

Software Engineering Department

Ort Braude College

Capstone Project Phase A – 61998

**ShopLINK**

**Authors:**

Julia Shbat – Julia.Shbat@e.braude.ac.il

Daniel Vardimon– Daniel.Vardimon@e.braude.ac.il

**Supervisor:**

Dr. Zakharia Frenkel

**24-2-D-27**

**Table of Contents**

[1. INTRODUCTION 5](#_Toc124978537)

[2. RELATED WORKS 6](#_Toc124978540)

[2.1. Zoko 6](#_Toc124978541)

[2.2. Zixflow 6](#_Toc124978542)

[2.3. Interakt 6](#_Toc124978543)

[3. TECHNOLOGY REVIEW 7](#_Toc124978544)

[3.1. Whapi 7](#_Toc124978545)

[3.2. Twilio 7](#_Toc124978546)

[3.3. Tax Authority 7](#_Toc124978547)

[3.4. MeaningCloud Text Classification 7](#_Toc124978547)

[4. SYSTEM FUNCTIONALITIES 8](#_Toc124978548)

[Table 4.1. FUNCTIONALITIES 8](#_Toc124978545)

[Table 4.2. Non-FUNCTIONALITIES 9](#_Toc124978546)

[5. TECHNOLOGY CHOICES 10](#_Toc124978548)

[5.1 BACK-END NODEJS 10](#_Toc124978549)

[5.1.1. ADVANTAGES OF NODE.JS 10](#_Toc124978550)

[5.1.2. DISADVANTAGES OF NODE.JS 11](#_Toc124978551)

[5.2. BACK-END MONGODB 11](#_Toc124978552)

[5.2.1 ADVANTAGES OF MONGODB 11](#_Toc124978553)

[5.2.2 DISADVANTAGES OF MONGODB 11](#_Toc124978553)

[5.3. FRONT-END REACT-JS 12](#_Toc124978554)

[5.3.1 ADVANTAGES OF REACT-JS 12](#_Toc124978555)

[5.3.2 DISADVANTAGES OF REACT-JS 12](#_Toc124978556)

[5.4 VISUAL STUDIO CODE FRAMEWORK 12](#_Toc124978557)

[5.4.1 FEATURES 12](#_Toc124978558)

[5.4.2 ADVANTAGES OF VISUAL STUDIO CODE 13](#_Toc124978555)

[5.4.3 DISADVANTAGES OF VISUAL STUDIO CODE 13](#_Toc124978556)

[5.5 EXPRESS-JS 13](#_Toc124978557)

[6. TECHNOLOGY CHALLENGES 14](#_Toc124978561)

[6.1. USER INTERFACE AND USER EXPERIENCE 14](#_Toc124978562)

[6.2. SCALABILITY 14](#_Toc124978563)

[6.3. PERFORMANCE 14](#_Toc124978564)

[6.4. SECURITY 1](#_Toc124978565)5

[7. PROJECT STRUCTURE 15](#_Toc124978566)

[7.1. MERN Stack 15](#_Toc124978567)

[8. Our Solution 16](#_Toc124978569)

[8.1 AUTOMATED RESPONSE 16](#_Toc124978570)

[8.1.1 AUTOMATED RESPONSE PROBLEM 16](#_Toc124978571)

[8.1.2 AUTOMATED RESPONSE SOLUTION 16](#_Toc124978572)

[8.2 CUSTOMER SEGMENTATION 17](#_Toc124978573)

[8.2.1 CUSTOMER SEGMENTATION PROBLEM 17](#_Toc124978574)

[8.2.2 CUSTOMER SEGMENTATION SOLUTION 17](#_Toc124978575)

[8.3 PRODUCT CATALOG MANAGEMENT 18](#_Toc124978579)

[8.3.1 PRODUCT CATALOG MANAGEMENT PROBLEM 18](#_Toc124978580)

[8.3.2 PRODUCT CATALOG MANAGEMENT SOLUTION 18](#_Toc124978581)

[8.4 TAX COMPLIANCE 20](#_Toc124978579)

[8.3.1 TAX COMPLIANCE PROBLEM 20](#_Toc124978580)

[8.4.2 TAX COMPLIANCE SOLUTION 20](#_Toc124978581)

[9. SOFTWARE ENGINEERING DOCUMENTS 21](#_Toc124978583)

[9.1 DIAGRAMS 21](#_Toc124978584)

[9.1.1 Package diagram 21](#_Toc124978585)

[9.1.2 FlowChart diagram 22](#_Toc124978586)

[9.2 GUI 23](#_Toc124978594)

[9.2.1 LOGIN 23](#_Toc124978595)

[9.2.2 MANAGER SCREENS 23](#_Toc124978596)

[9.2.2.1 MESSAGES SCREEN 23](#_Toc124978597)

[9.2.2.2 ANALYTICS SCREEN 23](#_Toc124978598)

[9.2.2.3 CUSTOMER MANAGEMENT 25](#_Toc124978599)

[9.2.2.4 PRODUCT CATALOG 26](#_Toc124978600)

[9.2.2.5 BROADCAST MESSAGES 26](#_Toc124978601)

[10. TEST PLANS 28](#_Toc124978625)

[10.1. LOGIN TESTS 28](#_Toc124978626)

[10.2. MANAGER TESTS 28](#_Toc124978627)

[11. REFERENCES 31](#_Toc124978630)

**Abstract.** *With recent increases in online purchases, many businesses allow customers to contact them through WhatsApp for support and marketing. This creates a big burden on the business and may overload the business' WhatsApp, making it difficult to keep up with the customers. The business owners can have difficulties seeing the customers messages on time, and this may cause problems with communication and availability for the customers, and may compromise the service quality of the business. Along with these difficulties, there is a need for methodically sending the correct advertisements to the corresponding customers according to their interests. There are some solutions that exist, but those don't exactly solve all of the problems, especially they are not able to automatically select groups of users, for example, for sending some marketing information.*

*In our project we are going to develop a new product with a standard set of functionalities as well as involving the clustering and the topic extraction algorithms for proceeding of the chats with users for creating different groups of users.*

*The standard functionalities include a comprehensive dashboard for managing customer interactions, automated message responses to ensure timely communication, detailed analytics to monitor engagement, and tools for organizing customer conversations using labels and tags.*

**1. INTRODUCTION**

With the surge in online shopping, businesses increasingly use platforms like WhatsApp for customer interactions, leveraging its convenience and popularity. This shift, however, presents significant challenges. The sheer volume of messages can overwhelm businesses, causing delays in response times and leading to potential dissatisfaction among customers. Effective communication is crucial for maintaining service quality, and any delay can negatively impact the customer experience.

Additionally, businesses face the challenge of targeting the right customers with appropriate advertisements. As customer bases grow, it becomes increasingly difficult to manually manage and personalize marketing efforts. There is a need for an efficient, automated solution that not only manages customer communications but also intelligently clusters customers based on their interests for targeted marketing.

Currently, various tools and platforms provide basic solutions for managing customer interactions and marketing via WhatsApp. These tools often include simple automated responses and basic customer segmentation features. However, they typically lack advanced capabilities for grouping customers based on detailed interest analysis. Manual processes are still prevalent, which can lead to inefficiencies and missed opportunities in personalizing customer engagement.

Our proposed application addresses these limitations by integrating advanced communication methods and data analysis techniques. Utilizing a chatbot, the system interacts with customers, gathering data on their interests through natural conversations. This information is stored in a database, where it undergoes text analysis using keywords and deep learning algorithms. These algorithms cluster customers into groups based on shared interests, enabling highly personalized marketing efforts.

By automating the clustering process and tailoring advertisements to specific customer groups, businesses can improve response times and enhance the relevance of their marketing messages. This leads to a more streamlined communication process and boosts overall customer satisfaction.

The primary stakeholders in this solution are business owners and their customers. Business owners gain a powerful tool to manage customer interactions more efficiently, reducing the manual workload and improving marketing precision. This not only enhances operational efficiency but also positively impacts the business's bottom line by increasing customer engagement and retention.

Customers, on the other hand, benefit from more timely responses and receive advertisements tailored to their specific interests, leading to a more satisfying and personalized shopping experience. This improved interaction fosters customer loyalty and strengthens the relationship between businesses and their clientele.

**2. RELATED WORKS**

The WhatsApp Business API enables businesses to send bulk messages and connect multiple devices to a single WhatsApp Business account, a feature not available with the standard WhatsApp Business App. This capability is essential, as WhatsApp statistics indicate that approximately 175 million users contact companies via WhatsApp daily.

**2.1. Zoko**

Zoko is a marketing application that allows selling products over WhatsApp. It offers features such as broadcasting messages, automated responses, and sales tracking, which help businesses streamline their communication and marketing efforts.The application also offers tools for making a product catalog on WhatsApp, a chat screen and an AI chatbot for communicating with customers, as well as tools for analytics to help sellers and business owners track and monitor their sales. Zoko includes integration with a few other tools for marketing such as FlowHippo and Shopify, adding the benefits of such tools to the app.

**2.2. Zixflow**

Zixflow is a powerful platform designed to streamline document review and approval processes for businesses. It offers features such as automated workflows, real-time collaboration, and detailed tracking of document statuses, which enhance efficiency and reduce manual tasks. Zixflow's robust security measures ensure that sensitive documents are protected, and its comprehensive reporting tools provide valuable insights into workflow performance. It allows users to segment their audience, add tags and create custom contact lists for WhatsApp broadcasts, and offers a centralized inbox and automated messaging, helping the users with customer support.

**2.3. Interakt**

Interakt is a versatile app designed to enhance customer engagement through seamless integration with WhatsApp. It offers features such as automated workflows and real-time messaging, which streamline communication and improve efficiency. Interakt's advanced security protocols ensure that customer data is protected, and its robust analytics tools provide valuable insights into user interactions. Additionally, it allows businesses to set up WhatsApp chatbots with Chat GPT with automated replies and automated tasks, run WhatsApp ads and AI generated ads with auto target, and offers CRM and shared inbox for improved customer support. Interakt integrates with many platforms including Shopify, Google Sheets and more.

**3. TECHNOLOGY REVIEW**

**3.1. Whapi**

API Reference: https://whapi.readme.io/reference/getproducts

Price: Free (Limited Sandbox Plan), 11$/month (Developer Plan)

The Whapi API provides a versatile set of tools for managing and interacting with product data within a business environment. In our app, we will use this API to retrieve, create, update, and delete product information, manage products by Contact ID, and send catalogs directly to customers. This integration will allow us to deliver accurate product details, automate product-related inquiries, and enhance the overall customer experience by ensuring seamless access to up-to-date product information.

**3.2. Twilio**

API Reference: https://www.twilio.com/docs/conversations

Price: $0.0053/Conversation, $0.005/Message

The Twilio Conversations API facilitates the management of multi-channel messaging, allowing businesses to organize and automate customer interactions across platforms like WhatsApp. In our app, we will use this API to automate responses, manage conversations, assign participant roles, and gather message insights.

**3.3. Tax Authority**

API Reference: https://govextra.gov.il/taxes/innovation/home/api/

Price: Free (Government API)

The Tax Authority API provides access to tax-related data and services, allowing businesses to interact with the tax system programmatically. In our app, we will use this API to facilitate customer transactions, ensuring compliance with local tax regulations and streamlining the financial process.

**3.4. MeaningCloud Text Classification**

API Reference: https://learn.meaningcloud.com/developer/text-classification/2.0/doc

Price: 25,000 words/month (Hebrew), 50,000 words/month (English)

The MeaningCloud Text Classification API categorizes text into predefined categories based on its content, enabling automated tagging and organization of text data. In our app, we will use this API to classify customer messages into specific topics or categories, which will help us tailor responses, organize conversations, and target users with relevant marketing content.

**4.  SYSTEM FUNCTIONALITIES**

Our system provides a number of features that are designed to help businesses track and manage their communication with customers. From a centralized space for responding to messages to detailed insights and sales metrics, the system offers vast capabilities for streamlining the business marketing process. In addition, the app contains a tag system, allowing the user to quickly and efficiently search for customer messages by tag and helps the user effortlessly navigate through thousands of WhatsApp messages.

**Table 4.1. FUNCTIONALITIES**

|  |  |
| --- | --- |
| **Functionality** | **Description** |
| **Login** | Allows employers to connect to ShopLINK app. |
| **Connecting to WhatsApp** | Allows employers to connect to the WhatsApp of the business. |
| **Tag System** | Gives the customer messages tags and allows the user to sort or filter by tag |
| **Sales Insights** | Shows sales data related to different categories, allowing to sort and filter based on category |
| **Customer Chat** | Shows the chat history with all customers, allowing for manual viewing and responding to messages from customers. |
| **Product Catalog** | Lays out the products available for sale along with prices, highlights recommended products for the user based on their interests. |
| **Broadcast** | Send broadcast messages on WhatsApp to many customers at once |

|  |  |
| --- | --- |
| **Non-Functionality** | **Description** |
| **Interface Usability** | The system features an intuitive and easy-to-use graphical interface. |
| **Sales Data Real-Time Accuracy** | The system ensures accurate and real-time sales data for insights. |
| **Data Privacy and Security Compliance** | The system is secure, ensuring data privacy and protection, especially for customer conversations. |
| **Multi-Language Compatibility** | The system supports multi-language capabilities to cater to a global audience. |
| **Speed** | The system is optimized for fast performance. |

**Table 4.2. Non-FUNCTIONALITIES**

**5. TECHNOLOGY CHOICES**

**5.1 BACK-END NODEJS**

Node.js is an open-source, cross-platform environment that allows JavaScript to run server-side. It's a popular choice for a wide range of projects! Node.js runs outside of the browser using the V8 JavaScript engine, which powers Google Chrome. This gives Node.js a significant performance edge.

A Node.js application runs in a single process without creating a new thread for each request. Most Node.js libraries are built using non-blocking patterns, and its standard library includes asynchronous I/O primitives that prevent JavaScript code from blocking. This means Node.js doesn't wait for I/O operations, like reading from the network or a database, to complete before continuing its execution. Instead, it resumes once the response is received.

This approach simplifies the complexity of managing thread concurrency, which can be a major source of bugs, allowing Node.js to efficiently handle thousands of simultaneous connections with just one server. Additionally, Node.js is advantageous for the millions of frontend developers who are familiar with JavaScript, enabling them to write both server-side and client-side code without having to learn a new language.

Now, let's examine the key advantages and disadvantages of using Node.js for server-side development.

#### 5.1.1. ADVANTAGES OF NODE.JS

1. High performance for real-time applications.
2. Scalable solutions for modern applications.
3. Cost-effective with full-stack JavaScript.
4. Strong community support to streamline development.
5. Simple to learn and adapt quickly.
6. Facilitates the creation of cross-functional teams.
7. Enhances application response time and overall performance.
8. Reduces time-to-market for your applications.
9. Extensibility to meet custom requirements.
10. Quick caching reduces loading times.
11. Supports the development of cross-platform applications.

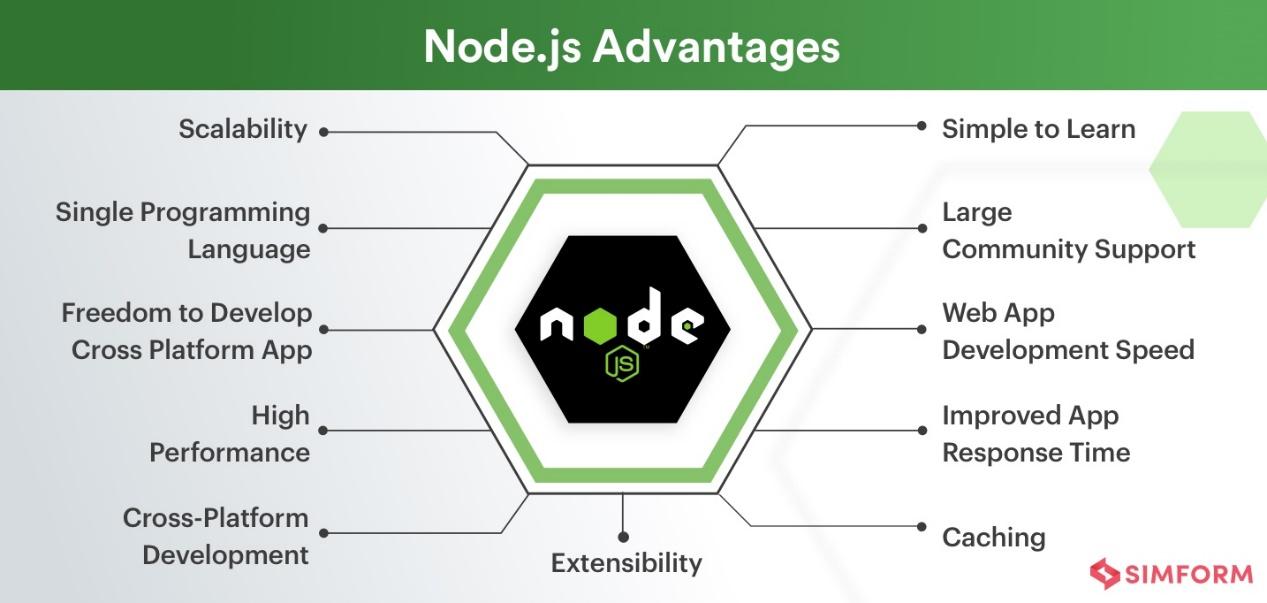


Figure 1 Node.js Advantages*, Reference 1*

#### 

#### 5.1.2. DISADVANTAGES OF NODE.JS

1. Reduced performance when handling heavy computational tasks.
2. Unstable API can lead to frequent code changes.
3. Asynchronous programming in Node.js can make code maintenance challenging.
4. Insufficient library support can pose risks to your code—choose wisely.
5. High demand for Node.js developers with relatively few experienced professionals.

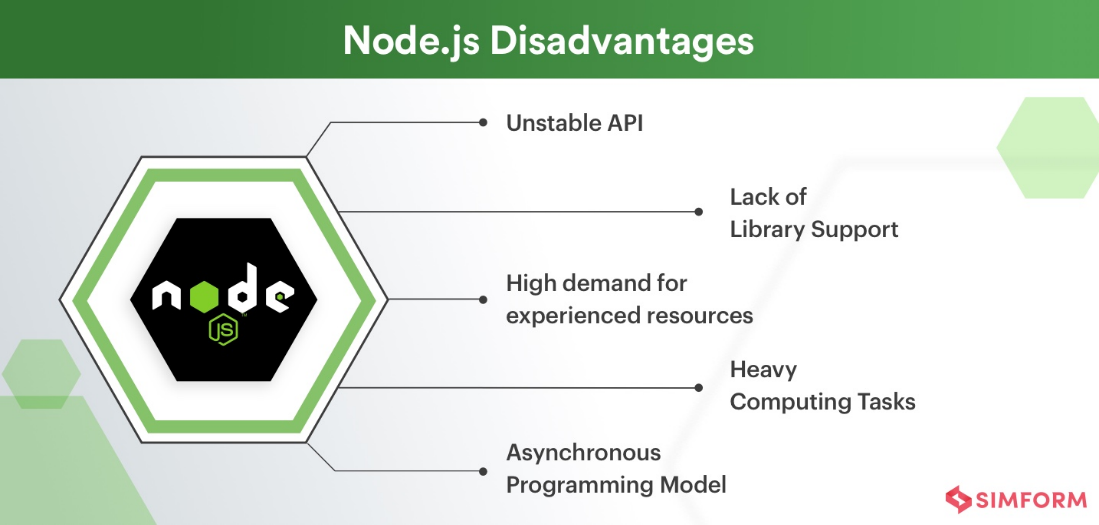


Figure 2 Node.js Disadvantages*, Reference 1*

תמונה שמכילה גופן, לוגו, גרפיקה, עיצוב

התיאור נוצר באופן אוטומטי

**5.2. BACK-END MONGODB**

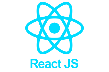
MongoDB is an open-source, cross-platform, document-oriented database. As a NoSQL database, MongoDB stores data in JSON-like documents, which may or may not adhere to a predefined schema. The database is developed by MongoDB Inc. under the Server-Side Public License (SSPL), which some distributions consider non-free.

**5.2.1 ADVANTAGES OF MONGODB**

1. Schema-less − MongoDB is a document database where a single collection can contain various documents, each with different fields, content, and sizes.
2. Clear structure of individual objects.
3. No need for complex joins.
4. Powerful querying capabilities. MongoDB supports dynamic queries on documents through a document-based query language that is nearly as robust as SQL.

#### 5.2.2 DISADVANTAGES OF MONGODB

1. Limited support for complex transactions − MongoDB doesn't naturally support multi-document transactions, which can be a limitation for applications requiring complex operations across multiple documents.
2. High memory usage − MongoDB's approach to data storage and indexing can lead to higher memory consumption compared to relational databases.

**5.3. FRONT-END REACT-JS **

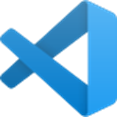
React.js is a popular JavaScript library used for building interactive and dynamic user interfaces, particularly single-page applications. It focuses on creating reusable UI components, which make development more efficient and maintainable. React's virtual DOM ensures high performance by minimizing direct interactions with the real DOM, leading to faster updates and rendering. It also enables seamless integration with other libraries or frameworks, making it highly versatile for front-end development.

**5.3.1 ADVANTAGES OF REACT-JS**

1. Component-Based Architecture: React promotes reusability by breaking the UI into modular components, which simplifies maintenance and scalability.
2. Virtual DOM: React’s virtual DOM improves performance by efficiently updating and rendering only the components that change, rather than the entire page.
3. Strong Community and Ecosystem: With a large developer community, extensive documentation, and plenty of third-party libraries, React has strong support for building complex applications.

**5.3.2 DISADVANTAGES OF REACT-JS**

1. Learning Curve: React has a steeper learning curve, especially for new developers, due to the concepts like JSX, component lifecycle, and state management.
2. Overhead for Small Applications: For small, simple projects, React can be overkill, as its advanced features and architecture might not be fully utilized.
3. State Management Complexity: As applications grow in size, managing the state across multiple components can become complex, often necessitating additional libraries like Redux or Context API for effective state management.

**5.4 VISUAL STUDIO CODE FRAMEWORK**

Visual Studio Code (VS Code) is a source-code editor developed by Microsoft, built on the Electron framework, and available for Windows, Linux, and macOS. It offers a range of features such as debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and integrated Git support.

**5.4.1 FEATURES**

Visual Studio Code is a versatile source-code editor compatible with various programming languages, including C#, Java, JavaScript, Go, Node.js, Python, C++, C, Rust, and Fortran. It is built on the Electron framework, which is commonly used for developing Node.js web applications running on the Blink layout engine. The editor component, codenamed "Monaco," is the same as the one used in Azure DevOps (formerly Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code provides basic support for most popular programming languages. This support includes features like syntax highlighting, bracket matching, code folding, and customizable snippets. Additionally, VS Code comes with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, and offers debugging capabilities for Node.js. Developers can further extend its functionality by installing extensions available on the VS Code Marketplace, which add support for additional languages and tools.

#### 5.4.2 ADVANTAGES OF VISUAL STUDIO CODE

1. Cross-platform availability − VS Code runs on Windows, Linux, and macOS, providing a consistent development environment across different operating systems.
2. Extensibility − The rich ecosystem of extensions available on the VS Code Marketplace allows developers to customize and enhance their coding environment to fit their specific needs.
3. Integrated Git support − VS Code includes built-in Git support, allowing for seamless version control directly within the editor.
4. IntelliSense − Provides intelligent code completion, parameter info, and quick info for supported languages, improving developer productivity.
5. Lightweight and fast − Despite its powerful features, VS Code is known for being lightweight and responsive, making it suitable for both small and large projects.
6. Active community and frequent updates − With an active community and regular updates from Microsoft, VS Code continuously evolves with new features and improvements.

#### 5.4.3 DISADVANTAGES OF VISUAL STUDIO CODE

1. Resource consumption − As an Electron-based application, VS Code can be more resource-intensive compared to native editors, which may affect performance on lower-end systems.
2. Limited built-in features − While VS Code is feature-rich, some advanced features require additional extensions, which may not always integrate seamlessly or perform optimally.

**5.5 EXPRESS-JS**

Express.js is a lightweight and flexible web application framework for Node.js that simplifies the process of building web servers and APIs. It provides a set of robust features for handling HTTP requests, middleware, routing, and creating RESTful services. Express is unopinionated, meaning it doesn’t enforce any specific project structure, giving developers the freedom to organize code as they see fit. Its simplicity, combined with the vast ecosystem of Node.js libraries, makes Express a popular choice for building scalable, high-performance web applications.

**6. TECHNOLOGY CHALLENGES**

As we develop ShopLINK, we face several technology challenges that must be addressed to ensure the platform's success. The first challenge we face in a competitive market is delivering services that match or exceed the current standards of speed and quality offered by similar products. Ensuring that our platform performs at least as efficiently as established alternatives is crucial for user retention and satisfaction. Additionally, we are introducing a new feature related to text clustering and topic extraction, which presents its own set of challenges. Adapting existing algorithms to our specific problem domain and the available data is a complex task that requires careful consideration of accuracy, scalability, and performance. Another important aspect is the design of a user-friendly graphical interface (GUI) for this new service. The interface needs to strike a balance between simplicity, making it easy to use, and flexibility, accommodating various user requirements effectively.

**6.1. USER INTERFACE AND USER EXPERIENCE**

In today's competitive market, web applications must be customer-friendly and responsive on various screen sizes, particularly with the widespread use of smartphones. At ShopLINK , we recognize the importance of user experience, which is why we have chosen React Native for our front-end development. React Native enables us to update and re-render only the specific component that has changed, rather than the entire page, improving both efficiency and user experience. Additionally, we are utilizing Material-UI components and styles to streamline development further and save time. Our back-end is powered by Node.js, with data stored in MongoDB, providing a flexible and scalable solution tailored to our needs.

**6.2. SCALABILITY**

As ShopLINK grows and attracts more users, it is crucial to ensure the platform can handle increased demand without suffering from performance issues or downtime. Scalability refers to a system's ability to manage rising traffic and workloads without a decline in performance. One way to enhance scalability is by using a distributed database system like MongoDB, which allows horizontal scaling across multiple servers.

By considering scalability from the beginning and implementing appropriate strategies, we can ensure ShopLINK can accommodate our expanding user base while maintaining a high-quality experience for all users.

**6.3. PERFORMANCE**

Performance is a critical aspect of any application, as it directly impacts how quickly and smoothly the application operates, influencing user satisfaction. We can enhance ShopLINK's performance by optimizing our code to reduce resource usage, implementing caching to decrease server load, and optimizing database queries for greater efficiency.

Additionally, utilizing a content delivery network (CDN) to serve static assets—such as images and CSS files—closer to our users can significantly improve loading times. By prioritizing performance from the outset and continuously monitoring and testing, we can ensure ShopLINK delivers a fast and seamless experience.

**6.4. SECURITY**

Securing our application is essential to protect user data and maintain the platform's integrity and reliability. Several measures can be taken to ensure ShopLINK's security, including implementing robust authentication protocols, using encryption to safeguard sensitive data, and regularly updating security protocols and software to address new threats. We should also consider incorporating security testing and monitoring processes to identify and mitigate potential vulnerabilities.

By prioritizing security from the start and taking proactive steps to protect the platform, we can ensure ShopLINK remains a trusted and secure application for our users.

**7. PROJECT STRUCTURE**

As technology evolves rapidly, user expectations and standards are continuously shifting. To keep up, developers strive to create applications that are more responsive, interactive, and enjoyable. A key approach to achieving this is through the use of technology stacks—carefully selected sets of functions and technologies designed to work seamlessly together. By leveraging advanced technology stacks, developers can ensure their applications are reliable, efficient, and capable of meeting the high demands of modern users.

**7.1. MERN Stack**

The MERN stack is a powerful combination of four technologies: MongoDB, Express.js, React.js, and Node.js. This stack allows developers to build a full-stack application with a cohesive 3-tier architecture (frontend, backend, database) using only JavaScript and JSON. By utilizing the MERN stack, developers can create robust and efficient applications that cater to a variety of needs and handle complex tasks effectively.

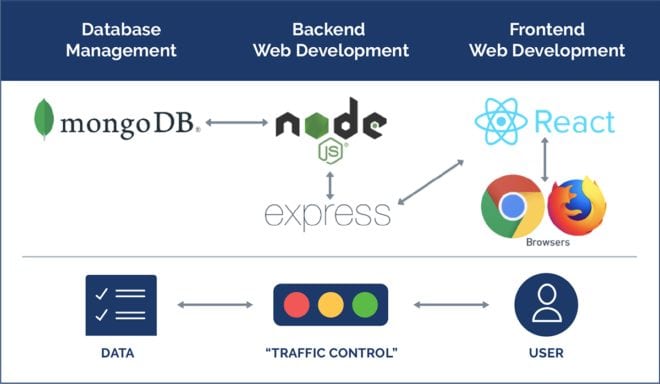


Figure 3 How does MERN work*, Reference 1**0*

**8. Our Solution**

**8.1 AUTOMATED RESPONSE**

**8.1.1 AUTOMATED RESPONSE PROBLEM**

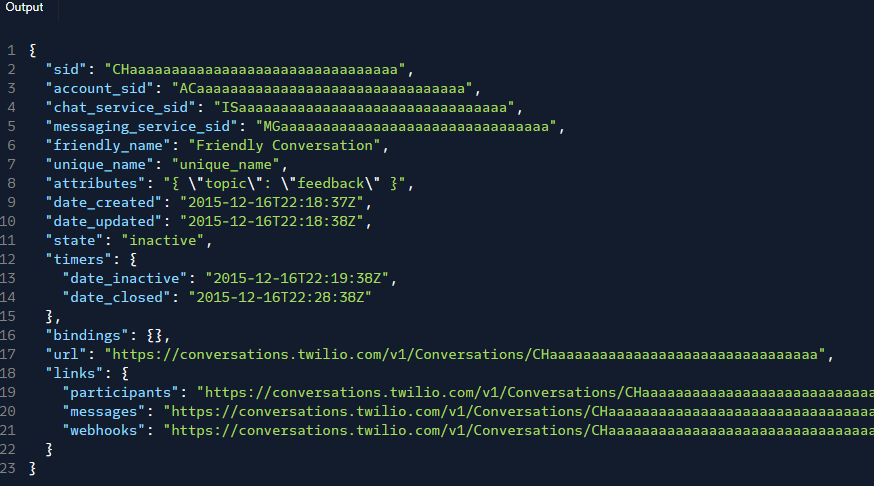
Handling a high volume of incoming messages efficiently is challenging, especially when dealing with repetitive inquiries.

**8.1.2 AUTOMATED RESPONSE SOLUTION**

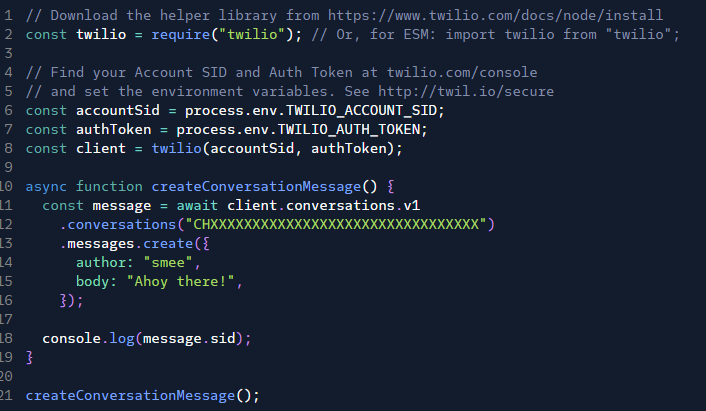
The application uses a chatbot, integrated with the Twilio Conversations API, to automatically manage and respond to customer messages on WhatsApp. The chatbot leverages natural language processing to analyze message content and provides either predefined or dynamically generated responses, ensuring timely and accurate replies to customer inquiries.

To create a conversation we use this code:  


The response (example):



To create a message:



To list all messages in a conversation:



**8.2 CUSTOMER SEGMENTATION**

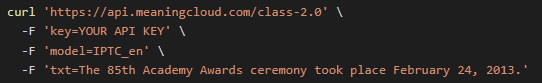
**8.2.1 CUSTOMER SEGMENTATION PROBLEM**

Automatically segmenting users based on their interactions to deliver personalized content is complex and not feasible manually due to the large data volume.

**8.2.2 CUSTOMER SEGMENTATION SOLUTION**

The application employs a Clustering Algorithm enhanced by the MeaningCloud Text Classification API. This API categorizes customer messages, allowing the algorithm to group users into segments based on their behavior and interests. These segments are then used to tailor marketing strategies and deliver targeted content to users.

Example:





**8.3  PRODUCT CATALOG MANAGEMENT**

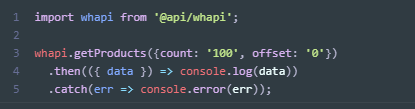
**8.3.1 PRODUCT CATALOG MANAGEMENT PROBLEM**

Maintaining an up-to-date and accurate product catalog is essential for providing customers with relevant product information.

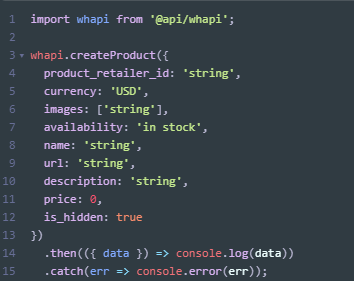
**8.3.2 PRODUCT CATALOG MANAGEMENT SOLUTION**

The application integrates the Whapi API to manage the product catalog, enabling the creation, retrieval, updating, and deletion of product data. This ensures that the product information is always current and can be efficiently shared with customers, particularly through WhatsApp.

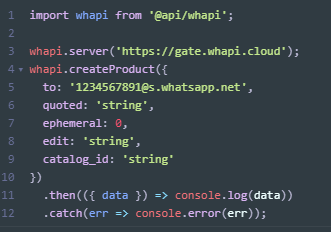
In order to get products from the catalog we can use the following API request:

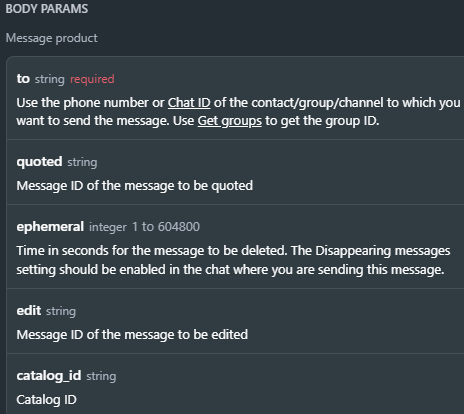


To create a new product:

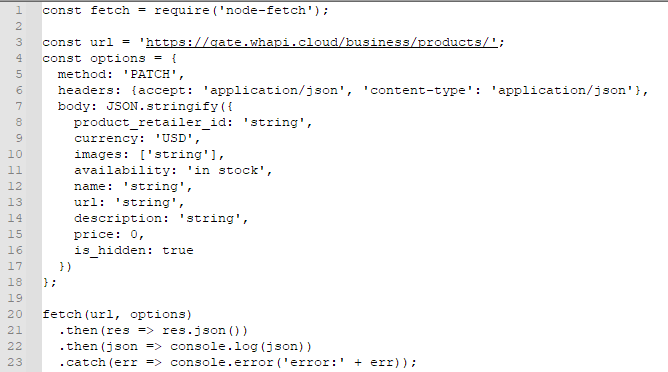


To send a product to a specific WhatsApp chat:





To update a product:



To delete a product:



**8.4  TAX COMPLIANCE**

**8.4.1 TAX COMPLIANCE PROBLEM**

Ensuring that all customer transactions comply with local tax regulations requires automated and accurate tax calculations.

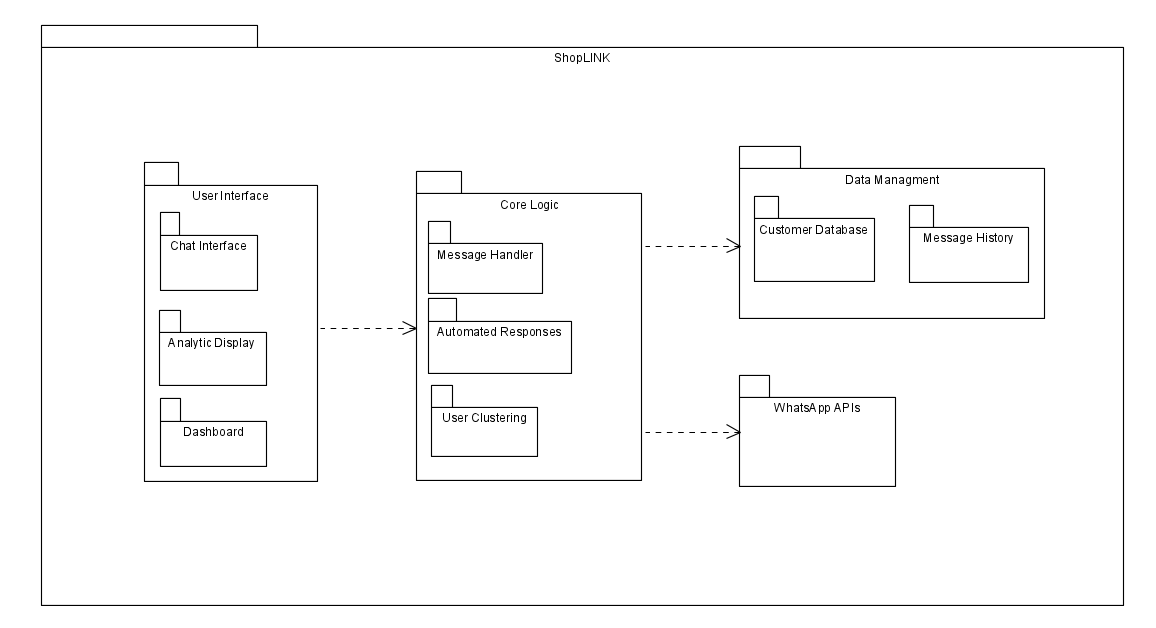
**8.4.2 TAX COMPLIANCE SOLUTION**

The application utilizes the Tax Authority API to automatically apply the correct tax rates to transactions. This integration ensures compliance with local tax regulations and streamlines the financial processing of customer transactions.

**9. SOFTWARE ENGINEERING DOCUMENTS**

**9.1 DIAGRAMS**

**9.1.1 Package diagram**



**9.1.2 FlowChart diagram**

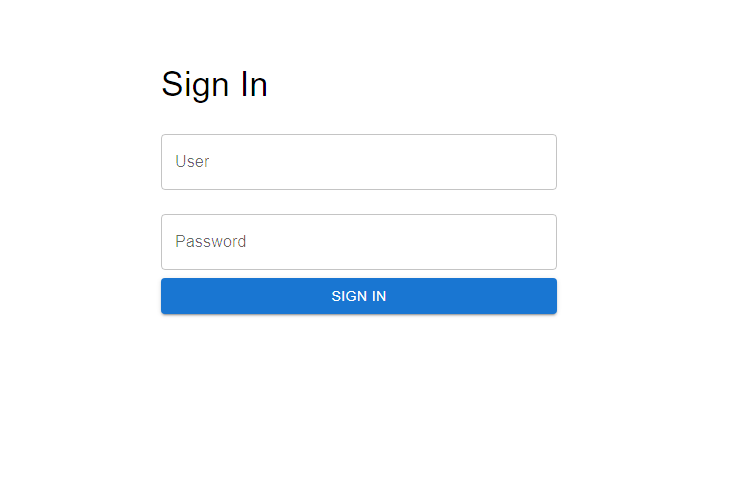
תמונה שמכילה טקסט, תרשים, צילום מסך, תוכנית

התיאור נוצר באופן אוטומטי

**9.2 GUI**

**9.2.1 LOGIN**

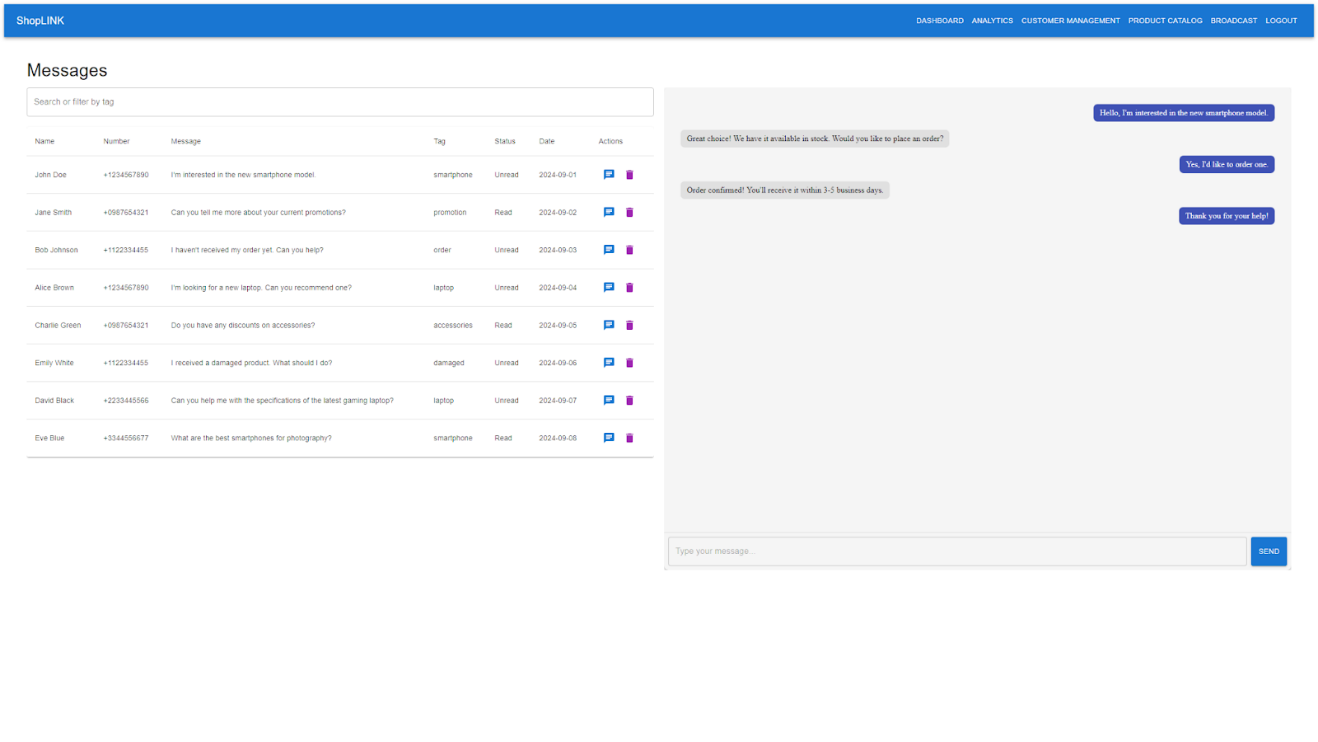
Users log into the application using a provided username and password.



**9.2.2 MANAGER SCREENS**

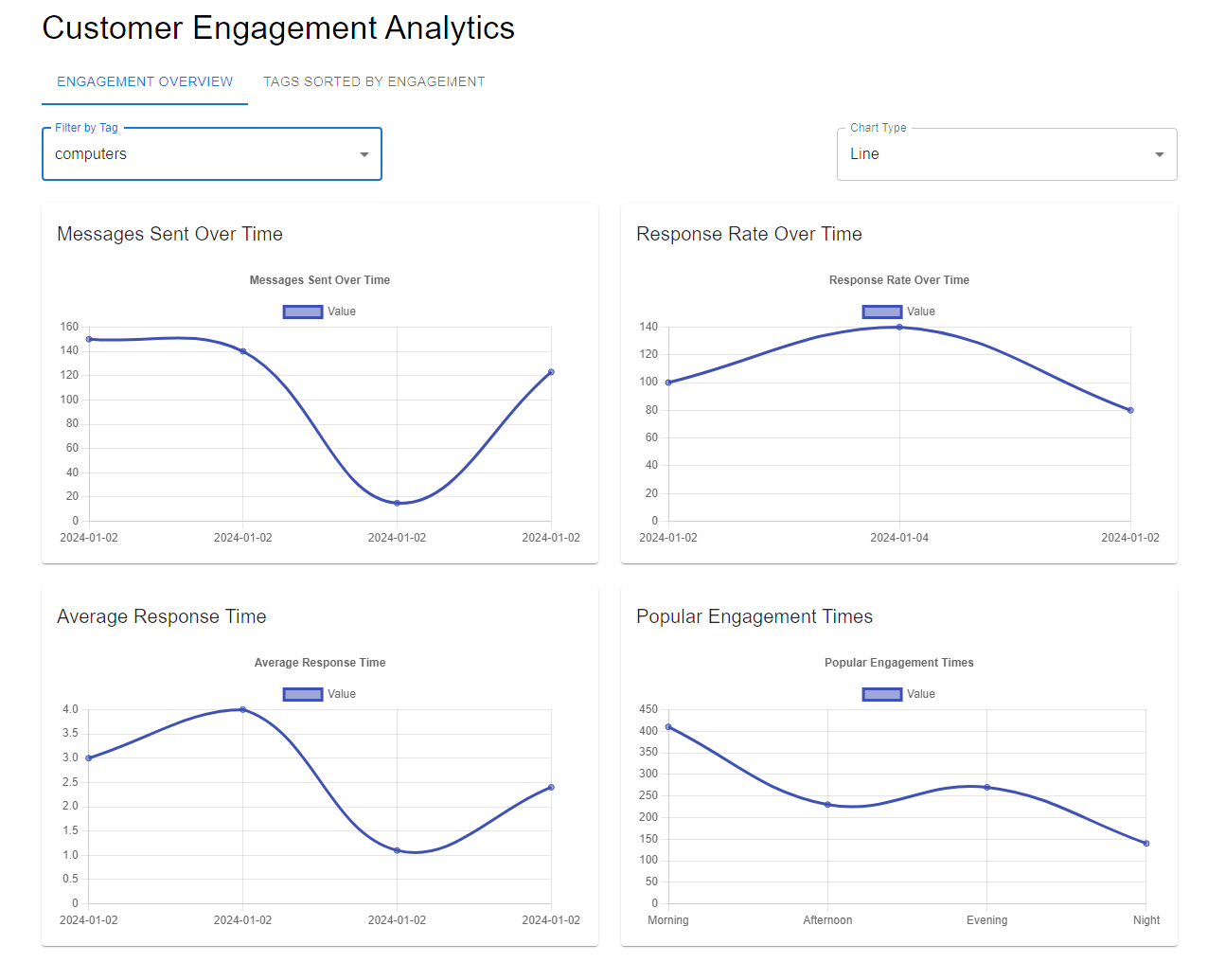
**9.2.2.1 Dashboard**

The dashboard is the main screen where managers view and respond to customer messages. It includes functionalities for filtering and searching messages by tag as well as viewing chat history.

****

**9.2.2.2 ANALYTICS SCREEN**

Managers access detailed analytics for monitoring customer engagement. The page includes multiple charts for various data insights, such as sales by category, engagement by tags, and popular engagement times. Charts can be filtered and sorted based on tags.



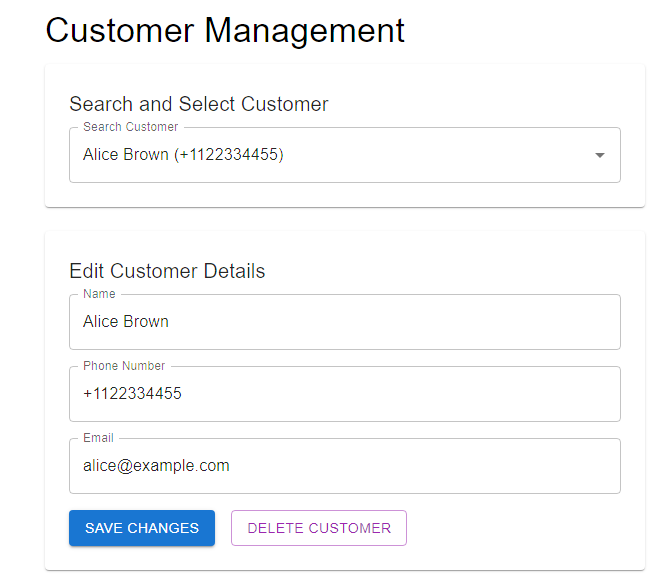
This tab displays a chart that sorts tags by total customer engagement.

A screenshot of a social media engagement

Description automatically generated

**9.2.2.3 CUSTOMER MANAGEMENT**

Managers can add and manage customer details.

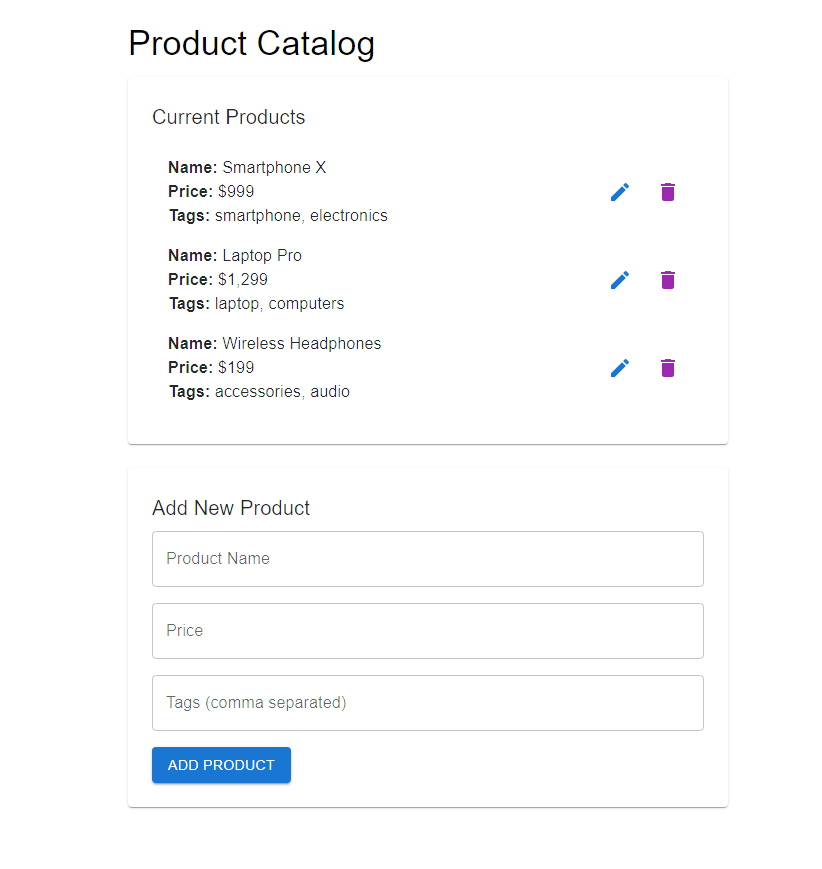


תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

**9.2.2.4 PRODUCT CATALOG**

Managers can manage the product catalog by adding, editing, or deleting products. Products can be assigned tags for labeling.

****

This tab allows to manage tags that are assignable to products. Using those tags we can cluster costumers, by assigning a tag to their messages using clustering APIs. This allows us to use targeted marketing on them based on the product tags they showed interest in.

**תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטי**

**תמונה שמכילה טקסט, צילום מסך, גופן, קו

התיאור נוצר באופן אוטומטי**

**9.2.2.5 BROADCAST MESSAGES**

Managers can send broadcast messages to selected groups of customers, or specific tags.

Using this page we can send targeted advertisements and announcement that are related to the selected tags to the customers clusters related to those tags.

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

**10. TEST PLANS**

**10.1. LOGIN TESTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Name | Description | Expected Results | Comments |
| Test valid login credentials | Enter a valid username and password to test the login feature | The user is logged in successfully |  |
| Test invalid login credentials | Enter an invalid username and password to test the login feature | The login feature rejects the credentials and displays an error message |  |
| Test valid username, no password | Enter a valid username and no password to test the login feature | The login feature rejects the credentials and displays an error message |  |
| Test no username, valid password | Enter a valid password and no username to test the login feature | The login feature rejects the credentials and displays an error message |  |

**10.2. MANAGER TESTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Name | Description | Expected Results | Comments |
| Add non-existing tag | Apply a new tag to a customer message. | The tag is successfully added to the message. |  |
| Add existing tag | Attempt to add an already existing tag to a message | The system does not add a duplicate tag, and the tag remains unchanged. |  |
| Add a tag with empty name | Try to create a tag without specifying a name | The system does not allow the tag to be added. |  |
| Filter messages by tag | Filter customer messages using a specific tag | Messages are filtered correctly based on the selected tag. |  |
| Add non-existing product | Fill out the form to add a new product to the catalog | Product is successfully added to the catalog. |  |
| Add existing product | Attempt to add a product already in the catalog. | System prevents adding a duplicate product. |  |
| Add product with missing fields | Add a product with missing name or price. | System does not allow the product to be added. |  |
| View customer chat history | Open the chat history for a specific customer | Customer chat history loads successfully. |  |
| Send broadcast message | Send a broadcast message to multiple customers. | The broadcast message is sent to all selected recipients. |  |
| Broadcast with empty message | Attempt to send a broadcast without a message body. | System prevents sending the broadcast. |  |
| View sales insights | Sort and filter sales insights by category. | Sales data is correctly sorted and filtered. |  |
| View analytics dashboard | Open the analytics dashboard to view engagement metrics. | The Analytics dashboard loads correctly with updated data. |  |
| Group customers by interests | Use clustering and topic extraction to group customers by interests. | Customers are grouped correctly based on their chat topics. |  |
| Filter messages by tag | Filter customer messages using a specific tag | Messages are filtered correctly based on the selected tag. |  |
| Add non-existing product | Fill out the form to add a new product to the catalog | The product is successfully added to the catalog. |  |
| Add existing product | Attempt to add a product already in the catalog. | The system prevents adding a duplicate product. |  |
| Add product with missing fields | Add a product with missing name or price. | System does not allow the product to be added. |  |
| View customer chat history | Open the chat history for a specific customer | Customer chat history loads successfully. |  |
| Send broadcast message | Send a broadcast message to multiple customers. | The broadcast message is sent to all selected recipients. |  |
| Broadcast with empty message | Attempt to send a broadcast without a message body. | System prevents sending the broadcast. |  |
| View sales insights | Sort and filter sales insights by category. | Sales data is correctly sorted and filtered. |  |
| View analytics dashboard | Open the analytics dashboard to view engagement metrics. | Analytics dashboard loads correctly with updated data. |  |
| Group customers by interests | Use clustering and topic extraction to group customers by interests. | Customers are grouped correctly based on their chat topics. |  |

1. **REFERENCES**
2. Advantages & Disadvantages of Node.js : Why to Use Node.js?, <https://www.simform.com/blog/nodejs-advantages-disadvantages/>
3. AI <https://chatgpt.com/>
4. AI <https://claude.ai/new>
5. Express <https://expressjs.com/>
6. Interakt application <https://www.interakt.shop/>
7. Introduction to NodeJS <https://nodejs.dev/en/learn/>
8. React <https://react.dev/>
9. What is MERN Stack?  [“https://kenzie.snhu.edu/blog/what-is-mern-stack/”](file:///C:\Users\danva\AppData\Local\Microsoft\Windows\INetCache\IE\XRZFXBX2\)
10. What Is MongoDB? <https://www.mongodb.com/company/what-is-mongodb>
11. Why did we build Visual Studio Code? <https://code.visualstudio.com/docs/editor/whyvscode>
12. Zixflow application <https://zixflow.com/>
13. Zoko application <https://www.zoko.io/>