LABORATORY REPORT

Application Development Lab (CS33002)

B.Tech Program in ECSc

Submitted By

Name:- Harsh Kumar

Roll No: 2230084

**Kalinga Institute of Industrial Technology
(Deemed to be University)
Bhubaneswar, India**

Spring 2024-2025

Experiment Number	5
Experiment Title	Web Scraper using LLMs
Date of Experiment	11-02-2025
Date of Submission	16 -02-2025

1. Objective:-

To create a web scraper application integrated with LLMs for processing scraped data.

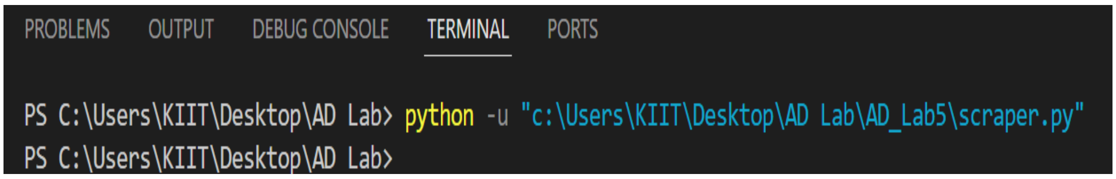
2. Procedure:- (Steps Followed)

1. Use Python libraries like BeautifulSoup and Requests to scrape web data.
2. You can also use LlamaIndex for Web Scraping and Ollama for open ended LLMs
3. Integrate LLMs to process and summarize the scraped information.
4. Develop a Flask backend for handling scraping tasks and queries.
5. Create an HTML/CSS frontend to initiate scraping (like the web page to scrape) and display results.
6. You can also take a topic and search the web for a web page and then scrape it.

3. Code:-

❖ *Use Python libraries like BeautifulSoup and Requests to scrape web data.*

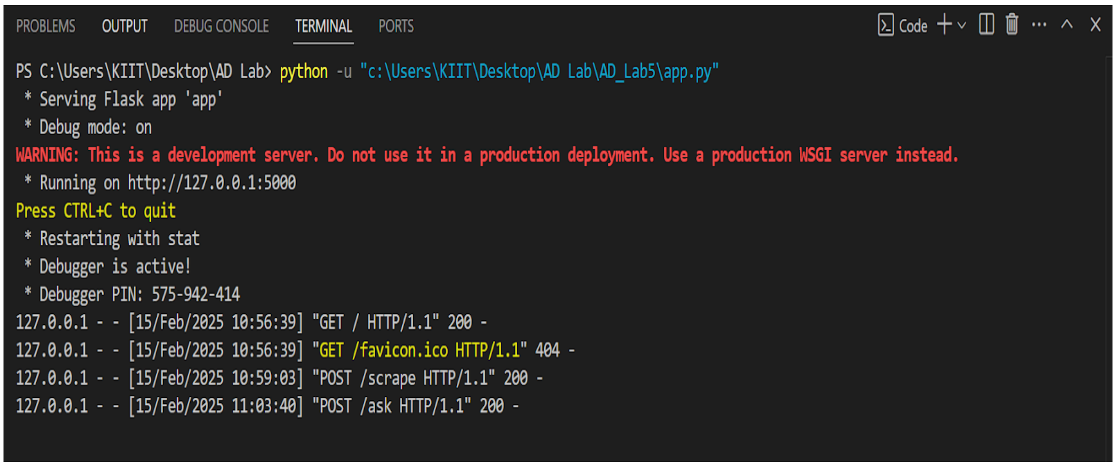
```
import requests
from bs4 import BeautifulSoup
import ollama

def scrape_website(url):
    try:
        response = requests.get(url)
        response.raise_for_status()

        soup = BeautifulSoup(response.text,
                        'html.parser')
                paragraphs = soup.find_all('p')
                text_content = "\n".join([p.get_text() for p
                        in paragraphs])
                    return text_content
            except requests.exceptions.RequestException as e:
                return f"Error: {e}"
```



```
PS C:\Users\KIIT\Desktop\AD Lab> python -u "c:\Users\KIIT\Desktop\AD Lab\AD_Lab5\scraper.py"
PS C:\Users\KIIT\Desktop\AD Lab>
```

❖ *Flask backend for handling scraping tasks and queries.*

```
from flask import Flask, render_template, request, jsonify
from scraper import scrape_website  # Removed summarize_content
import ollama

app = Flask(__name__)

scraped_text_global = ""

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/scrape', methods=['POST'])
def scrape():
    global scraped_text_global
    data = request.json
    url = data.get("url")

    if not url:
        return jsonify({"error": "URL is required"}), 400

    scraped_text = scrape_website(url)
    scraped_text_global = scraped_text

    return jsonify({"scraped_text": scraped_text})  # Removed summary

@app.route('/ask', methods=['POST'])
def ask():
    global scraped_text_global
    data = request.json
    question = data.get("question")

    if not question:
        return jsonify({"error": "Question is required"}), 400

    if not scraped_text_global:
        return jsonify({"error": "No scraped text available. Please scrape a website first."}), 400

    model = "llama3:latest"
    prompt = f"Based on the following text, answer the question:\n\n{scraped_text_global[:3000]}\n\nQuestion: {question}"

    response = ollama.chat(model=model, messages=[{"role": "user", "content": prompt}])

    answer = response['message']['content'] if 'message' in response else "Error in generating answer."
    return jsonify({"answer": answer})

if __name__ == '__main__':
    app.run(debug=True)
```



```
PS C:\Users\KIIT\Desktop\AD Lab> python -u "c:\Users\KIIT\Desktop\AD Lab\AD_Lab5\app.py"
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 575-942-414
127.0.0.1 - - [15/Feb/2025 10:56:39] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [15/Feb/2025 10:56:39] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [15/Feb/2025 10:59:03] "POST /scrape HTTP/1.1" 200 -
127.0.0.1 - - [15/Feb/2025 11:03:40] "POST /ask HTTP/1.1" 200 -
```

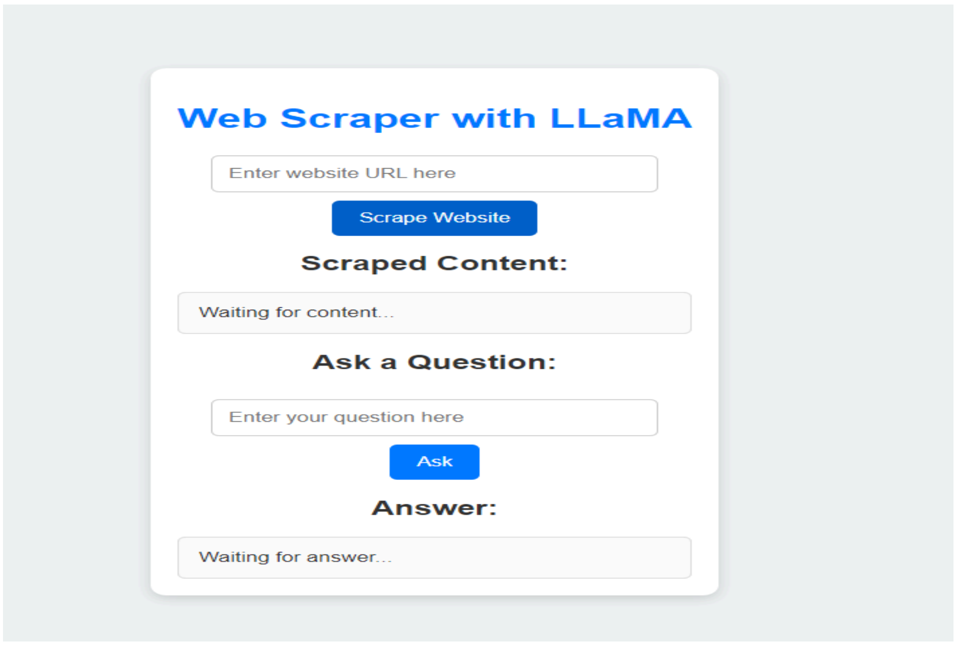
❖ *HTML/CSS frontend to initiate scraping (like the web page to scrape) and display results.*

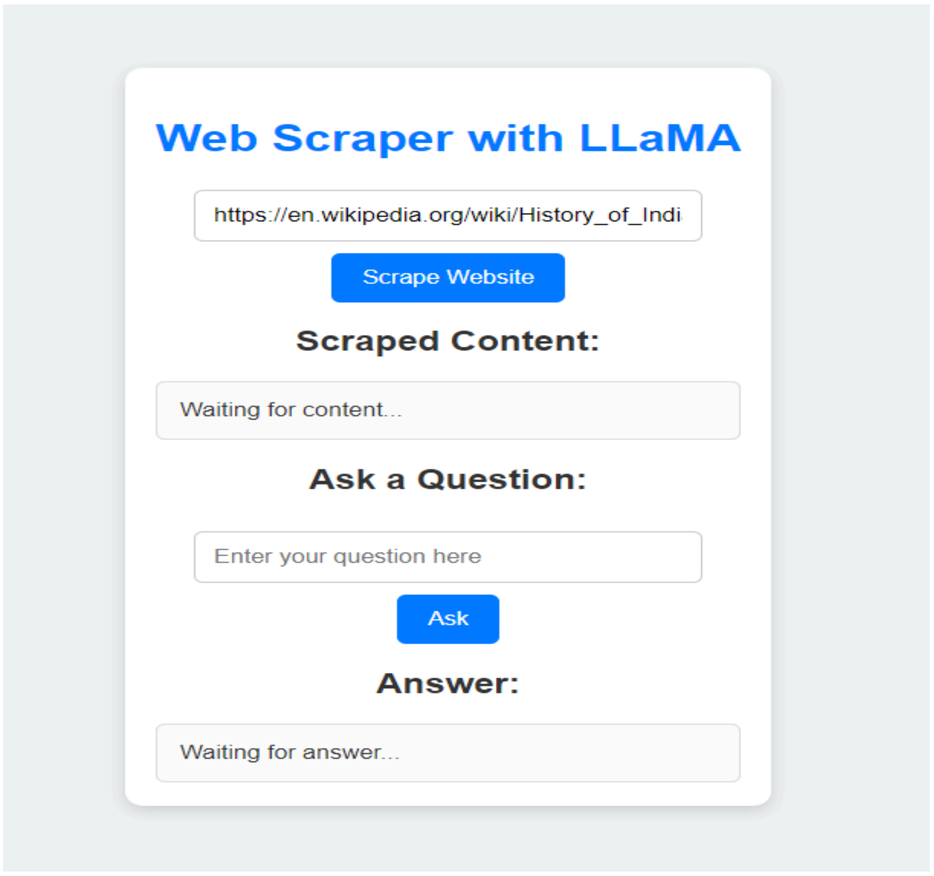
```
<!DOCTYPE html>
<html lang="en">
  <head>
  <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Web Scraper with LLaMA</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background-color: #eef2f3;
        color: #333;
        margin: 0;
        padding: 20px;
        display: flex;
        justify-content: center;
        align-items: center;
        min-height: 100vh;
    }
    .container {
        max-width: 700px;
        background: white;
        padding: 20px;
        border-radius: 12px;
        box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);
        text-align: center;
    }
    h1 {
        color: #007BFF;
        margin-bottom: 15px;
    }
    input[type="text"] {
        width: 80%;
        padding: 12px;
        margin: 10px 0;
        border: 1px solid #ccc;
        border-radius: 6px;
        font-size: 16px;
    }
    button {
        background-color: #007BFF;
        color: white;
        border: none;
        padding: 12px 20px;
        border-radius: 6px;
        cursor: pointer;
        font-size: 16px;
```

```
            transition: background 0.3s;
        }
        button:hover {
            background-color: #0056b3;
        }
        .content-box {
            background: #fafafa;
            padding: 15px;
            border-radius: 6px;
            max-height: 200px;
            overflow-y: auto;
            border: 1px solid #ddd;
            margin-top: 10px;
            text-align: left;
        }
        .scroll-button {
            position: fixed;
            bottom: 20px;
            right: 20px;
            background: #007BFF;
            color: white;
            border: none;
            padding: 10px 15px;
            border-radius: 50%;
            cursor: pointer;
            font-size: 18px;
            display: none;
        }
        .scroll-button:hover {
            background: #0056b3;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Web Scraper with LLaMA</h1>
        <input type="text" id="urlInput" placeholder="Enter website URL here">
        <button onclick="scrapeWebsite()">Scrape Website</button>

        <h2>Scraped Content:</h2>
        <div id="scrapedText" class="content-box">Waiting for content...</div>
        <h2>Ask a Question:</h2>
            <input type="text" id="questionInput" placeholder="Enter your question
here">
        <button onclick="askQuestion()">Ask</button>

        <h2>Answer:</h2>
        <div id="answerText" class="content-box">Waiting for answer...</div>
    </div>

    <button class="scroll-button" onclick="scrollToTop()">&#8679;</button>

    <script>
        function scrapeWebsite() {
            let url = document.getElementById("urlInput").value;
            if (!url) {
                alert("Please enter a URL.");
                return;
            }

            fetch('/scrape', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ url: url })
            })
            .then(response => response.json())
            .then(data => {
                                document.getElementById("scrapedText").textContent  =
data.scraped_text || "No content found.";
            })
            .catch(error => console.error('Error:', error));
        }

        function askQuestion() {
            let question = document.getElementById("questionInput").value;
            if (!question) {
                alert("Please enter a question.");
                return;
            }

            fetch('/ask', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ question: question })
            })
            .then(response => response.json())
            .then(data => {
                document.getElementById("answerText").textContent = data.answer ||
"No answer generated.";
            })
            .catch(error => console.error('Error:', error));
        }

        window.addEventListener("scroll", function() {
            let scrollButton = document.querySelector(".scroll-button");
            if (window.scrollY > 200) {
                scrollButton.style.display = "block";
            } else {
                scrollButton.style.display = "none";
            }
        });

        function scrollToTop() {
            window.scrollTo({ top: 0, behavior: "smooth" });
        }
    </script>
</body>
</html>
```

4. Results/Output:- Entire Screen Shot including Date & Time





Web Scraper with LLaMA

https://en.wikipedia.org/wiki/History_of_Indi

Scrape Website

Scraped Content:

Waiting for content...

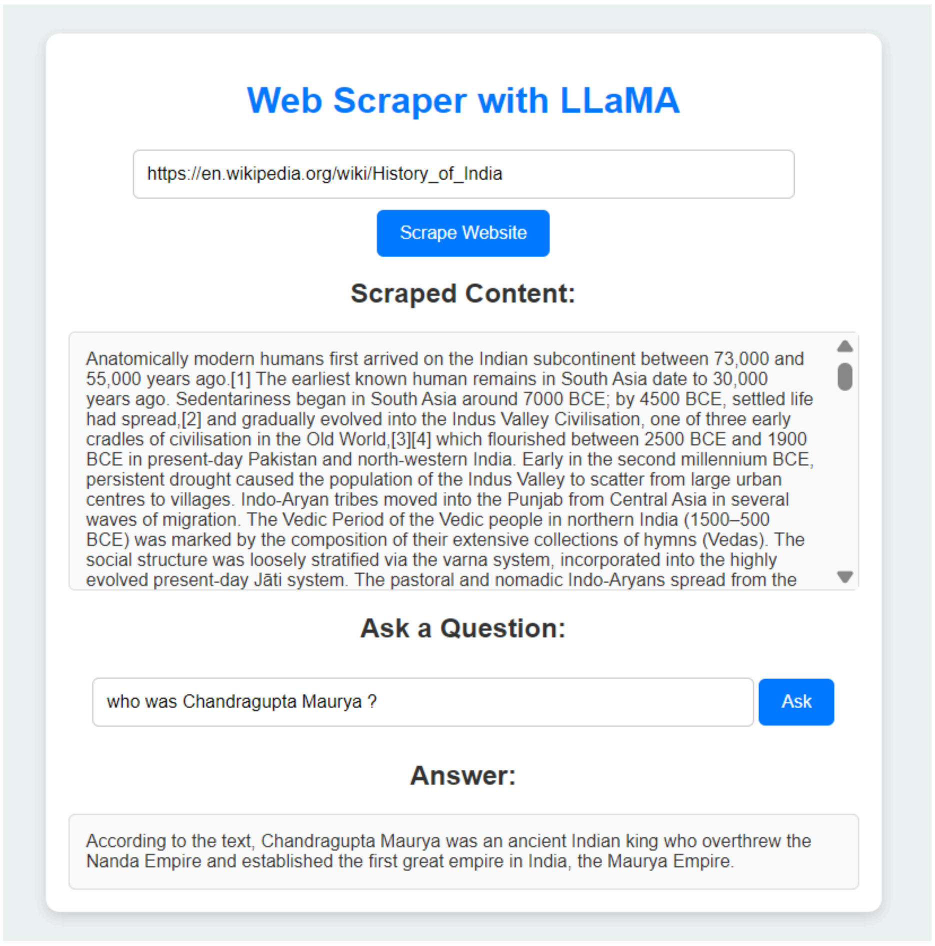
Ask a Question:

Enter your question here

Ask

Answer:

Waiting for answer...



Web Scraper with LLaMA

https://en.wikipedia.org/wiki/History_of_India

Scrape Website

Scraped Content:

Anatomically modern humans first arrived on the Indian subcontinent between 73,000 and 55,000 years ago.[1] The earliest known human remains in South Asia date to 30,000 years ago. Sedentariness began in South Asia around 7000 BCE; by 4500 BCE, settled life had spread,[2] and gradually evolved into the Indus Valley Civilisation, one of three early cradles of civilisation in the Old World,[3][4] which flourished between 2500 BCE and 1900 BCE in present-day Pakistan and north-western India. Early in the second millennium BCE, persistent drought caused the population of the Indus Valley to scatter from large urban centres to villages. Indo-Aryan tribes moved into the Punjab from Central Asia in several waves of migration. The Vedic Period of the Vedic people in northern India (1500–500 BCE) was marked by the composition of their extensive collections of hymns (Vedas). The social structure was loosely stratified via the varna system, incorporated into the highly evolved present-day Jāti system. The pastoral and nomadic Indo-Aryans spread from the

Ask a Question:

who was Chandragupta Maurya ?

Ask

Answer:

According to the text, Chandragupta Maurya was an ancient Indian king who overthrew the Nanda Empire and established the first great empire in India, the Maurya Empire.

5. Remarks:-

This experiment successfully combined web scraping with LLM-based processing to extract and analyze web data. Using BeautifulSoup and Requests, we scraped website content, while LlamaIndex and Ollama helped process and generate insights. A Flask backend handled scraping tasks, and an HTML/CSS frontend provided an interactive user interface. This approach enables efficient data extraction and querying, making it useful for various real-world applications.

Signature of the Student	Signature of the Lab Coordinator
(Harsh Kumar)	(Name of the Coordinator)