School of Future Tech

Case Study Report

on

Simulated Chat Bot

BY

Ankit raj Jha, Parthiv Kumar, Sairaj Jadhav, Sejal Singh

150096725050, 150096725095, 150096725067, 150096725045

1. Introduction to the Case Study

The importance of customer support for an online company cannot be overstated. Most customers have questions about their orders, refunds, and how to return an item. Hiring human representatives for 24/7 customer service has become more expensive and time-consuming.

In this case, we are developing a Simulated Customer Support Chat Bot using HTML, CSS, and JavaScript. The customer interacts with the chat bot in real time, and the chat bot replies to the user's message with predefined responses.

This project shows how basic skills, like string matching, DOM manipulation, and asynchronous delays, can come together to create a simple customer support system.

2. Problem Statement / Case Background (Abstract)

Background:

Many users expect quick replies when they visit a website. Waiting for email responses or human agents can be frustrating. A chatbot addresses this issue by quickly answering common questions.

Abstract:

This case study discusses the design and implementation of a browser-based Customer Support Chat Bot. The chatbot receives user messages, analyses the input with keyword matching, and replies with preset answers. The system uses JavaScript to handle user input, match strings, update the chat interface with the DOM, and mimic real chat behaviour through time delays. The chatbot assists users with questions about orders, shipping, refunds, payments, and general help.

3. Case Study Design

The chatbot system is designed with a simple and clear flow:

- **User Interface**
  A chat window with a message input box and send button.
- **User Input Handling**
  The user types a message and presses Enter or clicks Send.
- **Message Processing**
  The input message is converted to lowercase and checked against predefined keywords.
- **Bot Response Generation**
  If a keyword matches, the bot sends a relevant response.
  If no match is found, a default help message is shown.

- **Asynchronous Behaviour**
  A typing indicator and delay are added to make the chat feel realistic.

4. Methods & Algorithms Technology Applied in the Problem Statement / Case Study

Key Concepts Used

- **String Matching**
  The bot checks if the user message contains specific keywords like “order”, “refund”, or “shipping”.
- **DOM Manipulation**
  Messages are dynamically added to the chat area using JavaScript.
- **Asynchronous Delay**
  `set Timeout ()` is used to simulate thinking time before the bot replies.
- **Conditional Logic**
  Different responses are shown based on user input.

Technology Stack

- **HTML** – Structure of the chatbot interface
- **CSS** – Styling and layout of the chat UI
- **JavaScript** – Logic for chatbot behaviour and interaction
-

5. Problem Statement / Case Study Implementation Details and Snapshots.

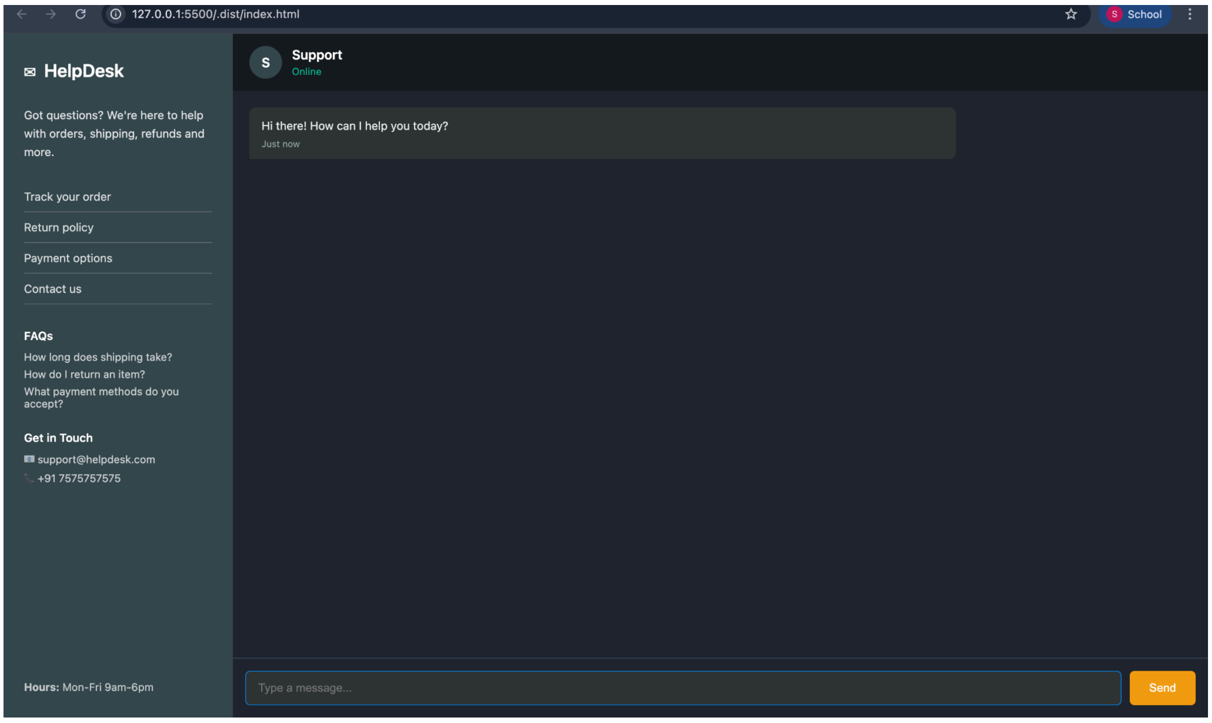
The project consists of three main files:

- **index.html**
  Contains the structure of the chatbot including input field, chat area, and buttons.
- **style.css**
  Used for styling the chatbot with a modern dark theme and responsive layout.
- **script.js**
  Handles user input, keyword matching, chat display, and bot responses.

Implementation Highlights

- User messages and bot messages are displayed differently.
- Order numbers entered as digits are detected and handled separately.
- A typing indicator is shown before the bot replies.
- The chat automatically scrolls to the latest message.

Snapshots:


127.0.0.1:5500/.dist/index.html

School

✉ HelpDesk

Got questions? We're here to help with orders, shipping, refunds and more.

Track your order

Return policy

Payment options

Contact us

FAQs

How long does shipping take?

How do I return an item?

What payment methods do you accept?

Get in Touch

support@helpdesk.com

+91 7575757575

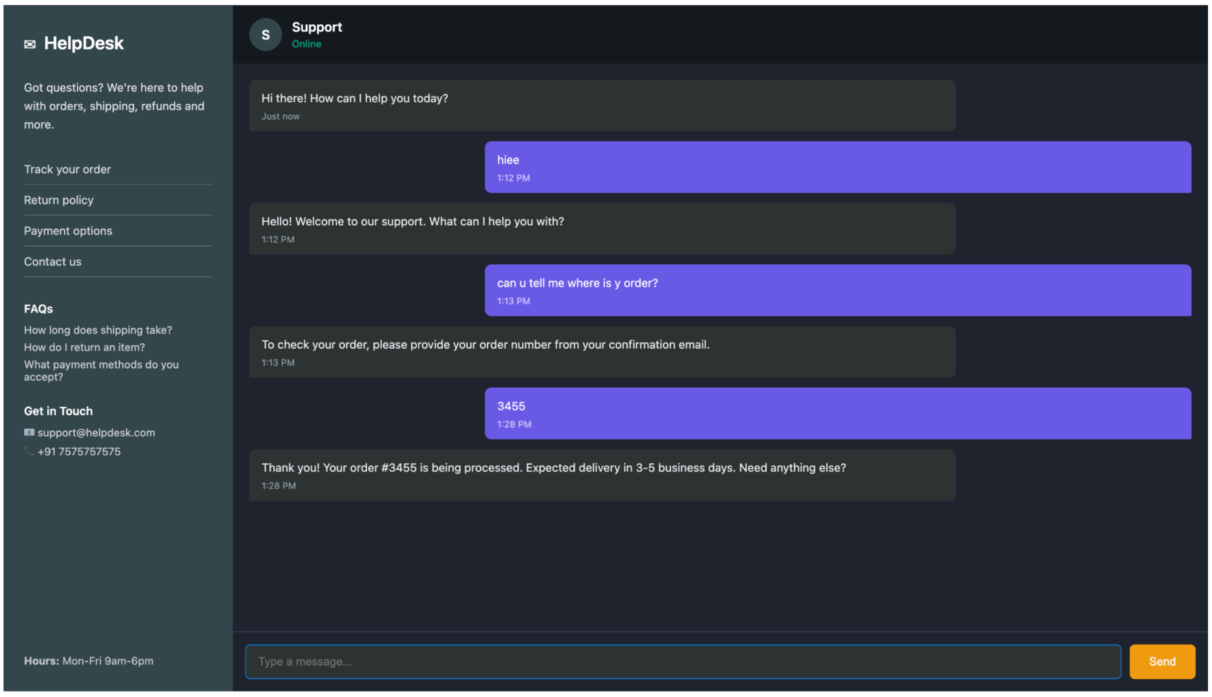
Hours: Mon-Fri 9am-6pm

S Support
Online

Hi there! How can I help you today?
Just now

Type a message...

Send


✉ HelpDesk

Got questions? We're here to help with orders, shipping, refunds and more.

Track your order

Return policy

Payment options

Contact us

FAQs

How long does shipping take?

How do I return an item?

What payment methods do you accept?

Get in Touch

support@helpdesk.com

+91 7575757575

Hours: Mon-Fri 9am-6pm

S Support
Online

Hi there! How can I help you today?
Just now

hiee
1:12 PM

Hello! Welcome to our support. What can I help you with?
1:12 PM

can u tell me where is y order?
1:13 PM

To check your order, please provide your order number from your confirmation email.
1:13 PM

3455
1:28 PM

Thank you! Your order #3455 is being processed. Expected delivery in 3-5 business days. Need anything else?
1:28 PM

Type a message...

Send

6. Problem Statement / Case Study Results and Conclusion.

Results

- The chatbot successfully responds to common customer queries.
- The user interface is clean, interactive, and easy to use.
- The typing delay makes the chatbot feel more realistic.
- The project correctly implements string matching and DOM concepts.

Conclusion

This case study shows how a simple **Customer Support Chat Bot** can be built using basic web technologies. Even without advanced AI or machine learning, useful chat systems can be created using predefined logic and keyword matching.
The project is a good example of applying JavaScript concepts to solve real-world problems and can be further improved by adding database support or AI-based responses.

7. References

- MDN Web Docs – HTML, CSS, JavaScript documentation
- JavaScript DOM Manipulation Tutorials
- Async Programming in JavaScript (`set Timeout`)
- Chatbot UI Design References