Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

Mahavadi Sesha Sai Sreenadh

2020MT13045

Mobile No: 9502757826

Email Id: 2020MT13045@wilp.bits-pilani.ac.in

Assignment-01: Custom AI Chatbot Using LLM

Approach, Observations, Software user guide and References are documented here.

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

Technical Specifications

API key will be provided to you by the instructor over email. Use it wisely for this project only and never distribute. Please note that same will be deactivated once submission date is over.

- You application should run on port **http://localhost:3000**
- Web stack should be restricted to

  backend: nodejs,

  frontend: react,

  database to store embeddings: excel

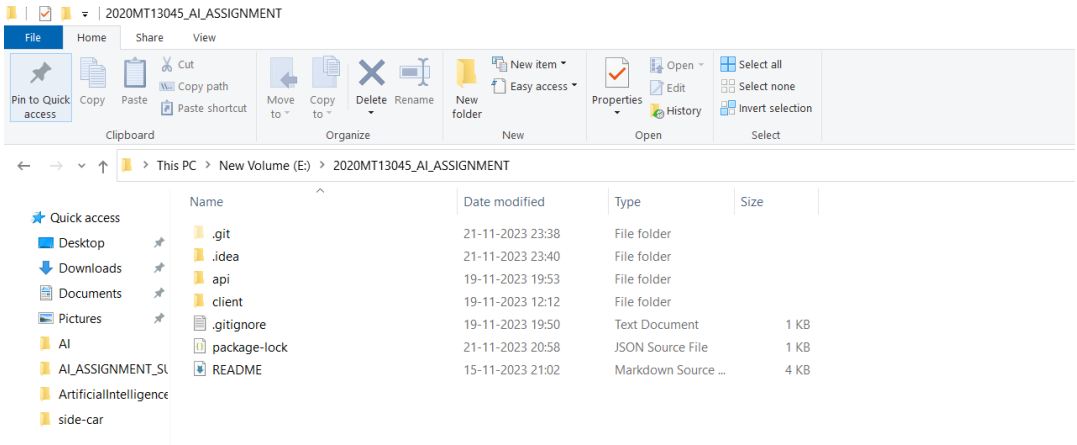
- For embedding use **text-embedding-ada-002** and for chat completion **text-davinci-003** model

2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

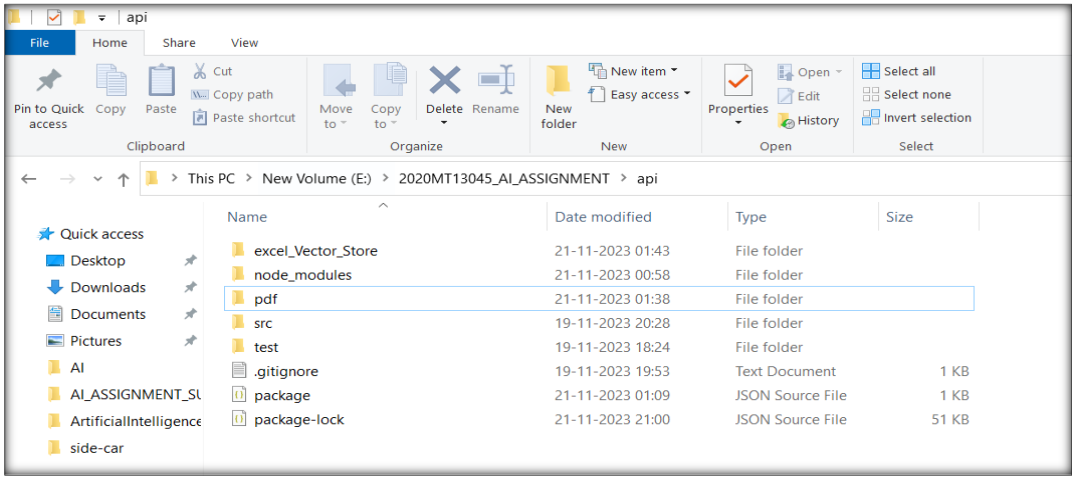
Folder structure



>> **"api"** folder is server side / backend node js component.

>> **"client"** folder is Front end React frame work based application.

Backend



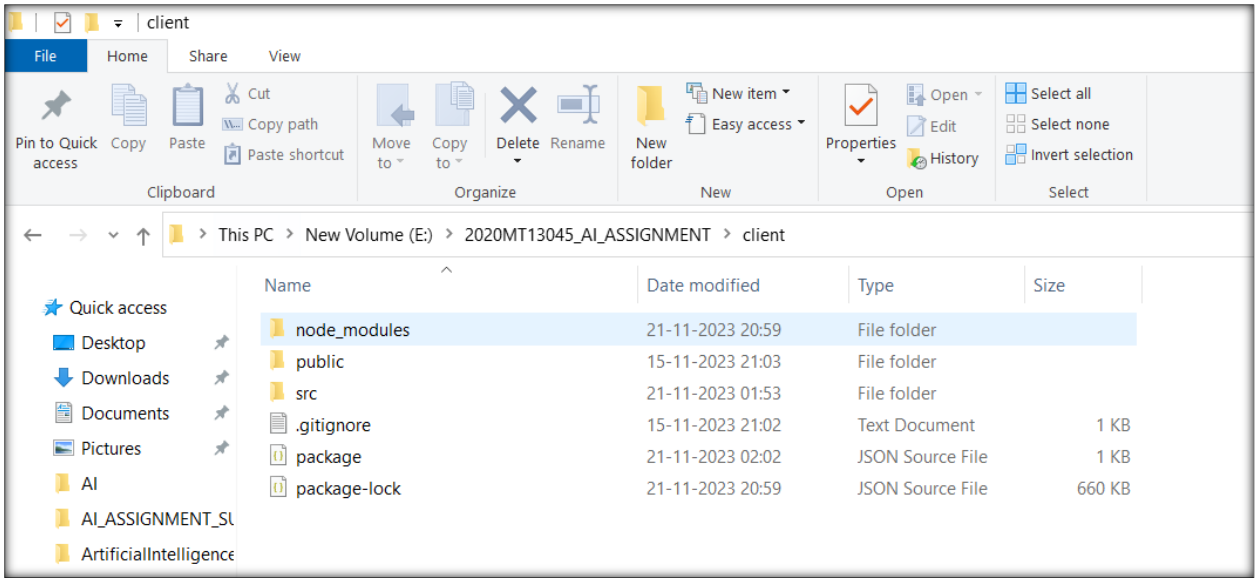
1) node js version installed to develop this project is V20.9.0
2) **npm install** command on this folder creates node modules required for running this component
3) **npm start** will run the backend project on http://localhost:9000/
4) **excel_vector_store** is the folder where vector embeddings created for PDF's are stored
5) **pdf** is the folder where uploaded PDF from front end are stored
6) **test** folder is a dependency of pdf reader dependency, it expects a sample pdf file to exist there for function with name **"05-versions-space.pdf"**

2020MT13045
MAHAVADI SESHA SAI SREENADH

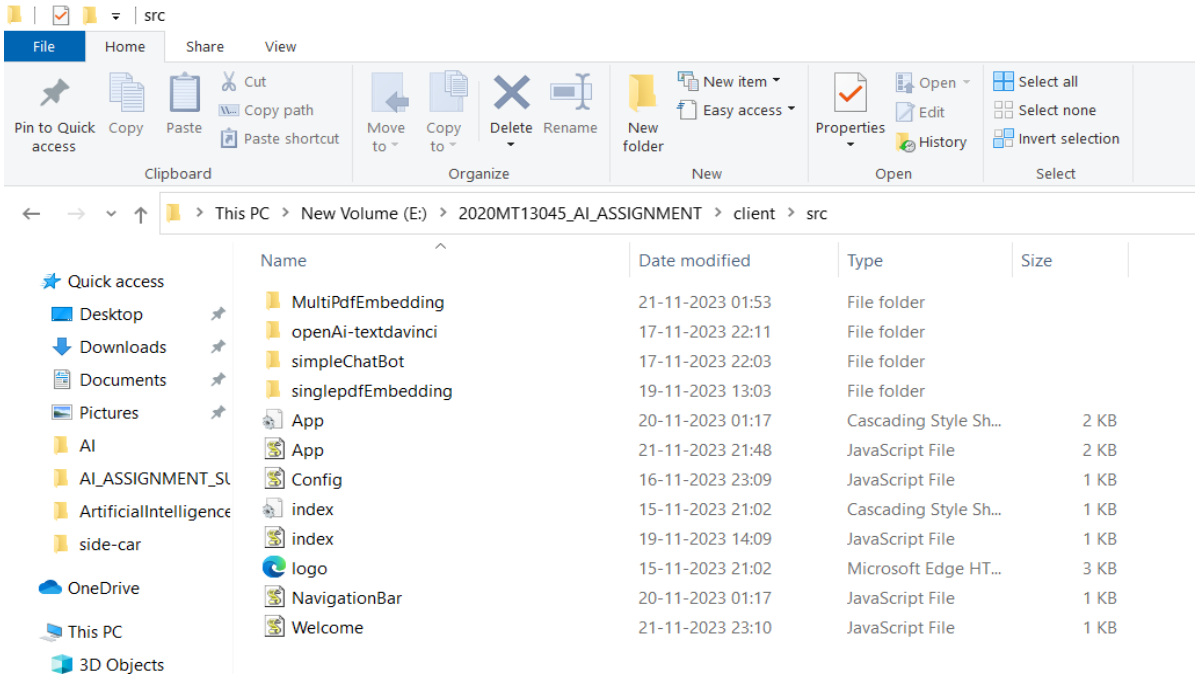
Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

Frontend



1) node js version installed to develop this project is V20.9.0
2) **npm install** command on this folder creates node modules required for running this component
3) **npm start** will run the backend project on **http://localhost:3000/**
4) **Separate packages** are created for the react components expected as per the requirement.



2020MT13045
MAHAVADI SESHA SAI SREENADH

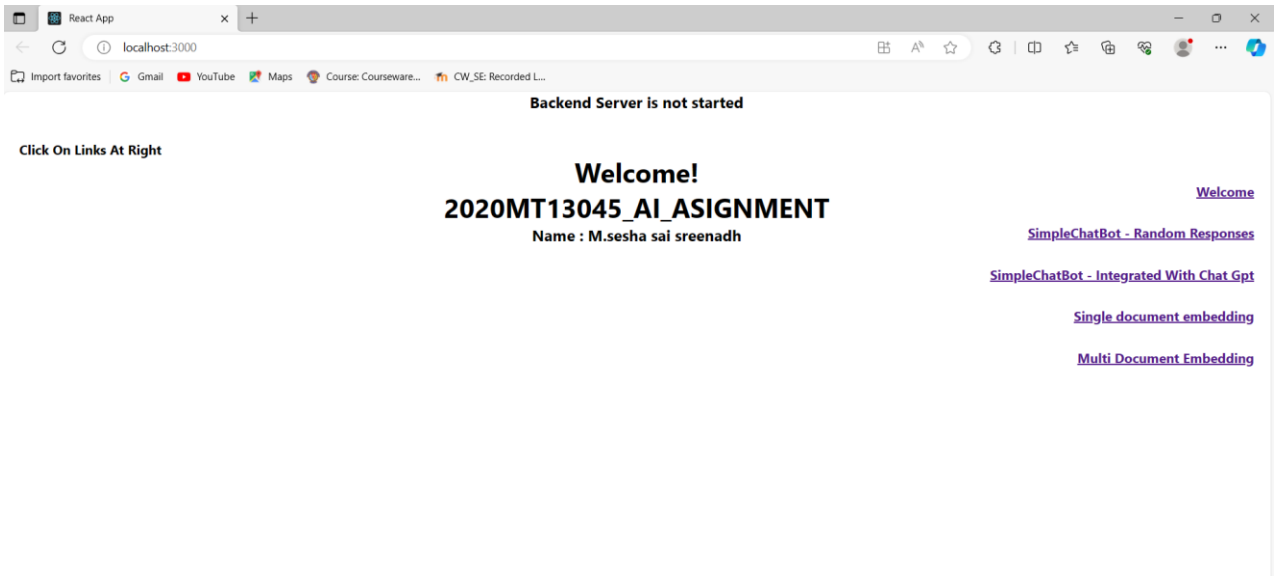
2020MT13045
MAHAVADI SESHA SAI SREENADH

Front end Server Start up

1) First **npm install** to be executed
2) Then **npm start** like below will start front end server and application will run on "http//localhost:3000"
3) Please notice its client folder here .

```
E:\2020MT13045_AI_ASSIGNMENT\client>npm start

> client@0.1.0 start
> react-scripts start

(node:15712) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:15712) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view client in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.29.241:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Front end home page



- Message on the top of screen states backend server not started so let's start backend server now

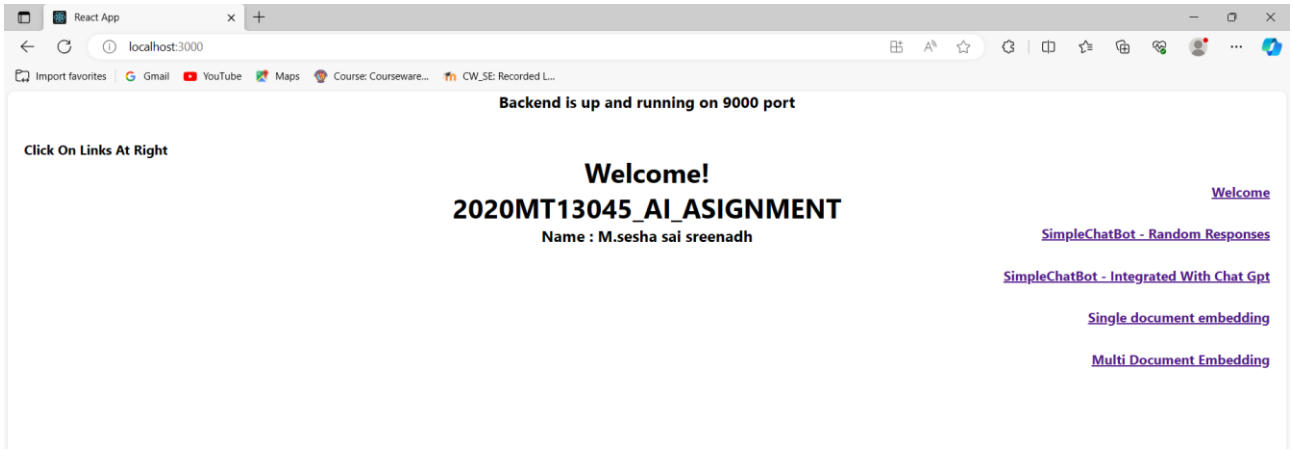
2020MT13045
MAHAVADI SESHA SAI SREENADH

Backend Server start up

1) **npm install** to be done for the first time .
2) **npm start** will start the backend server
3) **Please notice its api folder here**
4) **Refresh front end screen after this you can notice backend server related message as below**

npm start

```
E:\2020MT13045_AI_ASSIGNMENT\api>npm start

> api@0.0.0 start
> node ./src/app.js

Warning: TT: undefined function: 32
Warning: TT: undefined function: 32
```



Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

In this assignment we will **develop a custom conversational AI agent (Chatbot) using openAI APIs.** You know such systems can provide replies to user's queries. By custom we mean that the answers will be based on domain specific knowledge provided to the AI agent using a PDF. In this assignment, we will only focus on text-based conversations. This assignment has four parts.

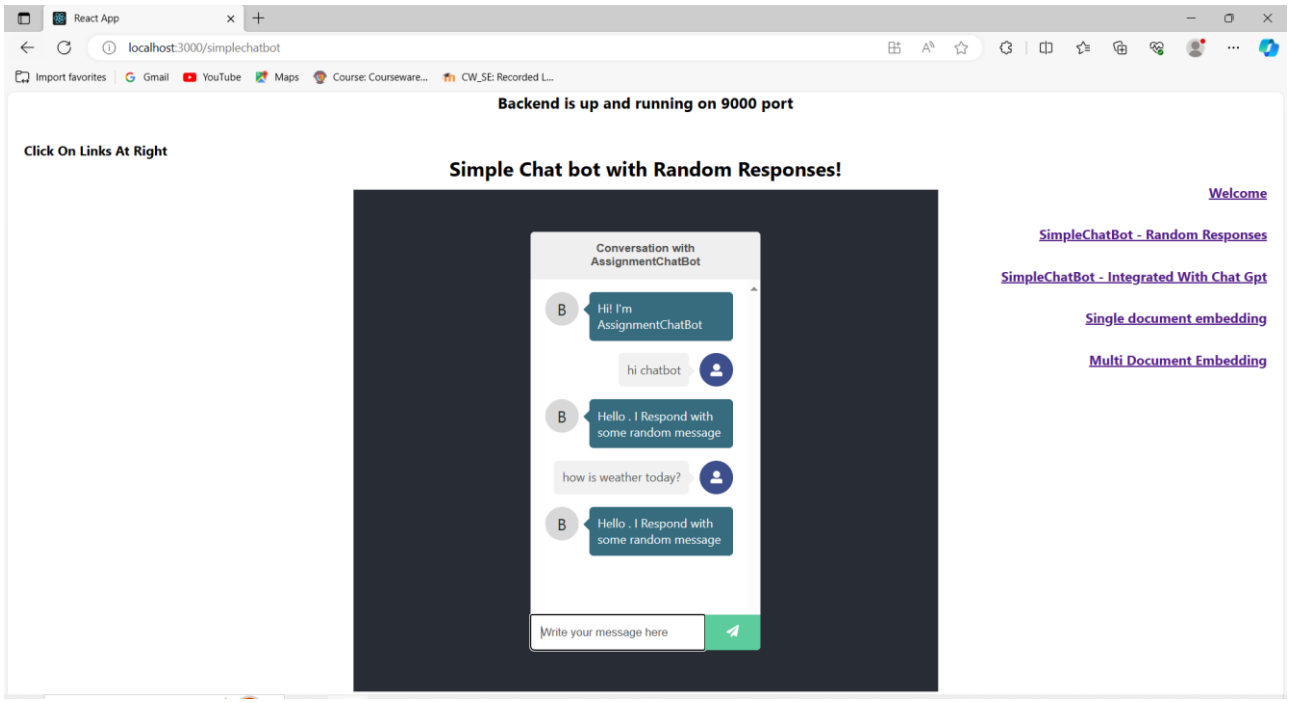
1. Develop a webapp, allowing user to chat with the AI agent by providing a place to enter his query and see the responses obtained by the system. This is similar to any instant messaging system where our responses are coming from the AI agent. **To test the functionality of this step the Chatbot can generate random responses.** [Marks 1%]

Approach:

1) Open-source Chat bot react component is used for Chat bot GUI
2) On a chat query or question from the user respond with a random message at the clilent level as if the message is received from server.

Screen shot of application with chat bot and random responses:

- **Please notice the route url's in address bar**



2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

2. Extend the webapp developed **above to integrate openAI.** Now the Chatbot will generate the responses using openAI API's similar to ChatGPT. **You can get details of using openAI API's from https://community.openai.com/.** You may have to create an account in openAI for this step. You can look on a short course on youtube "Building Systems with the ChatGPT API: A short course from OpenAI and DeepLearning.AI"1 . Also you can do a google search "How to Build an AI-Powered ChatBot with OpenAI, ChatGPT, Node.js, and React" and see the material available online2 . [Marks 2%]

Approach:

1) Open ai create chat completion API is used
   as tin the screen shot.
2) "text-davinci-003" model is used as suggested
3) This service responses will give the responses like chat gpt in our chat bot developed in step above

Screenshot:

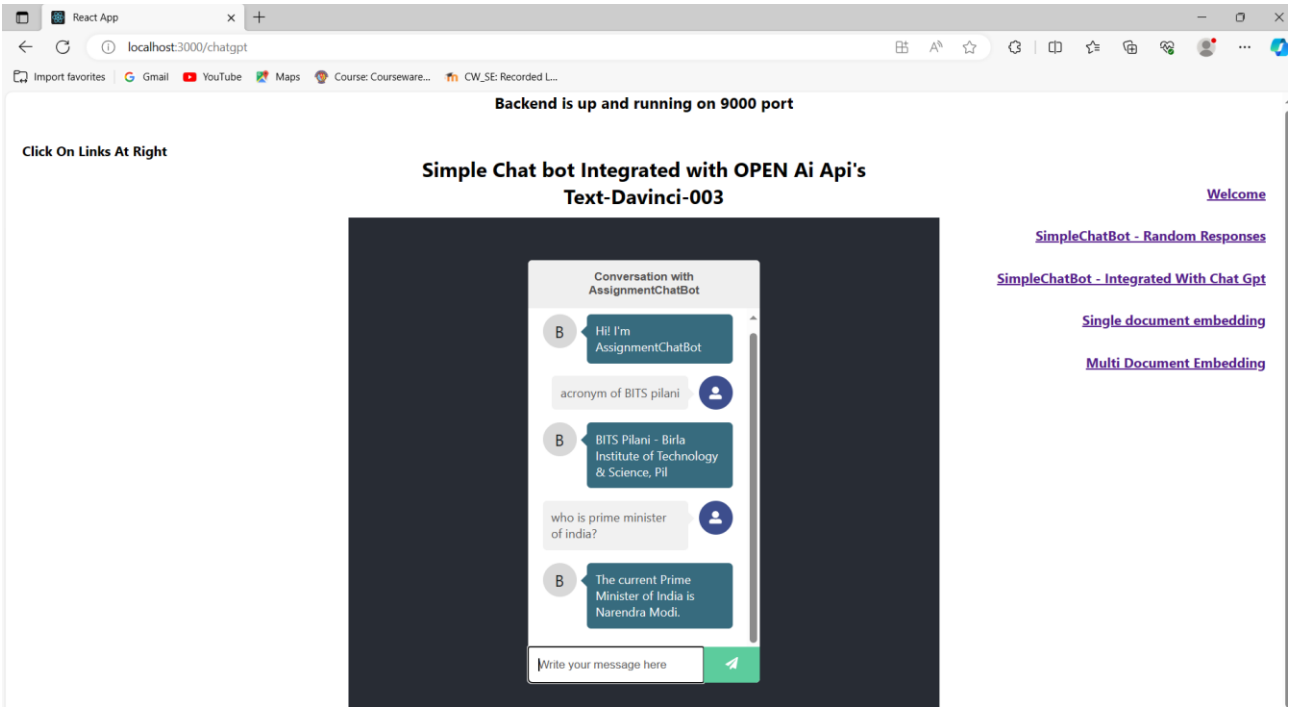
1) Notice the route in browser address bar **http://localhost:3000/chatgpt** in screenshot below.
2) Click on "`SimpleChatBot - Integrated With Chat Gpt`".
3) Sample chat can be found in screen shot below.

```
/* API 2 >> Ask for Simple Chat bot integrated with Chat completions API
app.post("/ask", async (req, res) => {
    const prompt = req.body.prompt;
    try {
      if (prompt == null) {
        throw new Error("Uh oh, no prompt was provided");
      }
      const response = await openai.createCompletion({
        model: "text-davinci-003",
        prompt,
      });
      /*-----------------------*/
       console.log(response);
      /*-----------------------*/
      const completion = response.data.choices[0].text;
      return res.status(200).json({
        success: true,
        message: completion,
      });
    } catch (error) {
      console.log(error.message);
    }
});
```

2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH



Server logs or responses from API of open API:

```
npm start
    upgradeOrConnect: false,
    parser: null,
    maxHeadersCount: null,
    reusedSocket: false,
    host: 'api.openai.com',
    protocol: 'https:',
    _redirectable: Writable {
      _writableState: [WritableState],
      _events: [Object: null prototype],
      _eventsCount: 3,
      _maxListeners: undefined,
      _options: [Object],
      _ended: true,
      _ending: true,
      _redirectCount: 0,
      _redirects: [],
      _requestBodyLength: 71,
      _requestBodyBuffers: [],
      _onNativeResponse: [Function (anonymous)],
      _currentRequest: [Circular *1],
      _currentUrl: 'https://api.openai.com/v1/completions',
      [Symbol(kCapture)]: false
    },
    [Symbol(kCapture)]: false,
    [Symbol(kBytesWritten)]: 0,
    [Symbol(kNeedDrain)]: false,
    [Symbol(corked)]: 0,
    [Symbol(kOutHeaders)]: [Object: null prototype] {
      accept: [Array],
      'content-type': [Array],
      'user-agent': [Array],
      authorization: [Array],
      'content-length': [Array],
      host: [Array]
    },
    [Symbol(errored)]: null,
    [Symbol(kHighWaterMark)]: 16384,
    [Symbol(kRejectNonStandardBodyWrites)]: false,
    [Symbol(kUniqueHeaders)]: null
  },
  data: {
    warning: 'This model version is deprecated. Migrate before January 4, 2024 to avoid disruption of service. Learn more https://platform.openai.com/docs/deprecations',
    id: 'cmpl-8NRlkJncn54yuj5kxqQ656HN7FpV2',
    object: 'text_completion',
    created: 1700597368,
    model: 'text-davinci-003',
    choices: [ [Object] ],
    usage: { prompt_tokens: 8, completion_tokens: 12, total_tokens: 20 }
  }
}
POST /ask 200 697.628 ms - 86
```

2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

2. Further extend the above Chatbot by customising it to generate responses confined to a **domain specific knowledge** gathered by supplying a **PDF to the system.** You have to use embedding model API's of OpenAI to **generate the embedding of a given PDF** and use it as a **database for providing domain specific responses.** Search on google "Customizing an OpenAI Chatbot With Embeddings" and visit page

[Marks 3%]

Approach towards the requirement:

1) Take a single pdf upload from user / front end application
2) With an api move the pdf to the "pdf" folder in the backend.
3) Use pdf reader to read and extract text from the pdf uploaded
4) Chunk the data to a limit of 600 per chunk , as in the free usage open api allows only 3 calls in a minute
5) **Restriction:** For the implementation of assignment point of view please restrict to test this application only with a **single page pdf**
6) Some of the sample pdf's are available here >> https://drive.google.com/drive/folders/1kwQPPTDBO7_0aU6-F_Q5RUrUuzp8bNR6
7) Each chunk of data will be passed to **openai.createEmbedding** api
8) Vector tokes will be created per chunk
9) These data will be store in a json format as in the screenshot in an (csv) format for future reference .
10) Csv files are stored under **"api"/"excel_vector_store "** folder

```
async function callEmbeddingService(chunk,index){
  var result = {};
  return new Promise((resolve) => {
      openai.createEmbedding({
            model:"text-embedding-ada-002",
            input: chunk,
      })
      .then((res) => {
        result = {
            index:index,
            embeddings:res.data["data"][0]["embedding"],
            tokens:res.data["usage"].total_tokens,
            inputText:chunk
          };
          console.log(result);
          return result;
      });
      setTimeout(() => {
          resolve(result);
      },1000);
  });
}
```

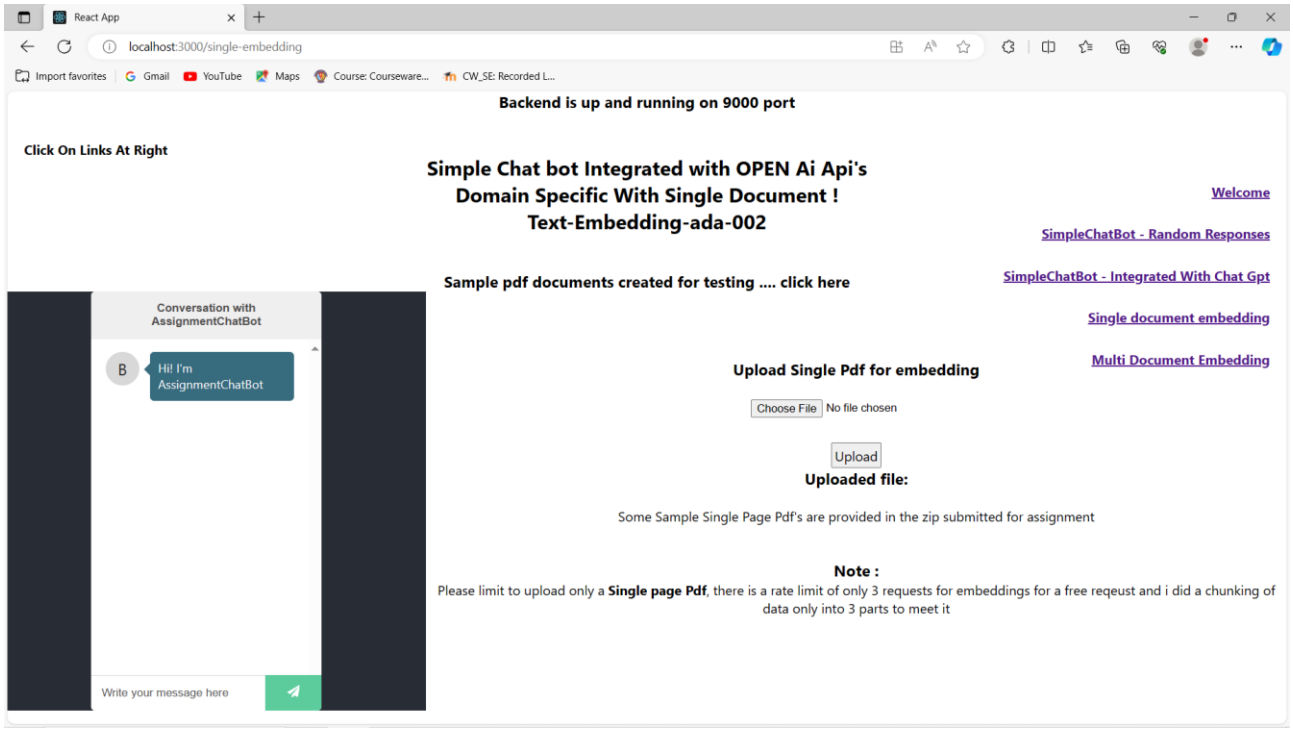
Screen shot of api to create embeddings in **app.js of api node module**

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

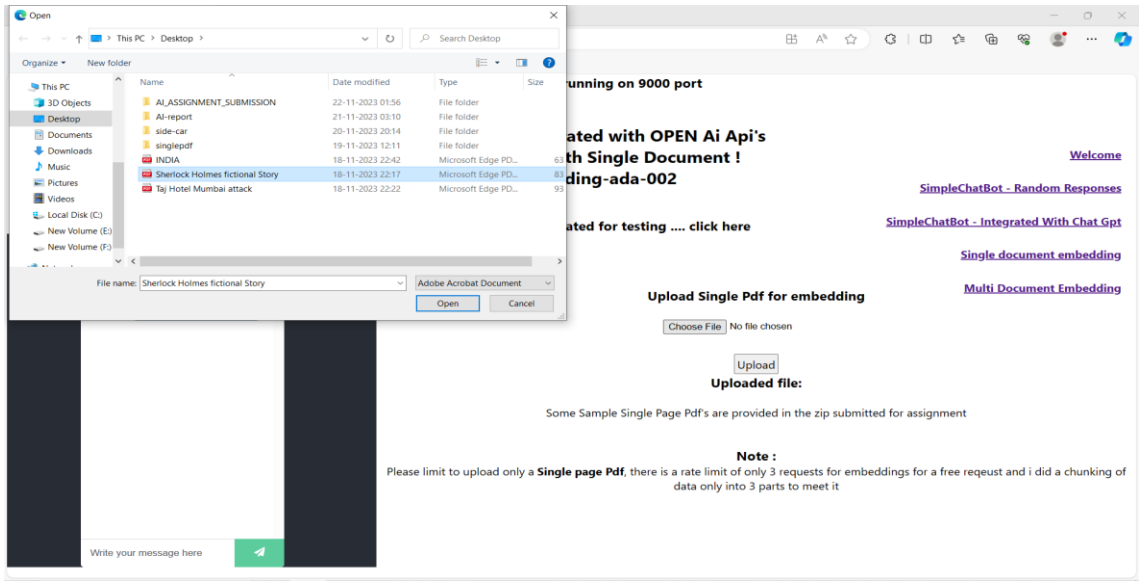
2020MT13045
MAHAVADI SESHA SAI SREENADH

Flow from front-end:

Step1 : Upload PDF



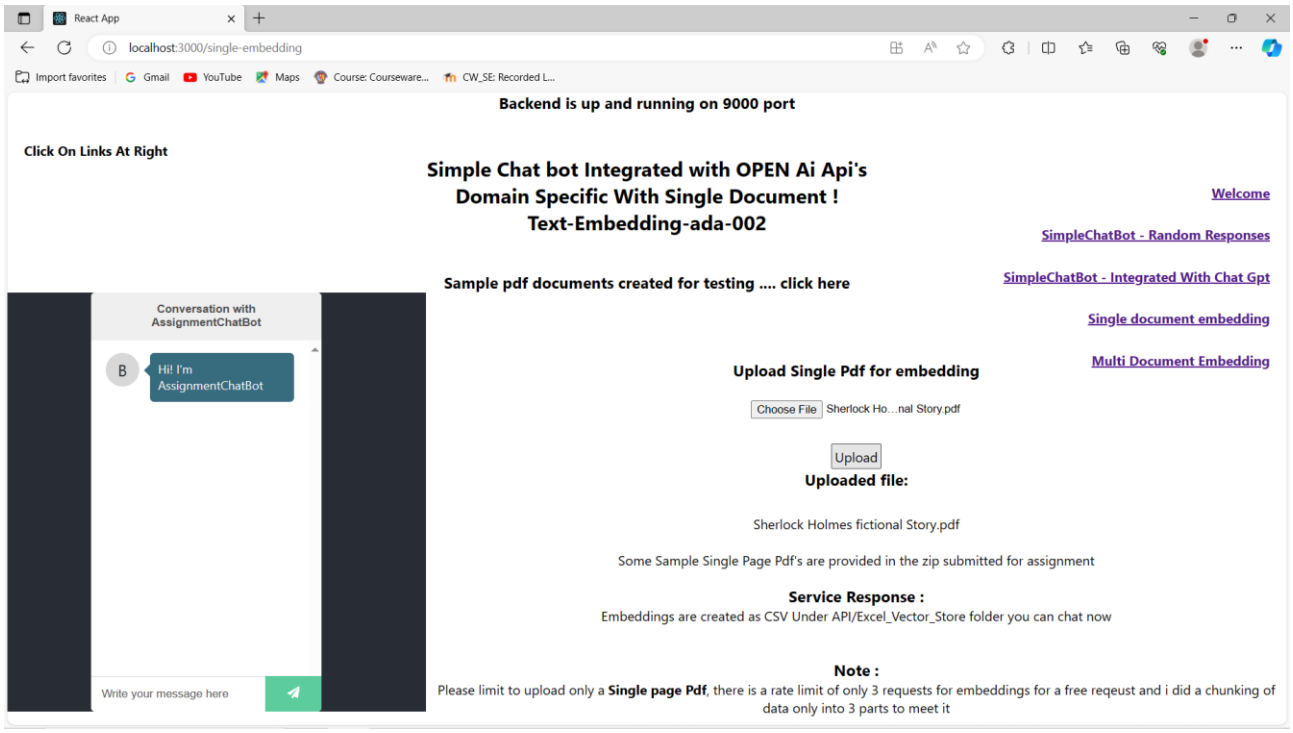
Step2 : “Sherlock Holmes fictional Story.pdf” is chosen to upload using chose file button



2020MT13045
MAHAVADI SESHA SAI SREENADH

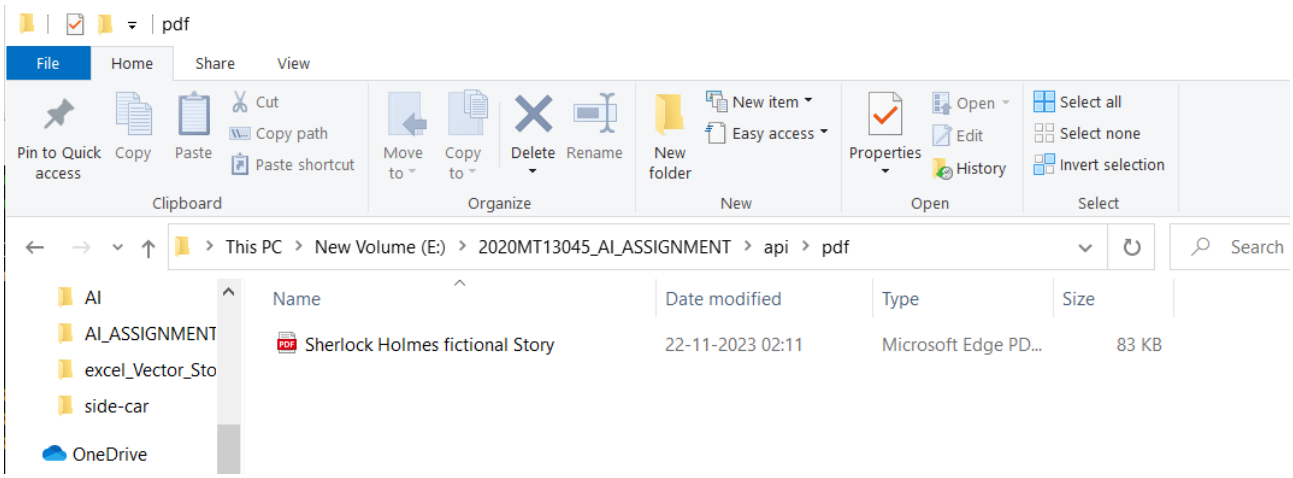
2020MT13045
MAHAVADI SESHA SAI SREENADH

Step 3: Click on upload button to upload



After upload please observe the service response is printed below .

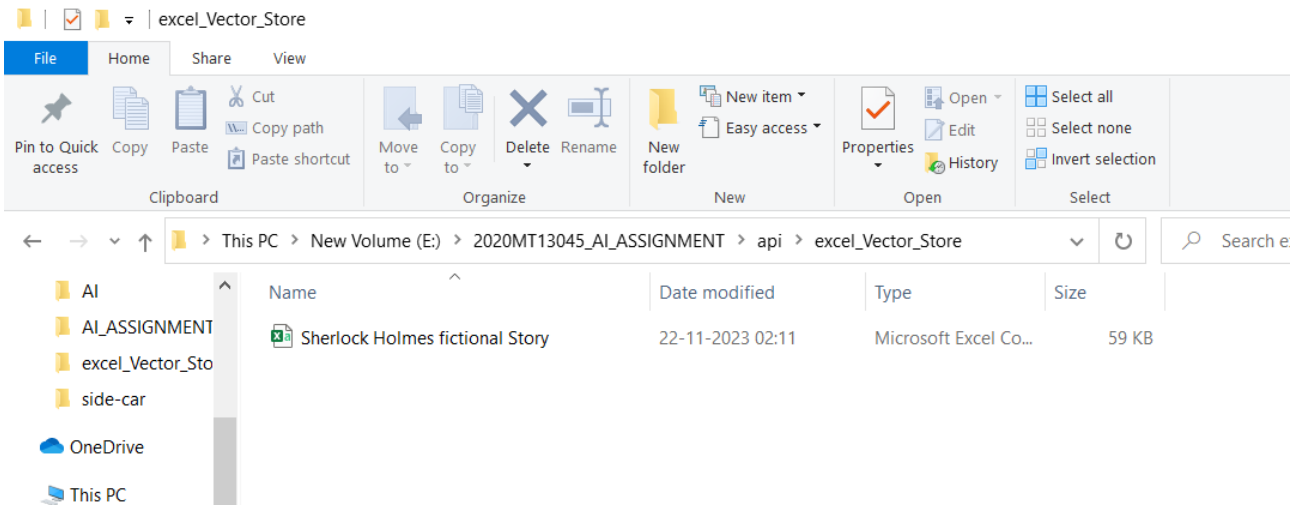
Step 4 : Pdf file is uploaded to backend



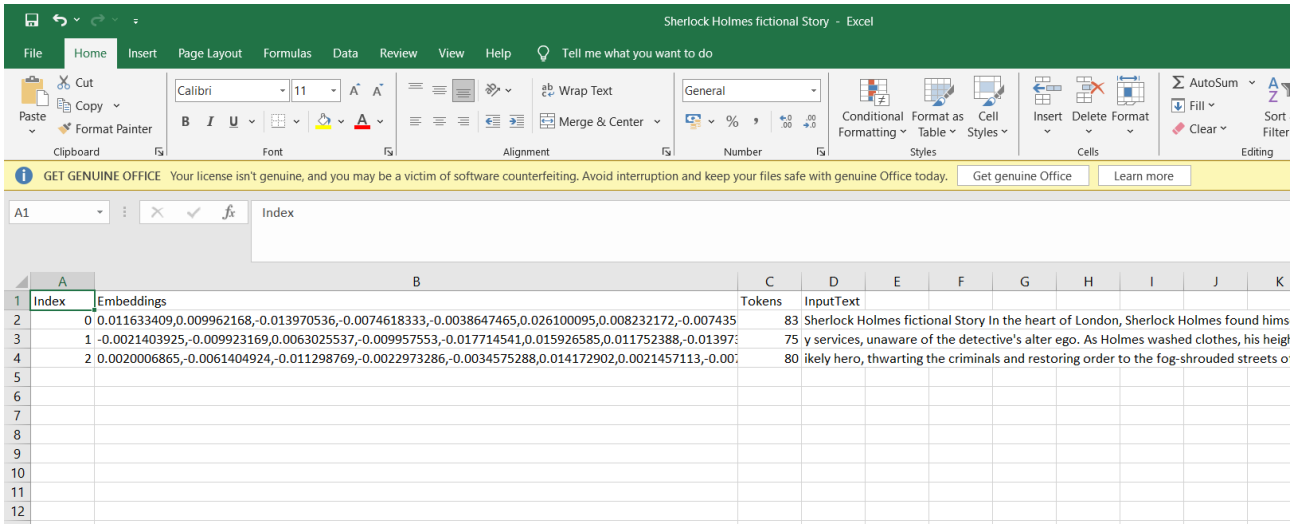
Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

Step 6 : Node backend read and chunked the backend data and crated vector tokens using Open AI api



Step 5 : csv file is created with vector tokens



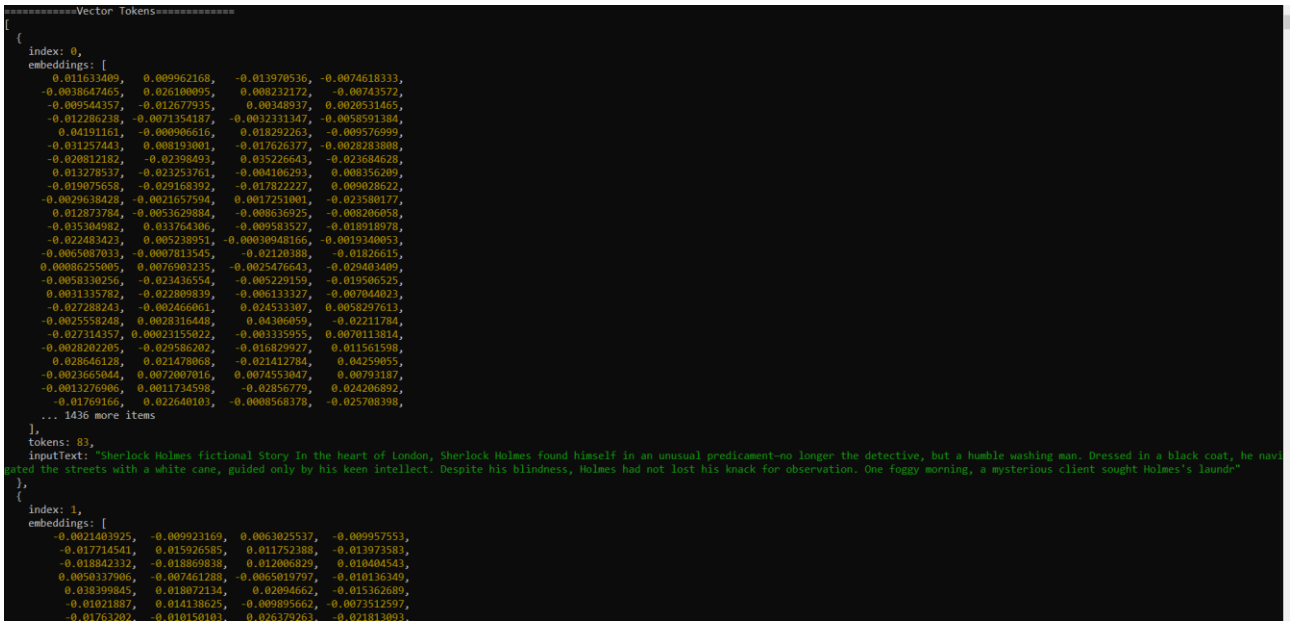
2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

Node JS backend screen shot :

Chunked the input text and vector tokens are created

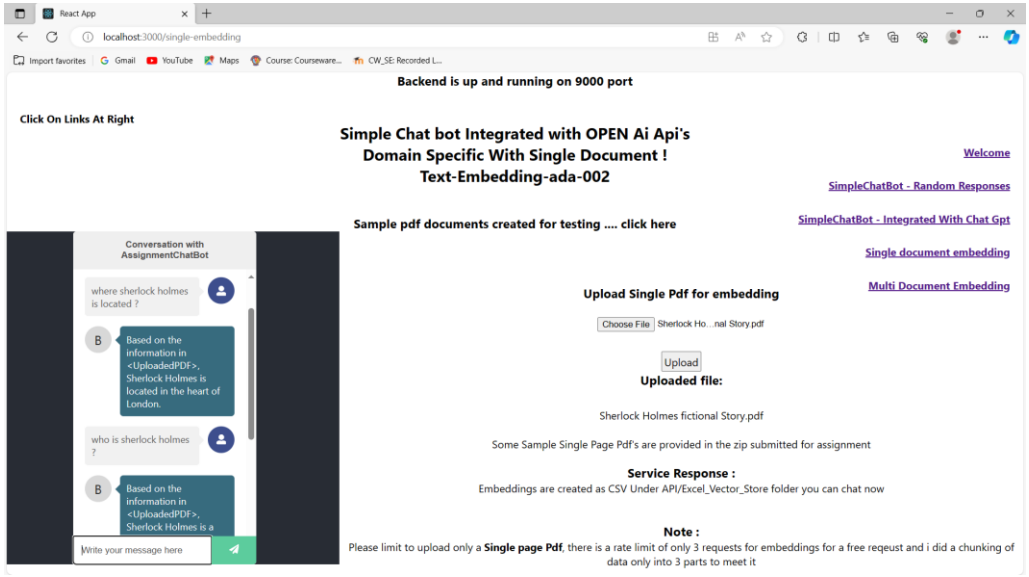


Step 6: upon chat the file uploaded will be set and used for context of chat

>> The chat will be restricted to the context of PDF

>> any questions other than that will be politely told out of context by chat bot

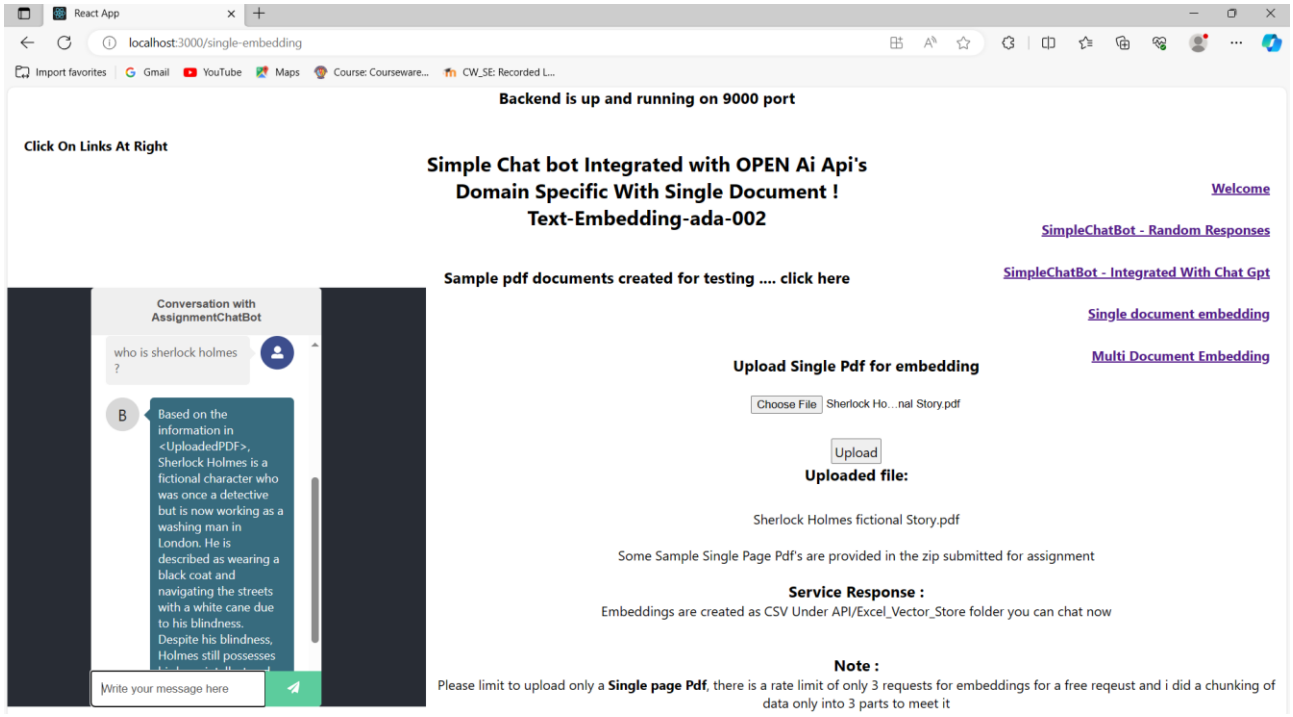
Please refer screen shots below



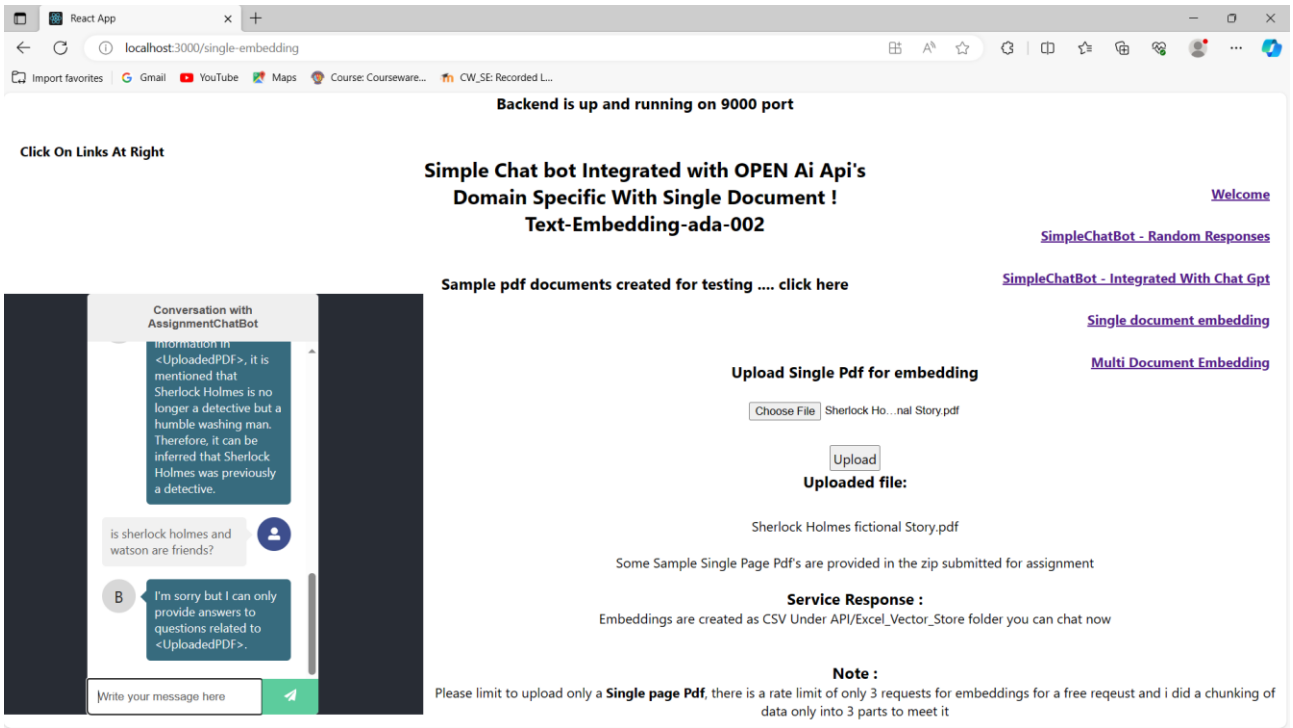
2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH



Chat bot unable to answer for the information out side the context of pdf

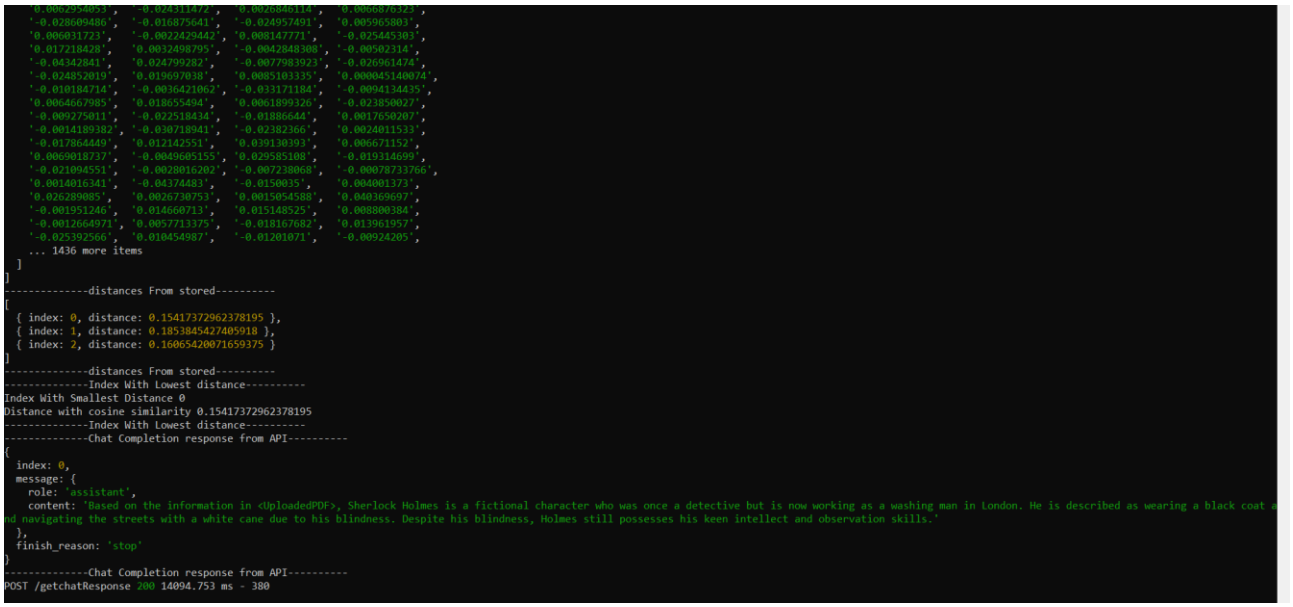


2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

Cosine Similarity and shortest distance calculation between question and stored data :



Create chat completion API from open API is used to create a response using the context

```
/* Now pass the chunk of text with lowest distance to completions API to get response */

//Chat Prompts are picked from an online reference
 const chatCompletionInPdfContext = await openai.createChatCompletion({
   messages: [
     { role: "system", content: "You are a helpful assistant" },
     {
       role: "assistant",
       content: "We are going to call the following set of information <UploadedPDF>:\n\n'
     },
     {
       role: "assistant",
       content: "If question is NOT related to <UploadedPDF> or NearForm respond with: 'I
     },
     {
       role: "assistant",
       content:"If there is NO relevant information in <UploadedPDF> to answer the questi
     },
     {
       role: "assistant",
       content: "If you provide an answer, use only the information existing in <Uploaded
     },
     {
       role: "user",
       content: "Question:"+question,
     },
   ],
   temperature: 0,
   top_p: 1,
   model: "gpt-3.5-turbo",
 });
```

2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

4. Integrate the uploading of PDF, generating embedding, and the AI agent into a single web application. The GUI of the app should allow users to browse and upload **multiple PDFs** and set the context for the conversation. Now the Chatbot should answer based on the supplied PDFs. [Marks 3%]

Approach :

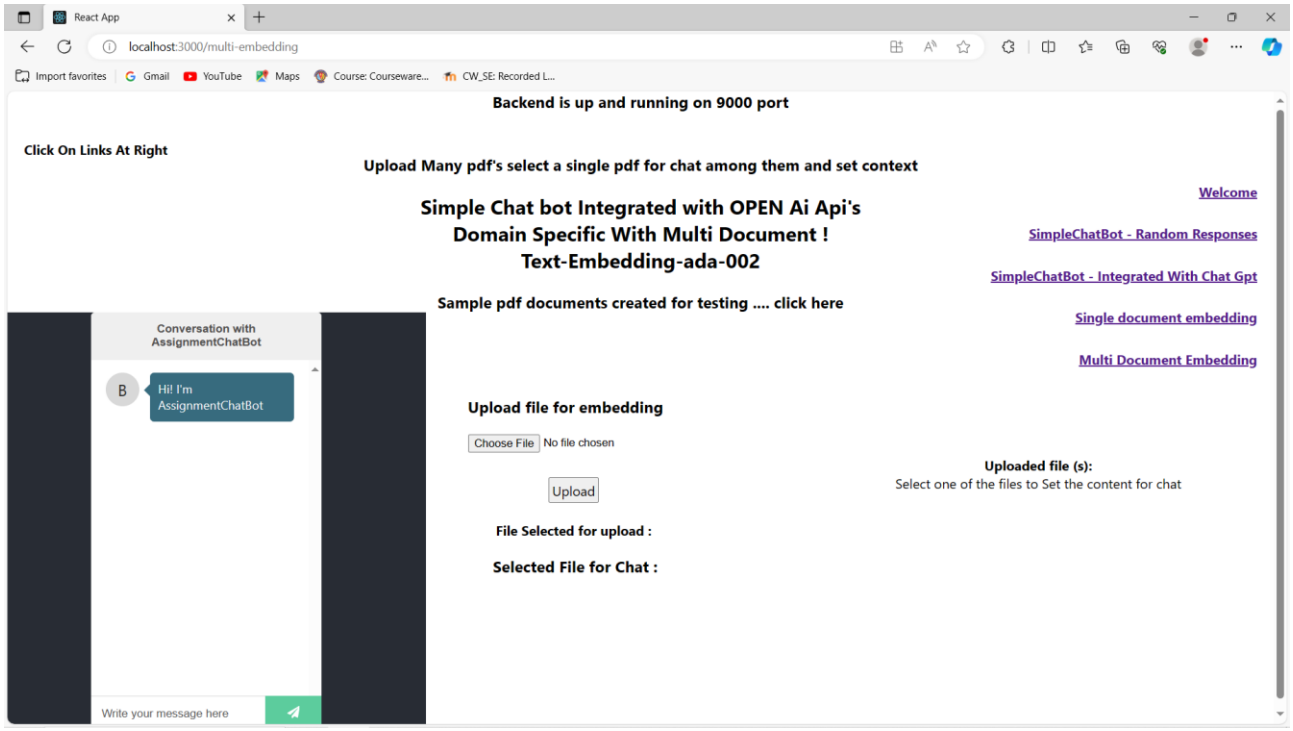
1) Allow to upload multiple pdf's
2) **Store all the uploaded files in backend along with the vector tokes created for them as CSV files**
3) Give a chance to select which file user intends to set as context for chat **using radio button against the uploaded file names**
4) **On the context of the file with checked radio button chat will happen**
5) **In the middle of the chat selection can be changed to change the context of chat .**

Note : As there is a rate limit of 3 requests per minute its better to restrict to 2 uploads and give a gap between first upload and second upload

(or) wait and upload

Step 1: Upload Multiple files into the application

1) Note the URL of ROUTE

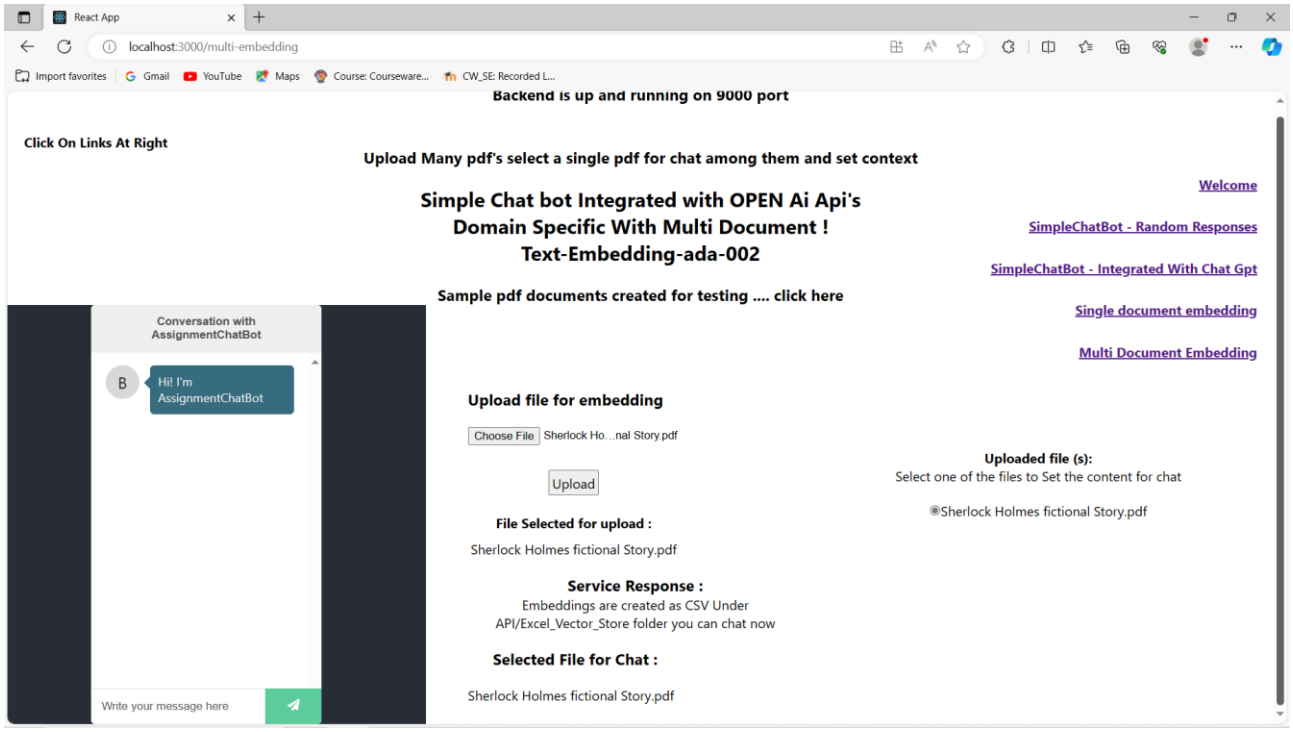


2020MT13045
MAHAVADI SESHA SAI SREENADH

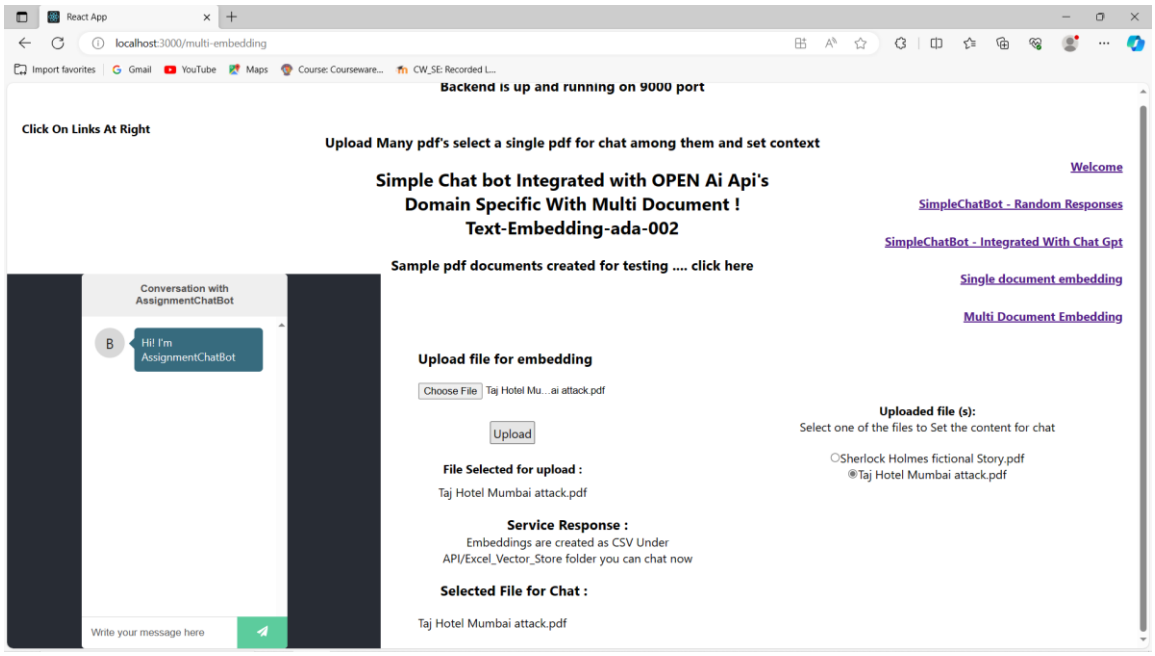
Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

Frist file upload



Second file upload



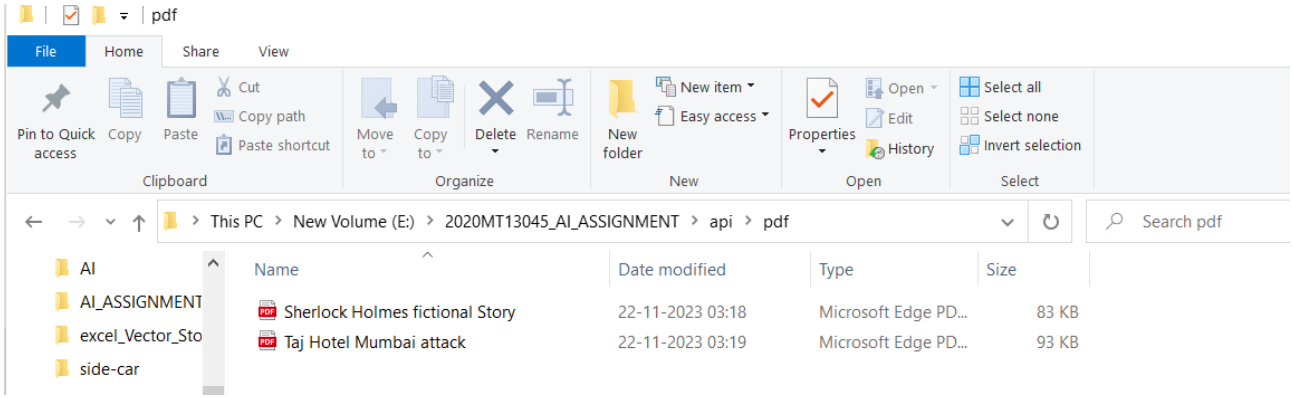
2020MT13045
MAHAVADI SESHA SAI SREENADH

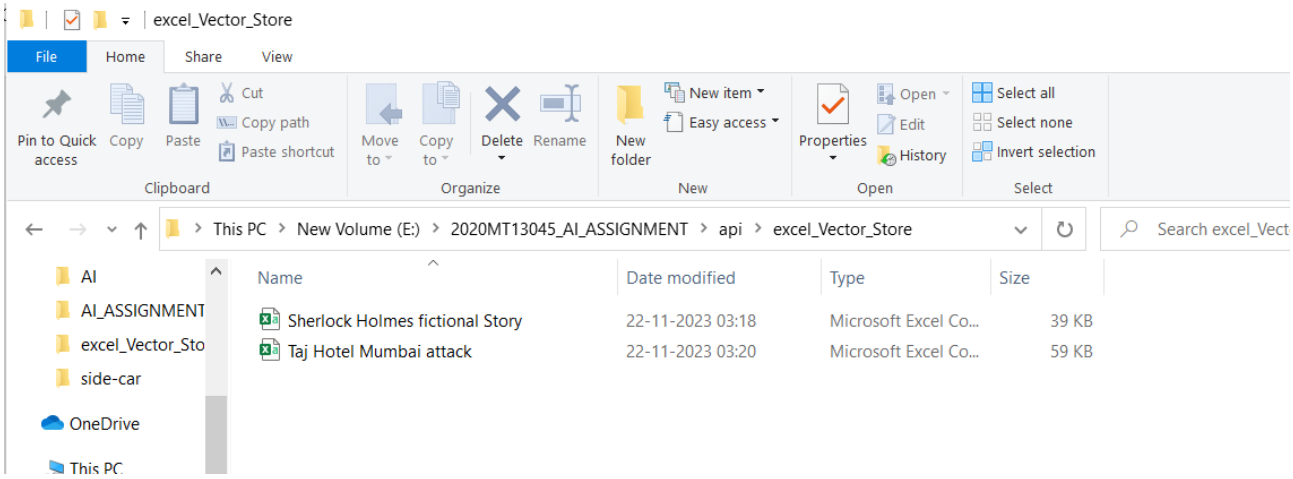
Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

Step 2 : Verify if files are uploaded and vector tokens are created well

Pdf's in server

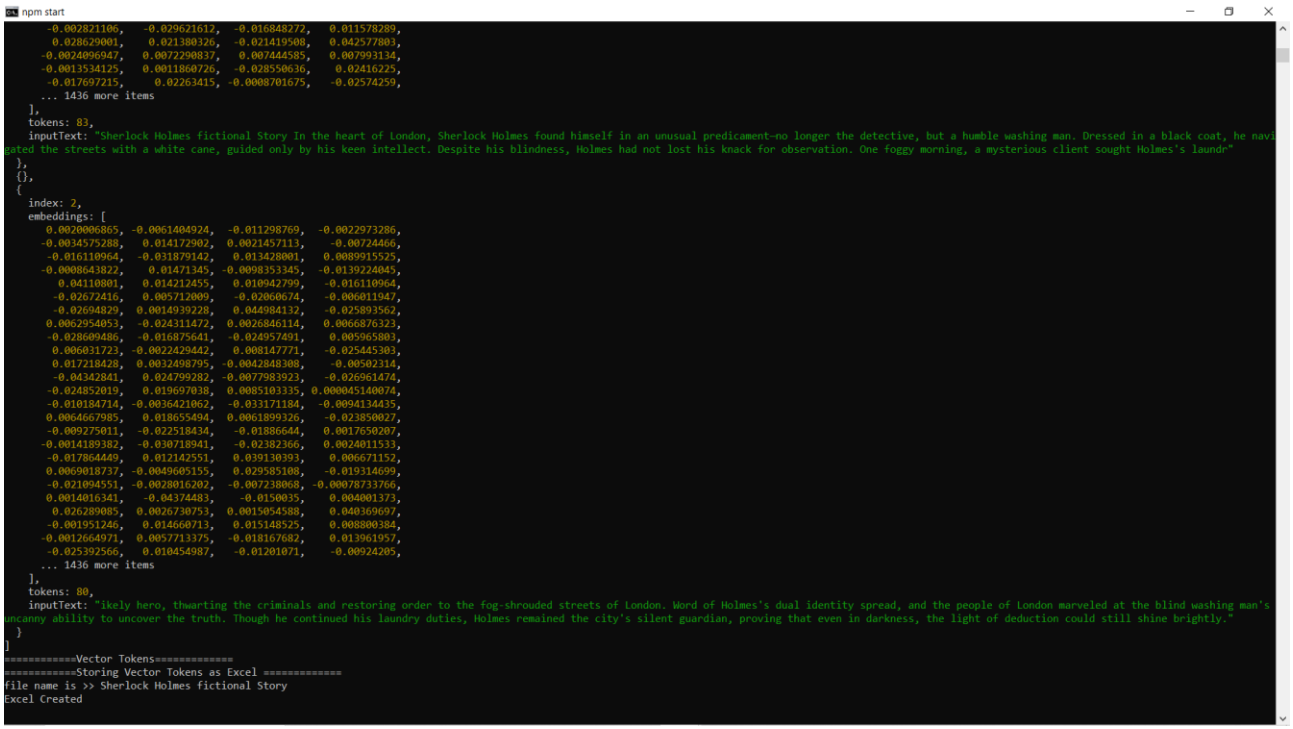


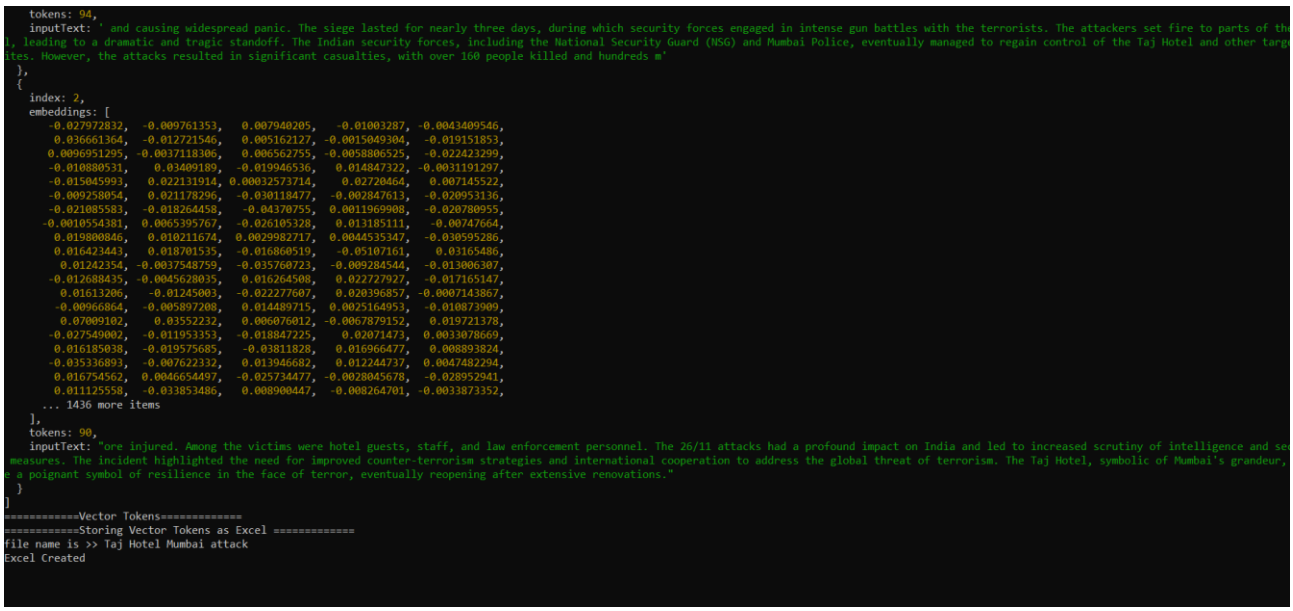


2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

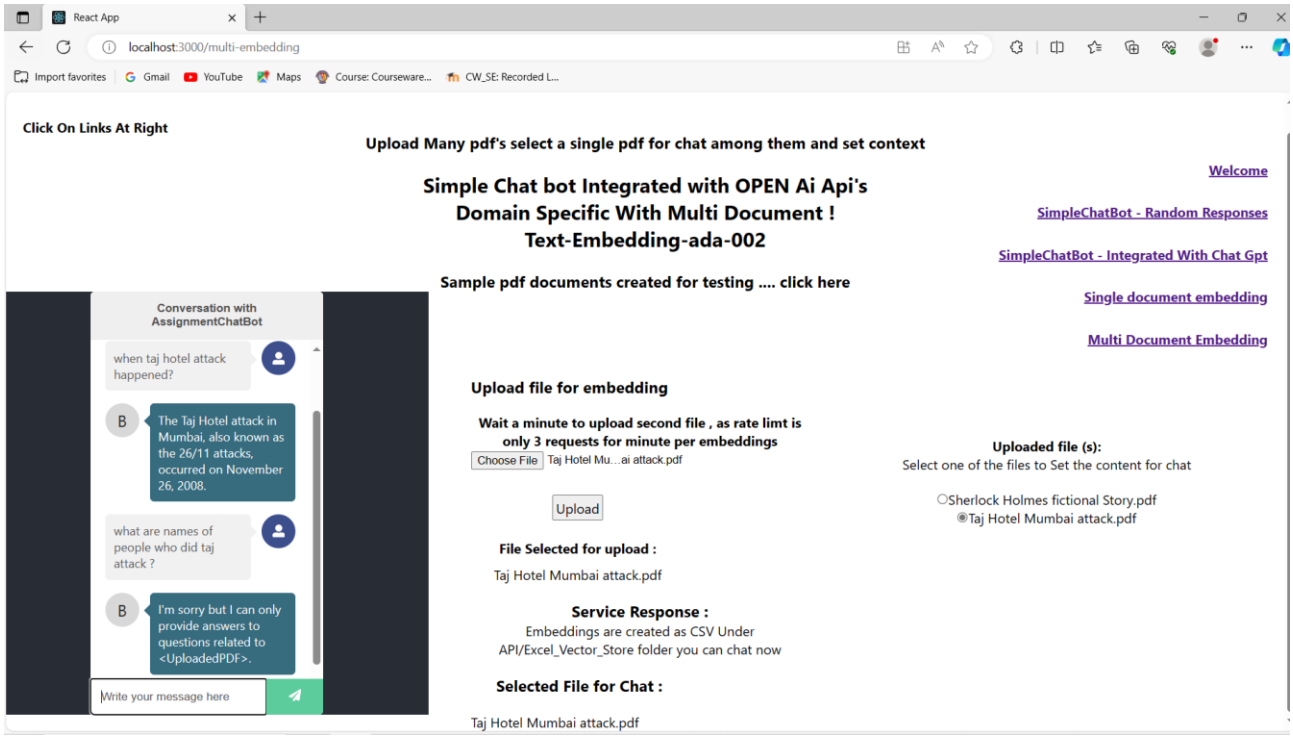




2020MT13045
MAHAVADI SESHA SAI SREENADH

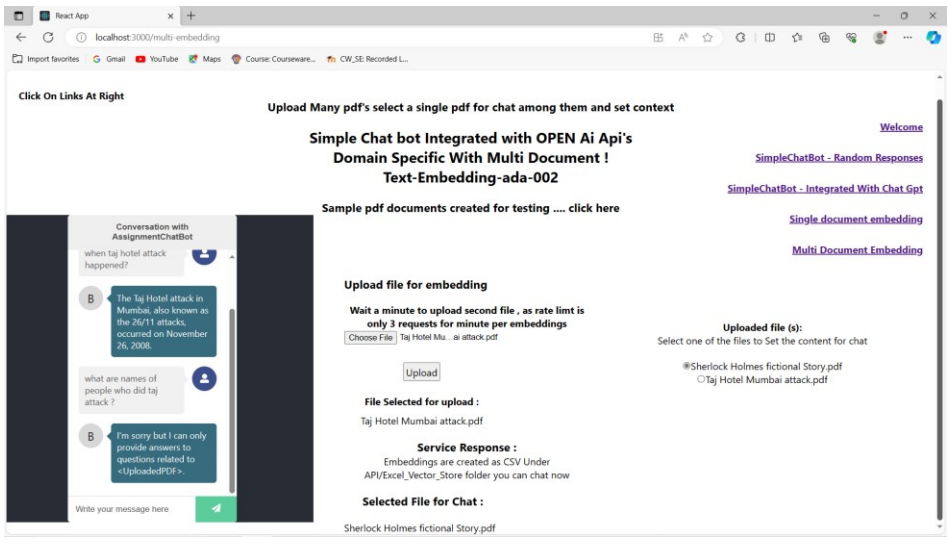
2020MT13045
MAHAVADI SESHA SAI SREENADH

Step 3 : Select a particular file and chat in the context of that file



Step 4 : Change the radio button and change the context of chat

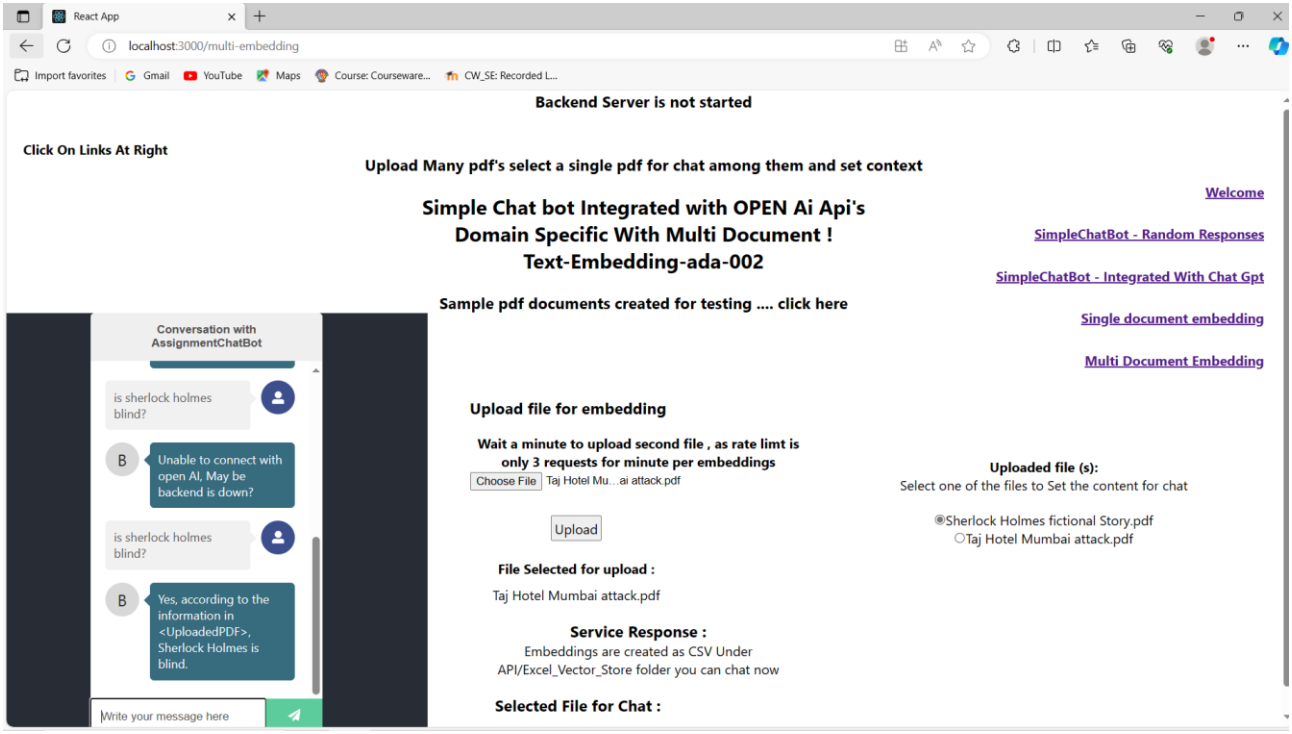
Changing selection to sherlock holmes file



2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH



Note : As there is a rate limit of 3 requests per minute its better to restrict to 2 uploads and give a gap between first upload and second upload

2020MT13045
MAHAVADI SESHA SAI SREENADH

Birla Institute of Technology and Science Pilani
IS/SE/SS ZC444: Artificial Intelligence
Assignment-01: Custom AI Chatbot Using LLM

2020MT13045
MAHAVADI SESHA SAI SREENADH

Make a single zip file consisting of 1) source code, 2) readMe file containing installation and execution instructions.

Upload zip file through the e-learn portal https://elearn.bits-pilani.ac.in/

• Create a short video of less than 5 min, demonstrating the working of your app. Share it's link in the readMe file and fill the form

https://forms.gle/dEAzpo38ULxejoRf7

[Marks 1%].

- **Zip file is uploaded in elearn portal**
- **Google Form provided above is filled**

2020MT13045
MAHAVADI SESHA SAI SREENADH