A

PROJECT REPORT ON

**QUESTION ANSWERING SYSTEM**

**By**

**ANSHUMAN AICH (CE-006) (18CEUOD012)**

**ARYA KAGATHARA (CE-053) (18CEUOD022)**

**MAHESH KANTARIYA (CE-055) (17CEUSF127)**

**B.Tech CE Semester-VI**

**Subject: System Design Practice**

**Guided by:**

Prof.Hariom A. Pandya

Assistant Professor

Dept. of Comp. Engg.

****

**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**

****

**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical / term work carried out in the subject of

**System Design Practice** and recorded in this journal is the

bonafide work of

**ANSHUMAN AICH (CE-006) (18CEUOD012)**

**ARYA KAGATHARA (CE-053) (18CEUOD022)**

**MAHESH KANTARIYA (CE-055) (17CEUSF127)**

ofB.Tech semester **VI** in the branch of **Computer Engineering**

during the academic year **2019-2020**.

Prof.Hariom A. Pandya Dr. C. K. Bhensdadia,

Assistant Professor, Head,

Dept. of Computer Engg., Dept. of Computer Engg.,

Faculty of Technology Faculty of Technology

Dharmsinh Desai University, Nadiad Dharmsinh Desai University, Nadiad

**CONTENTS**

|  |  |
| --- | --- |
| 1. | Abstract |
| 2. | Introduction |
| 3. | System Requirement Specifications |
| 4. | Design |
| 5. | Implementation Details |
| 6. | Testing |
| 7. | Screen-Shots |
| 8. | Conclusion |
| 9. | Limitation & Future Extension |
| 10. | Bibliography |

1. **ABSTRACT**

Gotcha is an interactive question answering technology designed to allow intelligence analysts and other users of information systems to pose questions in natural language and obtain relevant answers, or the assistance they require in order to perform their tasks. Our objective in Gotcha is to allow the user to submit exploratory, analytical, non-factual questions, such as “What is Data Structure?”, “Who is President of India?”

1. **INTRODUCTION**

Vast amounts of data are available on the Wikipedia. There are few techniques available to fetch the accurate information from these data. One of the effective way is the Question Answering [QA] Systems. QA system is a computer science discipline within the field of Information Retrieval and Natural Language Processing, which revolves around building systems that automatically answers questions posed by humans in a natural language.

This project centres on adding QA System to Gotcha, an open source search engine. The Summarizer in Gotcha fetches a short summary from the Wikipedia documents it crawls. Question-Answering System will extract the information stored in the summary and store it in Gotcha so that it can be used to answer questions. Extraction of information from the summary will be done by implementing various functionalities of natural language processing.

**Technology:**

* HTML
* CSS
* JS
* JQuery
* Python
* Ajax
* Django Framework

**Platforms:**

* x64-based Processor Windows 10
* i3-6006U CPU

**Tools:**

* Visual Studio Code

**3. SOFTWARE REQUIREMENT SPECIFICATIONS**

Functional Requirements

User :

R.1.1 : Question entering

Description : User can enter a question in the search field for the answer.

Input : Enter a question.

Output : answer and reference pages

R.1.1.1 : Ask question by voice

Input : Voice

Output : answer and reference pages

R.1.2 : Play answer

Description : User can here answer on answer page

Input : play command

Output : voice output

R.1.3 : Contact

Description : User can submit their queries to contact page.

Input : contact details

Output : success message

Non-Functional Requirements

R.1 : Security

R.2 : Admin Rights

Description : Admin shall be able to view and modify all information in system.

R.3 : Availability

Description : The system shall be available all the time.

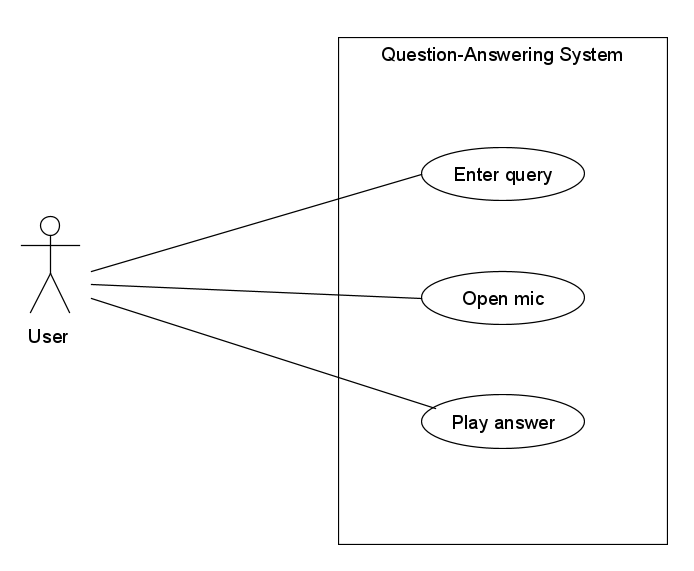
R.4 : Maintainability

Back Up : The system shall provide the capability to back up the data.

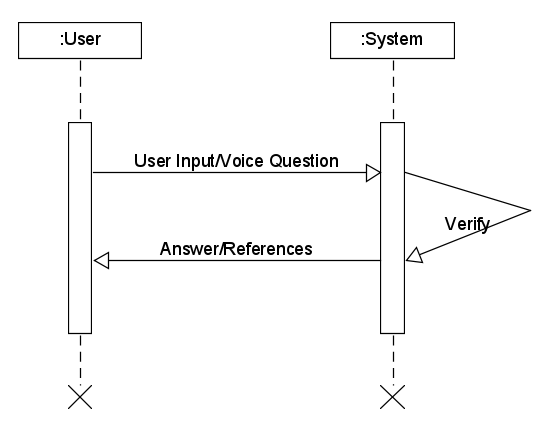
Error : The System shall keep all log of errors.

**4.DESIGN**

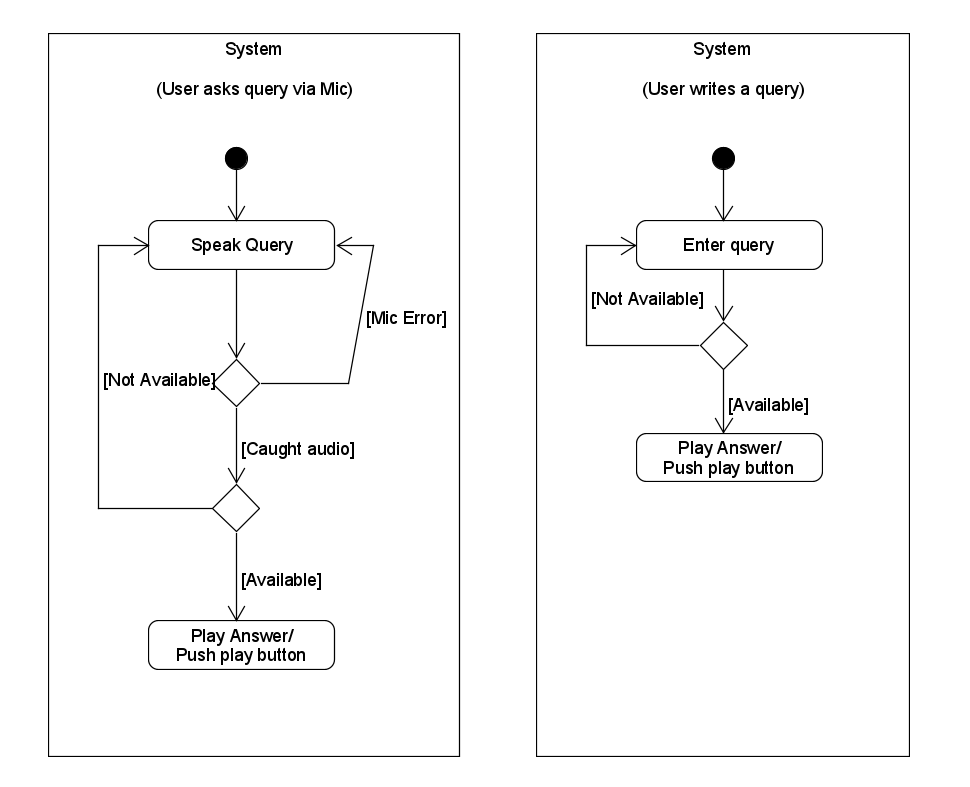
**Use Case Diagram**



**Sequence Diagram**



**Activity Diagram**



**­­­­­­­­**

**5. IMPLEMENTATION DETAILS**

Home Page :

As the home page is unveiled, it starts off with a loading animation. Here User can input a query or can ask via microphone.

Details Page (Answer Page):

As user asks a question he/she will be redirected to answer page. Also there play button, so user can hear the answer.

About Page:

About page.

Documentation Page:

Description that how this system works.

Contact Page :

Users can ask their queries and give feedback.

views.py :

Here all the business logic is written.

Functions:

* HomePageView (class)

Renders home page

* details

Renders detail page (Answer page).

Question related content is fetched from Wikipedia using wikipedia.page().content method and content is sent to prepare\_content\_QA function.

Answer is saved in .mp3 file also.

* documentation

Renders documentation page

* contact

Renders contact page

* about

Renders about page

* openmic (Ajax)

This function is used to open the microphone.

Voice input is taken using Google’s speech recogniser

* getwords (Ajax)

This function is used for suggestion to avoid spelling mistakes whenever user inputs a query.

It collects all the words in one list and returns.

* playsnd (Ajax)

This function is used to convert the answer into voice

Use of playsound function to play .mp3 file.

* askquestion (Ajax)

User can ask queries via contact page.

Use of smtplib to send user’s queries to admin.

prepare\_content.py

It takes a list of content fetched from wikipedia and summarizes these content using cosine similarity technique.

It takes sentences which has cosine similarity greater than 0.10 with asked question.

Modules :

* wikipedia : A Python library that makes it easy to access and parse data from Wikipedia.
* SpeechRecognition : To recognize speech.
* nltk : The Natural Language Toolkit (NLTK) is a Python package for natural language processing.
* gtts : A Python library and CLI tool to interface with Google Translate's text-to-speech API.
* playsound : It requires one argument - the path to the file with the sound you’d like to play. This may be a local file, or a URL.
* smtplib : This module defines an SMTP client session object that can be used to send mail to any Internet machine.
* scikit-learn: This is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.
* numpy : It is a powerful N-dimensional array object,useful linear algebra,Fourier transform and random number capabilities. It can also be used as an efficient multi-dimensional container of generic data.
* spaCy : It is a library for advanced Natural Language Processing in Python and Cython.
* msgpack : It is an efficient binary serialization format. It lets you exchange data among multiple languages like JSON.
* tqdm : It is used to make smart progess meters.
* regex : It is for regular expressions.
* torch : PyTorch is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing.

For Training the Model

* + Dataset : SQuAD dataset => 1) dev-v1.1.json

2) train-v1.1.json

* + Word2Vector file => glove.840B.300d.txt
  + Data pre-processing

prepro.py : For data cleaning and building vocabulary.

* + Model training:

train.py : rnnType – LSTM

Layers -- 4

optimizer – Adamax

learning rate – 0.1

hidden layers -- 128

* + Steps to train model :
    - * Preprocessing

Type command python prepro.py

* + - * Training

Type command python train.py –e epochs -bs batchsize

**6.TESTING**

1) What was the monetary value of the Nobel Peace Prize in 1989?

Answer : There was no data available on Wikipedia for the question.

2) Who was the first prime minister of india ?

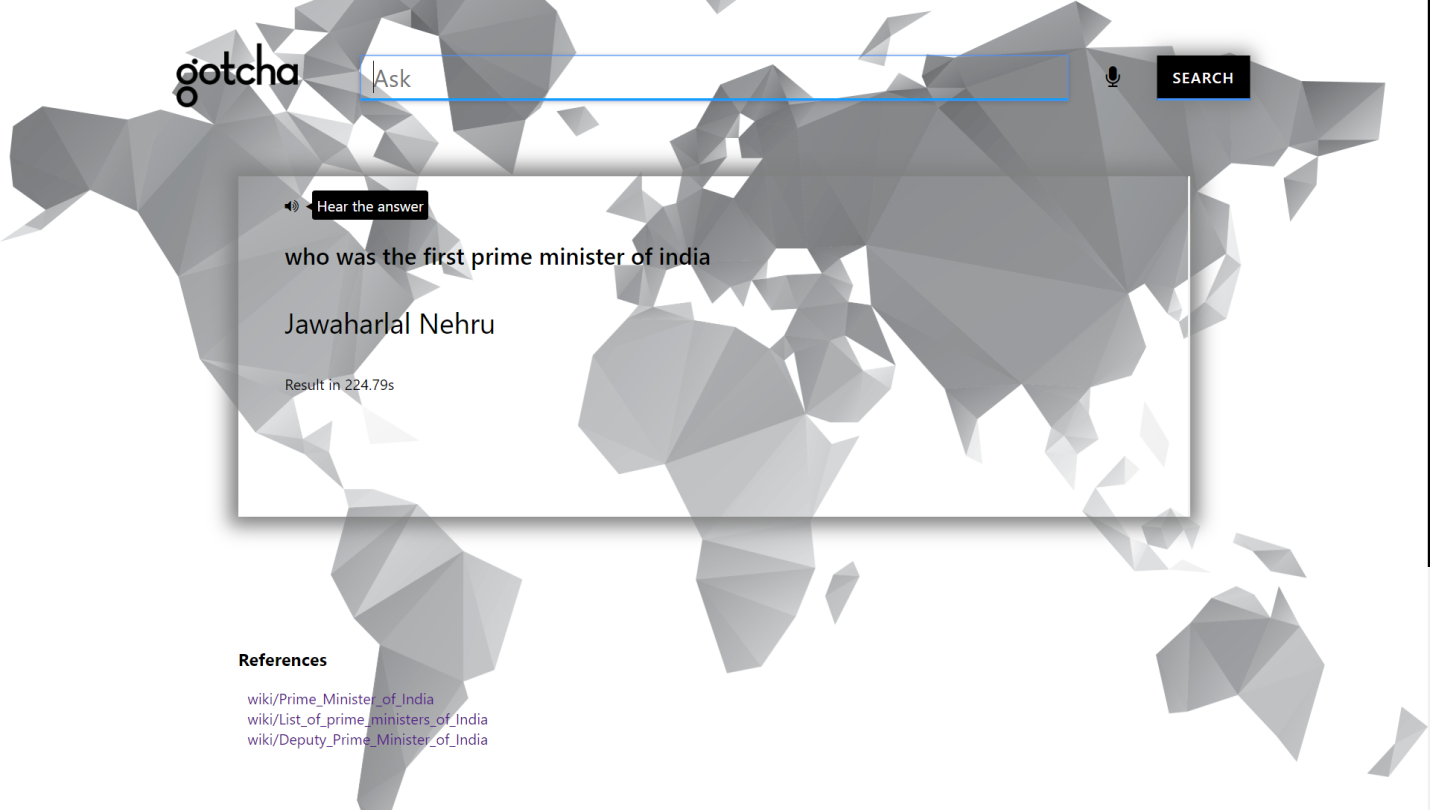
Answer : Jawaharlal Nehru

**7.SCREEN-SHOTS**

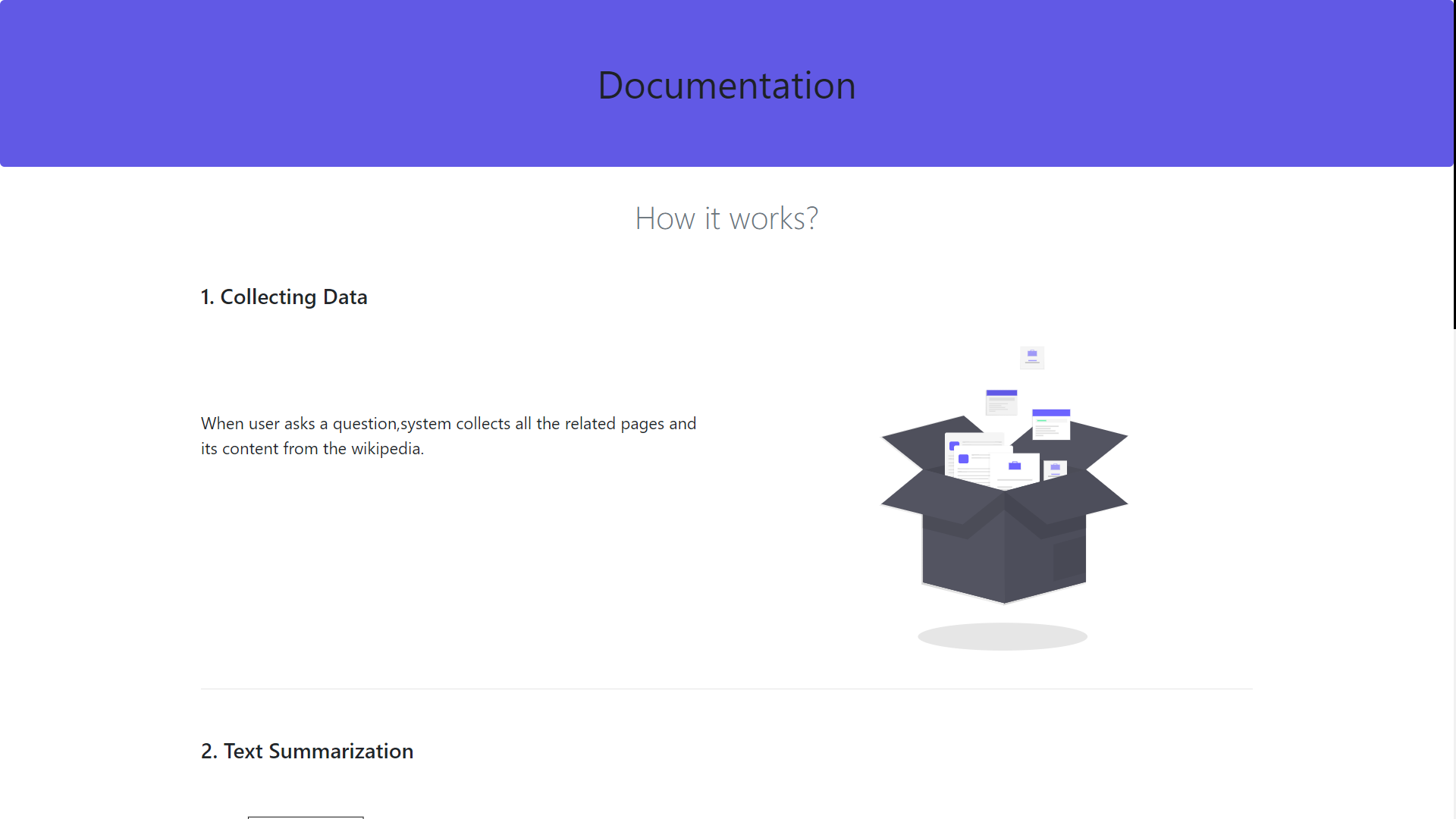
**Home Page**

****

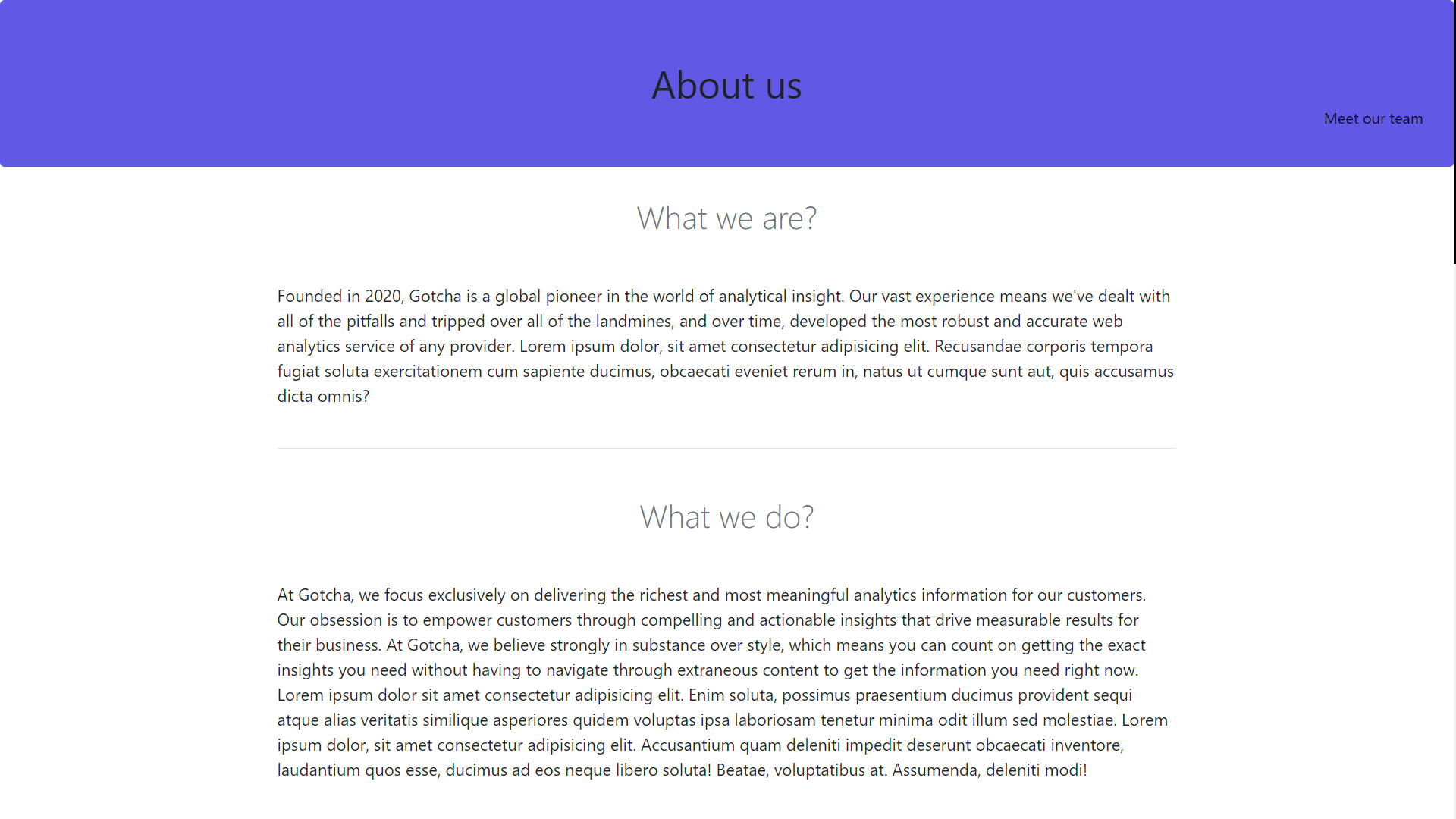
**Answer Page**



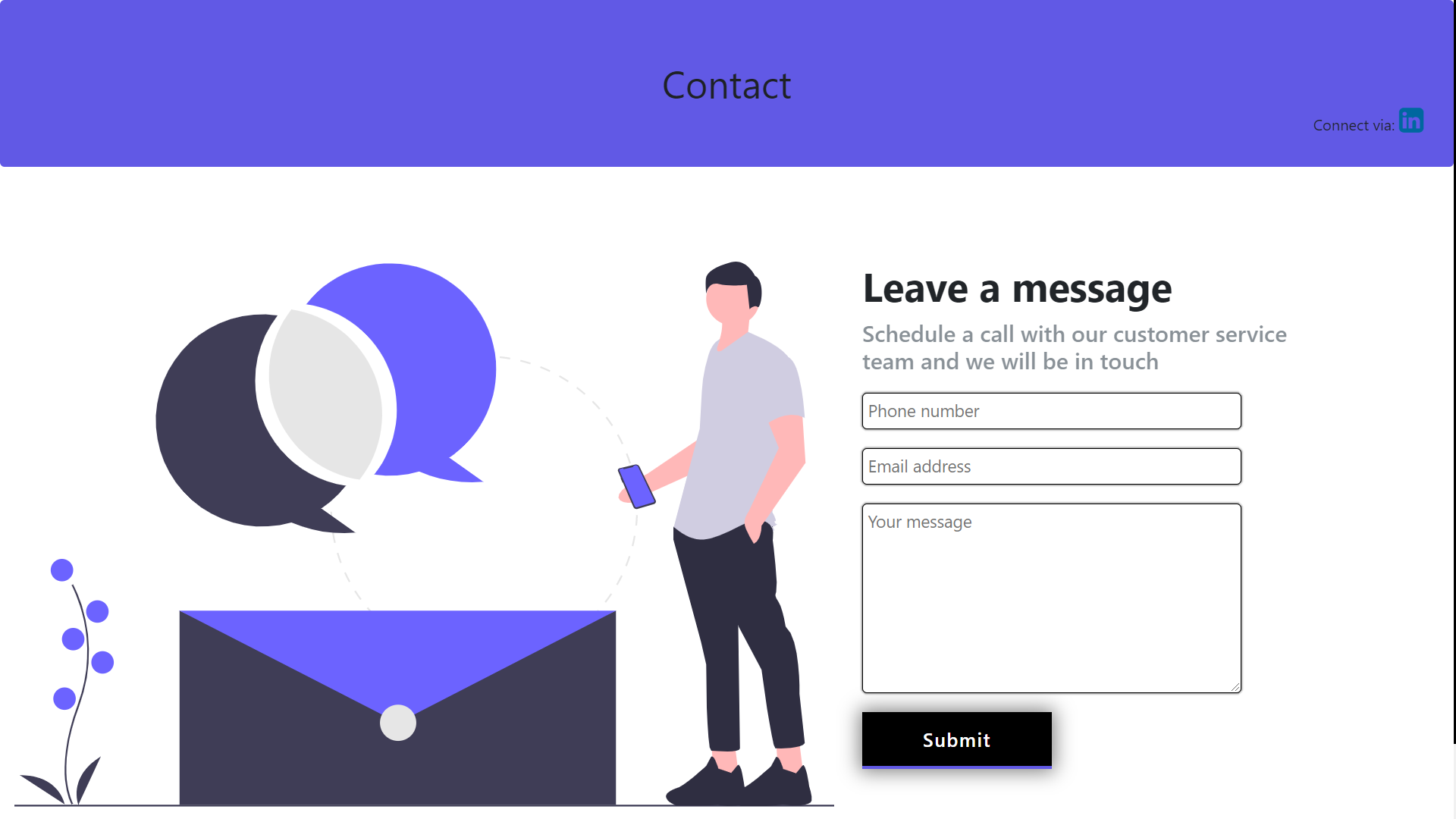
**Documentation Page**

****

**About Page**

****

**Contact Page**

****

**8.CONCLUSION**

We would like to conclude that , it is immense learning experience while preparing this project.

Our System dynamically searches for the query on Wikipedia and downloads the relevant pages.

We learned many machine learning and nlp concepts while working on this project.

**9.LIMITATION AND FUTURE EXTENSION**

**LIMITATION**

* Model is not trained completely.
* Users have to ask question in full sentence, for example “Who is prime minister of India” instead of “Prime minister of India” otherwise it will give definition of “prime minister”.

**FUTURE EXTENSION**

* Admin site.
* User will get related questions based on asked question.
* Model improvement.
* Users will be able to suggest their answers.

**10.BIBLIOGRAPHY**

<https://en.wikipedia.org/>

[https:// geeksforgeeks.org](https://www.geeksforgeeks.org)/

<https://www.python.org/>

<https://scikit-learn.org/>

<https://stackoverflow.com/>

<https://kaggle.com/>

<https://djangoproject.com/>

https://github.com/