TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

Abbreviation	Explanation
API	Application Programming Interface
COD	Cash On Delivery
RAM	Random Access Memory
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
IDE	Integrated Development Environment

CHAPTER 1

INTRODUCTION

While many would agree that chatbot has become a buzzword recently, but the concept has its existence from the time when people started developing ways to interact with computers. The first-ever chatbot was introduced even before the launch of personal computers. It was developed by MIT Artificial Intelligence Laboratory by Joseph Weizenbaum in 1966 and was named Eliza.

Eliza examined the keywords received as input and then triggered the output according to a defined set of rules. This methodology of generating output is still used by a number of chatbots.

The next chatbot was Parry, written by psychiatrist Kenneth Colby at the Stanford University in an attempt to simulate a person with paranoid schizophrenia. Then came the A.L.I.C.E, which was developed in 1995 by Richard Wallace. While A.L.I.C.E won the Loebner prize thrice, it failed to pass the Turing test. A Turing test examines whether or not a machine is able to think intelligently like humans
.

The next chatbot was Parry, developed by psychiatrist Kenneth Colby at the Stanford University. It was an attempt to simulate a person with paranoid schizophrenia. Then came the A.L.I.C.E, which was developed in 1995 by Richard Wallace. While A.L.I.C.E won the Loebner prize thrice, it failed to pass the Turing test. A Turing test examines whether or not a machine can think intelligently like humans.

After this, various virtual assistants were launched. Siri by Apple was the first one to introduce conversational assistants. The concepts gained popularity and soon after Google launched their Google Assistant for Android. Following the lead, Microsoft brought Cortana into existence.

Taking this a step forward, smart speakers were introduced which made voice conversation possible between humans and the bots. Amazon Alexa & Google Home represents a different category of conversational UI.

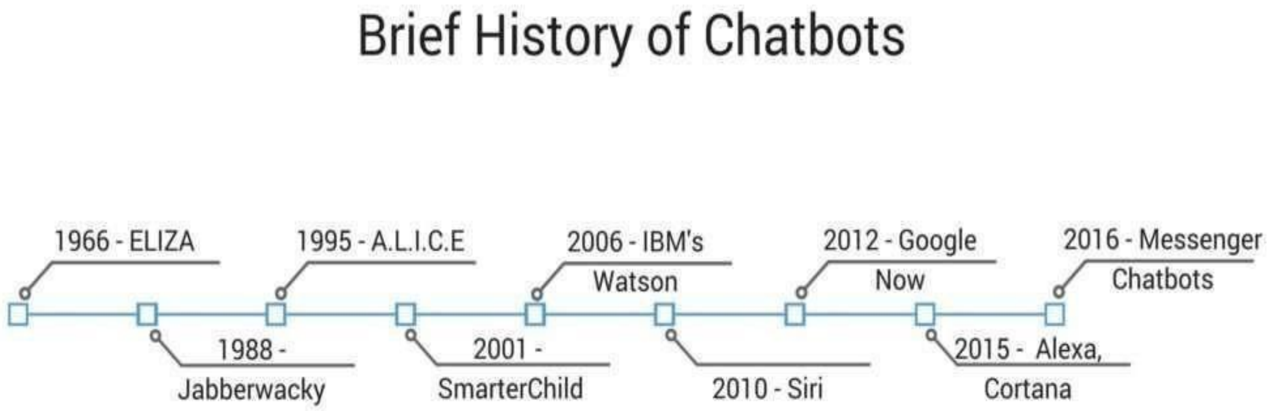


FIG 1.1

1.1 Statement of the problem

- Fulfilling the needs is a basic human quality. We humans acquire our needs ethically by buying it.
- Buying and selling of products and services had been done from times when started to civilize.
- Now buying a product is mainly done through ecommerce sites and there are thousands of ecommerce sites on the internet.
- So in this competitive field of business, we need to attract customers to be an imminent company. And to attract people we should have certain needed features that others do not possess and we should have it uniquely.

1.2 Objectives

The aims of this project are as follows:

- To allow the user to create the chatbot
- To allow our user to train the chatbot data
- To allow our user to modify the data of the chatbot

1.3 System Overview

Humans are constantly fascinated with auto-operating AI-driven gadgets. The latest trend that is catching the eye of the majority of the tech industry is chatbots. And with so much research and advancement in the field, the programming is winding up more human-like, on top of being automated. The blend of immediate response reaction and consistent connectivity makes them an engaging change to the web applications trend.

In general terms, a bot is nothing but a software that will perform automatic tasks. In other terms, a bot is a computer program that is designed to communicate with human users through the internet. This article will focus on the chatbot in the the class of bots that live on chat platforms and websites, i.e. chatbots.

The most natural definition of a chatbot is – a developed a program that can have a discussion/conversation with a human. For example, any user could ask the bot an inquiry or a statement, and the bot will respond or perform an activity as appropriate.

A chatbot interacts on a format similar to instant messaging. By artificially replicating the patterns of human interactions in machine learning allows computers to learn by themselves without programming natural language processing.

While a bot is a computer's ability to understand human speech or text short for chat robot. A chatbot is merely a computer program that fundamentally simulates human conversations. It allows a form of interaction between a human and a machine the communication, which happens via messages or voice command.

A chatbot is programmed to work independently from a human operator. It can answer questions formulated to it in natural language and respond like a real person. It provides responses based on a combination of predefined scripts and machine learning applications.

When it is asked a question, the chatbot will respond based on the knowledge database available to it at that point in time. If the conversation introduces a concept it is not programmed to understand, it will either deflect the conversation or potentially pass the communication to a human operator. Either way, it will also learn from that interaction as well as from future interactions. Thus, the chatbot will gradually grow in scope and gain relevance.

1.4 System Overview

1.4.1 Human Language processed by chatbot

A chatbot is like a normal application. There is an app layer, a database and APIs to call other external administrations. Users can easily access chatbots, it adds intricacy for the application to handle.

However, there is a common problem that must be tackled. It cant comprehend the plan of the customer. At the moment, bots are trained according to the past information available to them. So, most organizations have a chatbot that maintains logs of discussions. Developers utilize these logs to analyze what clients are trying to ask. With a blend of machine learning tools and models, developers coordinate client inquiries and reply with the best appropriate answer

1.4.2 Chatbot Trained

Training a chatbot occurs at a considerably faster and larger scale than human education. While normal customer service representatives are given a manual instruction which they must be thorough with, a customer support chatbot is nourished with a large number of conversation logs, and from those logs, the chatbot can understand what type of question needs, what kind of answers.

CHAPTER 2

LITERATURE REVIEW

The purpose of the literature review shapes the way that noting is done. A literature review is a survey of scholarly sources on a specific topic. It provides an overview of current knowledge, allowing you to identify relevant theories, methods, and gaps in the existing research.

And the purpose is to situate your study in the field – that is, to establish a space for the work you are going to do – and to find concepts and approaches that are helpful, that you can build on. It is also important to understand key debates and differences in the field so that you can position yourself in relation to them. The literature review thus typically discusses a field of knowledge production and key concepts and lines of argument within it.

The literature review is not finished in the first year of doctoral study, and the ways in which texts are used to develop a research proposal in year one may not be the same as their use in the final thesis text when the findings and argument are known. But whether it's early or later literature work, there is always nothing to do. There is one major maxim about noting – it's not rewriting the book or the article. That's a waste of time and it misses the point.

2.1 Relevant Works

Nordstrom’s chatbot talks with customers about what they’d like to purchase from the store. It is an inspiring one to take a look at the backgrounds of our projects. There are multiple ecommerce chatbots each with different purposes.

Lets see some more similar systems.

2.2 Staples

The office supply store uses Facebook Messenger to offer customers product suggestions based on their requests and past orders. Staples’ Facebook chatbot can also enable customers to complete their purchase from the chat.

Sephora

Sephora’s chatbot on the bot platform Kik offers users makeup tips and makes product suggestions based on their personal quiz answers about their makeup usage. It also redirects users to the Sephora app or site to complete purchases. The beauty company doesn’t stop there. Sephora also has a Facebook bot called Sephora Virtual Artist. This bot allows users to see what Sephora’s products would look like on them by imposing the makeup onto the user’s selfie

2.3 H&M

The clothing brand H&M created a chatbot on Kik that asks users questions about their style and offers photo options for users to select. With this information, the bot creates a fashion profile of each user to make outfit suggestions and direct the user to purchasing the clothing. Users can also create their own outfits and browse and vote for other users’ outfits on the bot for an interactive shopping experience.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Existing System

The existing e-commerce sites have multiple features that make them unique and attract customers. These features improve the user experience and enhance customer satisfaction. And also many sites have chatbot facility to communicate with customers and help them to understand customer needs.

Conversational commerce isn't just a cool-sounding concept, user research shows that buyers are more ready and willing than ever to shop online with bots. Here are a few reasons why your online business should be using a messaging app to host a bot and boost sales. Business applications of chatbots for consumer-facing goods are growing rapidly.

3.1.1 Disadvantages of Existing System

- Customers have to pay a fixed price for a product.
- Lack of complete knowledge about the product.
- Delayed answering of customer's questions about a product from the company side.
- Lack of real time like interactive shopping experience.
- No availability of bargaining options.

3.2 Proposed System

Though these features help the ecommerce sites to enhance the provided service, the future of bargaining for designing the final price of the product is absent in these sites. Our system consists of an artificial intelligence infused chat bot which helps in communicating with the customer and also allowing them to bargain with us. The customer can decide the price which the customer wishes to pay for the product.

3.2.1 Advantages of Proposed System

- Becomes more customer friendly and 24×7 services.
- Allows customers to decide product price.
- Clears customers doubts and answers their questions.
- Greater customer satisfaction and user experience.
- Both customers and company are profited.
- Attracts more customers towards our site

CHAPTER 4

SYSTEM SPECIFICATION

4.1 Hardware and Software Requirements

Processor : Any type of processor, suggested Intel

i3 RAM : Minimum 2GB.

OS : Any Operating System from recent

times. Browser : Any Internet browsers (chrome,

Firefox).

4.2 Feasibility Study

The work process of our ecommerce site “Shopem” is made under the feasibility of the reports made with the application based process to cover with the theme of the things to be feasible in it along with the reports which are not feasible are to be taken along to be treated to make it good and keep going in an feasible part. Thus the feasible study report of the process is analysed and reported.

4.3. Requirement Analysis

After the extensive analysis of the problems in the system, we are familiarized with the requirement that the current system needs, The requirement that the system needs is categorized into the functional and non-functional requirements.

These requirements are listed below.

4.3.1 Functional Requirements

Functional requirements are the functions or features that must be included in any System to satisfy the business needs and be acceptable to the users, Based on this, the functional requirements that the system must require are as follows. The system should be able to generate replies to messages of the customer from the database. The system should be able to scan the user prompt as input and print the reply as output.

4.3.2 Non-Functional Requirements

Non-functional requirement is a description of features, characteristics and attributes of the system as well as any constraints that may Limit the boundaries of the proposed system. The non-functional requirements are essentially based on the performance information, economy, control and security efficiency and services.

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 HTML

The HyperText Markup Language, or HTML(HyperText Markup Language) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages.

Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

5.2 Java Script

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced by Java, the syntax is more similar to C and is based on European Computer Manufacturers Association (ECMA) Script, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a web page has loaded without communicating with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server.

Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the Hypertext Markup Language (HTML) of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser. JavaScript functions can be called within <script> tags or when specific events take place. Examples include on Click, on Mouse Down, on Mouse Up, on Key Down, on Key Up, on Focus, on Blur, on Submit, and many others. While standard JavaScript is still used for performing basic client-side functions, many web developers now prefer to use JavaScript libraries like j Query to add more advanced dynamic elements to websites.

Used in Web pages, Javascript is a "client-side" programming language. This means JavaScript scripts are read, interpreted and executed in the client, which is your Web browser. By comparison, "server-side" programming languages run on a remote computer, such as a server hosting a website Javascript has six primitives types: string, number, undefined, null, Boolean, and symbol. There is also a compound type or object. Interestingly, the primitive types are immutable and don't have properties. For example, because of the String object you can retrieve the length of a string. The code "a" length is evaluated as new String("a").length by the interpreter like magic. There are also the objects Number, Boolean, and Symbol which also add properties to its own primitives.

The types in JavaScript look simple and useless but knowing how they work is important. They help give a better understanding of the language and its behaviour. The importance of JavaScript as a web technology can be determined from the fact that it is currently used by 94.5% of all websites. As a client-side programming language, JavaScript helps web developers to make web pages dynamic and interactive by implementing custom client-side scripts. At the same time, the developers can also use cross-platform runtime engines like Node.js to write server-side code in JavaScript. They can even combine JavaScript, HTML5 and Cascading Style Sheets (CSS3) to create web pages that look good across browsers, platforms, and devices. There are also a number of reasons why each modern web developer must know how to leverage all benefits of JavaScript

5.3 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. Before CSS, nearly all presentational attributes of HTML documents were contained within the HTML markup. All font colors, background styles, element alignments, borders and sizes had to be explicitly described, often repeatedly, within the HTML. CSS let's authors move much of that information to another file, the style sheet, resulting in considerably simpler HTML.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

5.4 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is among the most popular projects on GitHub, with more than 142,000 stars, behind freeCodeCamp (almost

312,000 stars) and marginally behind Vue.js framework. Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden.

According to Twitter developer Mark Otto, A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company.Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps).

The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements.The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

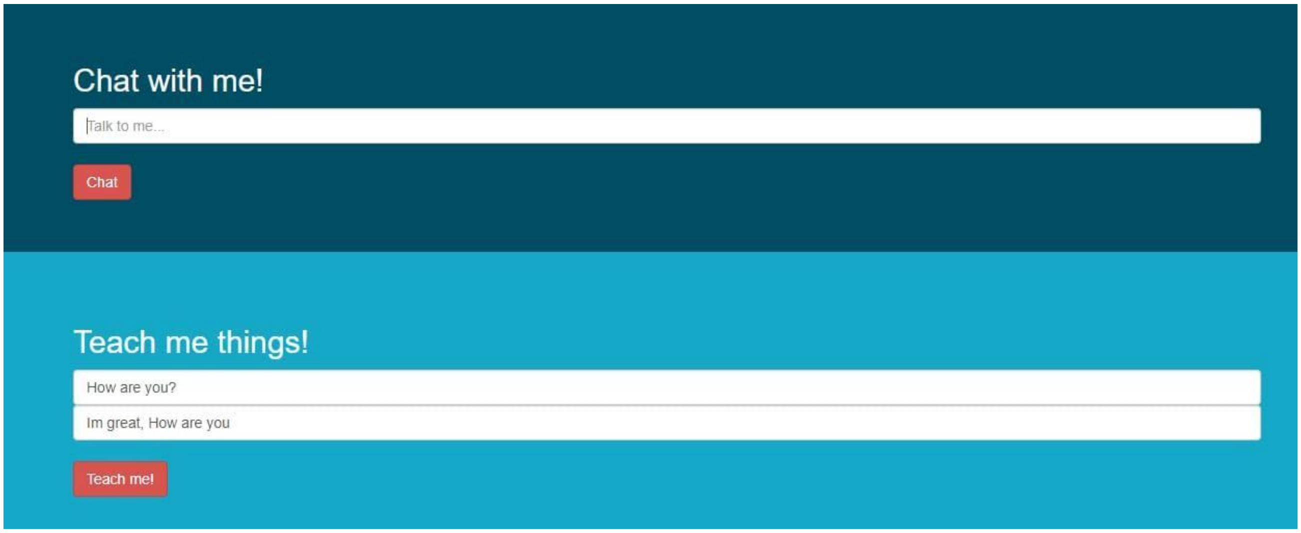
CHAPTER 6

OUTPUT & RESULTS

6.1. RESULTS

The project chatbot interacts on a format similar to instant messaging. By artificially replicating the patterns of human interactions in machine learning allows in the output a computers to learn by themselves without programming natural language

6.2 OUTPUT OF APPLICATION

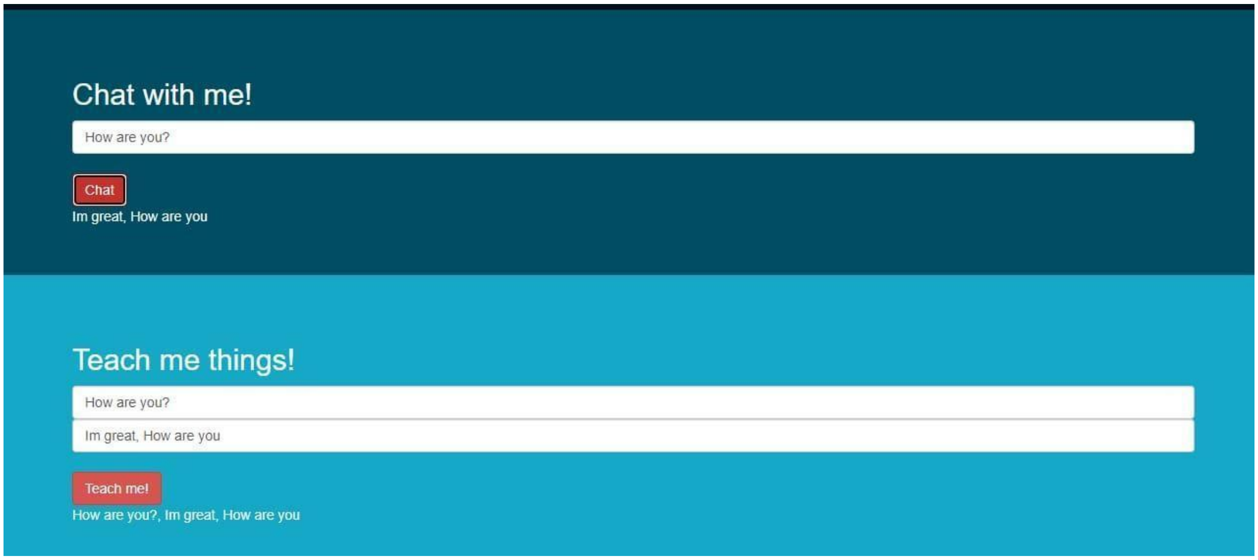


Chat with me!

Teach me things!

How are you?

Im great, How are you



Chat with me!

How are you?

Im great, How are you

Teach me things!

How are you?

Im great, How are you

How are you?, Im great, How are you

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1. CONCLUSION

- They give your company a face.
- You are available – immediately.
- They can increase your sales.
- They lay the foundation for your conversational marketing strategy.
- You can gain insights into customer behavior.
- They have a wide range of possible applications.

7.2. FUTURE WORKS

- A chatbot is an artificial intelligence that simulates a conversation with a user through apps or messaging. ChatBot are the future AI , as they provide the closest to a natural conversation between humans and machines, The progress we have made with chatbots thus far is, nonetheless , astonishing

- The chat bot usage in ecommerce sites will be a great success in future.

APPENDIX

SOURCE CODE

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

<head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" media="screen"
href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.5/css/bootstr
ap.min.css">
        <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
        <link rel="stylesheet"
href="http://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.4.0/css/font-aweso
me.min.css">



        <title>Chat Bot</title>
        <!-- -->

        <style>
                .title {
                        background-color: #c2454c;
                        color: #021119; text-align:
                        center; padding: 100px;
                          -webkit-background-size: cover;
 -moz-background-size: cover;
 -o-background-size: cover; background-size:
 cover;


                }
        .letsSpeak {
            background-color:
            #021119; color: white;
            padding-top: 50px;
            padding-bottom: 50px;
            padding-left: 100px;
            padding-right: 100px;
        }

        .knowledge { background-color:
            #004e64; color: white;
            padding-top: 50px;
            padding-bottom: 50px;
            padding-left: 100px;
            padding-right: 100px;
        }

        .teach { background-color:
            #16a8c7; color: white;
            padding-top: 50px;
            padding-bottom: 50px;
            padding-left: 100px;
            padding-right: 100px;
        }

        a { color: black;

        } button { background-color:
        #c2454c;
        }
    </style>

</head>

<body>

    <div class="title" id="title">
    <h2>Hi, I'm a chat bot.<br/> Say hi below</h2>
    <a id="icon" href="#letsSpeak"><i class="fa fa-angle-double-down"
style="font-size:48px"></i></a>
```

```
    </div>

    <div class="letsSpeak" id="letsSpeak">
        <h2>Talk to me!</h2>
        <p>Click this button and say 'Hello There!'</p>
        <button type="button" class="btn btn-danger" id="send"
onclick="annyang.start()" value="Reset Form">
            <!--adds ear icon to button-->
            <i class="material-icons">hearing</i></button>

        <!--machine response in output-->
        <div id="clickedChat">
            <div id="output"></div>
        </div>
    </div>

    <div class="knowledge">
        <h2>Chat with me!</h2>
        <input type="text" class="form-control" placeholder="Talk to
me..." id="userInput">
        <br/>
        <button type="button" class="btn btn-danger" id="send"
onclick="whatSaid()" value="Reset Form">Chat</button>
        <br/>
        <div class="response" id="botRespo"></div>
    </div>



    <div class="teach">
        <h2>Teach me things!</h2>
        <input type="text" class="form-control" placeholder="When you
say..." id="helpLearn">
        <input type="text" class="form-control" placeholder="How
should I respond?" id="aboutIt">
        <br/>
        <button type="button" class="btn btn-danger" id="send"
```

```
onclick="teachMe()">Teach me!</button>
        <br/>
        <div id="learnlog"></div>

    </div>


    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.11.3/jquery.min.js"></scri pt>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.5/js/bootstrap
.min.js"></script>
    <script
src="//cdnjs.cloudflare.com/ajax/libs/annyang/2.3.0/annyang.min.js"></script >

    <script> function
        init() {
            //init function, calls local storage which has a stringify'd //array
called facts                  //need to combine facts (string) with
                              vocab array savedValue =
                              localStorage.getItem('facts');

        }



        //annyang, need to figure out how to make multiple commands
run

        var commands = {
            // annyang will capture anything after a splat (*) and pass it
to the function.
            // e.g. saying "Show me Batman and Robin" is the same as
calling showFlickr('Batman and Robin');
            'show me *tag': showFlickr,

            // A named variable is a one word variable, that can
fit anywhere in your command.
```

```
            // e.g. saying "calculate October stats" will call
calculateStats('October');
            'calculate :month stats': calculateStats,

            // By defining a part of the following command as optional,
annyang will respond to both:
            // "say hello to my little friend" as well as "say hello friend" 'say
            hello (to my little) friend': greeting
        };

        var showFlickr = function (tag) { var url =
            'http://api.flickr.com/services/rest/?tags=' + tag;
            $.getJSON(url);


        }

        var calculateStats = function (month) {
            $('#stats').text('Statistics for ' + month);
        } var greeting = function ()

        {

        $('#greeting').text('Hello!');

        }


        if (annyang) {
            // Let's define our first command.
            //First the text we expect,
            //and then the function it should call
            var commands = {
                'Hello there': function () { var content
                    = ' ' content += '<br/>' content
                    += 'Hi, how are you??' content
                    += '<br/>'
                    content += 'I\'m a young bot called Larry.' content
                    += '<br/>'
```

```
                                        content += 'Unfortunately, my ears aren\'t great
(yet), so until I learn more this is all I can hear/say.' content +=
                                    '<br/>'
                                      content += 'But, you can type in the boxes below
to chat with me more!';

                                    console.log(commands);

document.getElementById('output').innerHTML = content;

                              }
                         }

                    };

                    // Add our commands to annyang annyang.addCommands(commands);


                    //need to turn off my microphone </script>
          <script> function gettingJSON() {
                    document.write("jquery loaded");

$.getJSON("http://api.openweathermap.org/data/2.5/weather?q=London&AP
PID=ee6596241130f193adf1ba90e625cc10", function (json) {
                              document.write(JSON.stringify(json));
                         });
                    }

                    //Checks for enter press
                    $(document).keypress(function (e) { if
                         (e.which == 13) { whatSaid();

                         }
                    });


                    //defines date so we can call later var
                    now = new Date();
          //this is what the bot knows var
          vocabulary = [
['hi', 'hello'],
['what is your name', 'My name is Larry, Im a young bot'],
['how are you', 'Im great, How are you'],
['where are you', 'Boulder, Colorado'],
['where is that', 'Near the Rocky Mountains in the United States'],
['where am i', 'you\'re also in Boulder'],
['who created you', 'You did!'],
['what is a bot', 'I\m a bot'],
['what is your favorite movie', 'Harry Potter perhaps?'],
['what day is it', now],
['where am i from', 'Madison Wisconsin'],
['i love you', 'Aw, thanks. I love you too!'],
['who is the coolest dog', 'Nora of course!'],
['would you like to go mountain bicycling', 'I\'m a computer silly'],
];



          function whatSaid() {



               //this creates a variable input that is the userInput var
               input = document.getElementById('userInput');

               //this defines a variable userSaid (which is what the user
said) //that gets the value of the Input variable var
               userInput = input.value;


               //this strips the punctuation and the spaces from user input
               //punctuation marks I want to strip
               var punctRE =
/[\u2000-\u206F\u2E00-\u2E7F\\'!"#$%&()*+,\-.\/:;<=>?@\[\]^_`{|}~]/g; //fixes
               the spaces after I remove a bunch of punctuation var spaceRE
               = /\s+/g;

               var str = userInput;
               //use string replace to strip the text var
               saidStripped = str.replace(punctRE,
'').replace(spaceRE, ' ');
               //convert to lower case
               var input = saidStripped.toLowerCase()



               var notUnderstood = "I'm sorry, I don\'t understand";


                                             for (var i = 0; i < vocabulary.length; i++)
                                                   { console.log('I am comparing ' +
with ' + input);                                        vocabulary[i][0] + '

                                                   if
                                                         (voca
                                                         bulary
                                                         [i][0]
                                                         ==
                                                         input)
                                                         { var
                                                         respo
                                                         =
                                                         vocab
                                                         ulary[i
                                                         ][1];

document.getElementById('botRespo').innerHTML = respo;
                                  return;
                                 console.log(vocabulary[i][1]); }
                           else {


document.getElementById('botRespo').innerHTML = notUnderstood; }
```

```
    }

    //clears the textbox
    //document.getElementById('userInput').reset;
}

//try to write this to local storage
//add a teach me bar
```

```
        //this is a function that takes input function
        teachMe() {

            var input1 = document.getElementById('helpLearn'); var
            input2 = document.getElementById('aboutIt');

            var learn_input = input1.value console.log(learn_input);
            //strip punctuation and spaces //learn_input
            becomes learnStripped

            var punctuateRE =
/[\u2000-\u206F\u2E00-\u2E7F\\'!"#$%&()*+,\-.\/:;<=>?@\[\]^_`{|}~]/g;
            var space_RE = /\s+/g; var str2 = learn_input;
            var learnStripped = str2.replace(punctuateRE,
'').replace(space_RE, ' ');
            var learned = learnStripped.toLowerCase()
            var about_input = input2.value console.log(about_input);

            document.getElementById('learnlog').innerHTML =
learn_input + ', ' + about_input

            var learnArr = [learnStripped, about_input];
            console.log(learnArr);

            vocabulary.push(learnArr); console.log(vocabulary);



            //var learnArrString = learnArr.toString();
            //console.log(learnArrString);

            var vocabString = vocabulary.toString();
            console.log(vocabString);
            //first, need to stringify my array before saving it to local
storage

            localStorage.setItem('facts', vocabString);
        }

        init();
    </script>

</body>

</html>
```

REFERENCES

[1] "What is a chatbot?". techtarget.com. Retrieved 30 January 2017. https://www.engati.com/blog/what-are-chatbots.

[2] Luka Bradeško, Dunja Mladenić. "A Survey of Chabot Systems through a Loebner Prize Competition". S2CID 39745939.

[3] "The AI Revolution is Underway! – PM360". www.pm360online.com. Retrieved 2 July 2018.

[4] "CHATBOTS: BOON OR BANE?". bluelupin. Retrieved April 1, 2019.

[5] "Retail e-commerce sales CAGR forecast in selected countries from 2016 to 2021". Statista. October 2016. Retrieved 1 January 2018.

[6] Wienclaw, Ruth A. (2013) "E-Commerce." Research Starters: Business.

[7] "The Economic Impacts of NIST's Data Encryption Standard (DES) Program". National Institute of Standards and Technology. United States Department of Commerce. October 2001. Retrieved 21 August 2019.