**INDEX**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| Sno | Title | Pg no |
|  | Abstract |  |
| 1 | Introduction&Objectives | 1 |
| 1.1 | Objective and scope of the project | 2 |
| 1.2 | Over view of the project | 3 |
| 2 | Requirement Elication/Analysis | 5 |
| 2.1 | Existing System | 5 |
| 2.2 | Proposed System | 5 |
| 2.3 | Techonologies used in the system | 6 |
| 2.3.1 | Algorithms for Chatbot | 8 |
| 2.3.2 | Design | 9 |
| 2.3.3 | Software Enviorment | 11 |
| 2.4 | Feasibility Study | 34 |
| 2.5 | Economical Feasibility | 34 |
| 2.6 | Techincal Feasibility | 34 |
| 2.7 | Social Feasibility | 34 |
| 2.8 | Functional&Non Functional Requirements | 35 |
| 3 | System Analysis | 38 |
| 3.1 | SRS | 38 |
| 3.2 | Modules | 38 |
| 3.3 | Performace Requirements | 38 |
| 3.4 | Software Quality Attributes | 39 |
| 3.5 | Software And Hardware Requirement Specification | 39 |
| 3.6 | Software Design | 40 |
| 4 | System Design | 42 |
| 4.1 | Data Flow Modeling | 42 |
| 4.2 | UML Diagrams | 45 |
| 4.3 | Use Case Diagrams | 49 |
| 4.4 | Class Diagrams | 50 |
| 4.4.1 | Interaction Diagrams | 51 |
| 4.4.2 | Sequence Diagrams | 51 |
| 4.4.3 | Collabaration Diagrams | 52 |
| 4.4.4 | State Chart Diagram | 53 |
| 4.4.5 | Deployment Diagram | 54 |
| 4.4.6 | Component Diagram | 55 |
| 4.4.7 | Activity Diagram | 56 |
| 4.4.8 | Database Design | 57 |
| 4.4.9 | ER Diagram | 57 |
| 5 | Coding | 58 |

|  |  |  |
| --- | --- | --- |
| 6 | Implementation | 59 |
| 6.1 | Data Set(Question) | 62 |
| 7 | Input And Output Design | 63 |
| 8 | Testing | 66 |
| 8.1 | Testing Levels | 66 |
| 9 | Screens | 71 |
| 10 | Future Scope of the Project | 75 |
| 11 | Conclusion | 77 |
| 12 | Bibilography | 79 |

**ABSTRACT**

Chatbots receive increasing attention from media and industry, but at the same time it is not yet well known what chatbots really are, what they can be used for and how to create them. The goal of this work is to answer these three questions by analyzing existing platforms, products and technologies, and additionally developing an exemplary chatbot. Explaining what chatbots are, demystifying what to use them for and showing how to create them will help more people to be able to use and create chatbots and thereby accelerate the development of the chatbot ecosystem. Starting by defining fundamental terms, the first half of the work focuses on showing available platforms, products and technologies, while the second half guides through the development of an exemplary chatbot, including user interaction design and software architecture.

Hospitals are the most widely used means by which a sick person gets medical check-ups, disease diagnosis and treatment recommendation. This has been a practice by almost all the people over the world. People consider it as the most reliable means to check their health status. The proposed system is to create an alternative to this conventional method of visiting a hospital and making an appointment with a doctor to get diagnosis. This research intends to apply the concepts of natural language processing and machine learning to create a chatbot application. People can interact with the chatbot just like they do with another human and through a series of queries, chatbot will identify the symptoms of the user and thereby, predicts the disease and recommends treatment. This system can be of great use to people in conducting daily check-ups, makes people aware of their health status and encourages people to make proper measures to remain healthy. According to this research, such a system is not widely used and people are less aware of it. Executing this proposed framework can help people avoid the time-consuming method of visiting hospitals by using this free of cost application, wherever they are.

**Introduction/Objectives**

**1. Introduction/objectives**

**Introduction**

Since the past few decades, humans have been tirelessly working day in and day out that they fail to prioritize their health on a regular basis. In the longer run, this problem leads to jeopardizing the quality of life. Nevertheless, with the aid of Artificial Intelligence, we can now provide health care services to individuals at their convenience at reasonable prices. One of the biggest blessings we possess is a healthy body. A healthy body and enhanced quality of life is something each one of us looks up to. The primary focus of this paper is to provide these services to fulfill the above mentioned purpose. It is difficult to imagine our lives without high tech gadgets because they have become an essential part of our lives. Therefore the field of Artificial Intelligence is prospering due to the various applications of  
it in the research field. Disease prediction is one of the main goals of the researchers based on the facts of big data analysis which in turn improves the accuracy of risk classification based on the data of a large volume. [1] E-healthcare facilities in general, are a vital resource to developing countries but are often difficult to establish because of the lack of awareness and development of infrastructure. A number of internet users depend on the  
internet for clearing their healthcare based queries. We have designed a platform for providing online medical services to patients with a goal to provide assistance to healthcare  
professionals. The user can also seek medical guidance in an easier way and get exposure to various diseases and diagnosis available for it. In order to make communication more effective, we have implemented a chatbot for disease prediction. Chatbots are the human version of software that is based on AI and uses Natural language processing (NLP) to interpret and accordingly respond to the user. This study proposes the disease prediction chatbot using the concepts of NLP and machine learning algorithms. The prediction is carried out using KNN and Decision tree algorithms. KNN and Decision tree are a few of the most used classification algorithms that are frequently used in disease prediction. It is assisted with the NLP driven chatbot. [2] The wordnet and tokenization concepts of NLP are used. The use of tokenization is to split the given text into a list of words whereas WordNet is a lexical database of dictionary designed for natural language processing. The study also focuses on the use of the Optical Character Recognition tool named Tesseract which is used to extract text from the patient’s scanned pathology report. The generated text helps in translating the report in an easier manner by providing a graphical analysis of the test result.

* 1. OBJECTIVE AND SCOPE OF THE PROJECT

Objective

Explaining what chatbots are, demystifying what to use them for and presenting how to create them, will help more people to be able to use and create chatbots, and thereby, accelerate the development of the chatbot ecosystem. Innovation in technology and the creation of new solutions can help automating and simplifying more tasks, which gives people the opportunity to focus on more interesting issues and accomplish more things. Chatbots have the potential to simplify and automate many existing tasks and thereby accelerate the overall technological progress.

**Scope**

The Virtual Estate ChatBot will share the vital information about the real estate properties across various locations with the user. Our proposed system introduces chatbot into the idea of estate agency. Chatbots are software agents that interact with the user in a conversation. The main goal of their creation was to resemble a human being in the way they perform said interaction, trying to make the user think he/she is writing to another human being. Chatbots functions like a typical search engine; although they produce only one output instead of multiple outputs/results. But the basic process flow is the same where each time an input is entered, a new search will be done.

The user can search for the information of various properties wherever he or she is in- terested. The chatbot will then forward the details to the server. The server processes the request and provides details of suitable properties to the user. This application will prove to be beneficial to the user. It will save time and efforts both.

**1.2 Over view of project**

This work gives a general introduction to chatbots by explaining what they are, what they can be used for and how to develop them. No previous domain-specific knowledge is required. Lately as of writing topics around chatbots have received increasing attention from media and also numerous investments from different actors in the industry. At the same time not many potential users know about the existence of chatbots or about areas in which chatbots could be helpful assistance. The topic is equally unknown to developers. While the term chatbot is commonly used in media, the meaning mostly remains ambiguous. There is a need for further explanation of what chatbots are and further analysis to identify well suited applications for chatbots. Additionally to spreading knowledge about the potentials of chatbots and their use cases, more developers should be enabled to create new, innovative chatbots. The lack of knowledge can be solved by providing answers to the questions of what chatbots are, what benefits they bring and how to create them. An appropriate definition of chatbots can be given by analyzing the fundamental meaning of the term chatbot and by exploring past and current applications. Use cases of chatbots can be identified in existing products. Market trends and attributes of media and technology can be analyzed to find new potential scenarios for the usage of chatbots. Development is best explained by creating a real chatbot and by using it to present the general principles of the development process. Explaining what chatbots are, demystifying what to use them for and presenting how to create them, will help more people to be able to use and create chatbots, and thereby, accelerate the development of the chatbot ecosystem. Innovation in technology and the creation of new solutions can help automating and simplifying more tasks, which gives people the opportunity to focus on more interesting issues and accomplish more things. Chatbots have the potential to simplify and automate many existing tasks and thereby accelerate the overall technological progress.

The structure of this work follows the three main questions. To begin with, terminology is defined and applications are explored to form a definition and understanding of what chatbots are. Afterwards use cases of chatbots are identified not only through the collection of existing examples, but also through the exploration of future potentials by analyzing attributes of the relevant technologies. The second half of the work is a case study for the development of a chatbot. The presented example guides through the process of designing user interactions for a chatbot, and additionally explains architectural decisions and technological choices, which provide a basis for other developers to buildon when creating new chatbots in the futurey

**Requirment Elication / Analysis**

**2. Requirement Elicitation/Analysis**

**2.1 Existing System**

Before exploring new technology one should examine prior work and learn from past ideas, both succeed and also failed attempts. This section presents a selection of events from the last century, which introduced the ideas that formed the present definition of a chatbot. It is not an attempt to give an all-encompassing overview about the history of computing, instead the aim is to explain where the concept of chatbots and the interest of creating them originated from. In today’s fast-paced world, people tend to neglect their health which may result in a critical problem. Such a problem can be avoided by using the symptoms driven disease prediction application.

2.2 THE PROPOSED SYSTEM

This is an automated chat robot design to answer users frequently asked questions, earlier natural language processing techniques were using to design this robots but its accuracy of giving correct answer was less and now due to Deep Learning algorithms accuracy of giving correct answer increase, so here using python deep learning project we are building CHATBOT application to answer users questions.

To implement this technique first we train deep learning models with the train data (all possible question’s answers) and whenever users give any question then application will apply this test question on train model to predict exact answer for given question.

Earlier companies were hiring humans to answer user’s queries but by using this application we can answer user’s question without using any man power.

Chabot can be described as software that can chat with people using artificial intelligence. Chabot ’s are generally used to respond quickly to users. Chabot’s, a common name for automated conversational interfaces, present a new way for individuals to interact with computer systems. Traditionally, to get a question answered by a software programinvolves using a search engine, or filling out a form. A Chabot allows a user to simply ask questions inthe same manner that they would address a human. There are many well-known voice-based catboatscurrently available in the market: Google Assistant, Alexa and Siri. Chabot’s are currently being adoptedat a high rate on computer chat platforms.

This research intends to apply the concepts of natural language processing and machine learning to create a chatbot application. People can interact with the chatbot just like they do with another human and through a series of queries, chatbot will identify the symptoms of the user and thereby, predicts the disease and recommends treatment. To implement this project we are using python deep learning neural networks and NLTK (natural Language Processing API) to process train and test text data.

**Advantages:**

This system can be of great use to people in conducting daily check-ups, makes people aware of their health status and encourages people to make proper measures to remain healthy. According to this research, such a system is not widely used and people are less aware of it.

**2.3 Techonologies used in the system**

**PYTHON**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An [interpreted language](https://en.wikipedia.org/wiki/Interpreted_language" \o "Interpreted language), Python has a design philosophy that emphasizes code [readability](https://en.wikipedia.org/wiki/Readability" \o "Readability) (notably using [whitespace](https://en.wikipedia.org/wiki/Whitespace_character" \o "Whitespace character) indentation to delimit [code blocks](https://en.wikipedia.org/wiki/Code_block" \o "Code block) rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code" \o "Source lines of code) than might be used in languages such as [C++](https://en.wikipedia.org/wiki/C%2B%2B" \o "C++)or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)" \o "Java (programming language)). It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system" \o "Operating system). [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation" \o "Reference implementation) of Python, is [open source](https://en.wikipedia.org/wiki/Open_source" \o "Open source) software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation" \o "Python Software Foundation). Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type" \o "Dynamic type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management" \o "Memory management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm" \o "Programming paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming" \o "Imperative programming), [functional](https://en.wikipedia.org/wiki/Functional_programming" \o "Functional programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming" \o "Procedural programming), and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library" \o "Standard library)

**DJANGO**

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability" \o "Reusability)and "pluggability" of components, rapid development, and the principle of [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself" \o "Don't repeat yourself). Python is used throughout, even for settings files and data models.



Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete" \o "Create, read, update and delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Introspection_(computer_science)" \o "Introspection (computer science)) and configured via admin models



**2.3.1 Algorithms For Chatbot**

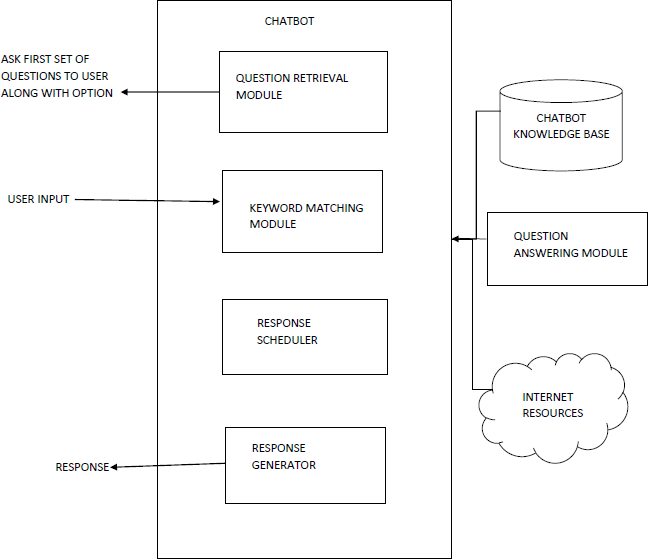
NLP (Natural language processing) and Machine Learning are both fields in computer science related to AI (Artificial Intelligence). Machine learning can be applied in many different fields. NLP takes care of “understanding” the natural language of the human that the program (e.g. chatbot) is trying to communicate with. This understanding enables the program e.g. chatbot) to both interpret input and produce output in the form of human language. Neural networks are one of the learning algorithms used within machine learning. They consist of different layers for analyzing and learning data.

* + - **Decision Tree Algorithm:** In this algorithm a decision tree is used to map decisions and their possible consequences, including chances, costs and utilities. This method allows the problem to be approached logically and stepwise to get to the right conclusion. An important algorithm that evolved from this algorithm is the Random Tree algorithm. This algorithm uses multiple trees to avoid overfitting that often occurs with using decision trees.
    - **Similarity Ranking Method:** Similarity ranking methods compare a given string to a set of strings and rank those strings in order of similarity. To produce a ranking, we need a way of saying that one match is better than another. This is done by returning a numeric measure of similarity as the result of each comparison. Alternatively, you can think of the distance between two strings, instead of their similarity. Strings with a large distance between them have low similarity, and vice versa.

**2.3.2 Design**

**Block Diagram**

Block diagram is a diagram showing in schematic form the general arrangement of the parts or components of a complex system or process, such as an industrial apparatus or an electronic circuit.



**Block Diagram For Chatbot**

As we can see in figure there are different modules of Chatbot Question retrieval mod- ule, keyword matching module, response scheduler, and response generator, question answering module, chatbot knowledge base and internet resources. The chatbot asks the first set of questions to the user with options, the user then replies with a choice and the chatbot then replies the users input with the best answer with the help of keyword matching module.

**Software Environment**

**2.3.3 Software Environment**

**What is Python :-**

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/" \t "_blank)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

**Advantages of Python** :-

Let’s see how Python dominates over other languages.

**1. Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

#### 2. Extensible

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

**3.Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

#### 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### 6. Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

#### 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you**[download Python](https://data-flair.training/blogs/install-python-windows/)** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

**10. Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

**11. Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

**Advantages of Python Over Other Languages**

1. **Less Coding**
2. Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### 2. Affordable

1. Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.
2. **The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

**Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **[machine learning](https://data-flair.training/blogs/machine-learning-tutorials-home/)**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### **Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait,what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**History of Python : -**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**What is Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

**Categories Of Machine Leaning :-**

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

**Applications of Machines Learning :-**

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# 

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/" \t "_blank), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### **How to start learning ML?**

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal.

**Step 1 – Understand the Prerequisites**

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding

**(a)Learn Linear Algebra and Multivariate Calculus**

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

**(b)Learn Statistics**

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it .Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

**(c)Learn Python**

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/" \t "_blank)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/" \t "_blank), [TensorFlow](https://www.tensorflow.org/" \t "_blank), [Scikit-learn](https://scikit-learn.org/stable/" \t "_blank), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as **[Fork Python](https://practice.geeksforgeeks.org/courses/fork-python" \t "_blank)** available Free on GeeksforGeeks.

**Step-2 Learn various ML concepts**

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

**(a)Terminologies of Machine Learning**

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

**(b) Types of Machine Learning**

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### 

### **Advantages of Machine learning :-**

**1.Easily identifies trends and patterns**

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

**2.No human intervention needed (automation)**

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

**3.Continuous Improvement**

As **[ML algorithms](https://data-flair.training/blogs/machine-learning-algorithms/)** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4.Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments

**5. Wide Applications**

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

**Disadvantages of Machine Learning :-**

**1. Data Acquisition**

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

**2. Time and Resources**

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

**3. Interpretation of Results**

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

**4. High error-susceptibility**

**[Machine Learning](https://en.wikipedia.org/wiki/Machine_learning)** is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software" \o "Free software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software" \o "Open-source software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)" \o "Library (computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning" \o "Machine learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks" \o "Neural networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google" \o "Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain" \o "Google Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License" \o "Apache License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license" \o "Open-source license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Install Python Step-by-Step in Windows and Mac :

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

**How to Install Python on Windows and Mac :**

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

**Download the Correct version into the system**

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: **[https://www.python.org](https://www.python.org/)**



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.



**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

**Verify the Python Installation**

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

**Check how the Python IDLE works**

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

**2.4.FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

**Three key considerations involved in the feasibility analysis are,**

* **ECONOMICAL FEASIBILITY**
* **TECHNICAL FEASIBILITY**
* **SOCIAL FEASIBILITY**

**2.5 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**2.6 TECHNICAL FESABILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**2.7 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**2.8 Functional And Non Functional Requirements**

**Functional Requirements**

Functional Requirements All necessary functional requirements can be extracted from the above defined user stories. First, a user needs to be able to add new vocabulary. There should not be any restrictions on what can be added and vocabulary should not be limited to single words, because in many cases it is more helpful to add whole phrases instead. Each vocabulary consists of the phrase the user tries to memorize and an explanation to help understanding the meaning of the phrase. Next, the chatbot should provide a way to revise vocabulary. There should be two possible modes for revising; one where users can click a button to tell whether they remembered the phrase correctly or not, and a second mode whereby users type out the phrase themselves. In each case the system should keep track of whether users knew the correct solution or not. Lastly, it is necessary to determine what to study next. A user should not be required to think about what or when to review vocabulary. The chatbot needs a system to decide the review time for each vocabulary, and ideally the user is notified when vocabulary is ready to be reviewed by sending a message to the user. These three main features can be seen as a sufficient minimal viable product, or MVP. For demonstration purposes it is desired to keep the product as simple as possible. The knowledge that can be taken from making decisions about the implementation and walking through the process of creating the chatbot,is mostly independent from this particular product and can be applied to the development of other chatbot products.

**Non Functional Requirements**

Non-functional Requirements Since this is a simple example, non-functional requirements remain minimal. Availability of the service is not a priority, but chatbot software can be scaled similar to other software, and redundancy can be used to ensure availability. Since messaging platforms act as intermediary between users and the chatbot software, most platforms also re-send missed messages in case the chatbot is unavailable. That the platform ensures availability, further lessens the priority to address it in the chatbot software itself. Similarly, security is not a main focus here, because the messaging platform itself already handles certain security-sensitive functionality such as authentication and encryption of communication. A production scenario, though, would require further care for securing the service. Performance is equally not a major concern. Because the scope of the example application is limited, the domain specific logic remains inexpensive in computation. The main performance bottleneck is the in 3.7 on page 20 mentioned aspect of networking and involved unknown parties. Employing performance-improving solutions for networking issues won’t be a part of the example chatbot, but performance can be improved by choosing geographically strategic located data centers for deploying the chatbot software

A more central requirement is reusability. Although the example focuses on solving a specific task, the software architecture should be designed in a way, that appropriate parts can be reused for other chatbots in the future. To ensure reusability the software should be documented, stable and extensible. Usability can be seen as the most important non-functional requirement. The focus of developing the example chatbot is to design a good user experience and to explore how interface and interaction design can be best accomplished with the given medium.

**System Analysis**

**3.System Analysis**

System analysis is the performance management and documentation of activities related to the life cycle phases of any software namely:

* The Study Phase
* The Design Phase
* The Development Phase
* The Implementation Phase
* The Testing Phase

Software Analysis starts with a preliminary analysis and later switches on to a detailed one. During the preliminary analysis the Analyst takes a quick look at what is needed and whether the cost benefits. Detailed analysis studies in depth all the cornered factors, which builds and strengthens the software.

3.1 SRS

SRS (Software Requirement Specification) is a document that completely describes what the proposed should do, without describing how the software does it.

**3.2 Modules**

This application consists of 4 modules

New user Registration: Using this module users can signup with the application.

Login: Using this module user can login to application

Chabot: Using this module users can interact with Chabot

Logout: Using this module users can exit from the application

* 1. **Performance Requirements**

1)The operation time should be small and the throughput should be high..

2)It should produce timely and accurate result.

**3.4 Software Quality Attributes**

1. **Maintainability** – Since it is directly associated with the database, so there is very little maintainability problem with this application.
2. **Easy to Learn –** Since there are less number of forms, this application is very easy to learn with user-friendly screens.
3. **Flexibility** – This application is very much flexible for future enhancements.

**3.5 Software and Hardware Requirement Specification**

**Software Requirements**

* Operating system : - Windows.
* Coding Language : python

**Hardware Requirements**

1. System : Pentium IV 2.4 GHz.
2. Hard Disk : 40 GB.
3. Ram : 512 Mb.

3.6 Software Design

System design is the second step in the system life cycle, in which overall design of the system is achieved. The functionalities of the system is designed and studied in this phase. The first step is designing of program specification. This determines the various data inputs to the system, data flow and the format in which output is to be obtained.

Design phase is a transmission phase because it is a transition from user oriented document to computer data. The activity in the design phase is the allocation of functions to manual operations, equipment and computer programs. Flow charts are prepared in the study time and is decomposed until all functions in the system perform evidently.

Design is a multi-step process that focuses on data structures, software architecture, procedural details( algorithms etc) and links between the modules. The design process goes through logical and physical stages. In logical design reviews are made linking existing system and specification gathered. The physical plan specifies any hardware and software requirement, which satisfies the local design.

Modularization of task is made in this phase. The success of any integrated system depends on the planning of each and every fundamental module. Usually a project is revised in step by step sequence. Inter-phase management of such module is also important. Software design methodology changes continually as new methods, better analysis and broader understanding evolve.

Various techniques for software design do exit with the availability of criteria for design quality. Software design leads three technical activities-design, code and test.

Each activity transforms information, which validates the software. The design system converts theoretical solution introduced by the feasibility study into a logical reality.

**System Design**

**4.System Design**

**4.1 Data Flow Modeling**

Data flow diagrams provide a graphical representation of how information moves between processes in a system. A Data Flow Diagram shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. A data flow diagram takes business processes and activities and uses them to create a clear illustration of how data flows through a system. DFDs represent the flow of data from external entities into a single system by moving and storing data from one process to another. Through the use of data flow diagrams, a system can be decomposed into subsystems, and subsystems can be further decomposed into lower-level subsystems. Each subsystem represents a process or activity in which data is processed. Once the lowest level is reached, processes can no longer be decomposed. Data flow modeling can be used to identify a variety of different things, such as:

Information that is received from or sent to other individuals, organizations, or other computer systems

Areas within a system where information is stored and the flows of information within the system are being modeled

The processes of a system that act upon information received and produce the resulting outputs

DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish.

DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub-processes

**Level 0 DFD For Chatbot**

**Chatbot**

User

DATABASE

Send Message

Send Response

**Figure 1:Level 0 DFD for Chatbot**

As we can see in figure 1 i.e. level 0 data flow diagram, the whole system is represented with the help of input, processing and output. The input to the Virtual Estate ChatBot is the service requested by the end user and the output is the response given

**Level 1.1 DFD For Chatbot**

**Chatbot**

DATABASE

Start chat

User

Send Reply

End chat

**Figure 2:Level 1.1DFD for Chatbot**

we can see in figure 2 i.e. level 1.1 data flow diagram, the bubble “Virtual Estate System” is shown in detail by various processes. The chatbot will initiate the chat with the user and also check with the MySQL database

**level 1.2 DFD For Chatbot**

**Chatbot**

User

DATABASE

Start chat

Send Reply

Ask Questions

Send Response

Endchat

**Figure 3 for DFD Chatbot**

As we can see figure 3 Data flow diagram, the user interacts with the system in a series of questions. The chatbot all asks questions to the user providing the user with multiple options. In this way, conversation will take place between the user and the chatbot.

**Level 1.3 DFD For Chatbot**

Process Response

RERereREssEresresResponse ssResponse

**Chatbot**

Save Reads

DATABASE

User

Start Chart

Get Reads raReads ReReads

Send Response

Figure 4 for DFD Chatbot

As we can see in figure 4 i.e level 1.3 data flow diagram, the user interacts with the system, and the system processes the requests, receives the data to be read and saves the data that is read. It gives a detailed description of the processes taking place between the chatbot and MySQL database.

**4.2 UML Diagrams**

UML stands for **Unified Modeling Language**. This object-oriented system of notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the [Rational Software Corporation](http://www.rational.com/" \t "_blank). These renowned computer scientists fused their respective technologies into a single, standardized model. Today, UML is accepted by the [Object Management Group (OMG)](http://www.omg.org/" \t "_blank) as the standard for modeling object oriented programs.

There are three classifications of UML diagrams:

* **Behavior diagrams**.  A type of diagram that depicts behavioral features of a system or business process.  This includes activity, state machine, and use case diagrams as well as the four interaction diagrams.
* **Interaction diagrams**.  A subset of behavior diagrams which emphasize object interactions.  This includes communication, interaction overview, sequence, and timing diagrams.
* **Structure diagrams**.  A type of diagram that depicts the elements of a specification that are irrespective of time.  This includes class, composite structure, component, deployment, object, and package diagrams.

### 

### **Types of UML Diagrams**

UML defines nine types of diagrams: class (package), object, use case, sequence, collaboration, statechart, activity, component, and deployment.

##### [Class Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Class-Diagrams)

Class diagrams are the backbone of almost every object oriented method, including UML. They describe the static structure of a system.

##### [Package Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Package-Diagrams)

Package diagrams are a subset of class diagrams, but developers sometimes treat them as a separate technique. Package diagrams organize elements of a system into related groups to minimize dependencies between packages.

##### [Object Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Object-Diagrams)

Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.

**[Use Case Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Use-Case-Diagram)**

Use case diagrams model the functionality of system using actors and use cases.

**[Sequence Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Sequence-Diagram)**

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

**[Collaboration Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Collaboration-Diagrams)**

Collaboration diagrams represent interactions between objects as a series of sequenced messages. Collaboration diagrams describe both the static structure and the dynamic behavior of a system.

**[Statechart Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Statechart-Diagrams)**

Statechart diagrams describe the dynamic behavior of a system in response to external stimuli. Statechart diagrams are especially useful in modeling reactive objects whose states are triggered by specific events.

**[Activity Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Activity-Diagram)**

Activity diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

**[Component Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Component-Diagram)**

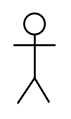
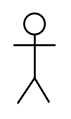
Component diagrams describe the organization of physical software components, including source code, run-time (binary) code, and executables.

**[Deployment Diagrams](http://www.smartdraw.com/resources/tutorials/UML-Deployment-Diagrams)**

Deployment diagrams depict the physical resources in a system, including nodes, components, and connections.

**Use Case Model**

**System name name**

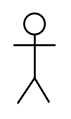
**Use Case1**

Actor Actor

**Use Case n**

**Use Case Diagram**

Register



logout

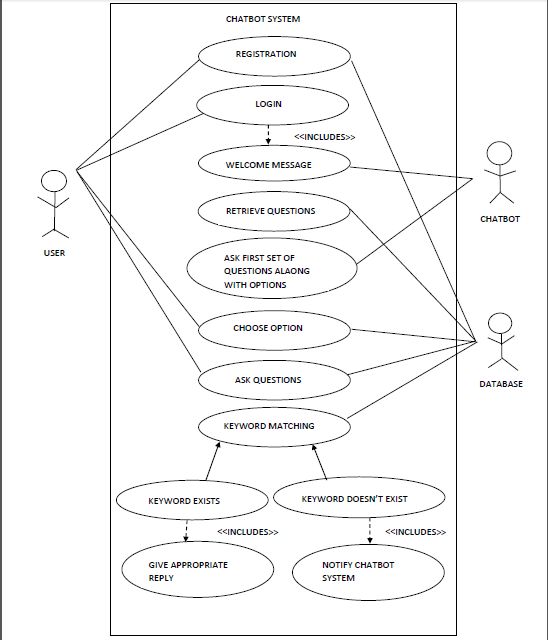
Chat with Chatbot

login

User

**4.3 Use Case Diagram**

**Overview of Use Case Diagram:**



**4.4 Class Diagram**

+User name

+password

+contact No

+Email id

+Adresss

User

**UU**

Data Base

+

**4.4.1 Interaction Diagrams**

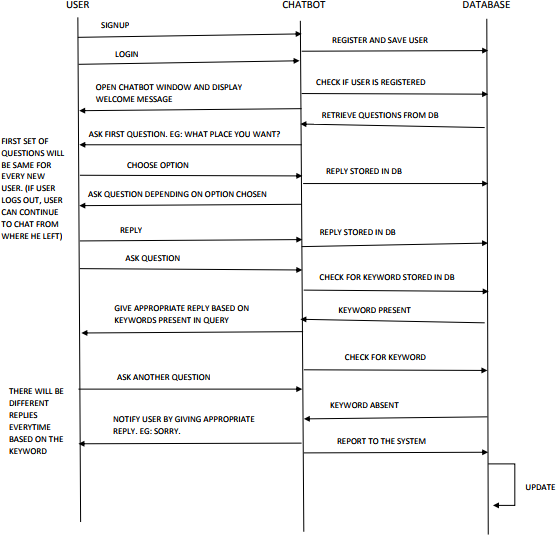
+Register()

+login()

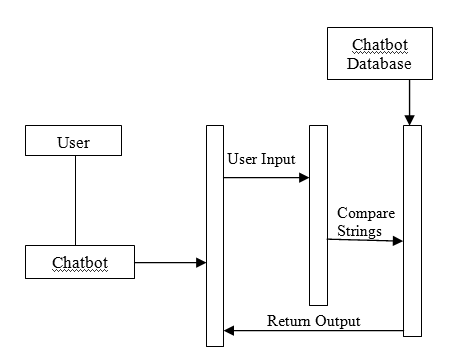
+chat with chatbot()

+log out()

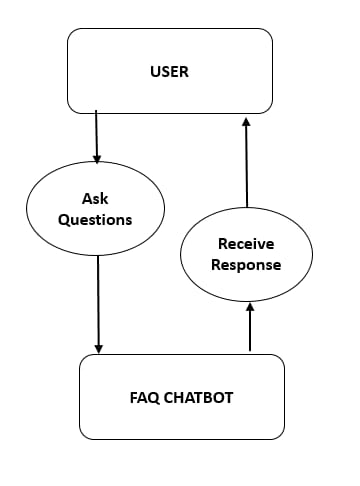
**4.4.2 Sequence Diagram**



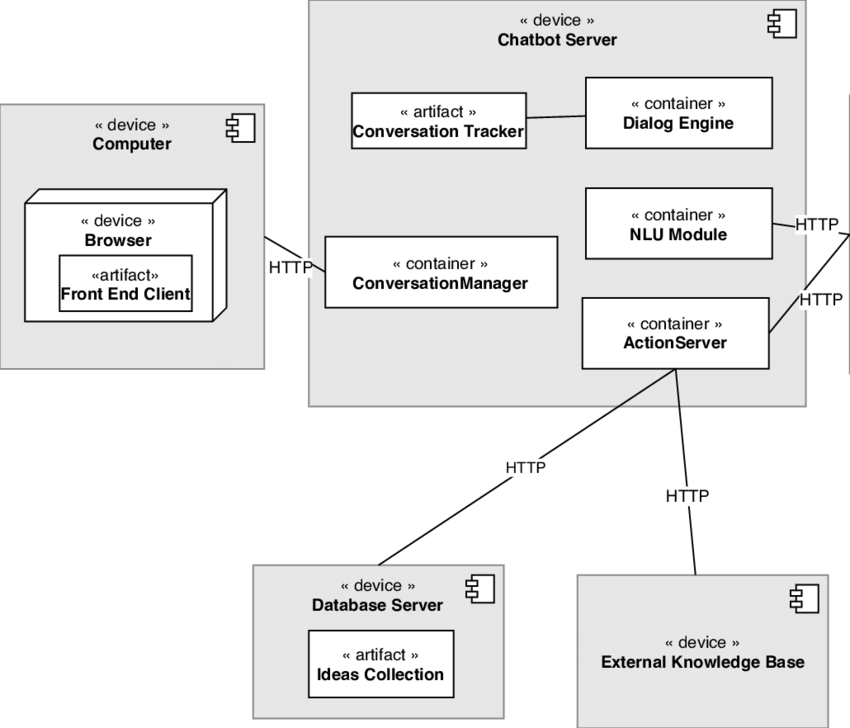
**4.4.3 Collabaration Diagram**



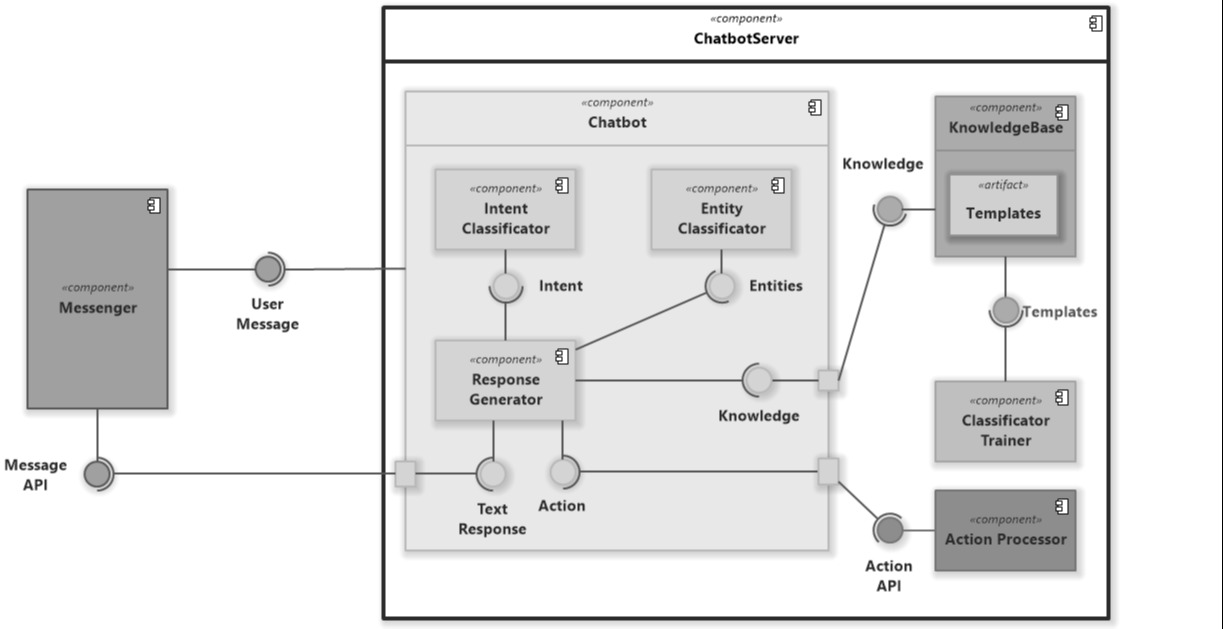
**4.4.4 State Chart Diagram**



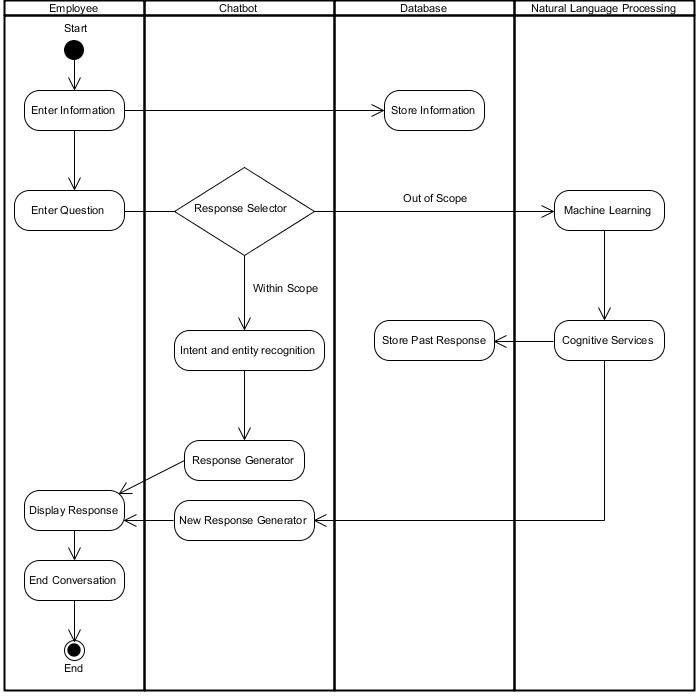
**4.4.5 Deployment Diagram**



**4.4.6 Component Diagram**



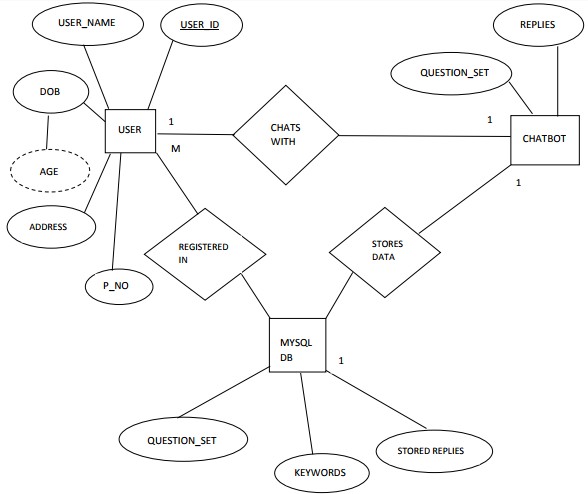
**4.4.7 Activity Diagram**



**4.4.8 Database Design**

**4.4.9 ER Diagram**

An Entity Relationship Diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data.



**Coding**

**6.Implementation**

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

import pymysql

from django.http import HttpResponse

import numpy

import tflearn

import tensorflow

import random

import json

import pickle

import nltk

from nltk.stem.lancaster import LancasterStemmer

stemmer = LancasterStemmer()

net = tflearn.input\_data(shape=[None, 46])

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, 6, activation="softmax")

net = tflearn.regression(net)

words = []

data = []

with open("C:/Python/Chatbot/MyChatBot/data.pickle", "rb") as f:

words, labels, training, output = pickle.load(f)

with open("C:/Python/Chatbot/MyChatBot/dataset/question.json") as file:

data = json.load(file)

model = tflearn.DNN(net)

model.load("C:/Python/Chatbot/MyChatBot/model/model.tflearn")

def bag\_of\_words(s, words):

bag = [0 for \_ in range(len(words))]

s\_words = nltk.word\_tokenize(s)

s\_words = [stemmer.stem(word.lower()) for word in s\_words]

for se in s\_words:

for i, w in enumerate(words):

if w == se:

bag[i] = 1

return numpy.array(bag)

def MyChatBot(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def User(request):

if request.method == 'GET':

return render(request, 'User.html', {})

def Logout(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def test(request):

if request.method == 'GET':

return render(request, 'test.html', {})

def Register(request):

if request.method == 'GET':

return render(request, 'Register.html', {})

def ChatData(request):

if request.method == 'GET':

question = request.GET.get('mytext', False)

results = model.predict([bag\_of\_words(question, words)])

results\_index = numpy.argmax(results)

tag = labels[results\_index]

str = "no result found"

for tg in data["intents"]:

if tg['tag'] == tag:

responses = tg['responses']

str = random.choice(responses)

print(question+" "+str)

return HttpResponse(str, content\_type="text/plain")

def UserLogin(request):

if request.method == 'POST':

username = request.POST.get('username', False)

password = request.POST.get('password', False)

index = 0

con = pymysql.connect("localhost","root","ROOT","chatbot")

with con:

cur = con.cursor()

cur.execute("select \* FROM register")

rows = cur.fetchall()

for row in rows:

if row[0] == username and password == row[1]:

index = 1

break

if index == 1:

context= {'data':'welcome '+username}

return render(request, 'UserScreen.html', context)

else:

context= {'data':'login failed'}

return render(request, 'User.html', context)

def Signup(request):

if request.method == 'POST':

username = request.POST.get('username', False)

password = request.POST.get('password', False)

contact = request.POST.get('contact', False)

email = request.POST.get('email', False)

address = request.POST.get('address', False)

db\_connection = pymysql.connect("localhost","root","ROOT","chatbot")

db\_cursor = db\_connection.cursor()

student\_sql\_query = "INSERT INTO register(username,password,contact,email,address) VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"')"

db\_cursor.execute(student\_sql\_query)

db\_connection.commit()

print(db\_cursor.rowcount, "Record Inserted")

if db\_cursor.rowcount == 1:

context= {'data':'Signup Process Completed'}

return render(request, 'Register.html', context)

else:

context= {'data':'Error in signup process'}

return render(request, 'Register.html', context)

**6.1 Data set (Question)**

{"intents": [

{"tag": "backache",

"patterns": ["backache","backache normal during the third","is backache normal during the third trimester of pregnancy"],

"responses": ["The extra weight you've gained is putting added pressure on your back, making it feel achy and sore. You might also feel discomfort in your pelvis and hips as your ligaments loosen to prepare for labor"],

"context\_set": ""

},

{"tag": "induction",

"patterns": ["what is labor induction","induction","labor"],

"responses": ["Induction can be the right choice for some women, but it has risks.And it doesn’t always work. If it doesn’t, you may need another induction or a c-section"],

"context\_set": ""

},

{"tag": "fevr",

"patterns": ["body temp increases","what's the deal with fever","fevr"],

"responses": ["many fish are also packed with healthy omega-3 fatty acids, which may be important in promoting your baby's brain health"],

"context\_set": ""

},

{"tag": "caffeine",

"patterns": ["drink caffeine","Can I drink caffeine while pregnant","caffeine"],

"responses": ["higher risk of miscarriage"],

"context\_set": ""

},

{"tag": "eat",

"patterns": ["two during pregnancy","eat for two during pregnancy","Do I really get to eat for two during pregnancy"],

"responses": ["The phrase eating for two implies that you get to take in double the calories you normally ingest. It's true that a normal-weight person should gain 25 to 35 pounds during a healthy pregnancy, but you don't need many extra calories to accomplish this"],

"context\_set": ""

},

{"tag": "fly",

"patterns": ["safe to fly","Is it safe to fly while pregnant","fly while pregnant"],

"responses": ["is generally safe for many mothers"],

"context\_set": ""

},

{"tag": "prenatal",

"patterns": ["prenatal care","prenatal","what is prenatal care"],

"responses": ["Prenatal care is the health care you get while you are pregnant"],

"context\_set": ""

},

{"tag": "need",

"patterns": ["why do I need prenatal care","care","need"],

"responses": ["Prenatal care can help keep you and your baby healthy. Babies of mothers who do not get prenatal care are three times more likely to have a low birth weight and five times more likely to die than those born to mothers who do get care."],

"context\_set": ""

},

{"tag": "doctor",

"patterns": ["how often should i see my doctor during pregnancy","often should i see my doctor","doctor"],

"responses": ["Your doctor will give you a schedule of all the doctor's visits you should have while pregnant. Most experts suggest you see your doctor"],

"context\_set": ""

},

{"tag": "month",

"patterns": ["About once each month for weeks 4 through 28","each month","month"],

"responses": ["If you are older than 35 or your pregnancy is high risk, you'll probably see your doctor more often."],

"context\_set": ""

},

{"tag": "drugs",

"patterns": ["What are the effects of drugs on an unborn child","effects of drugs","drugs"],

"responses": ["Studies show that using drugs -- legal or illegal -- during pregnancy has a direct impact on the fetus. If you smoke, drink alcohol, or ingest caffeine, so does the fetus. If you use marijuana or crystal meth, your fetus also feels the impact of these dangerous drugs -- and if you are addicted to cocaine, you're not only putting your own life on the line, but you are risking the health of your unborn baby."],

"context\_set": ""

},

{"tag": "castor",

"patterns": ["can you use castor oil while pregnant to bring on labor","castor oil","castor oil while pregnant","castor"],

"responses": ["Some people recommend a dose of castor oil to kick-start labor, but it tastes terrible and violent diarrhea might spoil the mood. It is not recommended."],

"context\_set": ""

},

{"tag": "urination",

"patterns": ["is frequent urination normal during the second trimester of pregnancy","frequent urination","frequent","urination"],

"responses": ["Your uterus will rise away from the pelvic cavity during the second trimester, giving you a brief break from having to keep going to the bathroom. Don't get too comfortable, though. The urge to go will come back during the last trimester of your pregnancy."],

"context\_set": ""

}

]

}

**7.Input And Output Design**

**INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**Testing**

**8. Testing**

**8.1 Testing levels**

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing presents an interesting anomaly of the software. During earlier definition and development phases, it was attempted to build software from abstract concept to a tangible implementation.

The testing phase involves the testing of the developed system using various set data. Presentation of test data plays a vital role in system testing. After preparing the test data the system under study was tested using test data. While testing the system by using test data errors were found and corrected. A series of tests were performed for the proposed system before the system was ready for implementation. The various types of testing done on the system are:

* Unit Testing
* Integration Testing
* User Acceptance Testing
* System Testing

Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. It comprises the set of test performed by the programmer prior to integration of the unit into larger system. The testing was carried out during the coding stage itself. In this step each module is found to be working satisfactorily as regards to the expected output from the module.

Each form is treated as a unit and tested thoroughly for bugs. The following is a list of some of the test cases :

1) In the login form, if a member does not enter a value for userId and password, then the user is prompted with the error message “userId and password should not be blank”.

2) In the login form, if a member enters wrong values for userId and password, then the user is prompted with the error message “Invalid userId and password. Try again.”.

3) In book Entry screen and new student, teacher screen, all the fields should have a value. Otherwise, the user is prompted with an appropriate error messages.

4) In book transactions form, member id, book no,. issue date, and return date are mandatory. If not provided, then the system will prompt the user with the error message “Fields should not be blank”.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover error associated within the interface. The objective is to take unit tested modules and build a program structure that has been dictated by design. All modules are combined in this step. The entire program is tested as whole. And chaos in interfaces may usually result. A set of errors is encountered in such a case.

The integration testing can be carried out using two methodologies:

# Top Down Integration

# Bottom Up Integration

The first one is done where integration is carried out by addition of major modules to minor modules. While Bottom Up integration follows combination of smaller ones to larger one. Here, Bottom Up Integration is followed. Even though correction was difficult because the isolation of causes is complicated by the vastness of the entire program, all the errors found in the system were corrected and then forwarded to the next testing steps.

The navigation among all the screens have been thoroughly verified so that the user of the system can move from one form to another form.

The connectivity between the forms and the database has been checked. In case of any malfunctions, the user will be informed about the problem.

User Acceptance Testing

User acceptance of a system is the key factor for the success of any system. The system under consideration was tested for users acceptance by constantly keeping

in touch with the perspective system user at the time of developing and making changes wherever required. This is done with the regards to the following points:

A system may be defined as a set of instructions combined in the same form and directed to some purpose.

Before any development is undertaken certain specifications are prepared which objectively describe the application system. The System specifications are made after consulting the end user managers of the relevant departments.

Software to be developed is planned on the basis of requirement of the user. The problem definition statement description of present situation and goal to be achieved by news system.

The success of system depends on how accurately a problem is defined, thoroughly investigated carried out through choice of solution. User need identification and analysis that are concerned with what the uses needs rather than what he/she wants. System explains how to perform specific activities or task, which does what and what.

System Testing

Testing the behavior of the whole software/system as defined in software requirements specification(SRS) is known as system testing, its main focus is to verify that the customer requirements are fulfilled.

System testing is done after integration testing is complete. System testing should test functional and non functional requirements of the software. The test types followed in system testing differ from organization to organization.

The project is executed and tested on the machines that satisfy the given hardware and software requirements. It was executed successfully with the specified hardware and operating system.

**Test case**

**Test case for Login form:**

|  |  |  |
| --- | --- | --- |
| **FUNCTION:** |  | **LOGIN** |
| **EXPECTED RESULTS:** |  | Should Validate the user and check his existence in database |

|  |  |
| --- | --- |
| **ACTUAL RESULTS:** | Validate the user and checking the user against the database |
| **LOW PRIORITY** | **No** |
| **HIGH PRIORITY** | **Yes** |

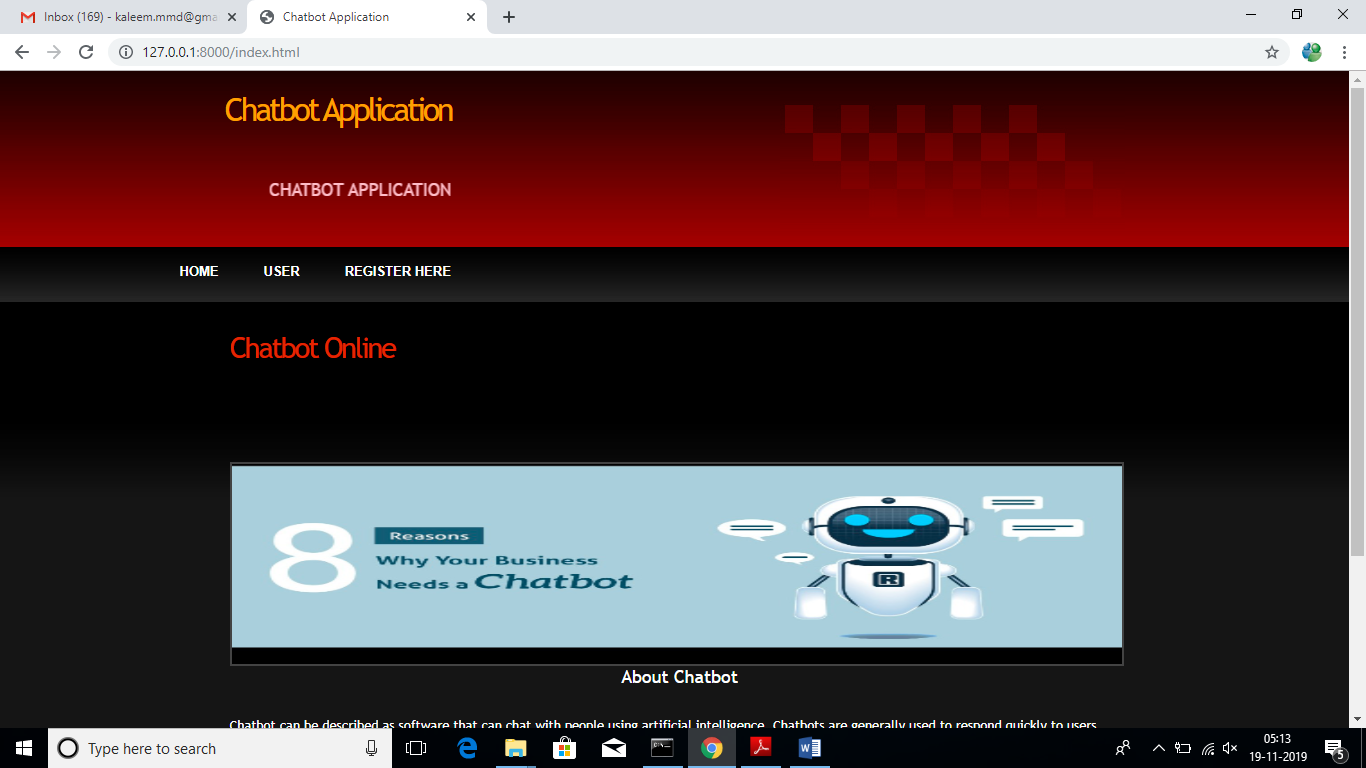
**Test case for User Registration form:**

|  |  |
| --- | --- |
| **FUNCTION:** | **USER REGISTRATION** |
| **EXPECTED RESULTS:** | Should check if all the fields are filled by the user and saving the user to database. |
| **ACTUAL RESULTS:** | Checking whether all the fields are field by user or not through validations and saving user. |
| **LOW PRIORITY** | **No** |
| **HIGH PRIORITY** | **Yes** |

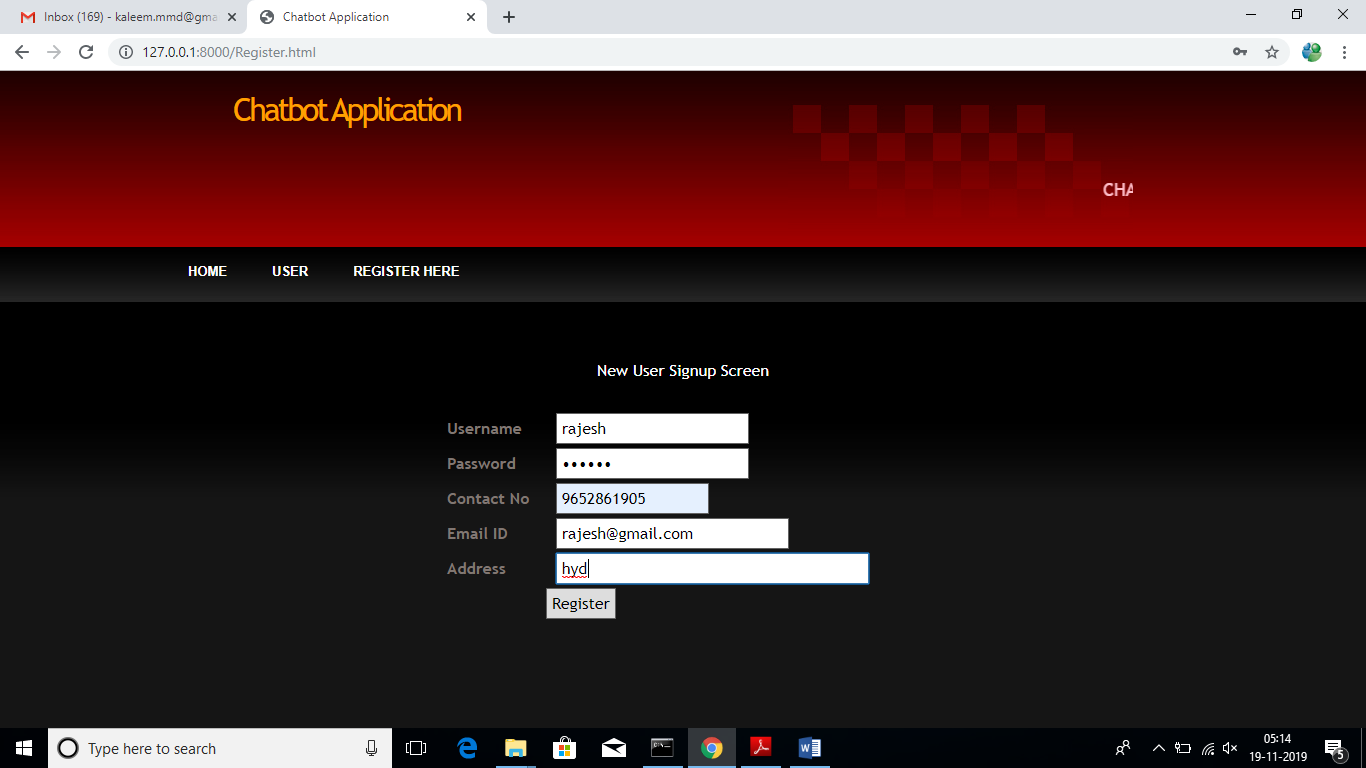
**Screens**

**9. Screens**

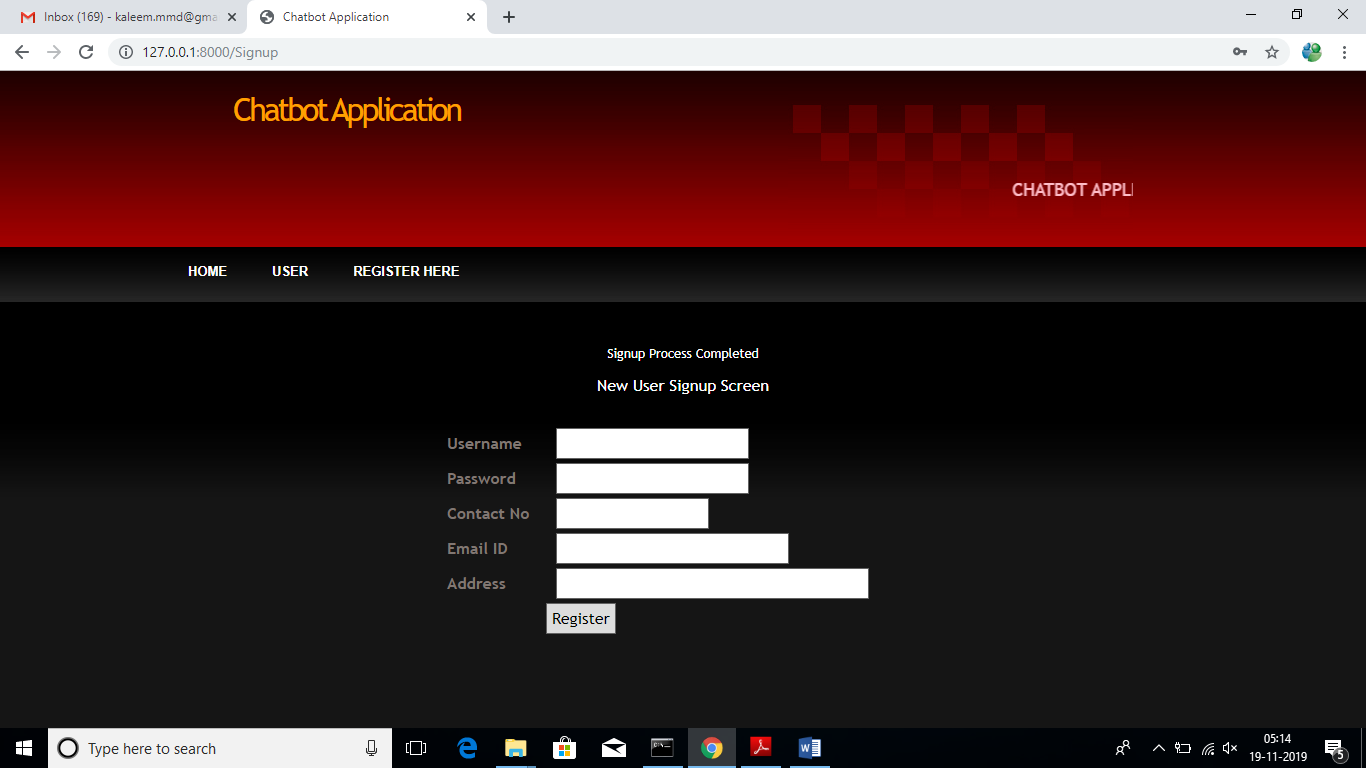
To run this project we need to install python and MYSQL and then deploy application on python DJANGO server. After setting up application we need to run on browser to get below screen



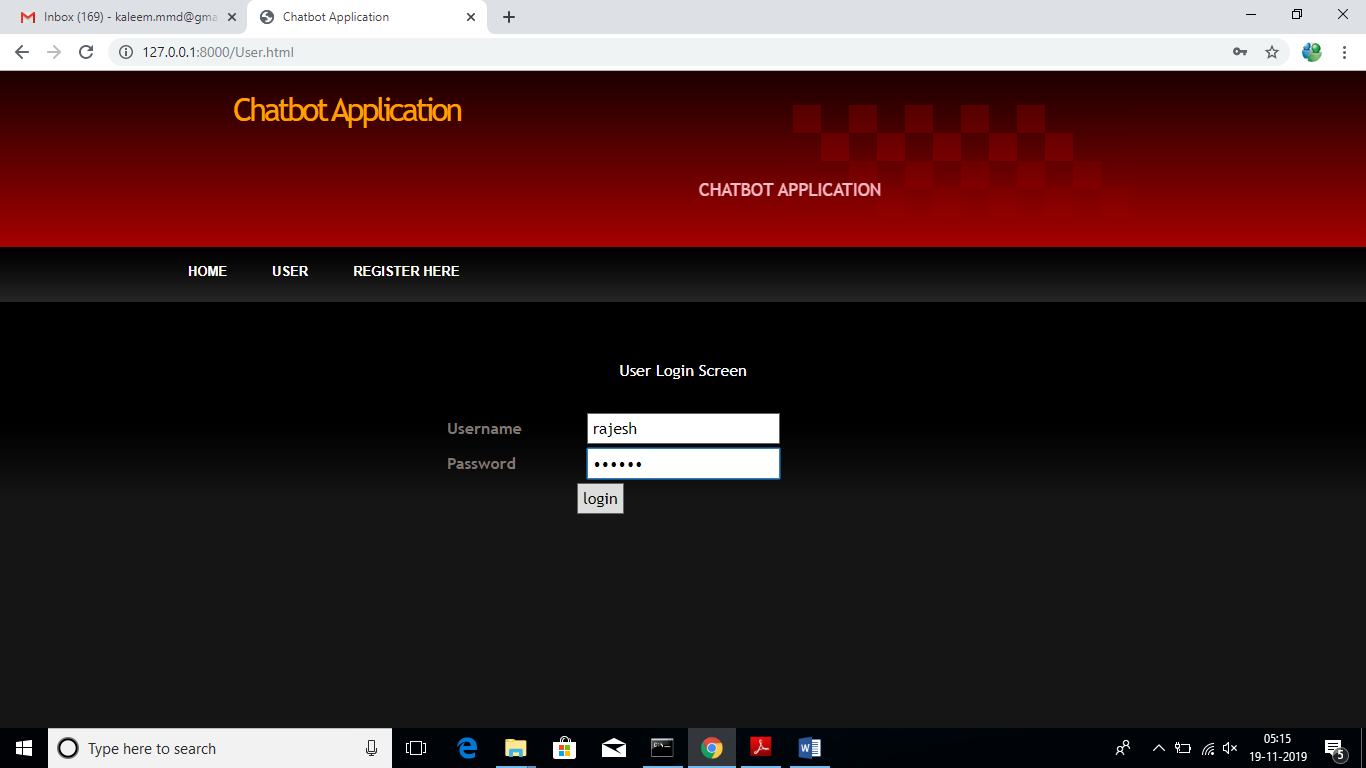
In above screen click on ‘Register Here’ link to add sign up with application



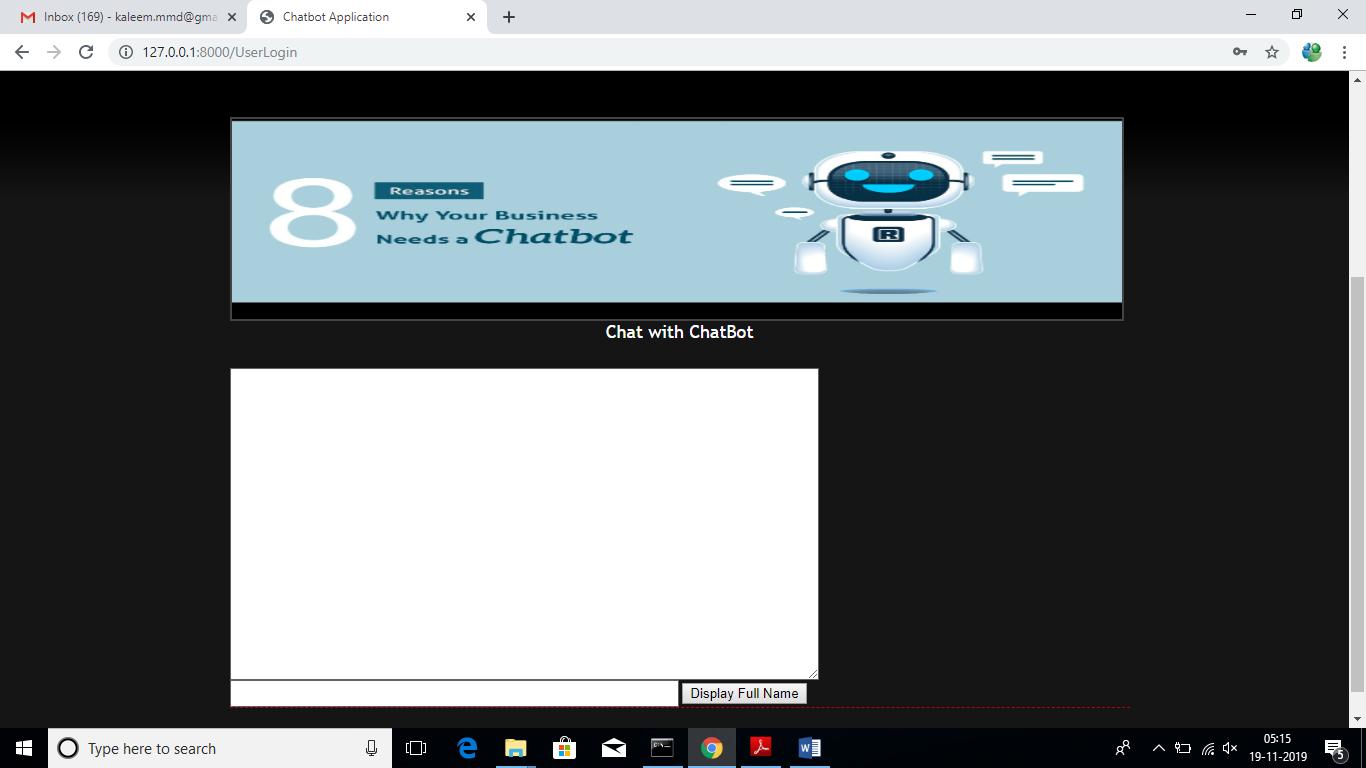
After signup will get below confirmation screen



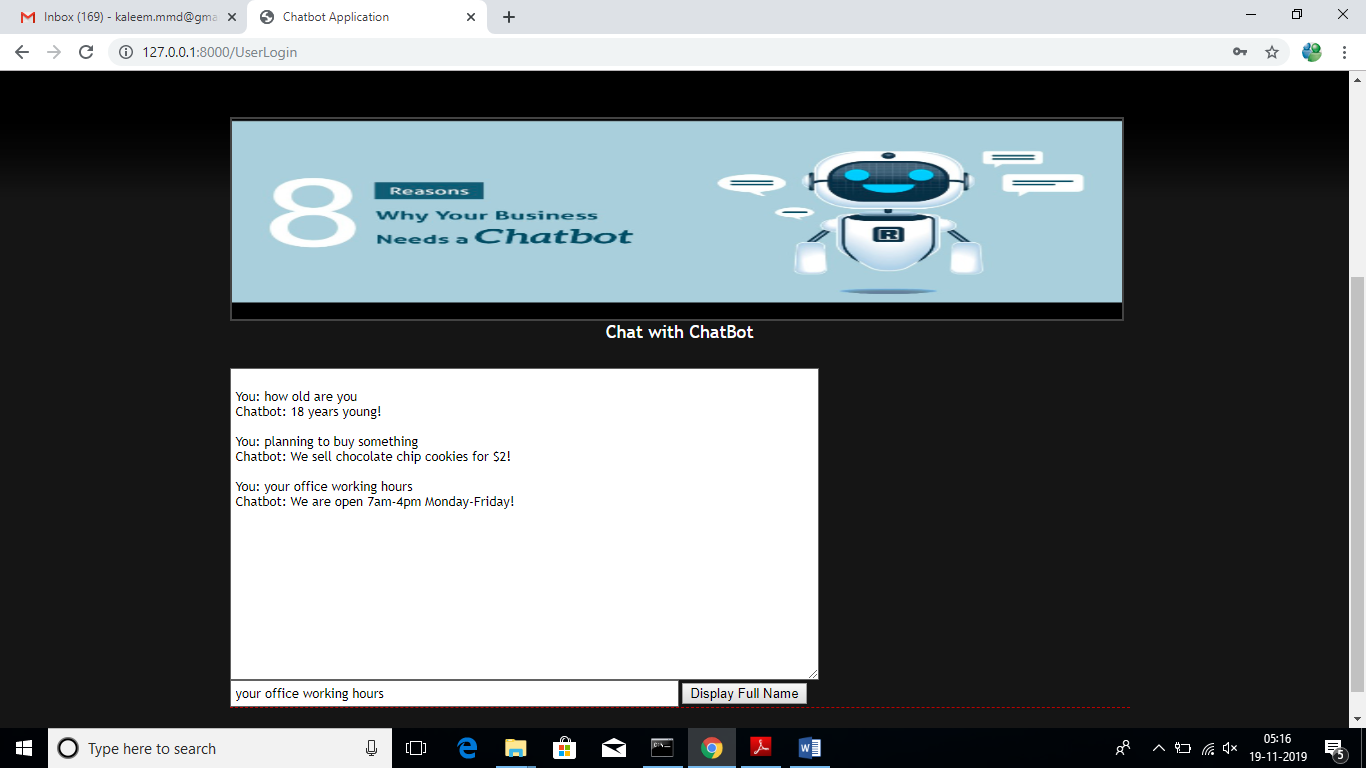
Now click on ‘User’ link to login to application



After login will get below user chat bot screen



In above screen user can enter questions and chat bot replies to that questions. See below screen



In above screen we can see I asked 3 questions and chat bot answer them correctly. Similarly we can ask any question and chat bot can answers those questions as long as those question answers are available inside training model of deep learning object.

Future Scope of the project

10. Future Scope of the project

We would further like to integrate the Chatbot system with Google Maps. This will assist the user to navigate to the various desired project locations. The Chatbot system can also be integrated with different websites to serve queries about different projects by different builders. Integrating it with leading real estate websites provides more options for the user, enabling him to choose the best suitable option from a wide range of projects.

**Conclusion**

**11 .CONCLUSION**

This work introduced the fundamentals of what chatbots are. It gave an overview about ideas, products and platforms, both, from the past and available today. The current interest in chatbots, potential use cases and limitations have been explored in detail. Different aspects of the implementation of a chatbot and working with conversational interfaces have been presented through the creation of an exemplary chatbot, which included interaction and user experience design, and a general, reusable software architecture for chatbots. While not all aspects can be covered within the context of this work, the goal was to give an overview about what chatbots are, their use cases and how to create them. This knowledge should help exploring further possibilities of chatbot usage and it should enable more developers to apply chatbots to new scenarios and thereby also improve human-machine interaction in general.

**Bibilography**

**12.Bibilography**

1. H. P. Levy, “Gartner Predicts a Virtual World of Exponential Change.” https://www.gartner.com/smarterwithgartner/ gartner-predicts-a-virtual-world-of-exponential-change/, 2018. [Online; 9- ¸Subat.-2018’de eri¸sildi.].

2.A. M. Turing, “Computing machinery and intelligence,” in Parsing the Turing Test, pp. 23–65, Springer, 2009.

3.L. Bradeško and D. Mladenic, “A survey of chatbot systems through a ´ loebner prize competition,” in Proceedings of Slovenian Language Technologies Society Eighth Conference of Language Technologies, pp. 34– 37, 2012.

4. M. v. Eeuwen, “Mobile conversational commerce: messenger chatbots as the next interface between businesses and consumers,” Master’s thesis, University of Twente, 2017.

5. AT&T, “Meet Atticus: The Entertainment Chatbot from AT&T.” http: //about.att.com/newsroom/meet\_atticus.html, 2018. [Online; 13- ¸Subat.- 2018’de eri¸sildi.].

6.T-Mobile, “Tinka: T-Mobile.” https://www.t-mobile.at/tinka/, 2018. [Online; 13- ¸Subat.-2018’de eri¸sildi.].

7.E. D. Liddy, “Natural language processing,” In Encyclopedia of Library and Information Science, 2001. 2nd Ed. NY. Marcel Decker, Inc.

8. “Facebook for Developers.” https://developers.facebook.com, 2018. [Online; 13- ¸Subat.-2018’de eri¸sildi.].