

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**PURWANCHAL CAMPUS**

**A**

**MINOR PROJECT REPORT**

**ON**

**“OPEN LEDGER USING BLOCKCHAIN”**

A PROJECT WAS SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND

COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE

REQUIREMENT FOR THE BACHELOR’S DEGREE IN COMPUTER ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PURWANCHAL CAMPUS,

DHARAN, NEPAL

SUBMITTED BY

ANUJ TIMSINA

PRAJWAL CHHETRI

SACHI KAMAT

SANDIP THAPA

August, 2021



**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**PURWANCHAL CAMPUS**

**A**

**MINOR PROJECT REPORT**

**ON**

**“OPEN LEDGER USING BLOCKCHAIN”**

Submitted By:

ANUJ TIMSINA

PRAJWAL CHHETRI

SACHI KAMAT

SANDIP THAPA

**ER. PRASANNA REGMI**

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

DHARAN, NEPAL

August, 2021

# APPROVAL LETTER

The undersigned certify that they have read and recommended to the Institute of Engineering for acceptance, a project report entitled "Open Ledger Using Blockchain" submitted by:

ANUJ TIMSINA

PRAJWAL CHHETRI

SACHI KAMAT

SANDIP THAPA

In partial fulfillment for a degree of Bachelor in Computer Engineering.

...............................

Supervisor

Er. Prasanna Regmi

................................

External Examiner

Prof. Dr. Subarna Shakya

……………………………….

Professor Mazhar Ali

Head of Department

Department of Computer and Electronics Engineering

# COPYRIGHT ©

The authors have agreed that the Library, Department of Electronics and Computer Engineering, Purwanchal Campus, Institute of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, Purwanchal Campus, Institute of Engineering in any use of the material of this project report. Copying or publication or the other use of this report for financial gain without approval of the Department of Electronics and Computer Engineering, Purwanchal Campus, Institute of Engineering and author’s written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head of Department

Department of Electronics and Computer Engineering,

Institute of Engineering, Purwanchal Campus,

Tribhuvan University, Nepal

# ACKNOWLEDGEMENT

First of all, we are very grateful to the Department of Electronics and Computer

Engineering, IOE Purwanchal Campus for providing us the opportunity to undertake this project by including it as a part of curriculum in B.E. Computer Engineering.

We are extremely thankful to Er. Prasanna Regmi, our Project Supervisor for providing us with his invaluable guidance and support throughout every phase of the project development. His useful suggestions and continuous motivation are sincerely acknowledged.

Also, we express sincere thanks to our colleagues for their positive advice and support and to all those who helped us directly or indirectly to complete this project on time.

ANUJ TIMSINA

PRAJWAL CHHETRI

SACHI KAMAT

SANDIP THAPA

# ABSTRACT

Data manipulation is often named as a serious threat to data integrity. Data can be tampered with, and malicious actors could use this to their advantage. Data users in various application domains want to be ensured that the data they are consuming are accurate and have not been tampered with.

To validate the integrity of these data, we describe a blockchain based hash validation method. The method assumes that the actual data is stored separately from the blockchain, and then allows a data identifier and a hash of these data to be submitted to the blockchain. The actual data can be validated against the hash on the blockchain at any time. Several use cases are described for blockchain-based hash validation, and to validate the method it is implemented inside an application audit trail to validate the audit trail data. This implementation shows that blockchain based hash validation is able to detect malicious and accidental changes that were made to the data.

Open ledger using blockchain allows users to view all transactions in a transparent way. Blockchain has the potential to be a game-changer in anti-corruption efforts. A blockchain is designed to operate in environments where trust in data/code is greater than trust in individuals or institutions. Records entered in the blockchain are transparent and immutable. When blockchains hold registries of physical items, trusted gatekeepers have to ensure that the physical reality and digital information correspond.

**Keywords**

NLP, Python, Keras, RNN, Pytorch

**Table of Contents**

[APPROVAL LETTER i](#_Toc72514284)

[COPYRIGHT © ii](#_Toc72514285)

[ACKNOWLEDGEMENT iii](#_Toc72514286)

[ABSTRACT iv](#_Toc72514287)

[LIST OF FIGURES viii](#_Toc72514288)

[1. INTRODUCTION 1](#_Toc72514289)

[1.1. Overview 1](#_Toc72514290)

[1.2 Problem Definition 1](#_Toc72514291)

[1.3 Objectives 2](#_Toc72514292)

[1.4 Scope of the project 2](#_Toc72514293)

[2. LITERATURE REVIEW 4](#_Toc72514294)

[2.1.1 History of Web Development 4](#_Toc72514295)

[2.2 Different Approaches of Website Development 5](#_Toc72514296)

[3. METHODOLOGY 6](#_Toc72514297)

[3.1 General Description 6](#_Toc72514298)

[3.2 System design 7](#_Toc72514299)

[3.3 Dataset Preparation 12](#_Toc72514300)

[3.3.1 Dataset Collection 12](#_Toc72514301)

[3.3.2 Voice to text conversion: 13](#_Toc72514302)

[3.4 Tokenization,Lemmatization: 13](#_Toc72514303)

[3.5 Model Training 14](#_Toc72514304)

[3.6 Identifying tags and Attributes: 15](#_Toc72514305)

[3.7 Render the HTML file: 15](#_Toc72514306)

[4. IMPLEMENTATION 16](#_Toc72514307)

[4.1 Software Development Life Cycle 16](#_Toc72514308)

[4.2 Requirement Analysis 17](#_Toc72514309)

[4.2.1 Functional Requirement 18](#_Toc72514310)

[4.2.1.1 Voice Recognition 18](#_Toc72514311)

[4.2.1.2 Tag Recognition 18](#_Toc72514312)

[4.2.1.3 Attribute Recognition 18](#_Toc72514313)

[4.2.1.4 Character Segmentation and Tokenization 19](#_Toc72514314)

[4.2.2 Non Functional Requirement 19](#_Toc72514315)

[4.2.2.1 Device Software Compatibility 19](#_Toc72514316)

[4.2.2.2 Offline Accessibility 19](#_Toc72514317)

[4.3 Feasibility Study 19](#_Toc72514318)

[4.3.1 Technical Feasibility 19](#_Toc72514319)

[4.3.2 Economic Feasibility 20](#_Toc72514320)

[4.3.3 Schedule Feasibility 20](#_Toc72514321)

[4.4 Programming Language 20](#_Toc72514322)

[4.4.1 Python and Text Editor 20](#_Toc72514323)

[4.4.2 HTML 22](#_Toc72514324)

[4.4.3 CSS 22](#_Toc72514325)

[4.4.4 Javascript 22](#_Toc72514326)

[4.5 GUI Development 23](#_Toc72514327)

[4.6 Project Scheduling 24](#_Toc72514328)

[5. CODE 25](#_Toc72514329)

[6. RESULT 30](#_Toc72514330)

[7. OUTPUT 34](#_Toc72514331)

[8. CONCLUSION 35](#_Toc72514332)

[9. LIMITATION AND FURTHER WORK 35](#_Toc72514333)

[4.7 Limitations 35](#_Toc72514334)

[4.8 Further Works 36](#_Toc72514335)

[10. REFERENCE 37](#_Toc72514336)

# LIST OF FIGURES

[Figure 3.1 Block diagram of working of system](#_1ksv4uv) 6

[Figure 3.2 Schematic diagram of system design](#_2jxsxqh) 7

[Figure 3.3 Use Case Diagram of MVMW system](#_z337ya) 8

[Figure 3.4 DFD level 0](#_3j2qqm3) 9

[Figure 3.5 DFD level 1](#_1y810tw) 10

[Figure 3.6 Sequence diagram of MVMW system](#_4i7ojhp) 11

[Figure 4.1 Software Development Life Cycle](#_1hmsyys) 15

[Figure 4.2 Iterative Methodology for software development](#_41mghml) 16

[Figure 4.4 Render page](#_41mghml)  22

Figure 5.4 Plot between Model accuracy and no. of epochs 31

Figure 7.1 Output 34

# 

# INTRODUCTION

## Overview

Voice to website development system aims at developing websites easily and efficiently with only the input of voice command without doing the single line of coding. It provides the means to overcome the drawbacks and deficiencies that web development without the knowledge of any programming languages is impossible.

The system is capable of processing the voice input given by the user, extract useful information from the command, and develop according to it. Nowadays web development is something that we can’t ignore in this digital world and has a huge scope in the future as well. Drag and drop concepts for the development of the website are there in the market but the development with the voice is something new and interesting.

In the future, this system can be used in the development of the website by people who have zero knowledge about programming.

The development process is generally divided into 2 parts:

1. Speech to text i.e conversion of the speech input given by the user to the text.
2. Rendering text i.e process the text input to create and render as the website.

## Problem Definition

The voice to website is the new emerging concept in the development of the website. Till now the drag and drop concept of website development has been used but still the development of the website from voice will be super cool.

The main problem with web development by coding from scratch is that it would take a lot of time in coding for a task that could be done in minutes. Likewise, only skilled manpower could perform the web development task as special programming skills and training are required. So the main concept of our project is to cope with this problem in web development providing the easy and effective approach

Also till now, AI has not been used in the development of the website i.e the concept of intelligent agents has not been applied which actually codes for us, in our case takes input as voice and code according to it.

## Objectives

**1.3.1 Project objectives**

The prime objective of the project being proposed is to design and build a system for Website Development with the help of voice command only .

The typical objectives are listed below:

* To develop the website using the voice command
* To facilitate the Website development easy and user-friendly
* To learn deep learning and its utilization in the project

1.3.2 Academic objectives Academically, the project is primarily focused on fulfilling the discipline of an engineering student as a computer engineer working on a project and gain experience as a team throughout the different phases of a project.

Some Typical academic objectives of the project are:

* To fulfill the requirements of the minor project of bachelor’s in computer engineering.
* To design and complete a functional project that integrates various course concepts.
* To develop various skills related to project management like teamwork, resource management, documentation and time management.
* To get hands-on experience of working in a project as a team.
* To learn about and become familiar with the professional engineering practices

## Scope of the project

For any business, a website matters more than anything, when it comes to reaching out to customers online. A website is the representation of the business online. These days, every business realizes the [need for having a website](https://acodez.in/top-reasons-why-startups-need-website/) and are putting in efforts to design and develop the best site for taking their products or services online. This is where we can see a great deal of scope for web development and design

One of the latest areas of AI is in web development . There are no templates or coding – all you have to do is provide instructions, and it will lay out what you want. It implements AI to make this process easier. So now, the time has come for a web developer to be an AI web developer, who knows better about the future wherein web development is more inclined toward giving the consumers what they need and not just giving them what the developer is capable of creating.

So our project “MY VOICE MY WEBSITE” can be one of the leading web development techniques in the near future with its modern concept of development.

# LITERATURE REVIEW

**2.1 History**

We are going to discuss the history of web development and Artificial Intelligence.

## 2.1.1 History of Web Development

In 1993 Tim Berners-Lee wrote the first version of HTML. He was a physicist and a computer scientist who implemented the first successful communication between a Hypertext Transfer Protocol (HTTP) client. He is essentially the inventor of the Internet we know and love today.HTML had many iterations throughout the next couple of years. HTML v2 was released in 1995, followed by v3 and v4 in 1997. Finally, that leads us to HTML5, which was released in 2011.

During the first year, HTML looked pretty bland. HTML v1 was an ugly baby. But, that all started to change in late 1994 when Håkon Wium Lie published the first draft of the Cascading HTML Style Sheets (CSS) proposal.

The first release of CSS stirred a little drama. It was perceived by many as being too simple for the task it was designed for. It was argued that to style documents, the power of a full programming language was needed.

After a year of back and forth, CSS made it into the recommendation of the World Wide Web Consortium (W3C). In 1997 CSS v2 was published and used more widely in browsers, including Netscape Navigator.

Back in 1995, a Netscape programmer named *Brandan Eich* developed a new scripting language called Mocha, which eventually became known as JavaScript in the modern world.Server-side JavaScript was also implemented in 1995, but the true champion of server-side JS is Node which was released in 2009.

That is a short history of the solid foundations (HTML, CSS, JavaScript) we use today for every web application.

**2.1.2 History of AI**

In the 1940s, a school of thought called “Connectionism” was developed to study the process of thinking. In 1950, a man named Alan Turing wrote a paper suggesting how to test a “thinking” machine. He believed if a machine could carry on a conversation by way of a teleprinter, imitating a human with no noticeable differences, the machine could be described as thinking. His paper was followed in 1952 by the Hodgkin-Huxley model of the brain as neurons forming an electrical network, with individual neurons firing in all-or-nothing (on/off) pulses. These events, at a conference sponsored by Dartmouth College in 1956, helped to spark the concept of Artificial Intelligence.

Today, AI technology is currently deployed in numerous sectors, including transportation, manufacturing, finance, health care, education, and urban planning. There is no place left where AI is not being used these days.

## 2.2 Different Approaches of Website Development

Different approaches for the development of web are:

**• Scripting (or Programmatic) Approaches**

The page is generated primarily from code written in a scripting language or a high level language. In this method, the website is completely developed from the scratch by coding using different languages required like HTML,CSS, Perl, PHP, Javascript, Python, Bash, Ruby, and M4by the programmer. For this the programmer should have the sound knowledge about the different languages mentioned above. Only the skilled manpower can use this method for web development. IOn the other hand, this method is time consuming as everything should be done from scratch.

**• Template Approaches**

The approach of using templates in the development of the web is one of the most easy and effective ways of website development. It utilizes a source object (the template) that focuses on formatting with a limited set of embedded programming constructs. In this, the developer uses the predesigned and programmed template in the development just by editing the content of the system without making any changes in the design. we shouldn't have to code from scratch , so the web development is faster this way but the user satisfaction is least in this approach.

**• Hybrid Approaches**

This approach is the combination of programmatic approach and template approach. The inbuilt template is used in the startup and different changes are made in design later on as per the requirements. So this approach is an effective way of web development as the development is sped up by the use of templates and different changes are made in the code by programming which helps in user satisfaction.

# 3. METHODOLOGY

## General Description

The system developed through this project can be viewed with six cases

1. Voice to Text Conversion- The first step is to accept the voice command from the user.This process is carried out by Google-Text-To-Speech Python Library.

2. Word Tokenization- It includes the steps after collecting and converting the voice in speech. here the sentence is fragmented and a certain token is assigned.

3. Predicting Parts of Speech- After the tokens are assigned, the role of individual token is identified.

4. Text Lemmatization, Identifying Stop Words, Dependency Parsing

5. Named Entity Recognition: Detect and Label nouns with real world concepts that they represent.

5.1. Synthesize the HTML tag

5.2 Understand the inner HTMLtag and attribute if necessary.

5.3 Store the incoming request in required form.

5.4 Display the HTML file as per requested tag and attribute.

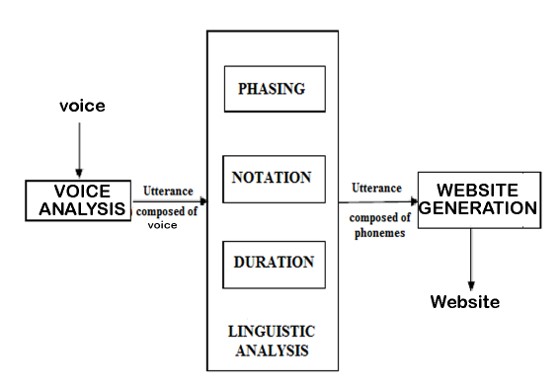


Figure 3.1 Block diagram of working of system

## System design

The schematic diagram of the developed systems is given below

3.2.1 Schematic diagram of system design

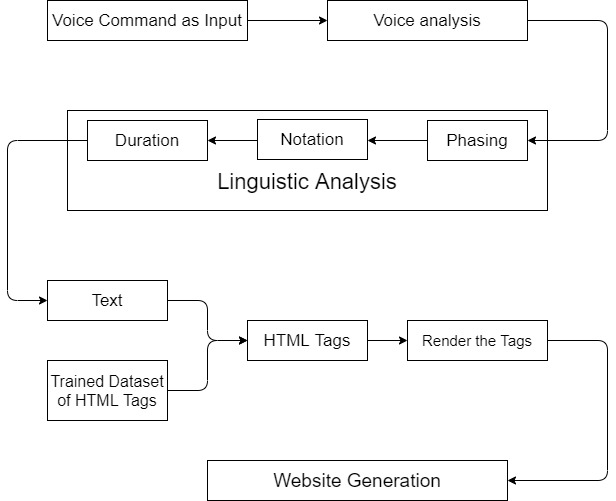


Figure 3.2.1 Schematic diagram of system design

3.2.2 Use Case Diagram of MVMW system

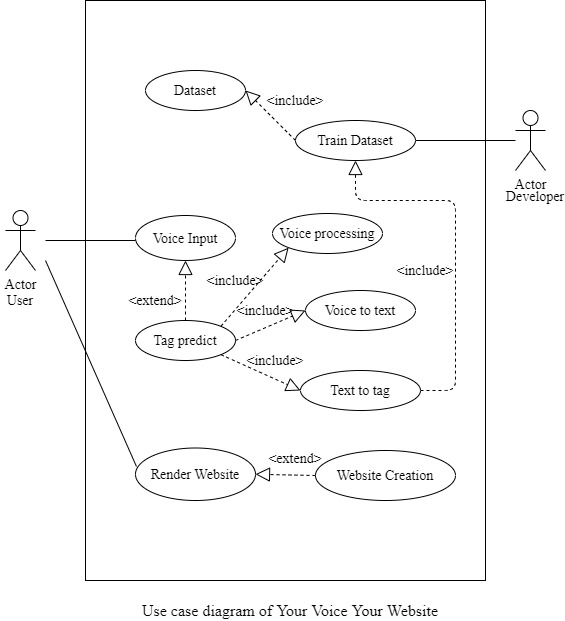


Figure 3.2.2 Use Case Diagram of MVMW system

3.2.3 DFD level 0

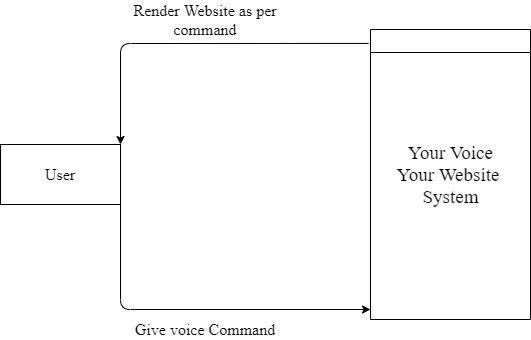


Figure 3.2.3 DFD level 0

3.2.4 DFD level 1

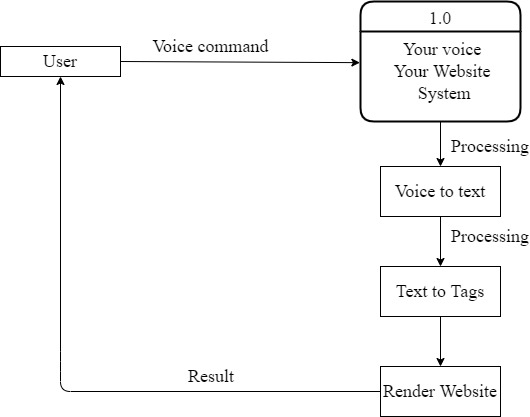


Figure 3.2.4 DFD level 1

3.2.5 Sequence diagram of MVMW system

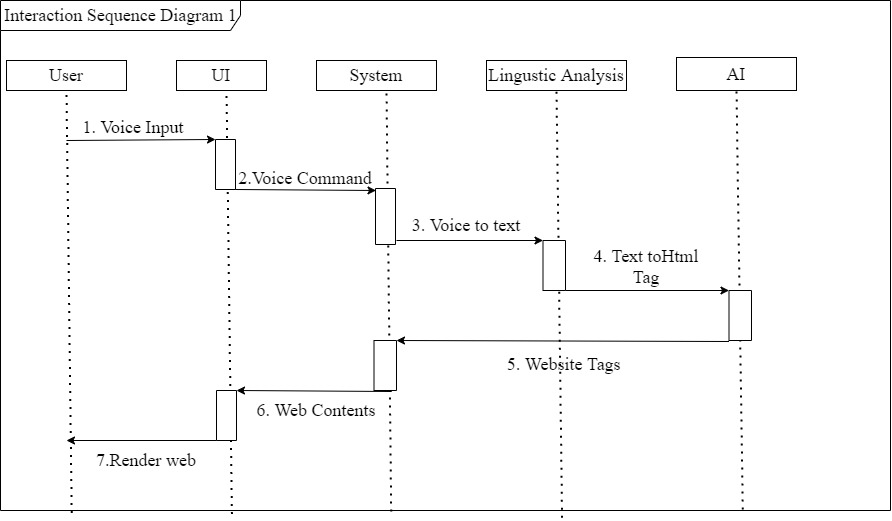


Figure 3.2.5 Sequence diagram of MVMW system

## Dataset Preparation

### Dataset Collection

We listed the HTML tag in json format. There are numerous HTML tags available . In this project we included 29 tags .

Along with tags their attributes are also listed.

Tags:

1.Anchor tag(a) 2. Heading tag(H1 to H6) 3.Image tag(img) 4. Footer(footer)

5. Horizontal Line(hr) 6. Line Break(br) 7. Fieldset(fieldset) 8.Textarea(textarea)

9.Dialog(dialog) 10.Input (input) 11. Form(form) 12. Italic(i)

13.Bold(b) 14. Strong(strong) 15. Label(label) 16. Caption(caption)

17. Navigation bar(nav) 18.Select(select)

19.Option(option) 20. Button(button) 21. Table(table)

22. Table row(tr) 23.Table Heading(th) 24. Table Data(td)

25. Anchor(a) 26.Paragraph(p) 27. Image(img)

28. Unordered List(ul) 29.Ordered List(ol)

\*\*tags along with their name.

Attributes included:

1. src: Source Attribute
2. class
3. id
4. style
5. action
6. placeholder
7. type
8. href :Hypertext Reference

### Voice to text conversion:

Here the input voice command is converted into text. This work is done by using a special python library called, Speech Recognition . It is a python library for recognizing speech with the support of several engines and APIs, online and offline. Finally the user inserted voice command is successfully converted into text for further processing.

## Tokenization,Lemmatization:

**Tokenization** is the process of removing sensitive data from your business systems by replacing it with an undecipherable token and storing the original data in a secure cloud data vault.

**Lemmatization** in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. It refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word.

## Model Training

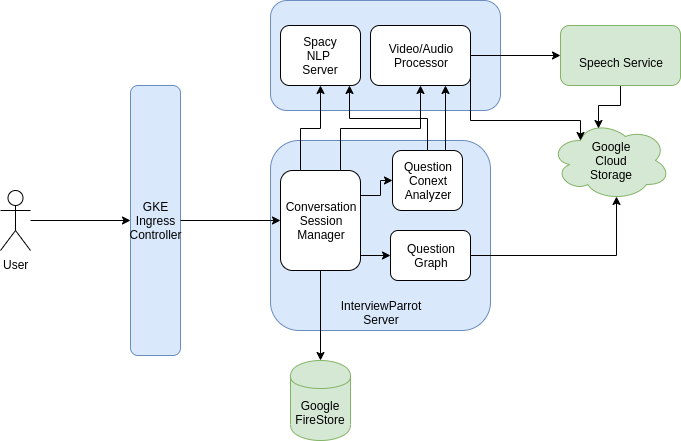


Figure 3.9 Use Case Diagram of NLP

## Identifying tags and Attributes:

The given voice command is evaluated after the model is trained. The model with probability of 80% predicts the actual tag that user is referencing.



## Render the HTML file:

Finally after the application understands the tag and its attribute, the system further processes to generate the required HTML code. Thus we render the frontend after the code is generated.

# IMPLEMENTATION

## Software Development Life Cycle

The Software Development Life Cycle (SDLC) is the process of developing the software in a fixed standard pattern. The SDLC process involves several distinct stages, including planning, analysis, design, building, testing, deployment and maintenance.



Figure 4.1 Software Development Life Cycle

[Source: www.javatpoint.com]

Among several SDLC methodologies, in our project we have used iterative methodology. Before using this, why would anyone consider an iterative approach? This model gives you a working version early in the process and makes it less expensive to implement changes. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

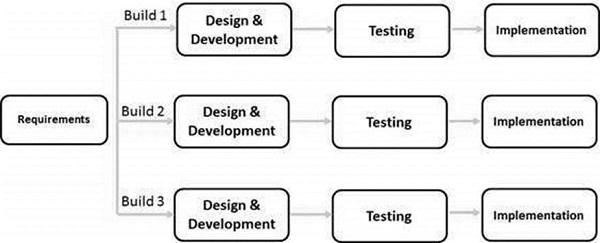


Figure 4.2 Iterative Methodology for software development

[source: www.javatpoint.com]

In this software development model, major requirements are defined; however, some functionalities or requested enhancements may evolve with time. Some working functionality can be developed quickly and early in the life cycle. Results are obtained early and periodically. Parallel development can be planned. Less costly to change the scope/requirements. Risk analysis is better and it supports changing requirements. Partial systems are built to produce the final system.

By following this model, we focused on building several components of the system in an incremental basis and finally those components are merged together to form a total functional system of Natural Language Processing and Deep Learning based frontend generation.

## Requirement Analysis

Requirement analysis, also called requirements engineering, is the process of determining user expectations for a new system that is going to be built. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements must be quantifiable, relevant and detailed.

The requirements for the project is the voice command that specifies the tag and attribute of the HTML element in clear voice. This identification of tag is done by training the model under certain epochs to obtain necessary accuracy.

### Functional Requirement

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs

#### Voice Recognition

Speech recognition is an [interdisciplinary](https://en.wikipedia.org/wiki/Interdisciplinary) subfield of [computer science](https://en.wikipedia.org/wiki/Computer_science) and [computational linguistics](https://en.wikipedia.org/wiki/Computational_linguistics) that develops [methodologies](https://en.wikipedia.org/wiki/Methodology) and technologies that enable the recognition and [translation](https://en.wikipedia.org/wiki/Translation) of spoken language into text by computers.

In our project, we should be able to understand what the user has given the input through the voice command and furthermore it should be able to convert the given voice command into the text.

#### Tag Recognition

Tags can be defined as the instructions which are being directly embedded in the text of an HTML document. The types of tags used in the HTML document are responsible to tell a web browser to do something (follow the instruction) instead of just displaying text. In an HTML document, all tag names are differentiated from other simple text. The tag names are enclosed in between angle brackets or a ‘less than’ and a ‘greater than’ symbol, (<) and (>).

In our project, we should be able to identify the tag based on the input text extracted from the voice command. We use artificial intelligence for the extraction of the appropriate tag from the text. For this we train the dataset which contains the tags and the appropriate text to point the tags.

#### Attribute Recognition

HTML attributes are a modifier of an HTML element type. An attribute either modifies the default functionality of an element type or provides functionality to certain element types unable to function correctly without them.

For the development of the webpage we should consider the different attributes like the background, background color, sources of images and so on. we should be able to recognize the attributes given by the user and work accordingly.

#### Character Segmentation and Tokenization

Character Segmentation is the most crucial step for text to tag identification. The selection of segmentation algorithms being used is the key factor in deciding the accuracy of tags in our system. If there is a good segmentation of characters, the recognition accuracy will also be high.

Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either words, characters, or subwords. So on the basis of the segmentation the tokenization task is performed.

### Non Functional Requirement

Non functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

#### Device Software Compatibility

Currently the application is capable of running on the web locally in PC’s. Furthermore it can work remotely on any platform.

#### Offline Accessibility

The application can’t be accessed offline as it is a web based application.

## Feasibility Study

Voice Recognition is one of the hottest topic in the current field of technology and science. Many researches have been carried out in this field from several decades ago to till today and many of them are under study to optimize the study.

### Technical Feasibility

All the technical feasibility needs for the project are met as the system requirement for the project tools are sufficient as per the recommendation by the tool’s vendor. Similarly development of new artificial intelligence and different prediction models have made it very simple for implementation of this recognition system. Today’s powerful computing processors and easy data collection software makes it more technically feasible.

### Economic Feasibility

The project is economically feasible to begin with as no expensive hardware and software components are required. Similarly all the tools and techniques to be used are open source and are easily available free of cost. Data collection is done by us which is economically and easily feasible.

### Schedule Feasibility

To develop the project, a proper timeline has been projected to complete relevant portions in a scheduled time period. Most of the necessary resources are searched on the web and are available to begin research on time. Also all the related software is easily available, which makes it more feasible.

## Programming Language

### Python and Text Editor

The entire work of this project is coded in Python programming language. Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in other high level languages. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library. The entire project was coded in ‘Sublime Text’ text editor. Sublime Text is a shareware cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free software licenses. Thus written Python codes were run in a python environment under the Windows operating system using the command line. Python allows the simple representation of large data matrix and vectors and also supports external libraries as required. Due to this feature of python, we were able to implement the algorithm successfully with minimum lines of codes. Thus, the major reasons behind the use of python programming language for this project are discussed below in.

**Software quality:** For many, Python's focus on readability, coherence, and software quality in general sets it apart from other tools in the scripting world. Python code is designed to be readable, and hence reusable and maintainable much more so than traditional scripting languages. The uniformity of Python code makes it easy to understand, even if you did not write it. In addition, Python has deep support for more advanced software reuse mechanisms, such as object-oriented programming (OOP) and function programming.

**Developer productivity:** Python boosts developer productivity many times beyond compiled or statically typed languages such as C, C++, and Java. Python code is typically one-third to one-fifth the size of equivalent C++ or Java code. That means there is less to type, less to debug, and less to maintain after the fact. Python programs also run immediately, without the lengthy compile and link steps required by some other tools, further boosting programmer speed.

**Program Portability:** Most Python programs run unchanged on all major computer platforms. Porting Python code between Linux and Windows, for example, is usually just a matter of copying a script's code between machines. Moreover, Python offers multiple options for coding portable graphical user interfaces, database access programs, web-based systems, and more. Even operating system interfaces, including program launches and directory processing are as portable is Python as they can possibly be.

**Support Libraries:** Python comes with a large collection of pre-built and portable functionality, known as the standard library. This library supports an array of application level programming tasks, from text pattern matching to network scripting. In addition, Python can be extended with both homegrown libraries and a vast collection of third-party application support software. Python 's third-party domain offers tools for website construction, numeric programming, serial port access, game development, and much. The list of libraries used in this project are:

1. Speech Recognition: It is a python library for recognizing speech with the support of several engines and APIs, online and offline
2. PyAudio: Bindings for PortAudio v19, the cross-platform audio input/output stream library
3. Torch: Tensors and Dynamic neural networks in Python with strong GPU acceleration. provide neuralnet.
4. pyttsx3: Text to Speech (TTS) library for Python 2 and 3. Works without internet connection or delay. Supports multiple TTS engines, including Sapi5, nsss, and espeak.
5. Numpy: NumPy is the fundamental package for array computing with Python.

**Component Integration:** Python scripts can easily communicate with other parts of an application, using a variety of integration mechanisms. Such integrations allow Python to be used as a product customization and extension tool. Today, Python code can invoke C and C++ libraries, can be called from C and C++ programs, can integrate with Java and .NET components, can communicate over frameworks such as COM and Silverlight, can interface with devices over serial ports, and can interact over networks with interfaces like SOAP, XML-RPC, and CORBA. It is not a standalone tool.

### HTML

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

### CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

### Javascript

JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

**4.4.5 Keras**

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It enables fast experimentation.

## GUI Development

For the development of a simple user interface we used HTML/CSS, we used the terminal for voice command input and rendering of the final result.

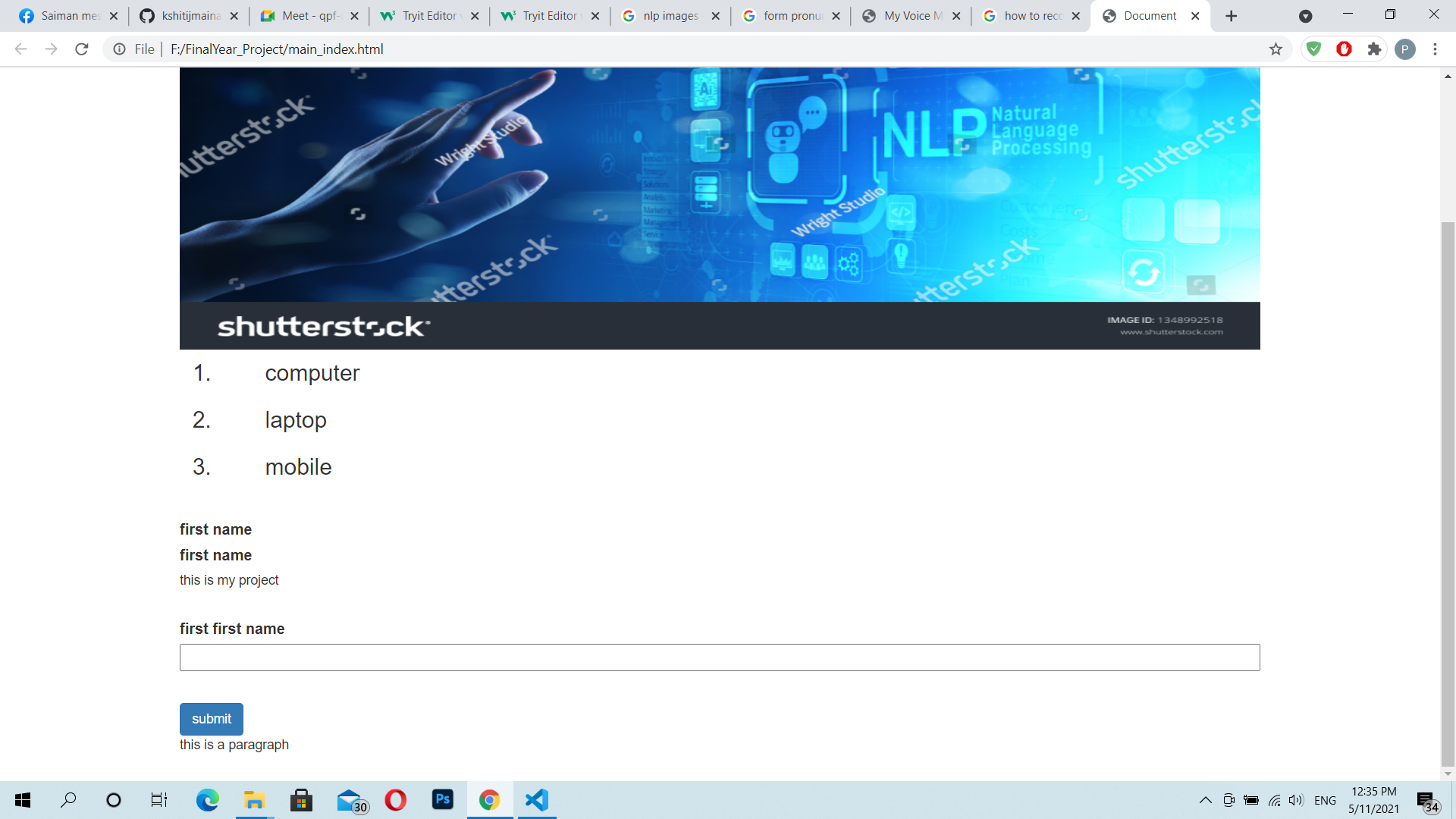


Figure 4.4 Render page

## Project Scheduling

Scheduling in project management is the listing of activities, deliverables, and milestones within a project. A schedule also usually includes a planned start and finish date, duration, and resources assigned to each activity. Effective project scheduling is a critical component of successful time management.

For our project it takes about 10 month for the completion of the work which is shown in the gantt chart below. We had started research work in august of previous year regarding our project and finally we are able to complete it in May of 2021

# 5. CODE

# Import the libraries

import json

import numpy

import torch

import torch.nn as nn

from torch.utils.data import Dataset, DataLoader

from tokenizationAndStemming import tokenize, stem, BagOfWords

from model import NeuralNet

# It is a data set for facilating the data access and use

class ChatDataset(Dataset):

def \_\_init\_\_(self, xTrain, yTrain):

self.nSample = len(xTrain)

self.xData = xTrain

self.yData = yTrain

def \_\_getitem\_\_(self, index):

return self.xData[index], self.yData[index]

def \_\_len\_\_(self):

return self.nSample

# 2> create collection according to input

def createCollection(fName, which, name):

allWords = []

tags = []

collection = []

key = 'intents' if which == 1 else 'attributes'

for value in fName[key]:

tag = value[name]

tags.append(tag)

for pattern in value['patterns']:

words = tokenize(pattern)

allWords.extend(words)

collection.append((words, tag))

ignoreWords = ['?', ';', ',', '.', '!', '/']

allWords = [stem(w) for w in allWords if w not in ignoreWords]

allWords = sorted(set(allWords))

tags = sorted(set(tags))

return collection, allWords, tags

# main operating area of training

def main():

# 1>#open the appropriate files

# Open the intents.json file in read mode

with open('intents.json', 'r') as f:

intents = json.load(f)

# open the attribute file as attribute

with open('attributes.json', 'r') as p:

attributes = json.load(p)

# 2># create collection from both files

collection1, allWords1, tags1 = createCollection(intents, 1, "tags")

collection2, allWords2, tags2 = createCollection(attributes, 2, "attr")

# 3># concatenate these to create the main collection and words

collection1.extend(collection2)

allWords1.extend(allWords2)

tags1.extend(tags2)

# 4># define xTrain and yTrain

xTrain = []

yTrain = []

# 5># Find out the bag of words of pattern and tags

for (patternSentence, tag) in collection1:

bag = BagOfWords(patternSentence, allWords1)

xTrain.append(bag)

label = tags1.index(tag)

yTrain.append(label)

# 6># Develop training data as array

xTrain = numpy.array(xTrain)

yTrain = numpy.array(yTrain)

yTrain = torch.tensor(yTrain, dtype=torch.long)

# 7>## Hyperparameters for further training

batchSize = 8

# input size is same as bagof words and xTrain contain bagofwords at each row

inputSize = len(xTrain[0])

hiddenSize = 8

outputSize = len(tags1)

learningRate = 0.001

numEpoch = 1000

# 8># dataset and trainLoader for data loading and preparation

dataSet = ChatDataset(xTrain, yTrain)

trainLoader = DataLoader(

dataset=dataSet, batch\_size=batchSize, shuffle=True)

# 9># build the model

model = NeuralNet(inputSize, hiddenSize, outputSize)

# 10# loss and optimizer

criterion = nn.CrossEntropyLoss()

optimizer = torch.optim.Adam(model.parameters(), lr=learningRate)

# 11># start the trainning loop

for epoch in range(numEpoch):

for (words, label) in trainLoader:

# forward pass

outputs = model(words)

loss = criterion(outputs, label)

# backward pass

optimizer.zero\_grad()

loss.backward()

optimizer.step()

if(epoch+1) % 100 == 0:

print(f'epoch {epoch+1}/{numEpoch},loss = {loss.item():.4f} ')

# print the final loss

print(f'final ::loss = {loss.item():.4f} ')

# save the model attributes in a file .pth extension

data = {

"modelState": model.state\_dict(),

"input\_size": inputSize,

"output\_size": outputSize,

"hidden\_size": hiddenSize,

"allWords\_size": allWords1,

"tags": tags1,

}

FILE = "data.pth"

torch.save(data, FILE)

print(f'training complete. file saved to {FILE}')

if \_\_name\_\_ == "\_\_main\_\_":

main()

# 6. RESULT

The main target of this project was to create a system that could recognize a user voice command that consists of HTML tags and attributes. Among all the processes the most important task was to obtain a model with the highest level of accuracy by training the dataset containing the relevant information.

The training of the system was performed on a computer system without running any external applications during the entire period of training with the specifications as mentioned below in the specification table.

Table 5.1 System Specification Table

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Particulars** | **System Specifications** |
| 1. | Computer Model | Dell Inspiron 5557 |
| 2. | Computer Processor | 2.3GHZ Intel Core  i5-5000U CPU |
| 3. | Installed RAM | 8.00GB(7.7GB usable) |
| 4. | Computer System Type | 64bit Windows Operating System |

The snapshot of different processes in model training with the datasets are mentioned below:

Figure 5.1 Separation of classes train sample and test sample

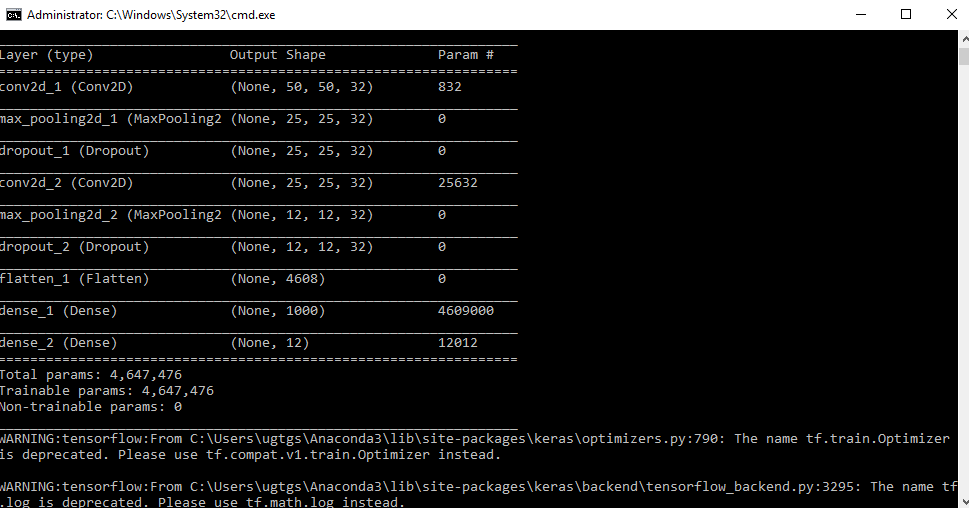


Figure 5.2 Loading train and test data in CNN model

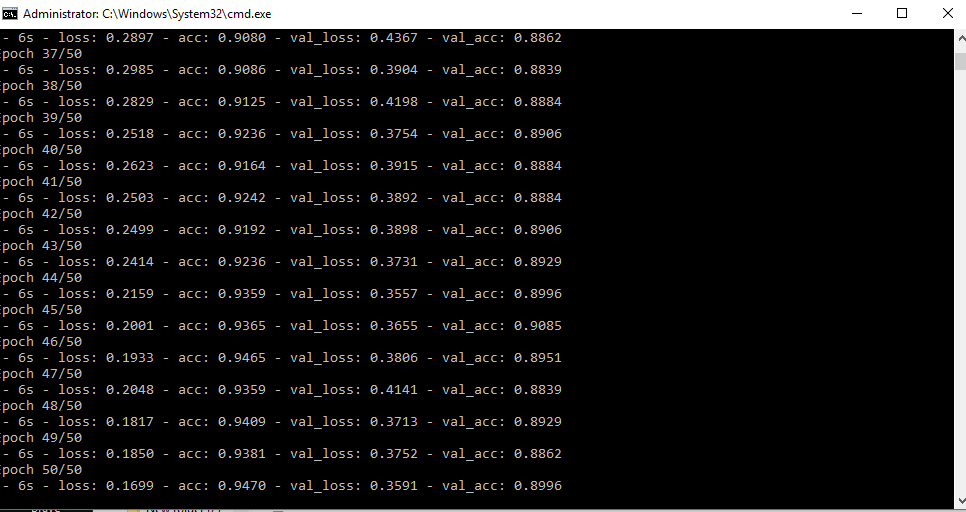


Fig.6.4. Snapshot of Accuracy

The final Analytics of the obtained visuals is shown below in the plot between model loss and total number of epochs and also the plot between model accuracy and total number of epochs, for which the Network was trained.

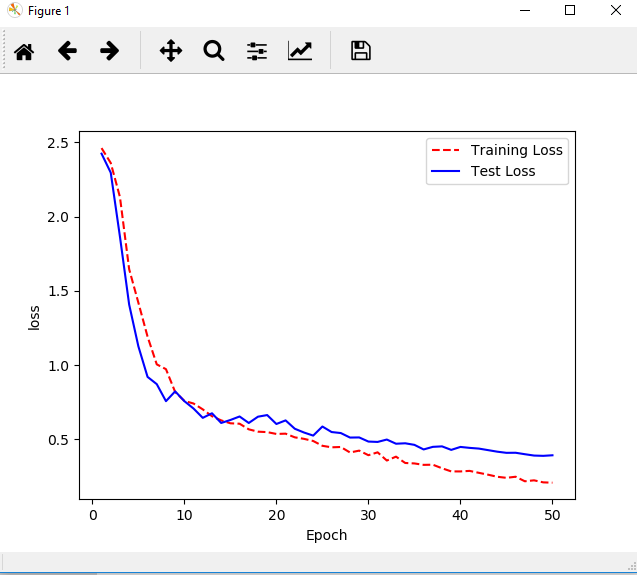


Figure 5.3 Plot between model loss function and no. of epochs

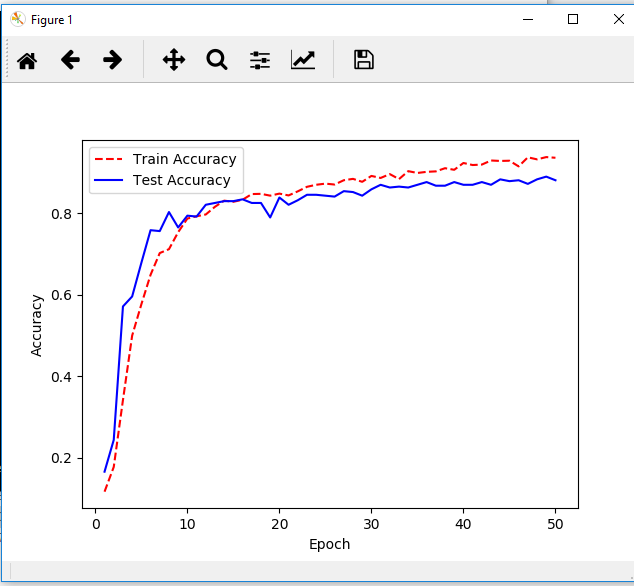


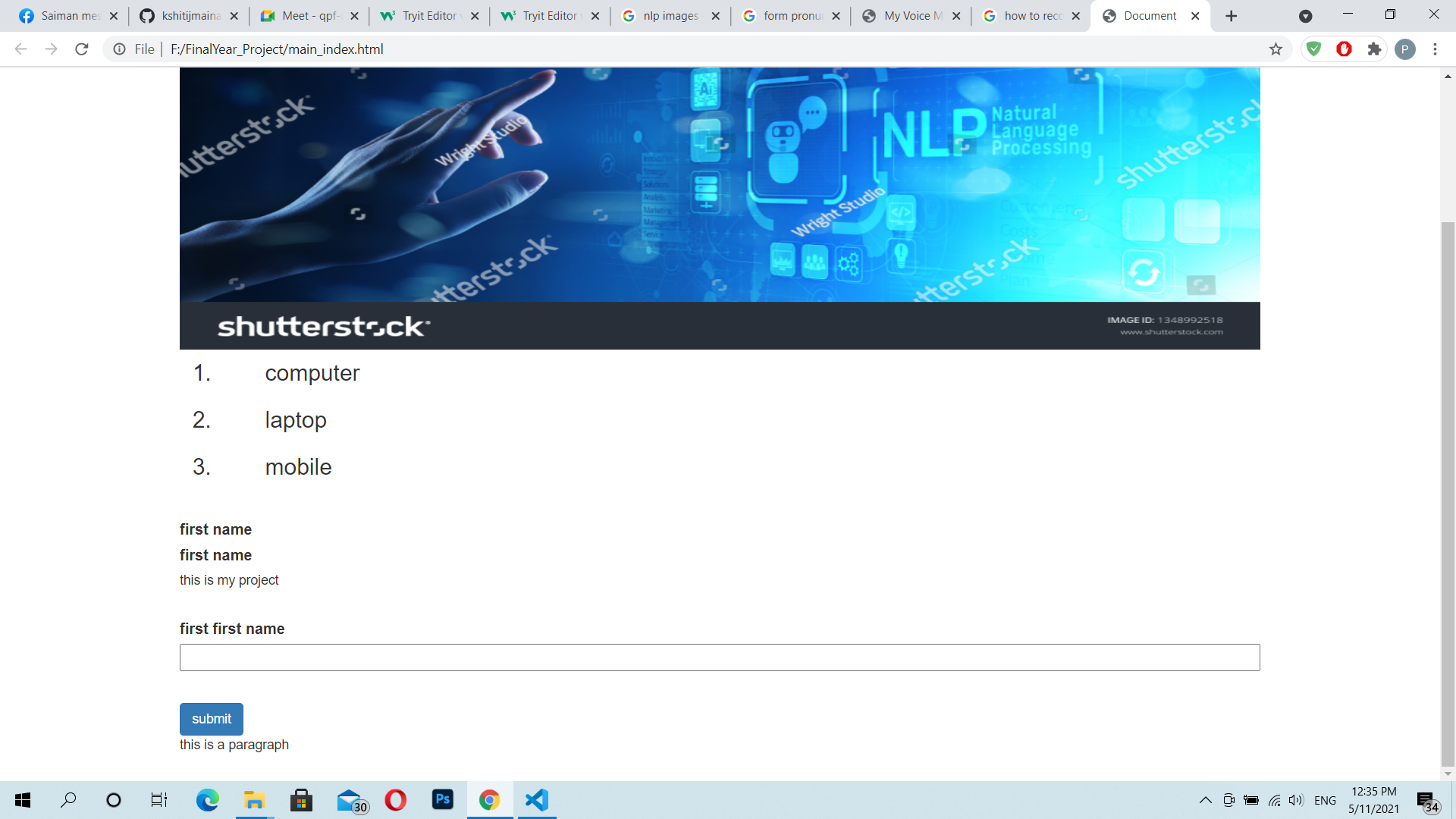
Figure 5.4 Plot between Model accuracy and no. of epochs

Thus, after training the system for 50 epochs, a stable hypothesis/model was obtained with accuracy of 94.7% and validation accuracy of 89.96%.

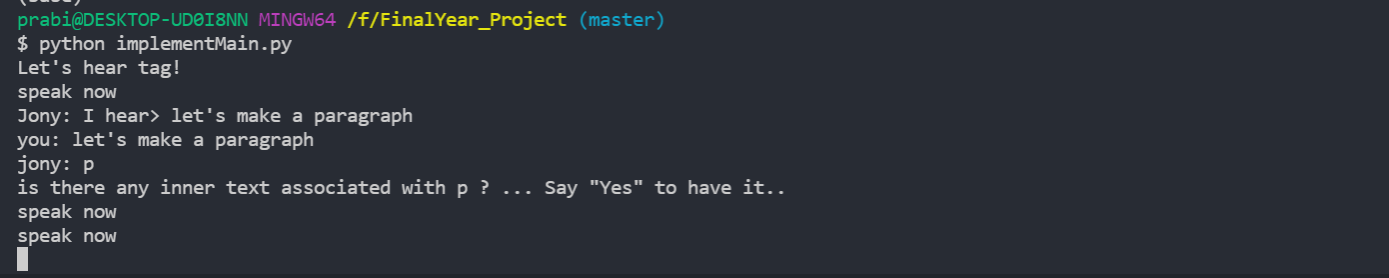
# 7. OUTPUT

The output of the project is obtained as below:

* UI render of code:



* terminal output of code:



# 8. CONCLUSION

This project work is a successful output of the course called ‘Major Project’ considered as the partial fulfillment of BE Computer Engineering at IOE. The main objective of the project was to input the user voice command, recognize the HTML tags and build the desired website. The system was implemented in python language and its performance was tested with clear voice input.

This system is one kind of AI program with the implementation of Natural Language Processing. With the model accuracy of 95% wew achieved success in Natural Language Processing.

This project work has been a great achievement for us even though some limitations still exist. With full obstacles we ought to complete the project in time.Finally, the overall result was satisfactory and we successfully carried out this project as a part of our course work and also developed the hands-on experience of working in a project.

# 

# 9. LIMITATION AND FURTHER WORK

Natural Language Processing has been quite an interesting field today. Many technical teams around the world are still working together to get satisfactory results. Several researches have been done and some are still ongoing in this field and due to advancement in technologies and efficient new models it has made possible to make further enhancement in this field to get more accurate results. As other projects, our project has also got limitations and the enhancements that can be made in future.

## Limitations

The voice recognition system has many defects such as inaccurate results if the voice is not of proper texture for example, if the voice contains disturbances then the results are always inaccurate. So the existing algorithm could be modified to produce better results. The main challenge was to understand and model the elements within variable context.

Words are unique but can have different meanings depending on the context resulting in ambiguity on the lexical, syntactic, and semantic levels.

## Further Works

The main objective of this project was to obtain a model with highest accuracy that can understand the voice command provided by the user and output the website as wished . But yet the model we have obtained is not perfectly tuned..

So, the basic tasks yet to be done are as:

1) Not all the Tags and Attributes are recognized.

2) Implementation in cross platform.

3) Increasing the number of training dataset

4) Performing the training session by altering different parameters and switching between the network architecture, different activation functions, loss functions and optimizers.

# 10. REFERENCE

[1] R.Elankavi, Dr.R. Udayakumar“Voice Command Execution with Speech Recognition and Synthesize” February 2017.

[2] R Ram Kumar, A Vimal Kumar, R Deepa, S Shalini, “Voice Command Execution with Speech Recognition and Synthesize ” IRJET, Vol. 5, Issue. March 2018.

[3] Association for Computer Linguistics “Bridging the Gap Between Training and Inference For Neural Machine Translation.

[4] Natural Language Processing, Wikipedia.

[5] Voice User Interface, Wikipedia.

[5] Speech Synthesis, Wikipedia.

[6] Google “A Neural Conversational Model”

[7]Bahdanau, D., Cho, K., and Bengio, Y. “Neural machine translation by jointly learning to align and translate” 2014.

[8] Jean, S., Cho, K., Memisevic, R., and Bengio, Y. “On using very large target vocabulary for neural machine translation”, 2014.

[11] Mikolov, T., Karafiat, M., Burget, L., Cernock`y, J., and Khudanpur, S. “Recurrent neural network based language model”, 2010.