E- COMMERCE HELP BOT

(Mini-Project)

A project report submitted to the Srinivas University as partial fulfilment for the award of the degree of

Bachelor of Technology in Cloud Technology and Information Security

Submitted By

HRITHWIK. K

USN: 1SU19CI012

Under the Guidance of

Mr. Daniel Selvaraj

Assistant Professor

Department of Cloud Technology & Data Science

College of Engineering and Technology

SRINIVAS UNIVERSITY

Mukka, Mangalore – 574146

January 2022

BONAFIDE CERTIFICATE

This is to certify that this project report entitled "**E- COMMERCE HELP BOT**" is submitted to Srinivas University College of Engineering and Technology, Mukka, is a bonafide record of work done by **HRITHWIK. K** under my supervision from 1ST November of 2022 to 30th of November 2022

Mr. Daniel Selvaraj
Professor

Prof. Daniel Selvaraj
Head of Department
Department of Cloud Technology and Data Science
Srinivas University, Mukka

Date:
Place: Mukka

TABLE OF CONTENT

ABSTRACT

The main aim of the paper is to introduce a online shopping app with integrated chatbot that saves the time of the user and help to resolve their query at a moment. Online shopping app consist of different functionalities and features such as proper categorization of the product, trending results, fashion, electronics, personal care, recent searched product etc. It also has a good interface that enhances the user experience with the application. that will save the time of the customer and customer/user will have clear idea about the product and will have the benefit to choose the product wisely. The chatbot will give the reply to the customer query in the form of “links of the product and details about the product”, “chat support”,” recommend product”, “customer services” and hence visually impaired people can use the app easily and will have the clear idea about the product. It also help the user who are not familiar with the products. The chatbot will provide relevant information to the user as per their query. This application works in various types of browsers as well as in various network.

1. INTRODUCTION

1.1 THE DOMAIN

Chatbot

A chatbot or chatterbot is a Software application used to conduct an on-line chat Conversation via text in lieu of providing direct contact with a live human agent. Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse, while none of them can pass the standard.

Chatbots are used in dialog system for various purposes including customer service, request routing, or information gathering. While some chatbot applications use extensive word-classification processes, natural-language processors, and sophisticated AI, others simply scan for general keywords and generate responses using common phrases obtained from an associated library or Database.

Ecommerce chatbots are computer programs that interact with website users in real time. They provide customer service, answer questions, recommend products, gather feedback, and track engagement.

There are three types of ecommerce chatbot platforms:

1. Artificial intelligence (AI) driven chatbots that use natural language processing and conversational AI. This mimics human behaviour and speech patterns.
2. Rules-based chatbots that offer answers to predefined questions that a user may ask.
3. Hybrid chatbots that offer AI and rules-based conversations.

1.2 THE PROBLEM

This is because ecommerce chatbots can:

1. **Answer FAQs**
2. **Engage customers**
3. **Automate sales**

Answer FAQs

Think of an ecommerce chatbot as an employee who knows (almost) everything. They're always available and never get tired of answering the same question. FAQ chatbots can answer questions, and push customers to the next step in their user journey.

This is a major time and money-saver for online retailers. They can outsource routine tasks and focus on personalized customer service. It also means that customers will always have someone (or something) on the other end of a chat window.

Engage customers

AI-powered ecommerce chatbots provide an interactive experience for users. They answer questions, offer information, and recommend new products and or services. And they do this at any time of the day.

The always-on nature of ecommerce chatbots is key to their effectiveness. Without one, retailers would miss the opportunity to interact with some users. This is a missed opportunity to create brand loyalty and land a sale.

Instead, chatbots ensure that retailers can always:
• Ask the customer if they need help
• Gather information about their needs and interests
• Make suggestions based on that information
• Guide them to similar content or products
• Walk them through the sales process
• Re-engage past customers to encourage repeat business

Automate sales

With ecommerce chatbots, retailers can:
• Walk customers through the buyer's journey
• Offer them customized product suggestions
• Inform them about sales and promotions
• Prompt them to the next stage in the checkout process

1.3 THE TECHNOLOGY

Rules-based chatbot software performs pre-programmed behaviors based on "playbooks" you create on the user interface's backend module. Like a digital assistant, rules-based chatbot technology can behave in a certain way based on click activities and simple event triggers like a "yes" or a "no" input. It may also detect a specific keyword or combination of phrases

For instance, you could program a rules-based chatbot to answer not only if someone chooses "black" or "white," but also if they say "I want a black item" – the chatbot's backend module will match the term black with a preconfigured rule.

Live chat is a type of chat system that sits on the webpage or in your mobile application and works as a consumer's window to your support team and contact center. Using this mechanism, chatbots incorporate routing capabilities to assign discussions in real-time.

When a customer needs to communicate with a representative from your team, the chatbot scans agent availability and routes the discussion request accordingly. It will connect the customer with someone who can help them with their problem – i.e., an agent with the right skills and knowledge. The chatbot also alerts the agent when there is a customer query and informs the customer about agent details like their name, waiting time, etc.

As you can see, these processes are relatively understandable, given that advancements in chatbot technology today are endless and readily accessible to users and developers alike.

2. SYSTEM ANALYSIS

2.1 LITERATURE REVIEW

A chatbot is an automated AI software program that allows for human-bot interaction. These conversations can be implemented through text interfaces and voice interfaces. Besides, chatbots are embedded AI features that accompany websites and messenger applications and, in some instances, serve as standalone bots. Chatbots have several synonyms such as ChatterBot, Chat Robot, talk bot, bot, IM bot, and virtual assistants. They can be ontology or pattern-based [1]. Businesses are increasingly using artificial intelligence in conjunction with chatbots to interact with customers to provide a more personalized service experience for customers. Examples of such organizations include Lyft, Fandango, Spotify, Sephora, MasterCard, Staples, and The Wall Street Journal [2–4].

In this work, a chatbot will be implemented to solve an e-commerce problem within an academic environment, specifically Covenant University, Ota. Therefore, the goal of this literature review is to study the application of chatbots in various contexts. In the section that follows, studies related to the application of chatbots in e-commerce and non-e-commerce contexts are examined to identify a gap concerning the concept of chatbots within the literature.
Text-based chatbots are bots whose primary mode of communication is through texting or messaging. They also come with additional features such as images, videos, and quick replies [17, 18]. Humans are sometimes unable to tell the difference between a text-based chatbot conversation and a human conversation during real-time conversations. However, methodologies, such as CAPTHCHA, keyword detection, and dialogue correlation [19] are used to unravel this. Furthermore, Mori et al. [20] suggest that text-based chatbots meet the reasonable answer criteria but lack emotions and attitude, which can be easily identified in human conversation. In light of advanced virtual assistants like Cortana and Siri, text-based messaging services are “cheap, fast, democratic and popular” and, especially for young people, the preferred way of communication [21].

Angga et al. [22] propose a chatbot design with an avatar (3D) and voice interaction elements to make the conversation more intelligible. Kraus et al. [23] investigated the factors for customers’ satisfaction in voice commerce and e-commerce. In the study, four factors were considered, which led to the author’s generation of four hypotheses. These factors include Recommendation Complexity, Recommendation Personalization, Convenience, and Transaction Process Efficiency.
NLP explores how computers can understand and manipulate natural language text or speech to do useful things [23]. Unlike ontology-based chatbots, pattern-based chatbots have preprogrammed responses, which makes their conversation unnatural [25].

2.2 EXISTING SYSTEMS

Slush – Answer FAQs in real time
Customers expect an immediate response when they reach out to a company for any support. To deliver real-time and high-quality support to a large number of customers is not an easy job. An AI enabled chatbot is the best option to handle the 'n' number of conversations with 24×7 engagement.

Vainu – Enrich customer conversations without form fill-ups
A combination of a perfect lead generation strategy and chatbots can bring your business a good number of leads. Filling up forms used to be the traditional method of generating sales leads.
Well, the chance of converting your customers via lead generation forms is between 2.5% to 5%.

Dominos – Deliver a smooth customer experience via Facebook messenger
Chatbots are no longer restricted to enterprises and different business verticals but it has significant use cases for consumers as well. 1 in 5 consumers would consider purchasing goods and services from a chatbot.
Dominos has built a Facebook chatbot to make the ordering process faster. The famed restaurant offers the easiest way to order a pizza from "Anywhere".

HDFC Bank – Help your customers with instant answers
The utility of financial chatbots is growing y the day as customers now expect prompt services all the time. When it comes to balancing finances or managing bills it can be a struggle to find the best solutions to navigate a numbers-and policy-heavy website. Banks can benefit themselves as well as their customers by implementing chatbot technology.

KLM Royal Dutch Airlines – Enhance your customer service with a multilingual FB bot
Artificial Intelligence has been a game-changer for the travel industry by helping companies to simplify travel arrangements and streamline business procedures. A travel chatbot is accessible 24×7, supports multiple languages and provides real-time
responses to travelers' queries, and is also available on the FB Messenger app.

Marriott Hotel – Meet your customer requests effectively
Indeed, hospitality is a vast industry, encompassing everything from transportation to restaurants but chatbots are excelling at all of them. This industry tightly focuses on customer service.
Here is a customer service chatbot example in the hospitality industry to get you started.
Marriott used chatbot implementation ideas and made them available to guests via text message. Bots allow guests to request basic hotel services, essentially acting

as an in-phone concierge. This exempts middleman involvement and enables requests to be met quickly and efficiently.

2.3 PROPOSED SYSTEM

In this web development tutorial, we will see how to create a basic chatbot using HTML, CSS, and vanilla JavaScript. This exercise is focused on JS fundamentals rather than any kind of Artificial Intelligence (AI). To make the process simpler and easier to learn, I am not using any third-party libraries. I will be building the chatbot from scratch by focusing on the basics of the JavaScript programming language.

Creating an HTML file
First, start by creating an HTML file. Let's create a file name index.html, containing the following code:

Here, we have created a div with id 'main' and inside it we have an input field in which the user can type text and communicate with the bot. This is the basic HTML set-up. We will be adding more functionalities along the way.

Next, we also need to add some layout styling to our webpage. So let us create a CSS file named style.css. Use the following stylesheet code:

Creating the .js File
Now, create a .js file and then set up some basic functionality into it with the following JavaScript code:

that adding the DOMContentLoaded listener here is ensuring that the JavaScript will load only when the HTML has finished rendering. It's good coding practice, by the way.

Some users find it a tedious task to click the send button, again and again, to send messages over a chat app. So we want to implement an instant messaging feature into this app that will enable the user to send the message just by pressing the enter key after they finish typing the message.

'Enter', we are telling the event listener that we only care about the enter key. We have replaced the console log with some statements which are performing these actions:

Assignment of the input field value to the local variable.
We also want to clear the form input field after submission, so we set the input field's value to the empty string.
Passing the input value to our next function 'output'

JavaScript Functions for Chatbots

In the next step, we will create some JavaScript functions that will make a bot.

First, we want to take control over the text the user types in the input form field. Whatever the user types in the input field, we want to make it a little more standard using regular expressions as done in the following code snippet:

The above regular expression will remove everything other than words, digits, and spaces. We also want to remove all rogue characters and everything that could make the matches difficult. So it can be done in the following ways:

Programming Bot Responses in JavaScript
Our next task is to work on the bot responses based on the triggers (user text). So I am going to create arrays of arrays that include utterances and the answers. For simplicity, I am going to keep them brief.

If you notice the indexes of each array and how they are lined up, you will see that the user text that matches an option at utterances[0] will respond with an option from answers[0], and so on. The alternatives array is, of course, for those input values that do not match in the first array. This explains how a basic chatbot works under the hood.

Chatbot Triggers and Responses
Now we can get familiar with how the triggers and responses look. The next step is to create a function that will compare these arrays:

After comparing arrays, our ‘output’ function finally calls the addChatEntry() function to make the conversation render on the webpage.

Updating the DOM
The last thing remaining now is to update the DOM. So we must create a function addChatEntry():

This function creates a thread of messages for both the bot and user fields, on the same page.

Create a JavaScript Chatbot
In the setTimeout function, we set the time delay by two seconds to make the impression that the bot thinks for a few seconds and then replies when someone talks to it.

2.4 HARDWARE AND SOFTWARE SPECIFICATIONS

windows requirement
Operating systemn: windows 8.1 or latest
Processor: intel pentiumor latest, amd ryzen
Memory: 2gb minimum, 4gb recommended
Storage type: solid state drive, hard disk drive
Storage size: 128gb
Application window size: 1024x680
Screen resolution: 1280x1024

Mac requirement
Operating systemn: macOS high sierra 10.13 or latest
Processor: intel pentiumor latest, Apple Silicon chip
Memory: 4gb recommended
Storage type: solid state drive, hard disk drive
Storage size: 128gb
Application window size: 1024x680
Screen resolution: 1280x1024

Linux Requirement
Operating systemn: 64-bitUbuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
Processor: intel pentiumor latest, amd ryzen
Memory: 2gb minimum, 4gb recommended
Storage type: solid state drive, hard disk drive
Storage size: 128gb
Application window size: 1024x680
Screen resolution: 1280x1024

3. SYSTEM DESIGN

3.1 MODULES DESCRIPTION

The website has the following flow:

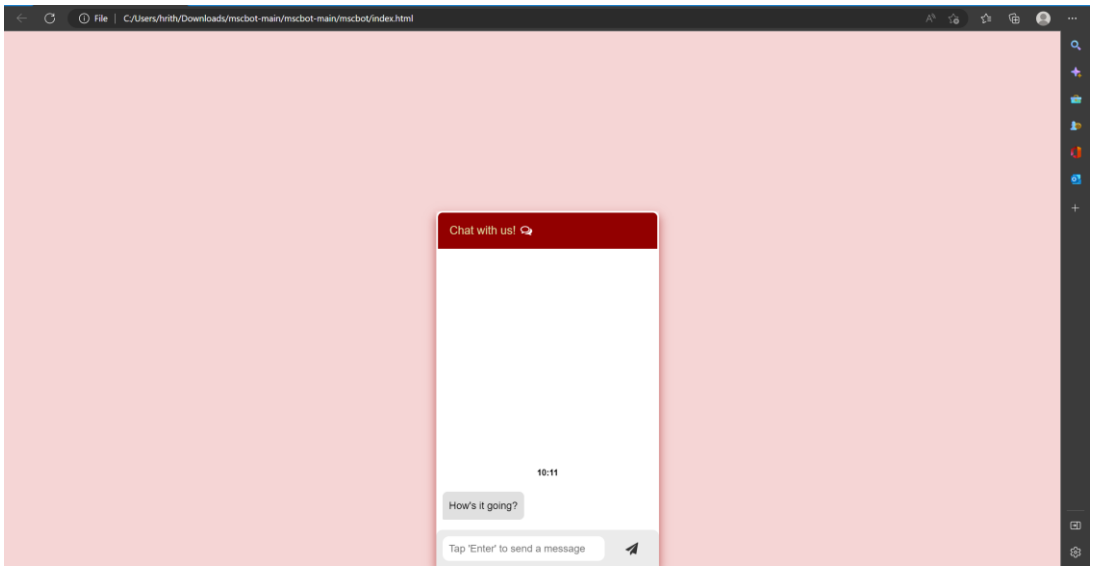
1. A user can chat with the bot varifing about the product and prizing and Other details.
2. Can view about the product.
3. Can view wheather the product is genune.

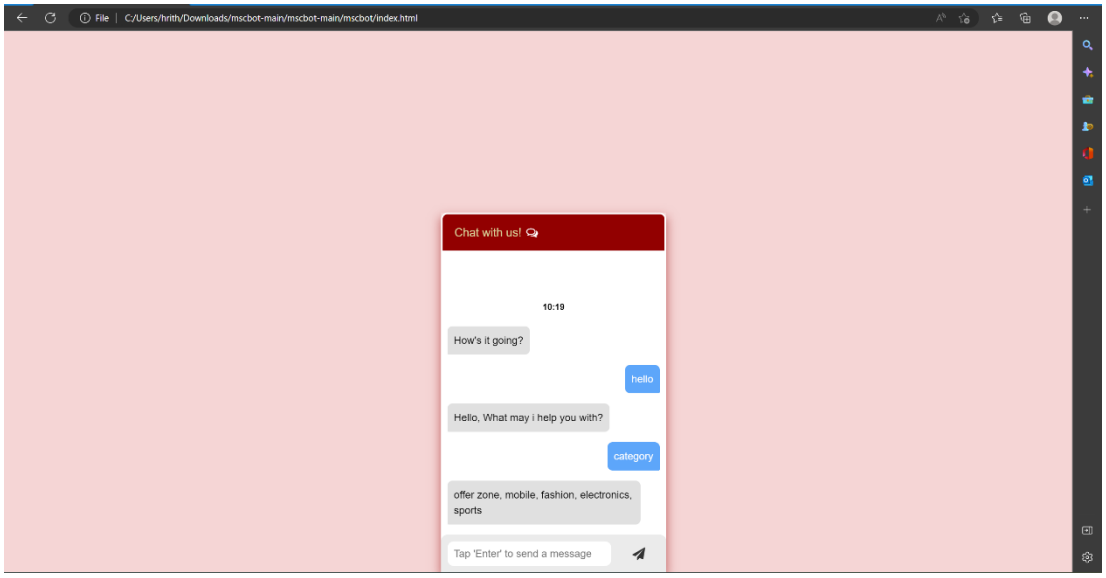
The webpage contains 1 page:

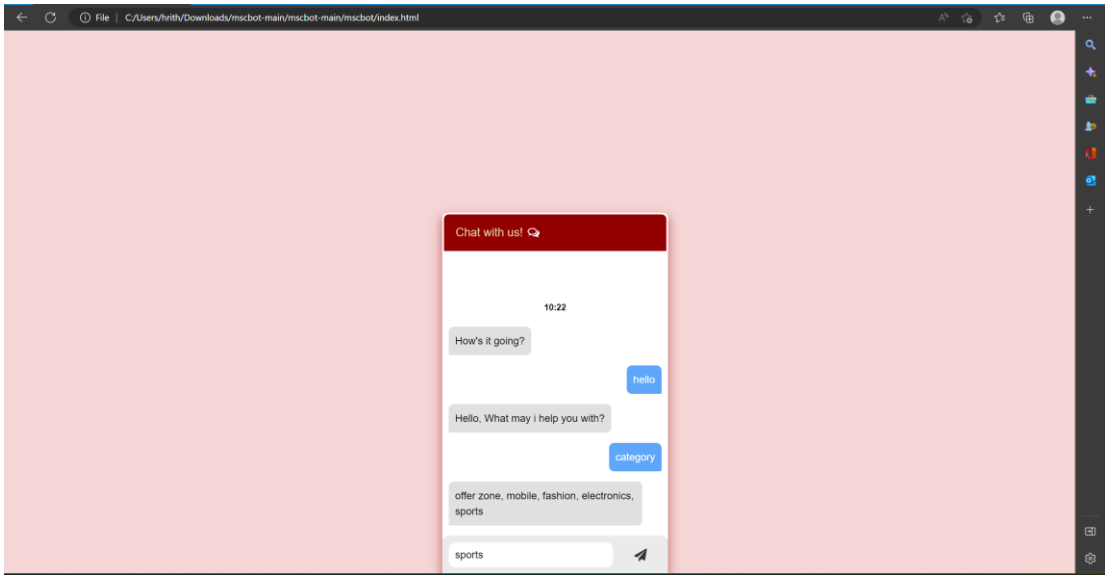
1. Home page with chatbot

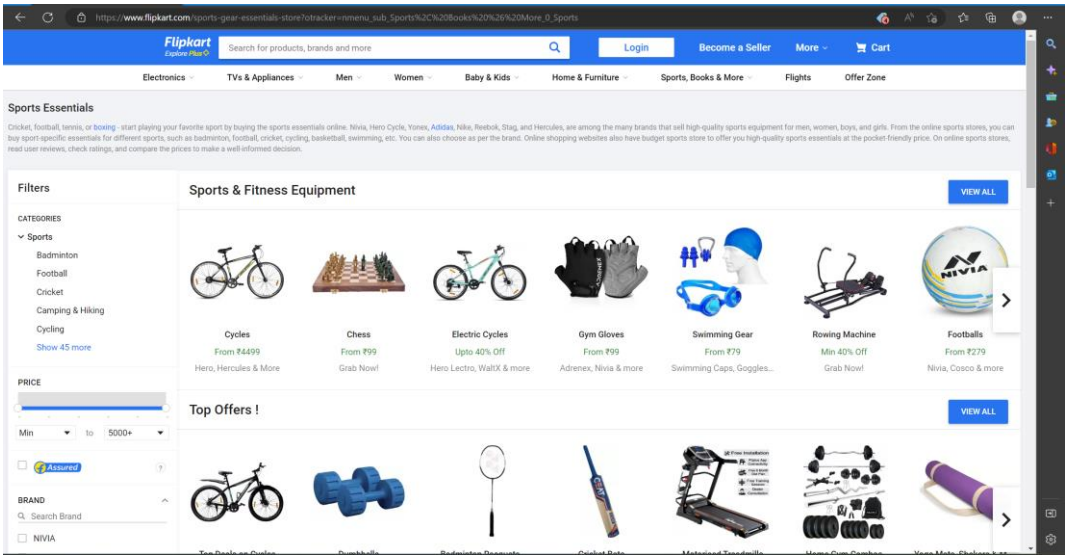
The Home Page has 5 elements:

1. Headline
2. collapsible chat box
3. send icon

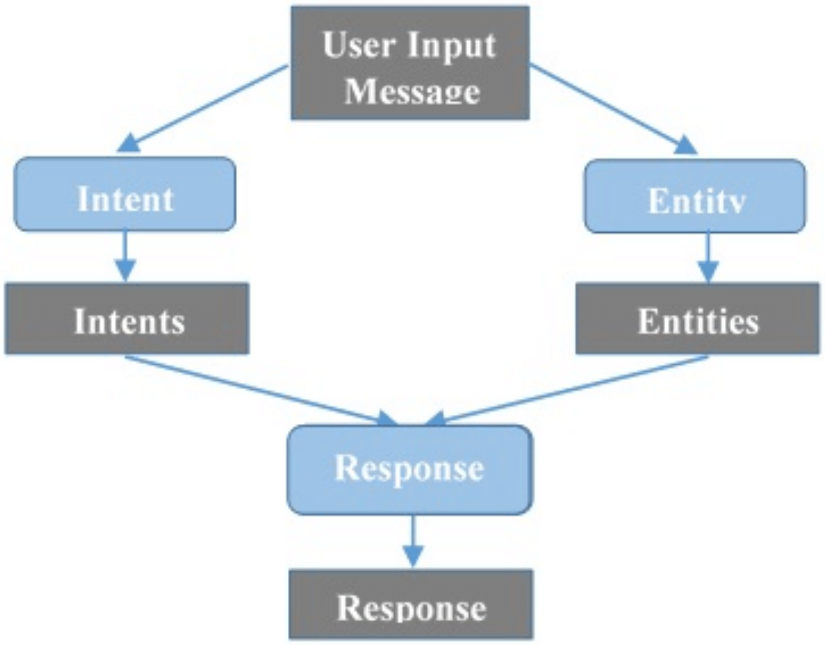


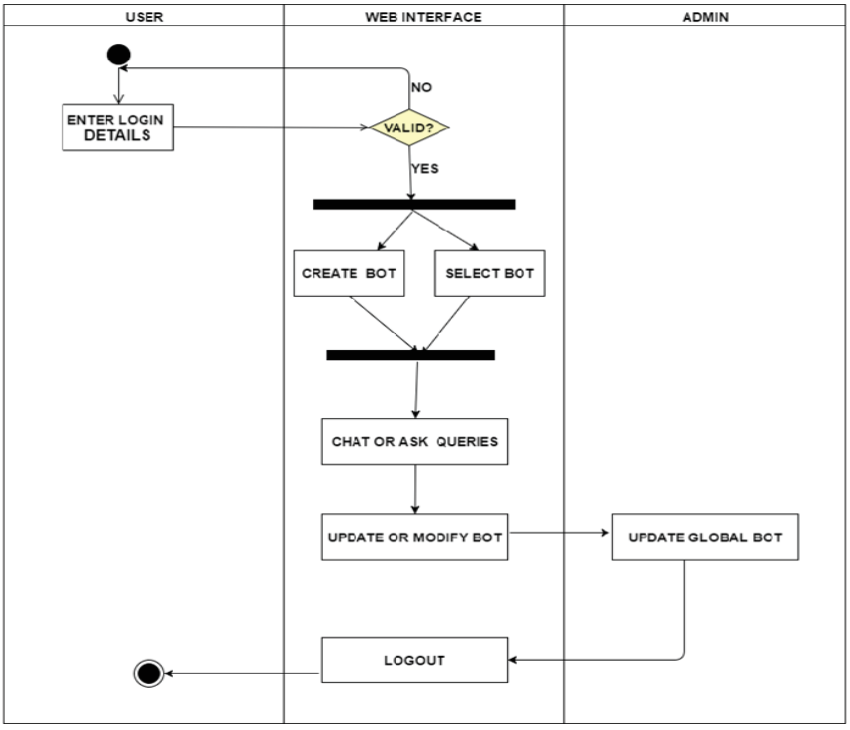

File | C:/Users/hrith/Downloads/mscbot-main/mscbot-main/mscbot/index.html
Chat with us!
10:19
How's it going?
hello
Hello, What may i help you with?
category
offer zone, mobile, fashion, electronics, sports
Tap 'Enter' to send a message


File | C:/Users/hrith/Downloads/mscbot-main/mscbot-main/mscbot/index.html
Chat with us!
10:22
How's it going?
hello
Hello, What may i help you with?
category
offer zone, mobile, fashion, electronics, sports
sports

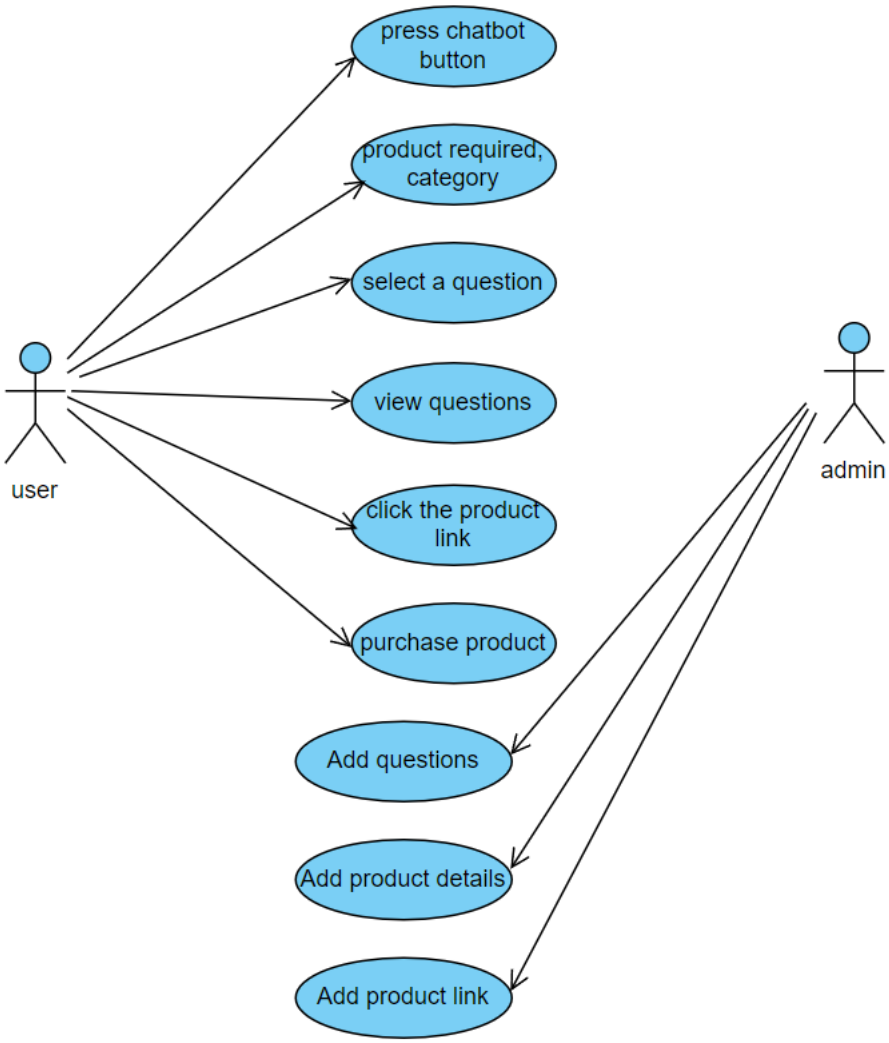

Flipkart
Search for products, brands and more
Login
Become a Seller
More
Cart
Electronics
TVs & Appliances
Men
Women
Baby & Kids
Home & Furniture
Sports, Books & More
Flights
Offer Zone
Sports Essentials
Filters
CATEGORIES
Sports
Badminton
Football
Cricket
Camping & Hiking
Cycling
Show 45 more
PRICE
BRAND
NIVIA
Sports & Fitness Equipment
VIEW ALL
Cycles
From ₹4499
Hero, Hercules & More
Chess
From ₹99
Grab Now!
Electric Cycles
Upto 40% Off
Hero Lectro, WaltX & more
Gym Gloves
From ₹99
Adrenex, Nivia & more
Swimming Gear
From ₹79
Swimming Caps, Goggles...
Rowing Machine
Min 40% Off
Grab Now!
Footballs
From ₹279
Nivia, Cosco & more
Top Offers !
VIEW ALL

3.2 ARCHITECTURE DIAGRAM

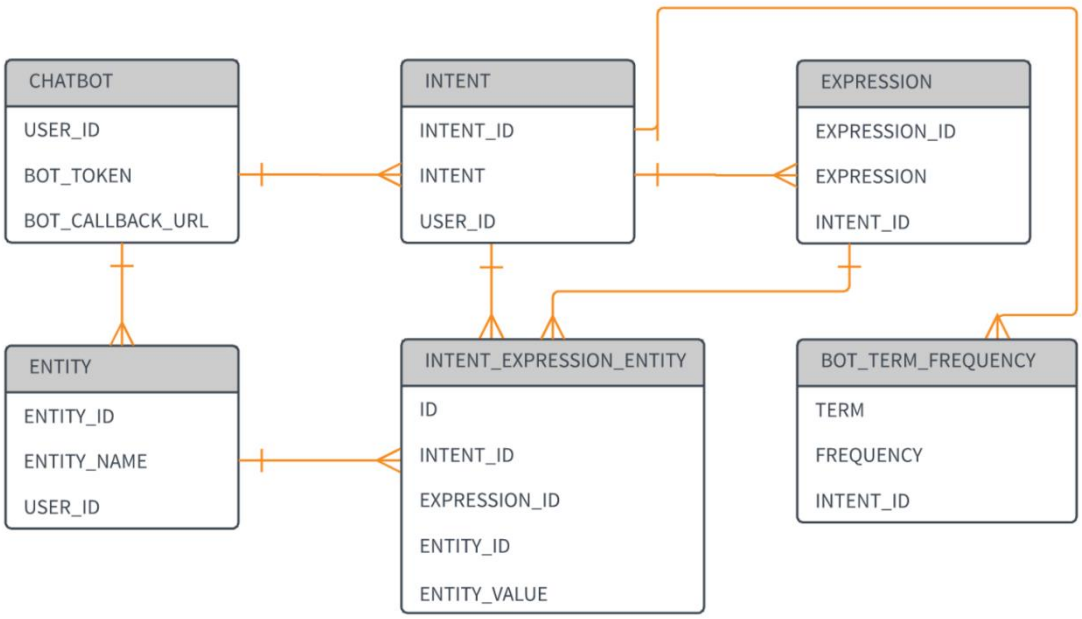




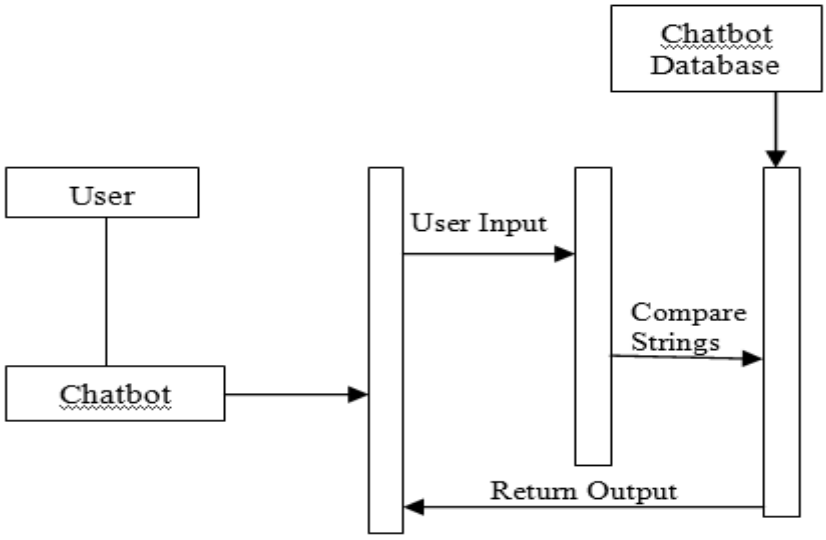
3.3 **USE CASES**


press chatbot button
product required, category
select a question
view questions
click the product link
purchase product
Add questions
Add product details
Add product link
user
admin

3.4 CLASS DIAGRAM



3.5 SEQUENCE DIAGRAM



4. IMPLEMENTATION

FRONT END: HTML

mscbot-main > mscbot > index.html > html > body > div.chat-bar-collapsible

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chat Bot</title>

    <link rel="stylesheet" href="static/css/chat.css">
    <link rel="stylesheet" href="static/css/home.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

</head>

<body>
    <!-- CHAT BAR BLOCK -->
    <div class="chat-bar-collapsible">
        <button id="chat-button" type="button" class="collapsible">Chat with us!
            <i id="chat-icon" style="color: #fff;" class="fa fa-fw fa-comments-o"></i>
        </button>

        <div class="content">
            <div class="full-chat-block">
                <!-- Message Container -->
                <div class="outer-container">
                    <div class="chat-container">
                        <!-- Messages -->
                        <div id="chatbox">
                            <h5 id="chat-timestamp"></h5>
                            <p id="botStarterMessage" class="botText"><span>Loading...</span></p>
                        </div>

                        <!-- User input box -->
                        <div class="chat-bar-input-block">
                            <div id="userInput">
                                <input id="textInput" class="input-box" type="text" name="msg"
                                    placeholder="Tap 'Enter' to send a message">
                                <p></p>
                            </div>

                            <div class="chat-bar-icons">
                                <i id="chat-icon" style="color: #333;" class="fa fa-fw fa-send"
                                    onclick="sendButton()"></i>
                            </div>
                        </div>

                        <div id="chat-bar-bottom">
                            <p></p>
                        </div>

                    </div>
                </div>

            </div>
        </div>

    </div>

</body>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="static/scripts/responses.js"></script>
<script src="static/scripts/chat.js"></script>

</html>
```

FRONTEND DESIGN: CSS CODE

mscbot-main > mscbot > static > css > # chat.css > .collapsible

```css
.chat-bar-collapsible {
    position: fixed;
    bottom: 0;
    right: 38%;
    box-shadow: 0 8px 16px 0 rgb(191, 87, 87);
}

.collapsible {
    background-color: rgb(146, 0, 0);
    color: rgb(219, 215, 157);
    cursor: pointer;
    padding: 18px;
    width: 350px;
    text-align: left;
    outline: none;
    font-size: 18px;
    border-radius: 10px 10px 0px 0px;
    border: 3px solid white;
    border-bottom: none;
}

.content {
    max-height: 0;
    overflow: hidden;
    transition: max-height 0.2s ease-out;
    background-color: #f1f1f1;
}

.full-chat-block {
    width: 350px;
    background: white;
    text-align: center;
    overflow: auto;
    scrollbar-width: none;
    height: max-content;
    transition: max-height 0.2s ease-out;
}

.outer-container {
    min-height: 500px;
    bottom: 0%;
    position: relative;
```

```
}

.chat-container {
    max-height: 500px;
    width: 100%;
    position: absolute;
    bottom: 0;
    left: 0;
    scroll-behavior: smooth;
    hyphens: auto;
}

.chat-container::-webkit-scrollbar {
    display: none;
}

.chat-bar-input-block {
    display: flex;
    float: left;
    box-sizing: border-box;
    justify-content: space-between;
    width: 100%;
    align-items: center;
    background-color: rgb(235, 235, 235);
    border-radius: 10px 10px 0px 0px;
    padding: 10px 0px 10px 10px;
}

.chat-bar-icons {
    display: flex;
    justify-content: space-evenly;
    box-sizing: border-box;
    width: 25%;
    float: right;
    font-size: 20px;
}

#chat-icon:hover {
    opacity: .7;
}

/* Chat bubbles */
```

```css
#userInput {
    width: 75%;
}

.input-box {
    float: left;
    border: none;
    box-sizing: border-box;
    width: 100%;
    border-radius: 10px;
    padding: 10px;
    font-size: 16px;
    color: #000;
    background-color: white;
    outline: none
}

.userText {
    color: white;
    font-family: Helvetica;
    font-size: 16px;
    font-weight: normal;
    text-align: right;
    clear: both;
}

.userText span {
    line-height: 1.5em;
    display: inline-block;
    background: #5ca6fa;
    padding: 10px;
    border-radius: 8px;
    border-bottom-right-radius: 2px;
    max-width: 80%;
    margin-right: 10px;
    animation: floatup .5s forwards
}

.botText {
    color: #000;
    font-family: Helvetica;
    font-weight: normal;
```

```
    font-size: 16px;
    text-align: left;
}

.botText span {
    line-height: 1.5em;
    display: inline-block;
    background: #e0e0e0;
    padding: 10px;
    border-radius: 8px;
    border-bottom-left-radius: 2px;
    max-width: 80%;
    margin-left: 10px;
    animation: floatup .5s forwards
}

@keyframes floatup {
    from {
        transform: translateY(14px);
        opacity: .0;
    }
    to {
        transform: translateY(0px);
        opacity: 1;
    }
}

@media screen and (max-width:600px) {
    .full-chat-block {
        width: 100%;
        border-radius: 0px;
    }
    .chat-bar-collapsible {
        position: fixed;
        bottom: 0;
        right: 0;
        width: 100%;
    }
    .collapsible {
        width: 100%;
        border: 0px;
        border-top: 3px solid white;
        border-radius: 0px;
    }
}
```

HOME PAGE DESIGN

mscbot-main > mscbot > static > css > # home.css > body

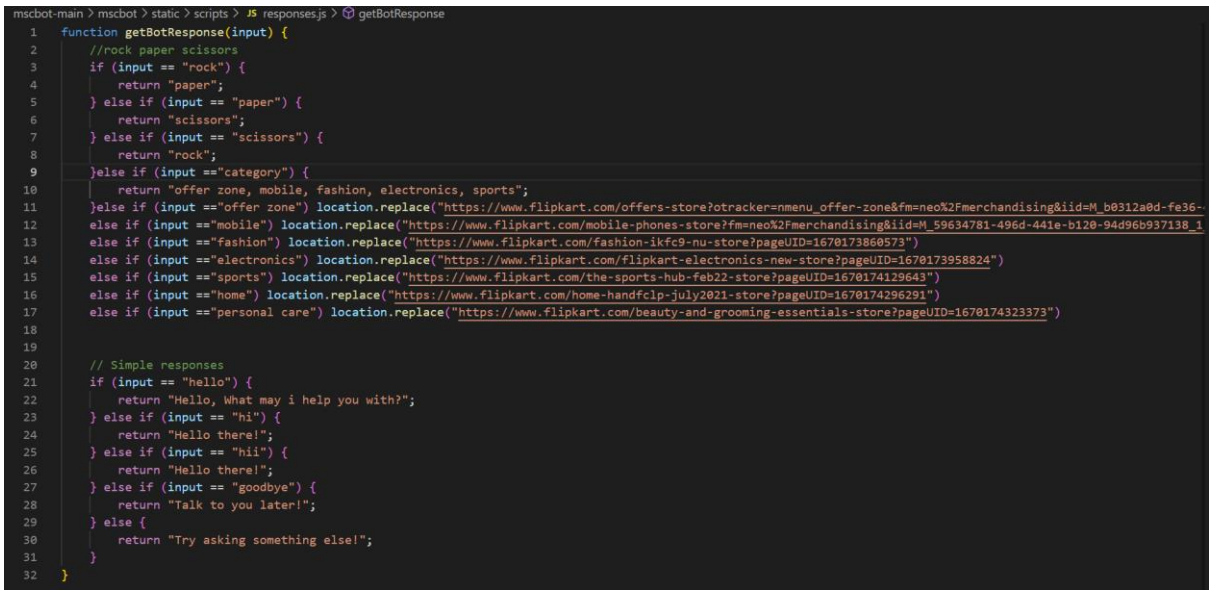
```
html {
    scroll-behavior: smooth;
    font-family: Helvetica, sans-serif, Arial;
}

body {
    margin: 0 auto;
    background-color: rgb(245, 213, 213);
    position: fixed;
    display: flex;
    align-items: center;
}
```

JAVA SCRIPT: CHAT

mscbot-main > mscbot > static > scripts > JS chat.js > getTime

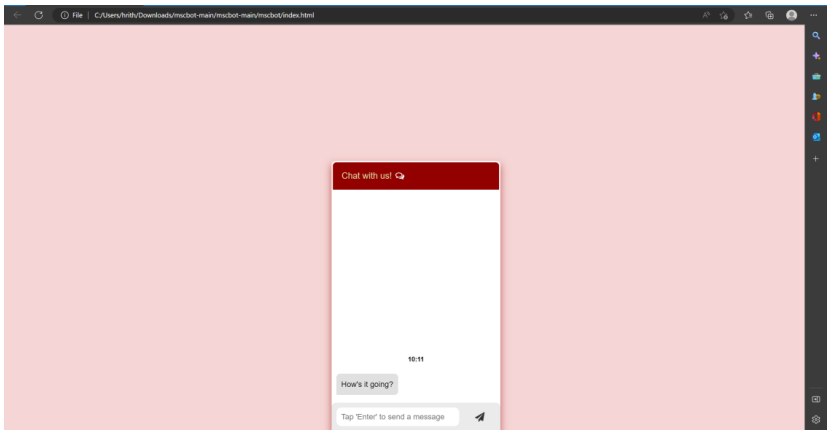
```
// Collapsible
var coll = document.getElementsByClassName("collapsible");

for (let i = 0; i < coll.length; i++) {
    coll[i].addEventListener("click", function () {
        this.classList.toggle("active");

        var content = this.nextElementSibling;

        if (content.style.maxHeight) {
            content.style.maxHeight = null;
        } else {
            content.style.maxHeight = content.scrollHeight + "px";
        }

    });
}

function getTime() {
    let today = new Date();
    hours = today.getHours();
    minutes = today.getMinutes();

    if (hours < 10) {
        hours = "0" + hours;
    }

    if (minutes < 10) {
        minutes = "0" + minutes;
    }

    let time = hours + ":" + minutes;
    return time;
}

// Gets the first message
function firstBotMessage() {
    let firstMessage = "How's it going?"
    document.getElementById("botStarterMessage").innerHTML = '<p class="botText"><span>' + firstMessage + '</span></p>';

    let time = getTime();

    $("#chat-timestamp").append(time);
    document.getElementById("userInput").scrollIntoView(false);
}

firstBotMessage();

// Retrieves the response
function getHardResponse(userText) {
    let botResponse = getBotResponse(userText);
    let botHtml = '<p class="botText"><span>' + botResponse + '</span></p>';
    $("#chatbox").append(botHtml);

    document.getElementById("chat-bar-bottom").scrollIntoView(true);
}

//Gets the text text from the input box and processes it
function getResponse() {
    let userText = $("#textInput").val();

    if (userText == "") {
        userText = "I love Code Palace!";
    }

    let userHtml = '<p class="userText"><span>' + userText + '</span></p>';

    $("#textInput").val("");
    $("#chatbox").append(userHtml);
    document.getElementById("chat-bar-bottom").scrollIntoView(true);

    setTimeout(() => {
        getHardResponse(userText);
    }, 1000)

}

// Handles sending text via button clicks
function buttonSendText(sampleText) {
    let userHtml = '<p class="userText"><span>' + sampleText + '</span></p>';

    $("#textInput").val("");
    $("#chatbox").append(userHtml);
    document.getElementById("chat-bar-bottom").scrollIntoView(true);
}

function sendButton() {
    getResponse();
}

function heartButton() {
    buttonSendText("Heart clicked!")
}

// Press enter to send a message
$("#textInput").keypress(function (e) {
    if (e.which == 13) {
        getResponse();
    }
});
```

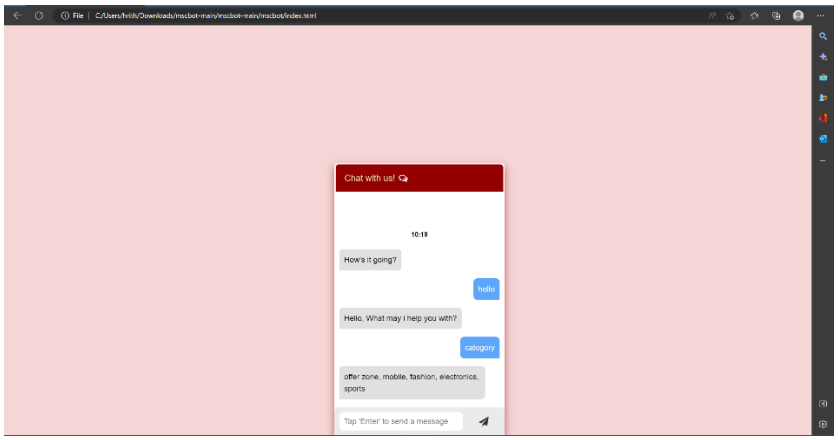
JAVASCRIPT RESPONSES

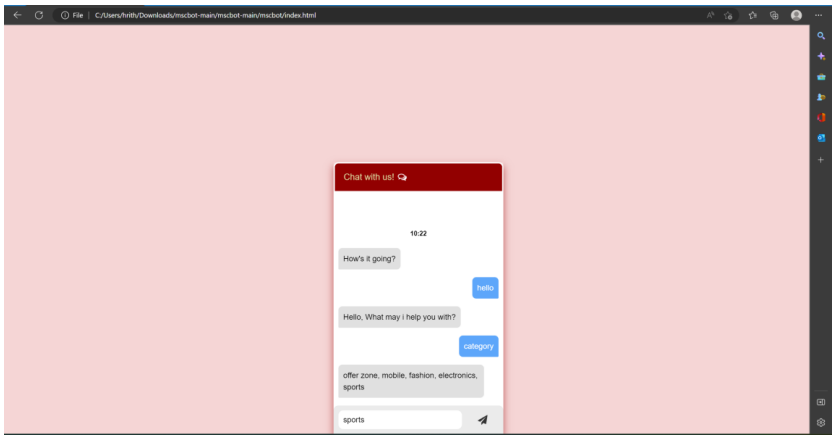


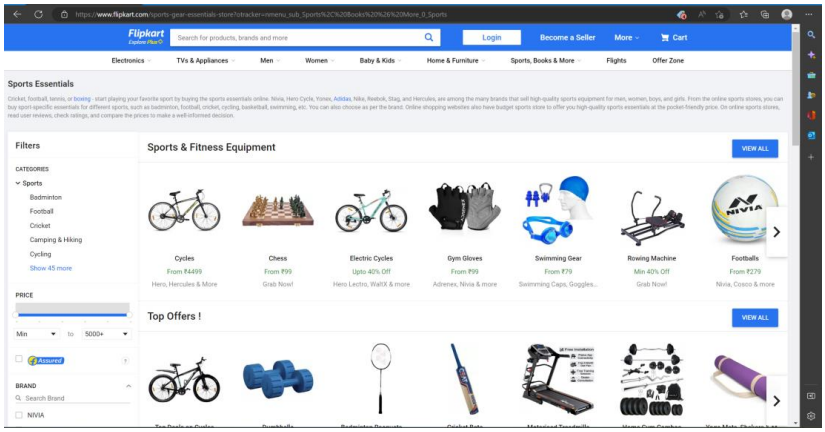
5. TESTING

- Testing chatbot
- Input: hello
- Input: category
- Input: sport








Flipkart
Search for products, brands and more
Login
Become a Seller
More
Cart
Electronics
TVs & Appliances
Men
Women
Baby & Kids
Home & Furniture
Sports, Books & More
Flights
Offer Zone
Sports Essentials
Filters
CATEGORIES
Sports
Badminton
Football
Cricket
Camping & Hiking
Cycling
Show 45 more
PRICE
Min
to
5000+
Assured
BRAND
Search Brand
NIVIA
Sports & Fitness Equipment
VIEW ALL
Cycles
From ₹4499
Hero, Hercules & More
Chess
From ₹99
Grab Now!
Electric Cycles
Upto 40% Off
Hero Lectro, WaitX & more
Gym Gloves
From ₹99
Adrenex, Nivia & more
Swimming Gear
From ₹79
Swimming Caps, Goggles...
Rowing Machine
Min 40% Off
Grab Now!
Footballs
From ₹279
Nivia, Cosco & more
Top Offers !
VIEW ALL

6. CONCLUSION AND FUTURE ENHANCEMENTS

The website contains all the basic functionalities that was set out in this project, but there always room for improvements. Some of those planned improvements are

1. This does not contains all types of different commands, add more responses
2. Better UI design and improving experiences
3. Adding features and products details
4. Files could also be deployed to cloud

7. REFERENCES

Alepis, E., & Virvou, M. (2011). Automatic generation of emotions in tutoring agents for affective e-learning in medical education. Expert Systems with Applications, 38(8): 9840–9847.

Ashok, G., Brian, C., Mithun, K., Shanu, S., Abhinaya, S., & Bryan, W. (2015). Using Watson for Enhancing Human-Computer Co-Creativity. AAAI Symposium: 22–29.

Avalverde, D. (2019). A Brief History of Chatbots. Perception, Control, Cognition. Retrieved March 9, 2019 from: https://pcc.cs.byu.edu/2018/03/26/a-brief-history-of-chatbots/

Ayedoun, E., Hayashi, Y., & Seta, K. (2015). A Conversational Agent to Encourage Willingness to Communicate in the Context of English as a Foreign Language. Procedia Computer Science, 60(1): 1433–1442.

Ben Mimoun, Mohammed Slim, & Poncin, I. (2015). A valued agent: How ECAs affect website customers' satisfaction and behaviors. Journal of Retailing and Consumer Services, 26: 70– 82.

Chatbot Magazine (2019). A Visual History of Chatbots. Retrieved March 9, 2019 from: https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2

Colace, F., De Santo, M., Lombardi, M., Pascale, L., Pietrosanto, A. (2018). Chatbot for E-Learning: A Cases Study. International Journal of Mechanical Engineering and Robotics Research Vol. 7, No. 5, September.

Egencia (2018). What is a Chatbot and How does it work? Retrieved March 9, 2019 from: https://www.youtube.com/watch?v=38sL6pADCog

Hattie, J. (2012). Visible learning for teachers: Maximizing impact on learning: Routledge. https://chatbotsmagazine.com/a-visual-history-of-chatbots-8bf3b31dbfb2

Lip ko, H. (2018). Meet Jill Watson: Georgia Tech's first AI teaching assistant. Retrieved on March 9, 2019 from: https://pe.gatech.edu/blog/meet-jill-watson-georgia-techs-first-ai-teaching-assistant.

Maruti Techlabs. (2018). Why can chatbots replace Mobile Apps immediately? Retrieved March 9, 2019 from: https://www.marutitech.com/why-can-chatbots-replace-mobile-apps-immediately/