**EMAIL SUPPORT CHATBOT**

A PROJECT REPORT

*Submitted by*

|  |  |  |  |
| --- | --- | --- | --- |
| **Roll Number** | **Registration Number** | **Student Code** | **Student Name** |
| **22013002546** | **22013002546 OF 2022-2023** | **BWU/BCA/22/167** | **Swastika Roy** |
| **22010301157** | **22013002547 OF 2022-2023** | **BWU/BCA/22/168** | **Subhadip Chakraborty** |
| **22010301164** | **22013002554 OF 2022-2023** | **BWU/BCA/22/175** | **Shreya Roy** |
| **22010301118** | **22013002504 OF 2022-2023** | **BWU/BCA/22/125** | **Swapnanil Goswami** |
| **22010301129** | **220130025180 OF 2022-2023** | **BWU/BCA/22/139** | **Soumodip Patra** |
| **22010301146** | **22013002536 OF**  **2022-2023** | **BWU/BCA/22/157** | **Satwik Chowdhury** |

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF COMPUTER APPLICATION**

*IN*

**Department of computational sciences**

**BRAINWARE UNIVERSITY**

*398, Ramkrishnapur Rd, near Jagadighata Market, Barasat, Kolkata, West Bengal 700125*



MAY &2025

**EMAIL SUPPORT CHATBOT**

*Submitted by*

|  |  |  |  |
| --- | --- | --- | --- |
| **Roll Number** | **Registration Number** | **Student Code** | **Student Name** |
| **22013002546** | **22013002546 OF 2022-2023** | **BWU/BCA/22/167** | **Swastika Roy** |
| **22010301157** | **22013002547 OF 2022-2023** | **BWU/BCA/22/168** | **Subhadip Chakraborty** |
| **22010301164** | **22013002554 OF 2022-2023** | **BWU/BCA/22/175** | **Shreya Roy** |
| **22010301118** | **22013002504 OF 2022-2023** | **BWU/BCA/22/125** | **Swapnanil Goswami** |
| **22010301129** | **220130025180 OF 2022-2023** | **BWU/BCA/22/139** | **Soumodip Patra** |
| **22010301146** | **22013002536 OF**  **2022-2023** | **BWU/BCA/22/157** | **Satwik Chowdhury** |

***In partial fulfilment for the award of the degree***

*of*

BACHELOR OFCOMPUTER APPLICATION

*In*

[Department of Computational Science]

**BRAINWARE UNIVERSITY**

*398, Ramkrishnapur Rd, near Jagadighata Market, Barasat, Kolkata, West Bengal 700125*



**BRAINWARE UNIVERSITY**

*398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125*

**Department of** **COMPUTATIONAL SCIENCE**

**BONAFIDE CERTIFICATE**

Certified that this project report “**EMAIL SUPPORT CHATBOT**” is the Bonafide work of”……. **Subhadip Chakraborty[**22010301157]**, Shreya Roy[**22010301164] **, Swapnanil Goswami[**22010301118]**, Swastika Roy[**22013002546] **, Soumodip Patra[**22010301129] **, Satwik Chowdhury** ” who carried out the project work under my supervision.

---------------------------- ------------------------------

**SIGNATURE SIGNATURE**

DR. JAYANTA AICH MR. SOURADEEP SORKAR

**HEAD OF THE DEPARTMENT SUPERVISOR**

**DEPARTMENT OF COMPUTATIONAL SCIENCE DEPARTMENT OF COMPUTAIONAL SCIENCE**

398, Ramkrishnapur Road, Barasat,

North 24 Parganas, Kolkata - 700 125

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to all those who have supported and guided me throughout the development of the Email Chatbot Project.

First and foremost, we are grateful to [**SOURADEEP SARKAR**], whose valuable insights and constant encouragement were instrumental in the successful completion of this project. We would also like to thank OUR HOD[ **DR. JAYANTA AICH** ],AND [SANDIP CHAKRABORTY],[ [**BRAINWARE UNIVERSITY**] for providing the necessary infrastructure and support.

Our heartfelt thanks go to my peers, friends, and family members who motivated me and helped me stay focused during the development phase.

This project has given me the opportunity to enhance my technical knowledge and practical skills in chatbot development, email automation, and natural language processing. We are confident that the experience gained through this work will help us in our future endeavours.

Thank you all.

**Signature of all members:**

**ABSTRACT**

In today’s fast-paced digital environment, timely and efficient customer support is essential for business success. This project presents the development of an Email Support Chatbot designed to automate and streamline responses to customer queries received via email. The system is capable of categorizing queries, extracting relevant information, and providing accurate, human-like responses. It also includes features such as keyword detection, template-based replies, and integration with email servers (e.g., Gmail, Outlook) to ensure smooth communication. By reducing the need for manual intervention, the chatbot enhances response times, improves customer satisfaction, and significantly reduces operational workload.

This Email Support Chatbot can be deployed across various industries, providing a scalable and cost-effective solution for handling customer service inquiries.

**TABLE OF CONTENTS**

**Chapter Title page no**

**CERTIFICATE II**

**ABSTRACT IV**

**LIST OF FIGURES VI**

**LIST OF TABLES VI**

**1. INTRODUCTION 7-8**

**2. OBJECTIVE 8**

**3. PLANNING 9-10**

**4. REQUIRMENT ANALYSIS 11-13**

**5. SYSTEM FLOW 14 - 18**

**6. PROPOSED DIAGRAM 19-20**

**7. SAMPLE CODE EXPERIMENTAL RESULT 21-31**

**8. FUTURE SCOPE 32-33**

**9. COLCUSION 34- 35**

**10. APPENDICES 36-39**

**11. REFERENCES 40**

**LIST OF TABLES**

1. **TABLE 1:**FUCTIONAL AND NON-FUCTIONAL REQUIRMENT……………………………………… 1

**LIST OF FIGURE**

1. **FIGURE 1:** CONTEXT LEVEL DIAGRAM……………………………………………………………..1
2. **FIGURE 2:** USE CASE DIAGRAM……………………………………………………………………….2
3. **FIGURE 3:** ACTIVITY DIAGRAM……………………………………………………………………….3
4. **FIGURE 3:** FLOW CHART …………………………………………………………………………………4

**Chapter-1**

**INTRODUCTION**

* 1. Objective of the Work:

Current business relies heavily on emails as a means of communication, and with this rapid dependence on digital communication, efficient customer support will be the key to success. Traditional customer support procedures are mostly conducted through human agents, which is time-consuming and resource-intensive. This could lead to delayed answering of customer inquiries and overall low satisfaction from customers. The project focuses on developing an intelligent chatbot designed and built specifically for dealing with frequently asked email support questions.

The exponential growth of online communication and the rising customer expectations for instant gratification necessitate innovative solutions for providing timely and effective support. Chatbots, powered by artificial intelligence and natural language processing, have emerged as a promising technology to revolutionize customer service interactions. By automating routine tasks and providing immediate assistance, chatbots can significantly enhance customer experience, reduce response times, and optimize support operations.

* 1. Scope of the Work :

This project will be concerning the development of an email-support chatbot to understand and interact with a comprehensive range of customers' inquiries toward email services. The chatbot will be set up to give users a fully human-like simulation of conversation by providing them with a seamless support experience. Applying advanced AI, the chatbot will be designed to understand queries from users and interpret them well, extract required information, and provide accurate and appropriate responses.

The project aims at creating a value tool that is likely to benefit businesses in achieving the most efficient and effective emailing support for customers. This includes improving customer satisfaction, reducing operation costs, and making the general business more effective by offering 24/7 accessibility with the assistance of the automated common inquiries and answers of the chatbot with reliable interface.

* 1. Technology Used:
     1. Hardware used:

The hardware device used for developing this project , are as follows:

Processor: 12th Generation RYZEN 5 7520U.

RAM: 8 GB

Solid State Drive: 512 GB

Screen: Resolution 1920 x 1080 x 144 hertz

Note : This is not the “System Requirements”.

* + 1. Software Used :

The software used for developing the project , are as follows :

Operating System : Windows 11 Home

Software : **Visual Studio Code 1.96.2**This is the primary code editor used to write and develop the HTML, CSS, and JavaScript code for the chatbot.

* + 1. Technologies :

**HTML:** The core language for structuring the web page, defining the content elements (header, chat messages, input field, div for section etc.), and providing the overall layout.

**CSS:** Used to style the appearance of the chatbot:

**Visuals:** Colors, fonts, borders, shadows, rounded corners, etc.

**Layout:** Using Flexbox for positioning and alignment of elements.

**Animations:** Creating visual effects like fading in, sliding in, and pulsing.

**JavaScript:**

**DOM Manipulation:** Interacting with the HTML elements (e.g., adding messages to the chat, handling user input).

**Event Handling:** Responding to user interactions (e.g., button clicks, key presses,media).

**Logic:** Implementing basic conversational logic (e.g., greeting responses, basic question-answering).

**Chapter - 2**

**OBJECTIVES**

This project will therefore aim at the development of an intelligent email support chatbot that helps to address all the challenges associated with the provision of timely and efficient customer support. The overall aim is to provide a useful tool that increases customer satisfaction and also streamlines the support operations.

General Objectives:

To create a chatbot that can understand and respond to common email support questions.

To enhance customer satisfaction by providing fast and efficient support, reducing wait times, and enhancing the overall user experience.

To reduce the workload of human support agents by automating routine tasks and handling a large volume of inquiries.

To prove the feasibility and potential benefits of using chatbot technology for email support.

Specific Objectives:

*Accuracy:* A minimum accuracy rate of 90% for the correct identification of user intent and relevant responses to common email support queries*.*

*Response Time:* Reduction in average response time for common email support queries by at least 50% compared to traditional support channels.

*User Satisfaction:* Customer satisfaction rating of at least 85% based on user feedback and surveys.

*Usability:* Create a friendly and intuitive chatbot interface easy to navigate for customers*.*

*Scalability:* Design the system so that the chatbot could be scaled to adapt to growing user inquiries and emerging customer needs.

*Integration:* Effectively integrate the chatbot into the existing email systems and relevant business l

*New features:* Add more screenshot to save your chats.

*Login-logout property :* it has a new login-logout security to access the chatbot*.*

**Chapter - 3**

**PLANNING**

*Project Goals:*

Provide basic automated support for common email-related inquiries. This includes issues like sending, receiving, account problems, settings, and security.

Improve user experience by offering a quick and accessible support channel.

Reduce the burden on human support agents by handling frequent and routine questions.

Enhance customer satisfaction by providing timely and helpful assistance.

*Target Audience:*

Users who encounter common email-related problems.

Individuals seeking quick resolutions to basic email issues.

*Scope:*

This project will develop a very basic chatbot with minimal conversation capabilities.

The chatbot will primarily provide information answers and refer users to resources.

Human interaction may be required for complex or highly specific questions.

**Technical Approach:**

*Frontend:*

HTML, CSS, and JavaScript for the user interface

CSS for styling and animations to make it visually appealing and user-friendly

JavaScript for handling user input, displaying messages, and implementing basic chatbot logic

*Backend Logic:*

JavaScript for core logic of the chatbot, namely:

getBotResponse() function:

* Analyses user input through regular expressions to recognize keywords and patterns.
* Returns suitable responses based on analysis.
* Handles greetings, farewells, and email-related common problems.
* Returns default responses for unknown or complex queries.

ChatMessage class:

Represents a single message (user or bot) with attributes for role (user/bot) and content.

displayMessage() function:

Adds messages to the chat window and scrolls to the bottom.

appendMessage():

messeage and media support for these chatbot.

getGreetingResponse():

Greet the users before intractions.

getFarewellResponse():

Represent as a farewell messages like goodbye, bye thank you etc.

hideTypingIndicator():

Hide the typing which is indicated.

readFileAsDataURL():

Represent how the data was chooses as several files.

toggleSound():

sound change purpose.

toggleSound():

Play audio facilities for this chatbot.

openSettings():

Here we can use the settings to system any changes.

.. Several methods are used to create a Regular chatbot..

*User Interface:*

Simple and intuitive design to easily interact with the user.

User and bot messages are displayed with visual cues.

**Project Timeline:**

*Phase 1: Planning and Design*

Define the scope and goals of the project.

Design the user interface and user experience.

Plan the logic and response patterns of the chatbot.

*Phase 2: Development*

Develop the HTML, CSS, and JavaScript code.

Implement the getBotResponse() function with core logic.

Test and debug the functionality of the chatbot.

*Phase 3: Testing and Refinement*

Conduct thorough testing to identify and fix bugs.

Gather user feedback and iterate on the design and functionality.

**Chapter - 4**

**REQUIRMENT ANALYSIS**

1. **FUNCTIONAL REQUIRMENT**

|  |  |
| --- | --- |
| Categorization | THEIR REQUIRMENT |
| *User Interaction:* | The chatbot should accept user input in the form of a text box.  The chatbot should display messages from users and bots clearly and in an orderly fashion. |
| *Core Chatbot Functionality:* | The core chatbot functionality should be able to comprehend and respond to most common email-based enquiry requests.  The chatbot needs to provide the correct and as concise relevant responses to the user's questions.  The chatbot should be functional for greetings and goodbyes.  The chatbot should have a default response for unknown or complex queries. |
| *Email-Specific Support:* | The chatbot should be able to provide information and support on the following:  *Greetings feature:(* hello, hi, hey, good (morning, afternoon, evening).  *Farewells greet*: (bye, goodbye, see you, later, exit).  *Email-Specific Issues:(* can't send, not sending, sending failed, not receiving, not getting, missing emails, account hacked, create account, forgot password, change password, update profile, account information*).*  *Account Recovery:(* recover account*)*  *Block sender:(* block sender*).*  *Setting & Features:(* *settings,* *signature,* *attachment,* *address,* *spam,* *phishing).*  *2FA Explanation:(* *what is 2fa)*  *Dark Mode Instructions:(* *dark mode,* *security,* *client).*  *Report a bug:(* *report a bug,* *account suspended)*  *New feature requests:(* *new feature).* |
| *General Inquiries:* | The chatbot should be able to handle general inquiries such as "contact us," "hours of operation," and "help." |
| *General Responses:* | The chatbot should be able to handle general inquiries such as "thank you," "thanks," and "sorry." |

1. **NON-FUNCTIONAL REQUIRMENT**

|  |  |
| --- | --- |
| Categorization | THEIR REQUIRMENT |
| *Usability:* | The chatbot interface should be user-friendly and easy to navigate.  The chatbot should provide clear and concise responses that are easy to understand.  The chatbot should be accessible to users with varying levels of technical expertise. |
| *Performance:* | The chatbot should respond to user queries promptly and efficiently. |
| *Reliability:* | The chatbot should be reliable and consistent in its responses. |
| *Maintainability:* | The code should be well-structured, documented, and easy to maintain and update. |
| *Security:* | The chatbot should not collect or store any sensitive user data.  The chatbot should be secure against probable vulnerabilities. |

**3. User Interface Requirements:**

A visually appealing and user-friendly interface

Clear display of user and bot messages.

An input field for user messages.

A "Send" button to submit the user messages.

**4. Technical Requirements:**

Front-end technologies: HTML, CSS, JavaScript. Basic JavaScript for handling user input, displaying messages, and simple logic.

Use of regular expressions for pattern matching in user input.

**Chapter - 5**

**SYSTEM FLOW**

**CONTEXT LEVEL DIAGRAM:**

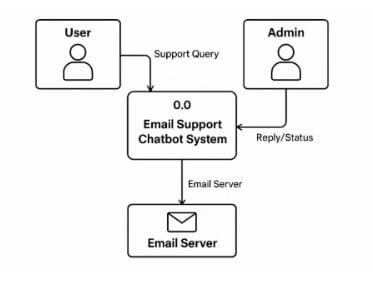
A **Context Level Diagram**, also known as a **Level 0 Data Flow Diagram (DFD)** or a **System Context Diagram**, is a high-level diagram that provides a bird's-eye view of a system and its interactions with the external environment. It represents the entire system as a single process (a circle or bubble) and shows the external entities (actors, systems, or organizations) that interact with it, along with the data flows (arrows) between them.

**USE OF THIS DIAGRAM:**

**Defining System Scope and Boundaries:** It clearly illustrates what is included within the system and what lies outside its boundaries. This helps in preventing scope creep and ensures everyone has a shared understanding of what the system will and will not do.

**Identifying External Interactions:** It helps to identify all the external entities that interact with the system, making it easier to understand the system's environment and dependencies.

**Facilitating Communication:** It provides a simple and non-technical overview of the system that can be easily understood by all stakeholders, including business users, developers, and project managers. This fosters better communication and alignment



**Fig:1. context level diagram**

**USE CASE DIAGRAM:**

A **Use Case Diagram** is a type of Unified Modeling Language (UML) diagram that visually represents the interactions between users (called **actors**) and a system to achieve specific goals. It provides a high-level overview of the system's functionality from the user's perspective, illustrating "what" the system does rather than "how" it does it.

**USE OF THIS DIAGRAM:**

**Gathering and Defining Requirements:** They help in identifying and documenting the functional requirements of the system from the user's perspective. By focusing on user goals, they ensure the system delivers value.

**Understanding System Scope:** The system boundary clearly defines what is part of the system and what is external, helping to manage the project scope.

**Facilitating Communication:** They provide a common language and visual representation that can be easily understood by both technical and non-technical stakeholders (users, developers, testers, managers). This improves communication and ensures everyone has a shared understanding of the system's functionality.

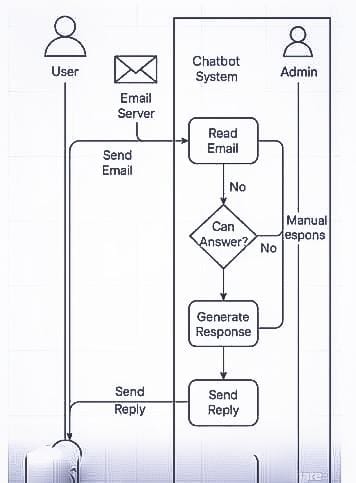


Fig:2. Use case diagram

**ACTIVITY DIAGRAM:**

An **Activity Diagram** is another type of behavioral diagram in the Unified Modeling Language (UML) that visually represents the flow of activities within a system, a business process, or a use case. It describes the sequence of actions, parallel activities, decision points, and the overall flow of control.

**USE OF THIS DIAGRAM:**

**Modeling Business Processes:** They are excellent for visualizing and understanding the steps involved in a business process, including workflows, decision points, and parallel activities.

**Modeling Use Case Behavior:** They can be used to elaborate on the flow of events within a use case, providing a more detailed view of how the system and actors interact.

**Modeling System Workflows:** They help in understanding the internal processes and the flow of control within a software system or a part of it.

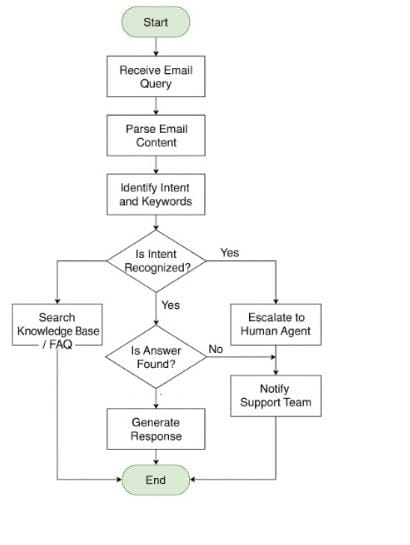


Fig:3. Activity diagram

**FLOWCHAT :**

A **Flowchart** is a visual representation of a process or algorithm, illustrating the sequence of steps and decisions involved. It uses standardized symbols connected by arrows to show the direction of flow. Think of it as a map that guides you through the steps of a process from start to finish.

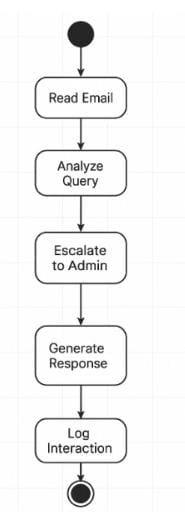
**USE OF THIS FLOW:**

**Visualizing Processes:** They provide a clear and easy-to-understand visual representation of complex processes, making them accessible to a wide audience.

**Documenting Processes:** They serve as a form of documentation, outlining the steps and decision points in a process for reference, training, and standardization.

**Analyzing Processes:** By mapping out a process, it becomes easier to identify bottlenecks, inefficiencies, redundancies, and areas for improvement.

Fig:3. Activity diagram

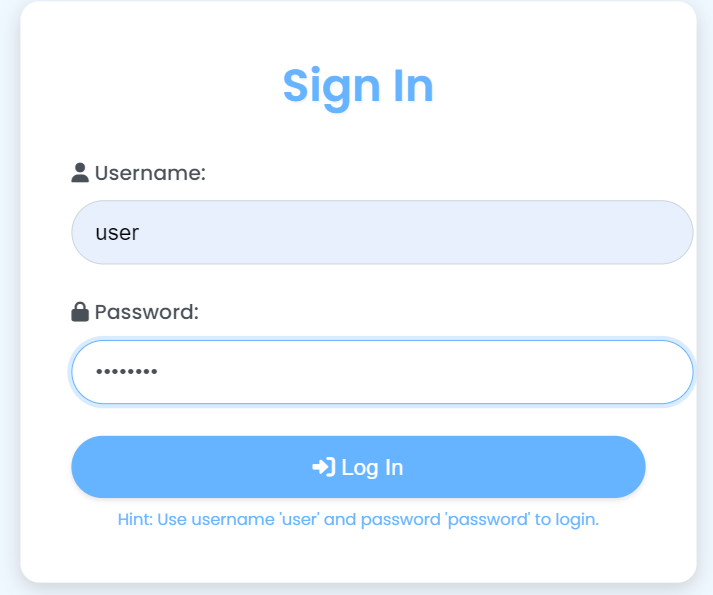


**CHAPTER -6**

**PROPOSED DESIGN**

**FRONT-END PAGE:**

**LOGIN PAGE:**



**USER INTERFACE PAGE:**



**Chapter - 7**

**SAMPLE CORE CODE AND EXPERIMENTAL RESULT**

**SAMPLAE CORE CODE:**

<script>

<script>

const chatMessages = document.querySelector('.chat-messages');

const userInput = document.querySelector('.user-input');

const sendButton = document.querySelector('.send-button');

const mediaButton = document.querySelector('.media-button');

const mediaInput = document.querySelector('.media-input');

const typingIndicator = document.querySelector('.typing-indicator');

const suggestedQuestions = document.querySelector('.suggested-questions');

const settingsButton = document.querySelector('.settings-button');

const settingsModal = document.querySelector('.settings-modal');

const settingsContent = document.querySelector('.settings-content');

//Get elements by ID

const darkModeSwitch = document.getElementById('darkModeSwitch');

const fontSizeSelect = document.getElementById('fontSizeSelect');

const languageSelect = document.getElementById('languageSelect');

const soundToggle = document.getElementById('soundToggle');

const closeButton = document.getElementById('closeButton');

const body = document.body;

const reportButton = document.getElementById('reportButton');

const changePasswordButton = document.getElementById('changePasswordButton');

const screenshotButton = document.getElementById('screenshotButton');

//Function to simulate Report

reportButton.addEventListener('click', function () {

alert("Reported! Thanks for Reporting");

});

//Function to simulate Password Change

changePasswordButton.addEventListener('click', function () {

alert("You can Change your Password Now!");

});

//Screenshot button listener

screenshotButton.addEventListener('click', function () {

html2canvas(document.querySelector(".chatbot-container")).then(canvas => {

const link = document.createElement("a");

link.download = "chatbot-screenshot.png";

link.href = canvas.toDataURL("image/png");

link.click();

});

});

function appendMessage(message, type, mediaUrl = null) {

const messageDiv = document.createElement('div');

messageDiv.classList.add('message', `${type}-message`, 'animate\_\_animated', 'animate\_\_fadeIn');

messageDiv.textContent = message;

if (mediaUrl) {

const mediaElement = mediaUrl.startsWith('data:image') ? document.createElement('img') : document

.createElement('video');

mediaElement.src = mediaUrl;

mediaElement.controls = mediaUrl.startsWith('data:video');

mediaElement.classList.add('message-media');

messageDiv.appendChild(mediaElement);

}

chatMessages.appendChild(messageDiv);

chatMessages.scrollTop = chatMessages.scrollHeight;

}

function botResponse(message) {

const lowerCaseMessage = message.toLowerCase();

// Greetings

if (/(hello|hi|hey|good (morning|afternoon|evening))/i.test(lowerCaseMessage)) {

return getGreetingResponse();

}

// Farewells

if (/(bye|goodbye|see you|later|exit)/i.test(lowerCaseMessage)) {

return getFarewellResponse();

}

// Email-Specific Issues

if (/can't send|not sending|sending failed/.test(lowerCaseMessage)) {

return "Possible reasons include: Network connectivity issues, recipient email address is incorrect, email server is temporarily down or has issues. Check all the things!";

}

if (/not receiving|not getting|missing emails/.test(lowerCaseMessage)) {

return "Confirm that the sender address is correct, check spam and junk mail folders, check email filtering settings. If all are fine, contact email admin support";

}

if (/account hacked/.test(lowerCaseMessage)) {

return "Change password immediately, review recent activity, enable 2FA, scan devices, and contact your email provider.";

}

if (/create account/.test(lowerCaseMessage)) {

return "Visit the email provider's website to create an account.";

}

if (/forgot password/.test(lowerCaseMessage)) {

return "Use the 'Forgot Password' link on the login page.";

}

if (/change password/.test(lowerCaseMessage)) {

return "Access your account settings to change your password.";

}

if (/update profile/.test(lowerCaseMessage)) {

return "Access your account settings to update your profile information.";

}

if (/account information/.test(lowerCaseMessage)) {

return "Log in to your account to view your account information.";

}

// Account Recovery

if (/recover account/.test(lowerCaseMessage)) {

return "Use recovery options like security questions or secondary email to verify identity. Contact customer support for help if required.";

}

// Block sender

if (/block sender/.test(lowerCaseMessage)) {

return "Right click the sender and select 'Block sender'. This will prevent the email sender from future emails";

}

// Setting & Features

if (/settings/.test(lowerCaseMessage)) {

return "Specify settings: forwarding, filtering, vacation responder, or other settings.";

}

if (/signature/.test(lowerCaseMessage)) {

return "Creating or editing your email signature? I can provide guidance.";

}

if (/attachment/.test(lowerCaseMessage)) {

return "Attaching files, sending large attachments, or receiving attachments?";

}

if (/address/.test(lowerCaseMessage)) {

return "Creating a new email address or changing your existing one?";

}

if (/spam/.test(lowerCaseMessage)) {

return "Identifying or blocking spam emails? I can provide tips and guidance.";

}

if (/phishing/.test(lowerCaseMessage)) {

return "Be wary of suspicious senders, check sender address, don't click links, hover over links, be cautious of urgent requests.";

}

// 2FA Explanation

if (/what is 2fa/.test(lowerCaseMessage)) {

return "2FA is Two-Factor Authentication, an extra layer of security requiring a code from your phone or email in addition to your password.";

}

// Dark Mode Instructions

if (/dark mode/.test(lowerCaseMessage)) {

return "Dark mode helps to save energy and to reduce eye strain. Go to settings and select the theme Dark to enable the dark mode";

}

if (/security/.test(lowerCaseMessage)) {

return "Use strong passwords, enable 2FA, update software, be cautious of what you share, and avoid suspicious emails.";

}

if (/client/.test(lowerCaseMessage)) {

return "Specify your email client (e.g., Outlook, Gmail, Apple Mail) for more specific assistance.";

}

// General Inquiries

if (/contact us/.test(lowerCaseMessage)) {

return "You can contact us by phone at [Phone Number] or by email at [Email Address].";

}

if (/hours/.test(lowerCaseMessage)) {

return "Our customer support hours are [Hours of Operation].";

}

if (/help/.test(lowerCaseMessage)) {

return "How can I assist you today? Please be more specific with your request.";

}

// General Responses

if (/thank you|thanks/.test(lowerCaseMessage)) {

return ["You're welcome!", "No problem, happy to help!"][Math.floor(Math.random() \* 2)];

}

if (/sorry/.test(lowerCaseMessage)) {

return "No worries, how can I assist further?";

}

// New feature requests

if (/new feature/.test(lowerCaseMessage)) {

return "We appreciate your suggestion! Please provide more details so we can consider it.";

}

// Report a bug

if (/report a bug/.test(lowerCaseMessage)) {

return "Thank you for reporting! Please provide details about the issue.";

}

if (/account suspended/.test(lowerCaseMessage)) {

return "Contact support immediately!";

}

// Default

return "I'm still learning. Rephrase or explore: Account security, sending/receiving emails, password assistance.";

}

function getGreetingResponse() {

return ["Hello!", "Hi there!", "Welcome!", "Good to see you!", "How can I help today?"][Math.floor(Math.random() \*

5)];

}

function getFarewellResponse() {

return ["Goodbye!", "See you later!", "Have a great day!", "Take care."][Math.floor(Math.random() \* 4)];

}

function showTypingIndicator() {

typingIndicator.style.display = 'block';

}

function hideTypingIndicator() {

typingIndicator.style.display = 'none';

}

async function sendMessage() {

const messageText = userInput.value.trim();

const mediaFile = mediaInput.files[0];

if (!messageText && !mediaFile) return;

let mediaUrl = null;

if (mediaFile) {

mediaUrl = await readFileAsDataURL(mediaFile);

}

appendMessage(messageText, 'user', mediaUrl);

userInput.value = '';

mediaInput.value = '';

showTypingIndicator();

setTimeout(() => {

hideTypingIndicator();

const response = botResponse(messageText);

appendMessage(response, 'bot');

if (isSoundEnabled) {

playSound(); // Play sound on new bot message

}

}, 800);

}

function readFileAsDataURL(file) {

return new Promise((resolve, reject) => {

const reader = new FileReader();

reader.onload = () => resolve(reader.result);

reader.onerror = () => reject(reader.error);

reader.readAsDataURL(file);

});

}

// Dark Mode Functionality

let isDarkMode = localStorage.getItem('darkMode') === 'true';

body.setAttribute('data-theme', isDarkMode ? 'dark' : '');

darkModeSwitch.checked = isDarkMode;

function toggleDarkMode() {

isDarkMode = !isDarkMode;

body.setAttribute('data-theme', isDarkMode ? 'dark' : '');

localStorage.setItem('darkMode', isDarkMode ? 'true' : 'false');

}

// Font Size Functionality

let currentFontSize = localStorage.getItem('fontSize') || 'normal';

setFontSize(currentFontSize);

fontSizeSelect.value = currentFontSize;

function setFontSize(size) {

localStorage.setItem('fontSize', size);

document.documentElement.style.setProperty('--font-size-normal', size === 'small' ? '0.85rem' : size ===

'large' ? '1.15rem' : '1rem');

}

// Language Functionality

let currentLanguage = localStorage.getItem('language') || 'en';

languageSelect.value = currentLanguage;

function setLanguage(lang) {

//This is for illustration only, actually implementing translation would be more complex.

localStorage.setItem('language', lang);

alert(`Language changed to ${lang}`);

}

// Sound Functionality

let isSoundEnabled = localStorage.getItem('soundEnabled') === 'true';

soundToggle.checked = isSoundEnabled;

function toggleSound() {

isSoundEnabled = !isSoundEnabled;

localStorage.setItem('soundEnabled', isSoundEnabled ? 'true' : 'false');

}

functiontoggleSound() {

const audio = new Audio(

'https://interactive-examples.mdn.mozilla.net/media/cc0-audio/t-rex-roar.mp3'); // Replace with your desired

audio.play();

}

function openSettings() {

settingsModal.style.display = 'block';

}

function closeSettings() {

settingsModal.style.display = 'none';

}

// Event Listeners

sendButton.addEventListener('click', sendMessage);

userInput.addEventListener('keydown', (event) => {

if (event.key === 'Enter') sendMessage();

});

mediaButton.addEventListener('click', () => mediaInput.click());

suggestedQuestions.addEventListener('click', (event) => {

if (event.target.tagName === 'BUTTON') {

userInput.value = event.target.dataset.question;

sendMessage();

}

});

settingsButton.addEventListener('click', openSettings);

closeButton.addEventListener('click', closeSettings);

darkModeSwitch.addEventListener('change', function () {

toggleDarkMode();

});

fontSizeSelect.addEventListener('change', function (event) {

setFontSize(event.target.value);

});

languageSelect.addEventListener('change', function (event) {

setLanguage(event.target.value);

});

soundToggle.addEventListener('change', function () {

toggleSound();

});

//Initialization - Load settings on startup

body.setAttribute('data-theme', isDarkMode ? 'dark' : ''); //Apply saved darkmode

fontSizeSelect.value = currentFontSize; //Apply saved font size

// Load settings on startup (better practice):

window.addEventListener('DOMContentLoaded', (event) => {

body.setAttribute('data-theme', isDarkMode ? 'dark' : '');

fontSizeSelect.value = currentFontSize;

soundToggle.checked = isSoundEnabled; //set sound toggled state to the value we loaded from localstorage on page load

});

// Close the modal if the user clicks outside of it.

window.addEventListener('click', (event) => {

if (event.target == settingsModal) {

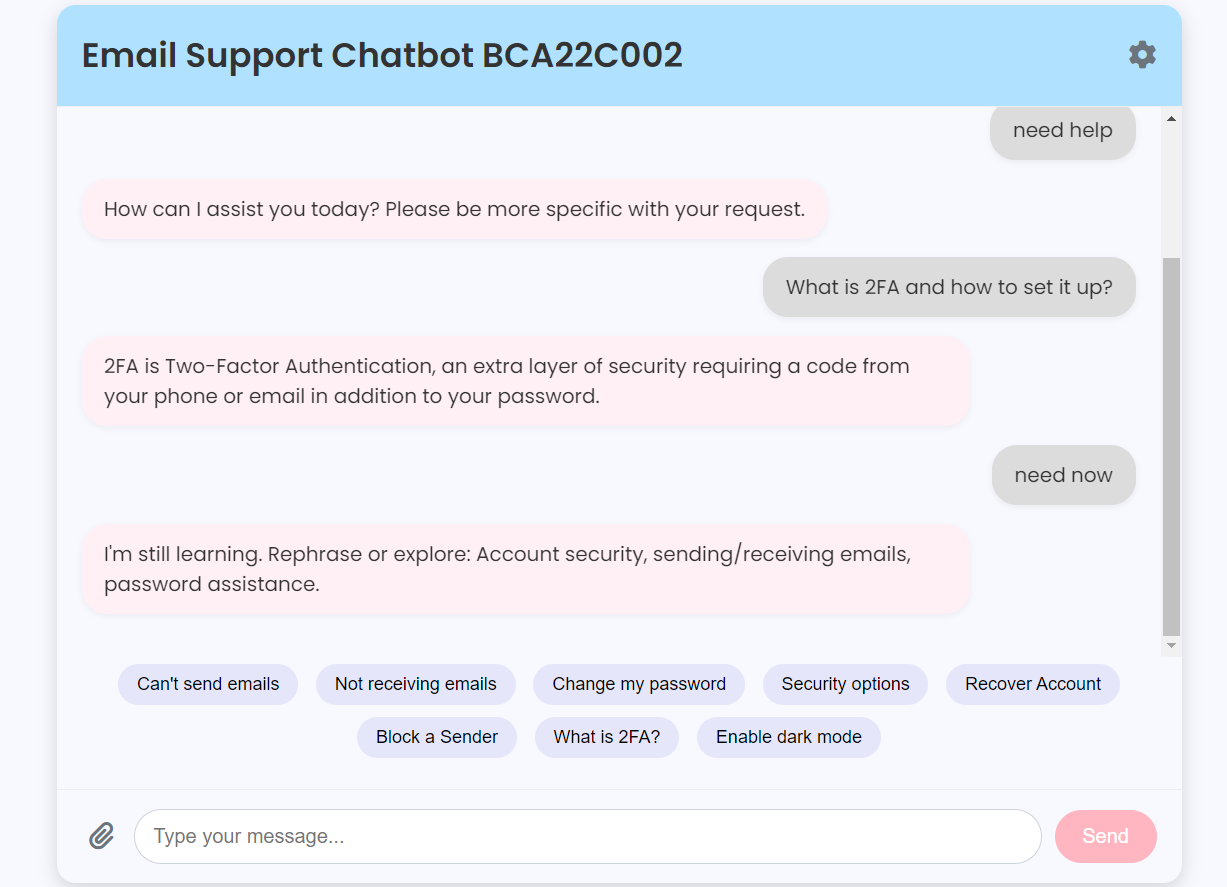
closeSettings();

}

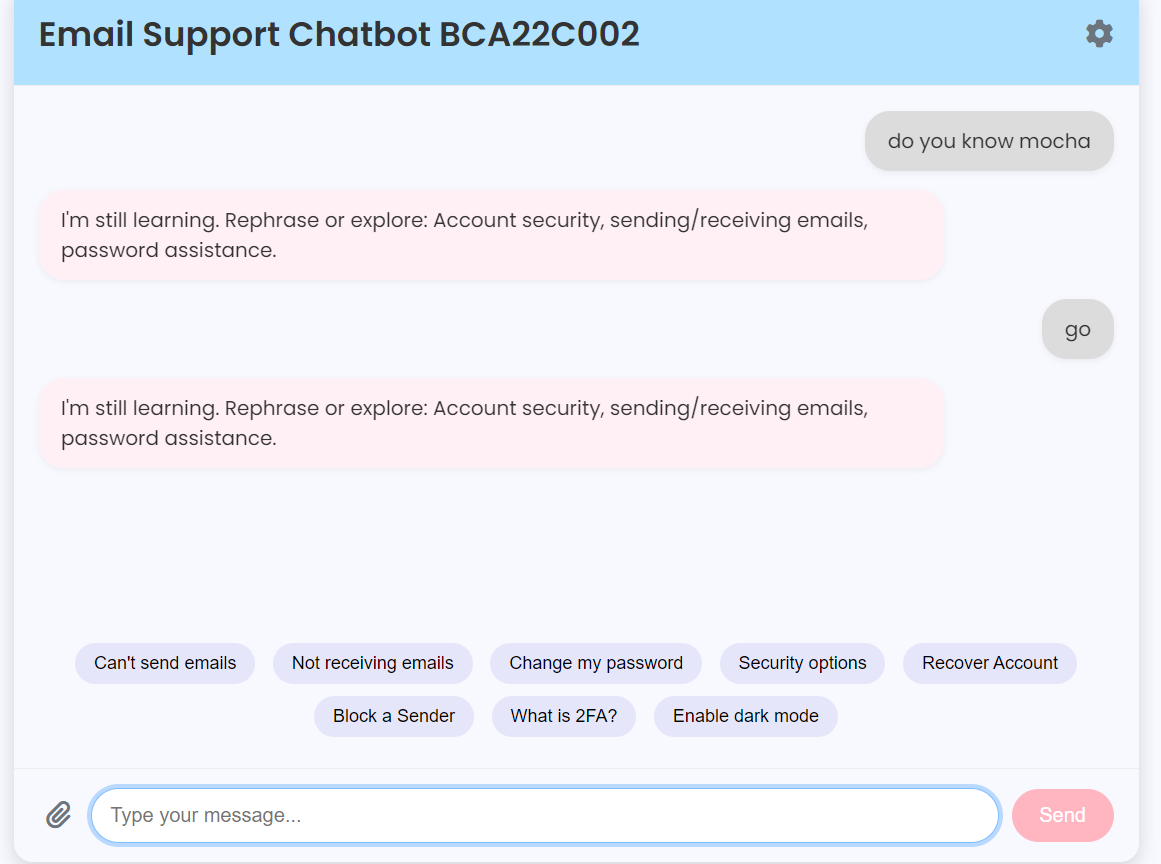
});

</script>

**EXPERIMENTAL RESULT ( Valid User Input) :**



**EXPERIMENTAL RESULT ( Invalid User Input) :**



**Chapter - 8**

**FUTURE SCOPE**

**1.Enhancements:**

*Natural Language Processing:*

Help the chatbot process more complex and subtle queries from users.

Develop features like sentiment analysis that would identify frustration or urgency in a user's message.

*Machine Learning:*

Allow the chatbot to learn and evolve its responses over time.

More advanced techniques for response generation include:

Dialogue management: keep context over a sequence of multiple user turns in a conversation.

Personalized responses: tailor the responses to individual users' needs and preferences.

*Integrate with External Services:*

Connect with email services, like Gmail or Outlook, to be able to give even more specific and personalized assistance.

Connect with knowledge base articles or FAQs to be able to provide more comprehensive and accurate information.

Integration with CRM( Customer relationship Management) systems for access to customer information and personalization of the interaction.

*Voice Interaction:*

Support for voice input and output for a more natural, intuitive experience.

*Multilingual Support:*

Support for multiple languages for a larger reach.

1. **Advanced Features:**

*Proactive Support:*

Identify potential email issues and proactively address them in advance (for example, inform users of suspicious emails, provide security alerts).

*Sentiment Analysis and Emotional Response:*

Detect and respond to user emotions (e.g., frustration, anger) with empathy and proper language.

*Personalized Recommendations:*

Provide personalized recommendations based on user history and preferences (e.g., suggest helpful articles, recommend security best practices).

**3. User Interface Enhancements:**

*Improved User Interface:*

More visually appealing and interactive design.

Integration with chat widgets for easy embedding into websites or applications.

*Multi-device Support:*

Ensure compatibility across various devices (desktops, laptops, mobile phones, tablets).

*Accessibility:*

Ensure that the chatbot is accessible to users with disabilities (for example, screen reader compatibility).

**4. Evaluation and Monitoring:**

Periodically evaluate the performance of the chatbot:

Track key metrics such as user satisfaction, response times, and resolution rates.

Collect user feedback and include it in future improvements.

Monitor the performance of the chatbot and identify areas for improvement.

**5.Security and Privacy:**

Maintain data privacy and security:

Implement adequate safety measures to guard user data

Comply with relevant data protection directives, such as GDPR.

**Chapter - 9**

**CONCLUSION**

This project shows a simple implementation of an email support chatbot. The chatbot effectively uses HTML, CSS, and JavaScript to create a user-friendly interface and provides automated responses to common email-related queries.

**Key Features:**

*Basic Functionality:*

The chatbot successfully handles basic interactions, including:

Receiving and processing user input.

Generating appropriate responses based on user queries.

Displaying user and bot messages in a clear and organized manner.

Handling greetings and farewells.

*Email Support:*

The chatbot deals with the most common issues related to emails, such as sending, receiving, account problems, settings, and security.

*User Interface:* The chatbot has a simple and intuitive user interface with basic visual elements.

**Limitations:**

*Limited Natural Language Understanding:*

The current implementation relies heavily on simple pattern matching. It may struggle to understand complex or nuanced user queries.

*Limited Context Awareness:*

The chatbot does not maintain context across multiple user turns in a conversation.

*Limited Personalization:*

The chatbot provides generic responses and lacks personalization for individual users.

*Basic Response Generation:*

The chatbot currently provides pre-defined responses. More sophisticated response generation techniques, such as machine learning, are needed to improve the quality and relevance of responses.

**Future Directions:**

*Improve Natural Language Understanding:*

Connect to advanced NLP libraries that increase the understanding of the user's intent and sentiment.

*Apply Machine Learning:*

Train a machine learning model that will result in more personal and relevant answers.

*Enhance User Interface*:

Improve visual attractiveness and usability.

*Expand Functionality:*

Add voice interaction, multilingual capabilities, and connect to other services.

*Reduce Limitations*:

Increase the complexity level of handling questions, improve the ability to remember context, and personalize the answers.

**Conclusion:**

This is a prototypical model of a simple email support chatbot. Basically simple, it still has vast potential for improvement regarding natural language understanding, response generation, and the general user experience. More development and tuning are required to build a genuinely efficient and helpful chatbot for email support.

**Chapter - 10**

**APPENDICES**

**Appendix A: JavaScript Code Comments**

// getBotResponse() function

function getBotResponse(userMessage) {

// ... (existing code)

// Check for specific email issues

if (/can't send|not sending|sending failed/.test(lowerCaseMessage)) {

// ... (response)

}

// ... (other checks)

// Default response

return "I'm still learning. Please rephrase your question or provide more details.";

}

function appendMessage(message, type, mediaUrl = null) {

const messageDiv = document.createElement('div');

messageDiv.classList.add('message', `${type}-message`, 'animate\_\_animated', 'animate\_\_fadeIn');

messageDiv.textContent = message;

if (mediaUrl) {

const mediaElement = mediaUrl.startsWith('data:image') ? document.createElement('img') : document

.createElement('video');

mediaElement.src = mediaUrl;

mediaElement.controls = mediaUrl.startsWith('data:video');

mediaElement.classList.add('message-media');

messageDiv.appendChild(mediaElement);

}

// Check for specific email issues

if (/can't send|not sending|sending failed/.test(lowerCaseMessage)) {

// ... (response)

}

// ... (other checks)

// Default response

return "I'm still learning. Please rephrase your question or provide more details.";

}

**Appendix B: Design Considerations**

*User Interface (UI) Considerations:*

Color scheme and fonts should be aesthetic and readable.

Layout should be clean, not cluttered, and easy to navigate.

It should be compatible with accessibility for people with disabilities, such as screen reader.

*User Experience (UX) Considerations:*

The chatbot should be intuitive and easy to understand.

Responses should be concise, informative, and easy to read.

The chatbot should give a clear and concise indication of its capabilities and limitations.

**Appendix C: Testing Scenarios**

*Basic Functionality:*

Test greetings and farewells.

Test common email-related issues (sending, receiving, account problems).

Test general inquiries (contact us, hours of operation).

*Edge Cases:*

Test with misspelled words or typos.

Test with ambiguous or nonsensical input.

Test with long or complex sentences.

*User Interface Testing:*

Test the responsiveness of the interface on different screen sizes.

Test the interface for accessibility by people with disabilities.

**Appendix D: Future Enhancements (Detailed)**

*NLP Integration:*

Use libraries such as spaCy, NLTK, or Transformers to enhance natural language understanding.

Integrate sentiment analysis to identify the emotions of the user and respond accordingly.

*Machine Learning:*

Train a machine learning model on a large dataset of user queries and responses.

Implement dialogue management for more context-aware conversations.

Enable personalized responses based on user history.

*Integration with External Services:*

Integrate with email providers like Gmail and Outlook for more specific help.

Connect to knowledge base articles or FAQs for more detailed information.

*User Interface Enhancements:*

Implement a more advanced and aesthetically pleasing UI.

Add features such as voice input and output.

Improve accessibility and cross-browser compatibility.

**Limitations:**

Need more efficient .

Access all the medias, voice recognitions, NLP system.

Location track and image recognitions.

**LIST OF THE REFERENCES :**

1. Marijn Haverbeke,(2019), ‘ Eloquent JavaScript 3rd Edition’.

“Eloquent JavaScript 3rd Edition" by Marijn Haverbeke, published in **2018** (though some sources might list 2019 due to its release timeframe), is a highly regarded book for learning JavaScript. It's known for its comprehensive approach, clear explanations, and its ability to take readers from the fundamentals of programming to more advanced concepts within the JavaScript language and its environments (browsers and Node.

* This book is very effective , to making our core code logics. And this book efficient for logical purpose.

1. Ethan Marcotte,(Second Edition-2018), ‘Responsive Web Design’.

The book you're referring to is **"Responsive Web Design" by Ethan Marcotte, Second Edition (published in 2014, although some sources might still refer to the initial impact of the first edition from 2011).** This book is a highly influential and foundational text in the field of web design.

* This book web base which is very helpful to make all the styling purpose. And it has best part of CSS.

1. Steve McConnell,(Second Edition-2004), ‘Code Complete: A Practical Handbook of Software Construction’.

"Code Complete: A Practical Handbook of Software Construction, Second Edition" by Steve McConnell, published in 2004, is widely considered a **classic and essential resource for software developers of all levels of experience.** It's not about the theoretical aspects of computer science but rather a deep dive into the practicalities of writing high-quality code.

* Here we work this book a reference of all web base projects , and all kind of logics and styling , modelling and which very increase to make out project   
  “EMAIL SUPPORT CHATBOT”.