.94
0.91
0.89
0.87
0.85
0.83
0.81
0.78
0.76
0.74
0.72
0.70
0.68
0.65
0.63
0.61
0.59
0.57
0.55
0.53
0.50
0.48
0.46
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0
Number of Samples

Customer chatbot User testing Feedback form results:

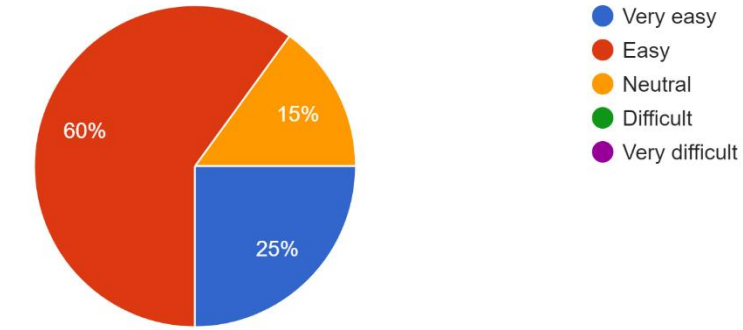
Overall, how satisfied are you with your experience with our chatbot?

20 responses



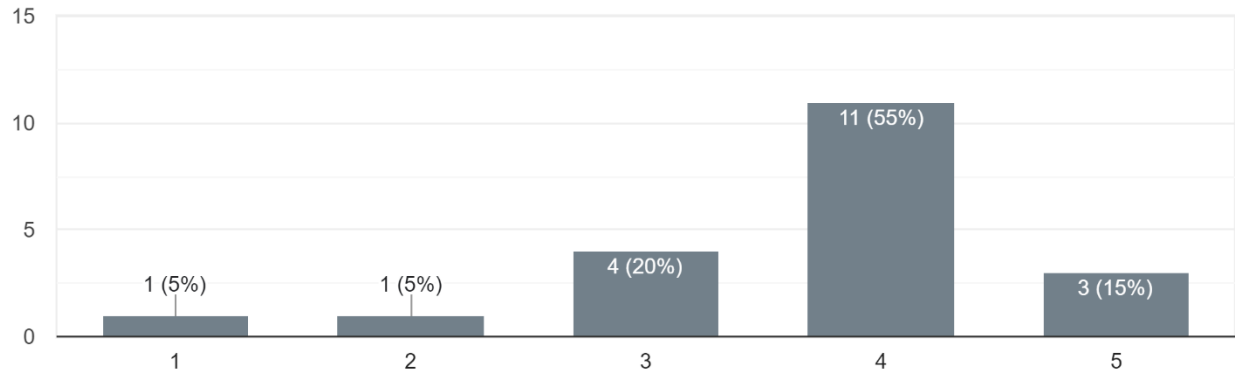
How easy was it to interact with the chatbot?

20 responses



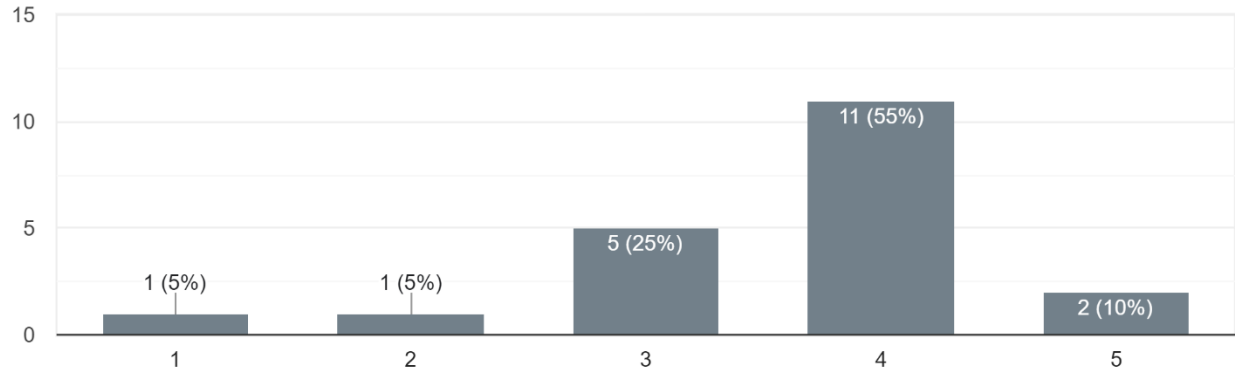
How visually appealing did you find the chatbot's interface?

20 responses



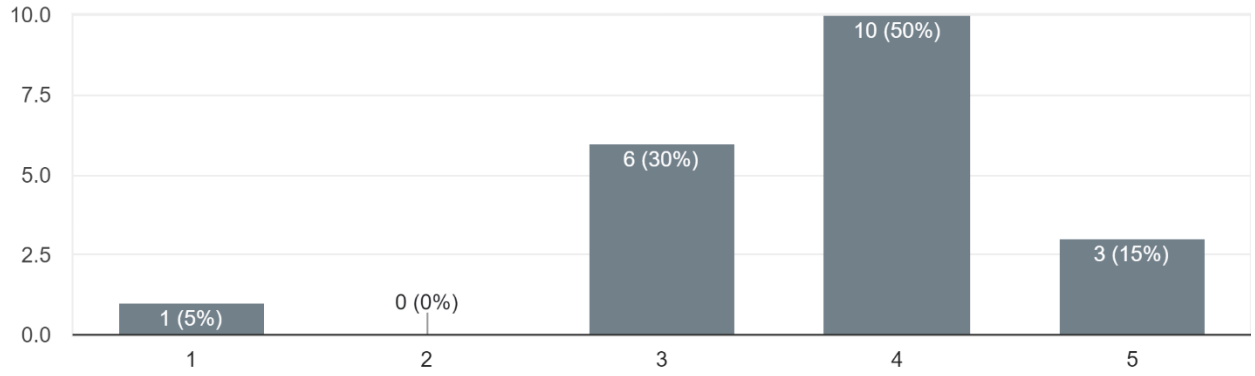
How clear and readable was the text on the chatbot's interface?

20 responses



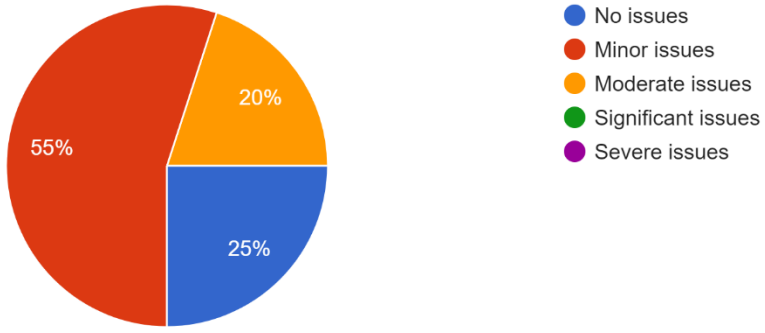
Was the chatbot's interface easy to navigate?

20 responses



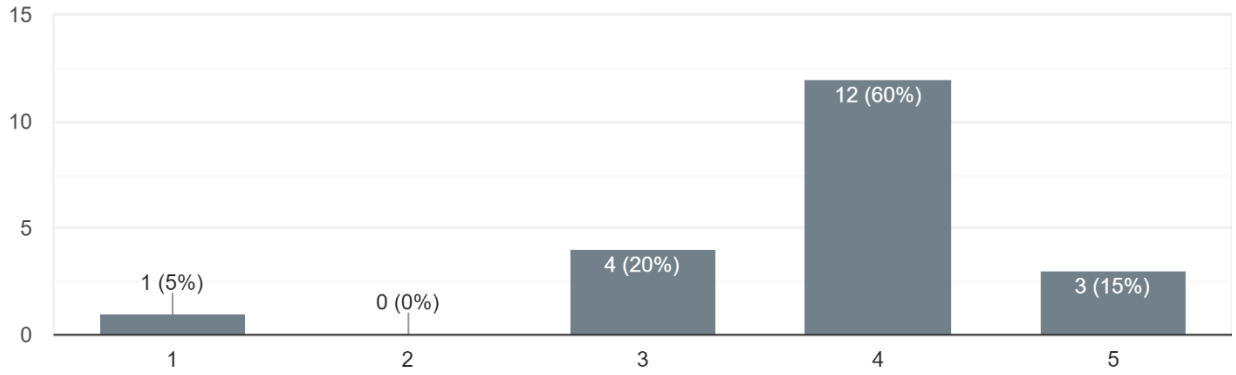
Did you experience any technical issues while interacting with the chatbot? (e.g., slow response time, errors)

20 responses



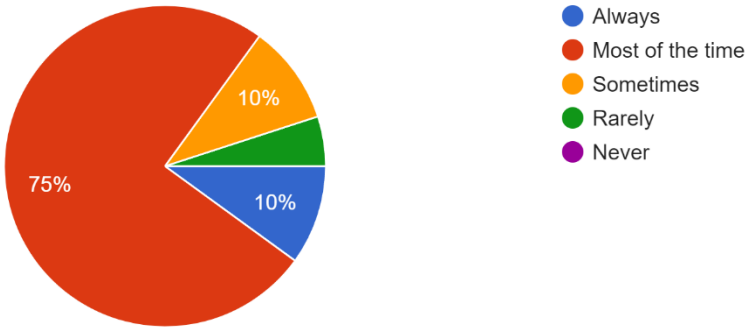
How satisfied are you with the speed of the chatbot's responses?

20 responses



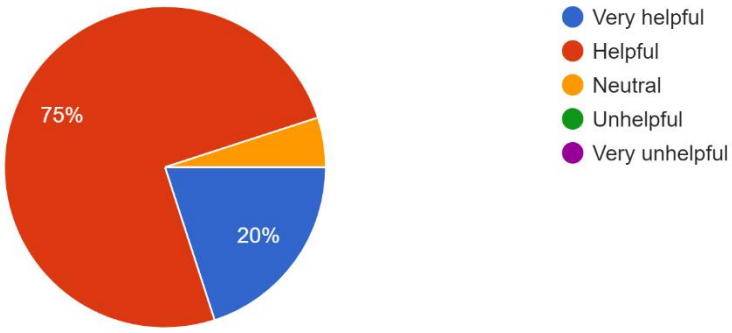
Did the chatbot understand your questions and provide relevant answers?

20 responses



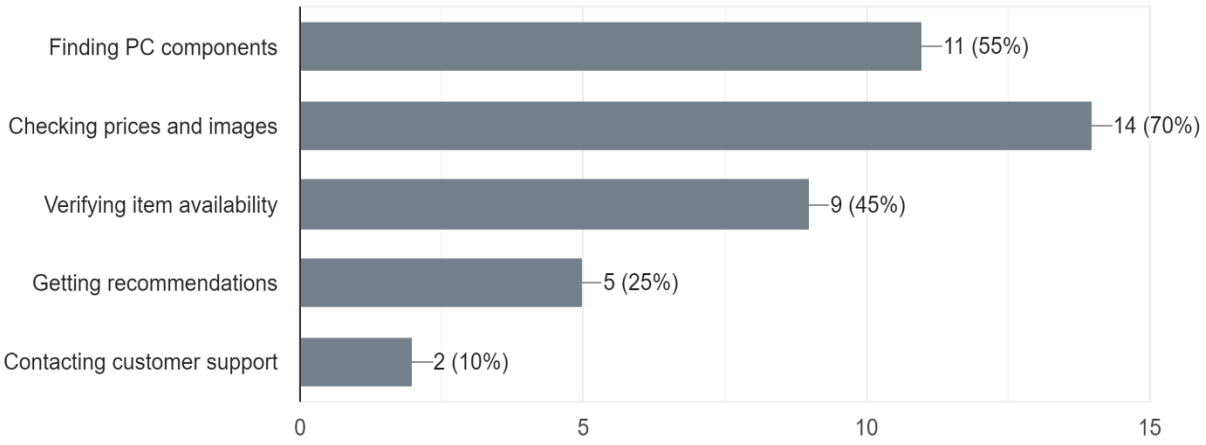
How helpful was the information provided by the chatbot?

20 responses



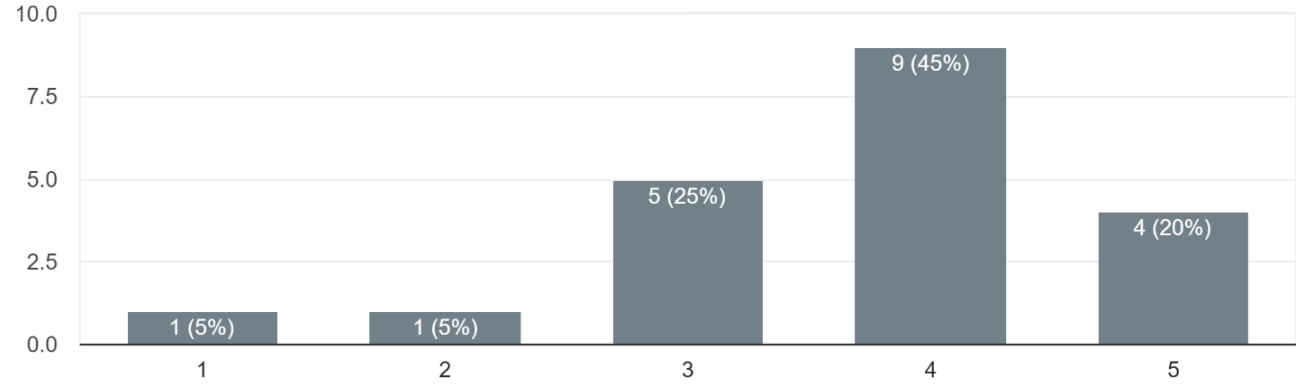
Which features did you find most useful?

20 responses



Did you find the chatbot's interface to be consistent across different devices (e.g., desktop, mobile)?

20 responses



Web interface responsive test

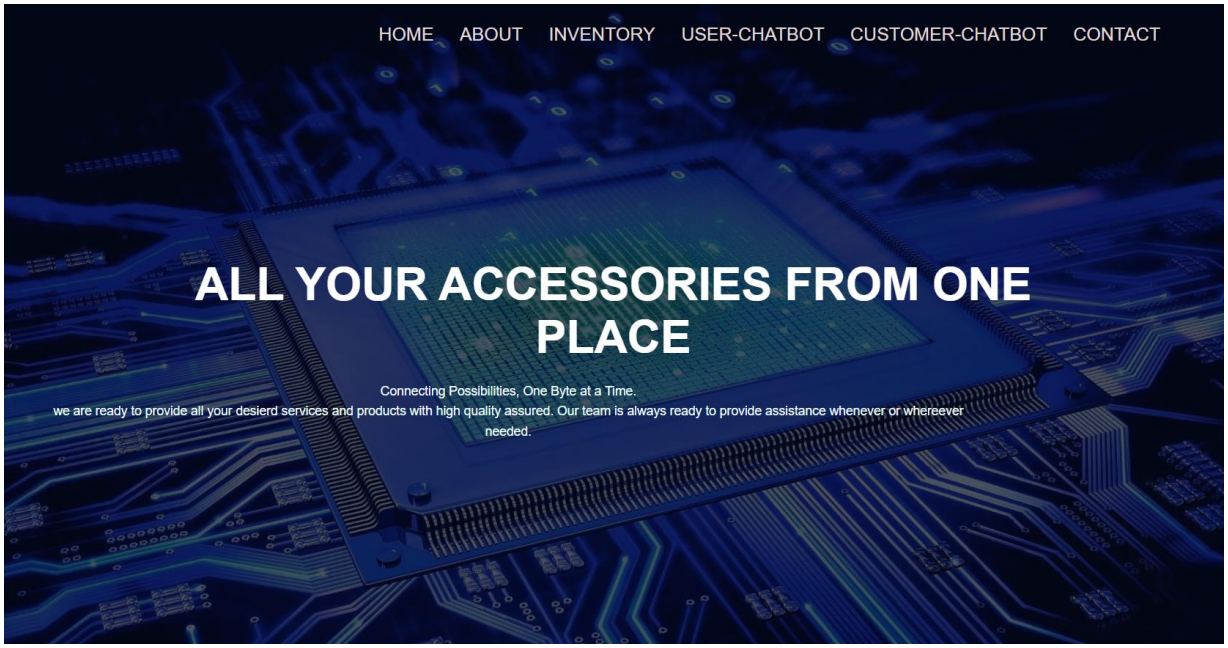


Figure 8: Web Interface Desktop View

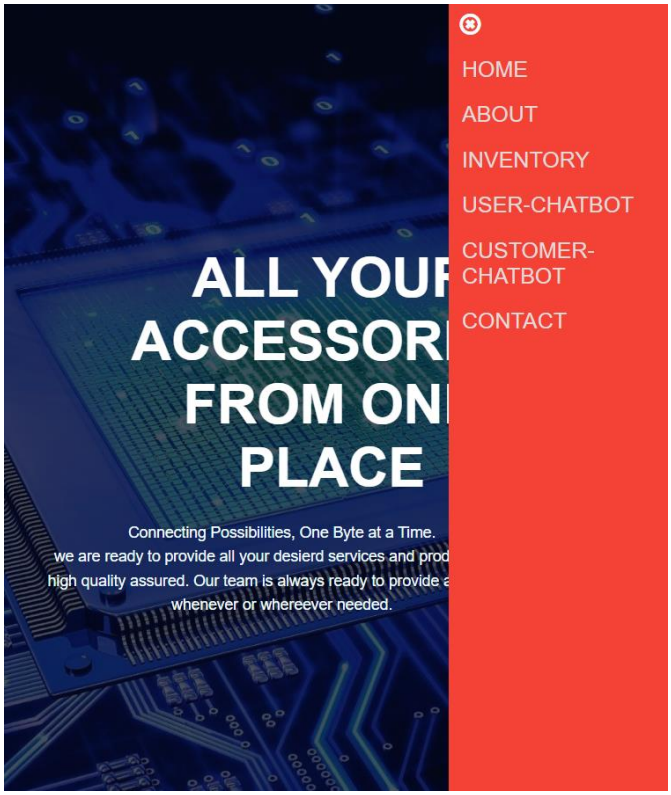


Figure 9 : Web Interface Mobile View

Customer chatbot user interface responsive test

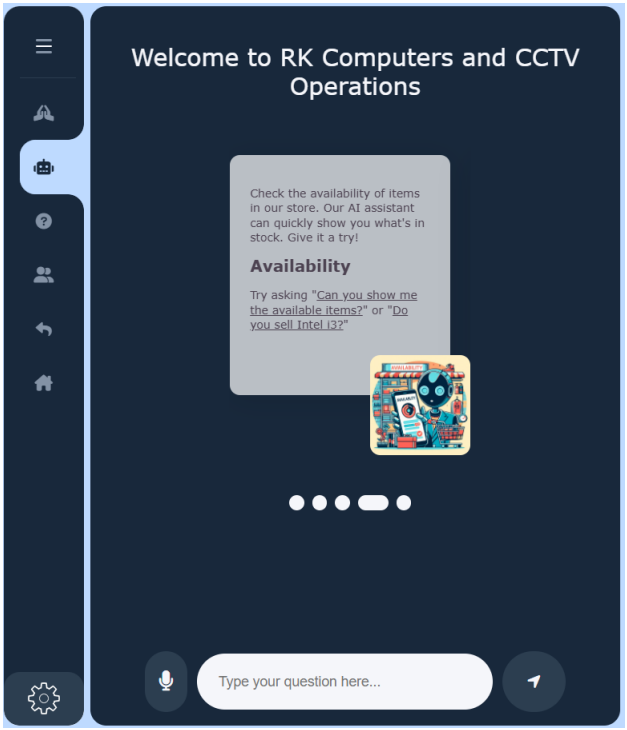


Figure 10: Customer Chatbot UI Mobile View

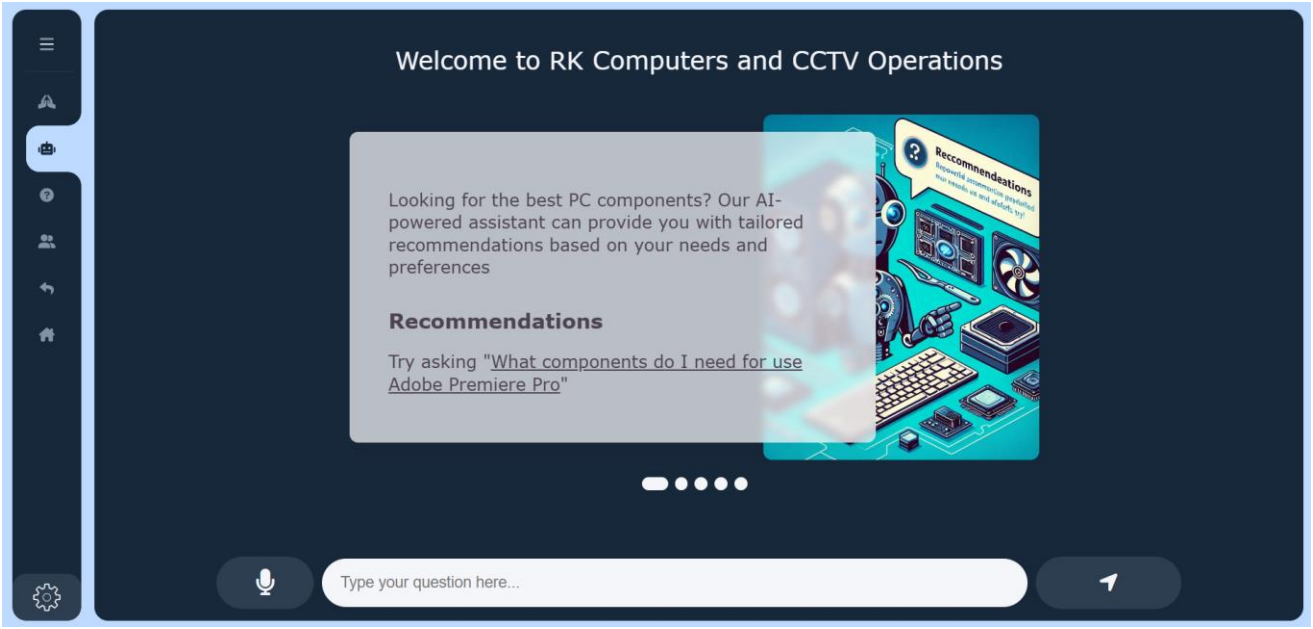


Figure 11: Customer Chatbot UI Desktop view