

Advanced Web Scrapping + Search

...

- Artjola Meli
- Shradha Godse

Project Structure

MySQL

MY_CUSTOM_BOT
Database

Central storage for
search URLs and
frequencies

Python Application

myapp.py script

- Interacts with the database to fetch and store data
- URL data extraction and data processing using **Selenium** for automated web scraping

Web Browser GUI (Manages user interactions)

index.html + results.html

- **index.html:** Takes user input for search terms
- **results.html:** Displays search results fetched from the database

Technologies Used

01

MySQL - Database

- Insert the search results
- Store the URLs and Frequency
- Display the URLs and Frequency

02

Flask Framework

- Routing and View Functions
- Integration with Selenium & MySQL
- Server Configuration & Error Handling

03

Selenium

- Web Browser Automation
- Fetching & Scraping URLs from Search Engines
- Handling Pagination

04

BeautifulSoup

- HTML Parsing
- Data Extraction
- Enhancing Data Quality

Duplicate Elimination Logic

Database Primary Key Constraint:

- **Primary Key on URL:**
 - The URL field is set as the primary key in the database.
 - This constraint automatically prevents duplicate URLs from being inserted.

In-Memory Check:

- **Session-Based List:**
 - During each session, maintain a list of URLs already collected and before adding a new URL, check if it already exists.
 - This immediate check ensures no duplicates within the same scraping session.

Advertisement Elimination Logic

- **HTML Tag Selection:**
 - Target tags (<h3>, <h4>) used for organic results.
 - Avoid structures or markers indicating ads.
- **CSS Selector Waiting:**
 - Wait for elements that indicate organic results.
 - Avoid partial results and late-loading ads.
- **URL Pattern Filtering:**
 - Extract URLs with patterns typical for organic results.
 - Skip URLs with ad-related parameters or structures.
- **Ad Marker Exclusion:**
 - Identify and skip elements indicating advertisements.

MY_CUSTOM_BOT Database

Table
Description

```
25 • DESCRIBE searchresults;
26
```

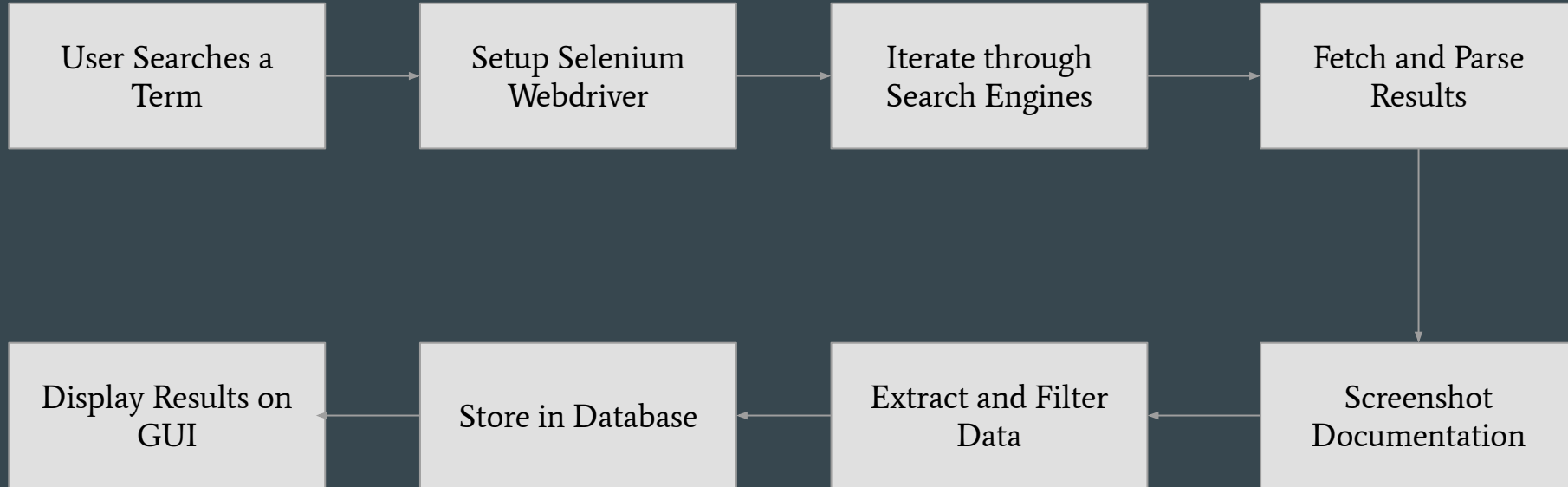
Field	Type	Null	Key	Default	Extra
URL	varchar(767)	NO	PRI	NULL	
Frequency	int	YES		1	
SearchTerm	varchar(255)	YES		NULL	

Table
Result

```
47 -- Table used for the project
48 • select * from searchresults order by Frequency DESC;
49
```

URL	Frequency	SearchTerm
https://health.usnews.com/best-hospitals/pediatric-rankings/cancer	33	Childhood cancer treatment best hospitals USA
https://www.stjude.org/media-resources/news-releases/2023-medicine-scienc...	33	Childhood cancer treatment best hospitals USA
https://www.beckershospitalreview.com/oncology/20-best-hospitals-for-pedi...	28	Childhood cancer treatment best hospitals USA
https://www.stjude.org/about-st-jude/honors-and-awards/best-childrens-ho...	23	Childhood cancer treatment best hospitals USA
https://cancercenters.cancer.gov/cancer-centers/st-jude-childrens-research...	22	Childhood cancer treatment best hospitals USA
https://www.beckershospitalreview.com/oncology/5-best-children-s-hospitals...	17	Childhood cancer treatment best hospitals USA
https://www.cancer.gov/research/infrastructure/cancer-centers/find/stjude	17	Childhood cancer treatment best hospitals USA
https://www.hopkinsmedicine.org/kimmel-cancer-center/cancers-we-treat/pe...	17	Childhood cancer treatment best hospitals USA
https://www.chop.edu/centers-programs/cancer-center	15	Childhood cancer treatment best hospitals USA

Implementation Workflow



Fetch URL function

```
def fetch_urls(searchterm):
    driver = setup_driver()
    search_engines = {
        'google': 'https://www.google.com/search?q=',
        'bing': 'https://www.bing.com/search?q=',
        'yahoo': 'https://search.yahoo.com/search?p=',
        'duckduckgo': 'https://duckduckgo.com/?q=',
        'dogpile': 'https://www.dogpile.com/search?q='
    }

    try:
        for engine, base_url in search_engines.items():
            urls_collected = []
            page = 0
            while len(urls_collected) < 30 and page < 1:
                url = f"{base_url}{searchterm}&start={page * 10}"
                driver.get(url)
                time.sleep(3)

                # Take a screenshot for each engine
                screenshot_path = f"C:/Users/Shradha Godse/Downloads/screenshots/{engine}_{searchterm.replace(' ', '_')}_page_{page}.png"
                driver.save_screenshot(screenshot_path)

                # Wait for the search results to load
                WebDriverWait(driver, 10).until(
                    EC.presence_of_element_located((By.CSS_SELECTOR, 'h3, h2'))
                )

                soup = BeautifulSoup(driver.page_source, 'html.parser')
                # Google and Bing commonly use <h3>, others might use <h2>
                results = soup.find_all(['h3', 'h2'])

                for result in results:
                    link = result.find('a', href=True)
                    if link:
                        href = link['href']
                        # Common URL patterns
                        if 'url?q=' in href or 'search?p=' in href:
                            parsed_url = urlparse(href)
                            href = parse_qs(parsed_url.query).get('q', [None])[0]
                            href = unquote(href) if href else None
                        if href and href not in urls_collected:
                            urls_collected.append(href)
                            if len(urls_collected) >= 30:
                                break
                    page += 1

            for url in urls_collected:
                save_search_results(url, searchterm)

    finally:
        driver.quit()
```


Saving Search Results

```
def save_search_results(url, searchterm, frequency=1):
    conn = connect_database()
    if not conn:
        print("Database connection failed")
        return
    try:
        cursor = conn.cursor()
        query = """
        INSERT INTO searchresults (URL, SearchTerm, Frequency)
        VALUES (%s, %s, %s)
        ON DUPLICATE KEY UPDATE Frequency = Frequency + 1;
        """
        cursor.execute(query, (url, searchterm, frequency))
        conn.commit()
    except mysql.connector.Error as e:
        print(f"Error in database operation: {e}")
    finally:
        if cursor:
            cursor.close()
        if conn:
            conn.close()
```

Display Search Results

```
@app.route('/results.html')
def results():
    search_term = request.args.get('search_term', '')
    conn = connect_database()
    cursor = conn.cursor()
    query = """
    SELECT URL, Frequency FROM searchresults
    WHERE SearchTerm = %s ORDER BY Frequency DESC;
    """

    cursor.execute(query, (search_term,))
    data = cursor.fetchall()
    cursor.close()
    conn.close()
    return render_template('results.html', data=data, search_term=search_term)

if __name__ == "__main__":
    app.run(debug=True, host='0.0.0.0', port=8000)
```

index.html

```
55 </style>
56 .btn:hover {
57     background-color: #007bff; color: white; border: 1px solid #007bff;
58 }
59 .image-container {
60     margin-bottom: 20px;
61 }
62 .image-container img {
63     max-width: 100%;
64     height: auto;
65     border-radius: 10px;
66 }
67 </style>
68 </head>
69 <body>
70 <div class="container">
71 <div class="image-container">
72 
73 </div>
74 <div class="header">Welcome to My Custom Search</div>
75 <form action="/" method="post" class="search-bar">
76 <div class="form-group">
77 <input type="text" class="form-control" id="search" name="search" placeholder="Search..." required>
78 <button type="submit" class="btn">Search</button>
79 </div>
80 </form>
81 </div>
82 </body>
</html>
```

results.html

```
<body>
<div class="container">
<div class="header">

<h1>Search Results for: "Childhood cancer treatment best hospitals USA"</h1>
</div>
<div class="table-responsive">
<table class="table">
<thead>
<tr>
<th style="width: 80%;">List of Hospitals</th>
<th style="width: 20%;">Frequency of Search Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>{% for url, frequency in data %}
<tr>
<td><a href="{{ url }}" target="_blank">{{ url }}</a></td>
<td>{{ frequency }}</td>
</tr>
<tr>
<td colspan="2">{% endfor %}
</td>
</tbody>
</table>
</div>
<a href="/" class="btn btn-primary">New Search</a>
</div>
</body>
</html>
```

GUI Search Page



Welcome to My Custom Search

Search

GUI Results Page

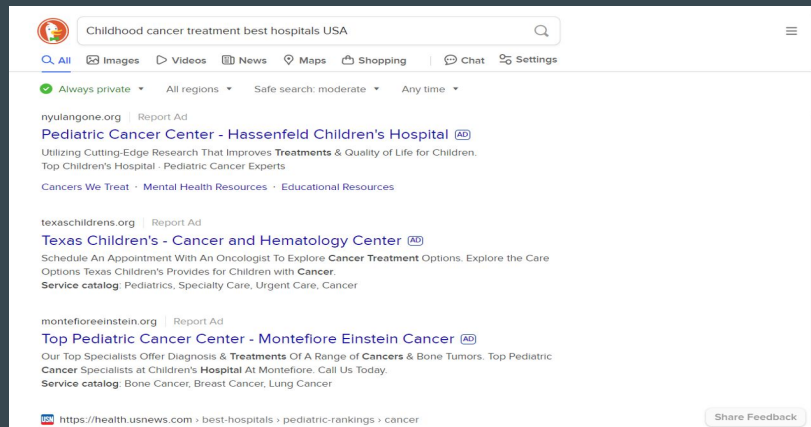
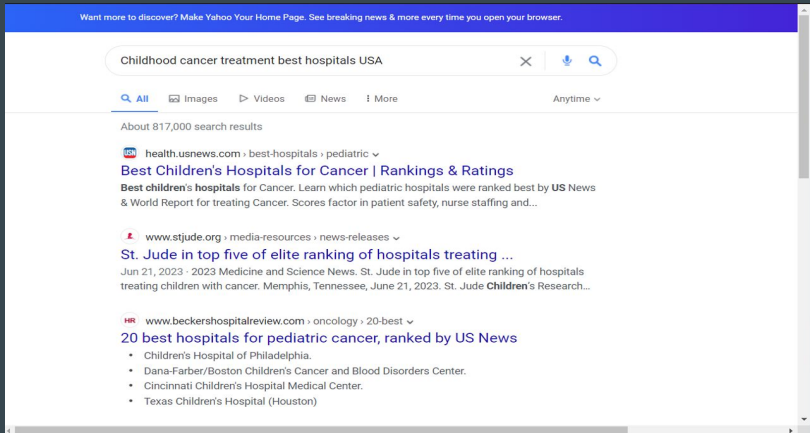
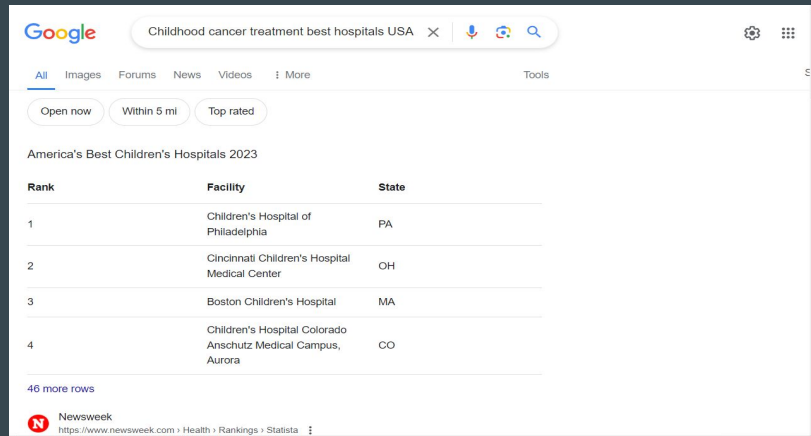
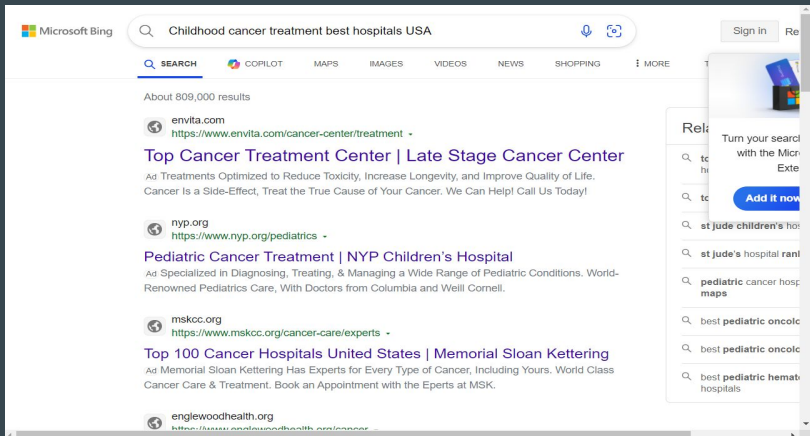


Search Results for: "Childhood cancer treatment best hospitals USA"

List of Hospitals	Frequency of Search Term
https://health.usnews.com/best-hospitals/pediatric-rankings/cancer	33
https://www.stjude.org/media-resources/news-releases/2023-medicine-science-news/st-jude-ranked-in-top-five-hospitals-treating-children-with-cancer.html	33
https://www.beckershospitalreview.com/oncology/20-best-hospitals-for-pediatric-cancer-ranked-by-us-news.html	28
https://www.stjude.org/about-st-jude/honors-and-awards/best-childrens-hospital-for-cancer.html	23
https://www.bing.com/aclk?lId=e86ntdZyLChOvmvqD5oTelzTVUCUwDqQNdoF-UXHifQ8Re_-WoUOSQuQUhf1IC_BldUiivbwCK5AjmLI13EhCakfnbKbqWK07MJzb6Qk5vGTMJr2inYb6pw02Er0VV3CsUyGGfBbDhanZkEd4FnWUzKuAxYubcmzG3EGNsUrwaqwSzLtg2Fhx60Os2Mrs7fMjhF2G-eOA&u=aHR0cHMIM2ElMmYIMmZ3d3cubnlwLm9yZyUyZnBIZGIhdHJpY3MIMmZjYw5jZXItY2FyZSUzZnBrX21IZGI1bSUzZGNwYyUyNnBrX3NvdXJjZSUzZGJpbmclMjZwa19jYW1wYWIubUzZENyb3VkX1NIYXJjaF9CaW5nX05ZRE1BX1BIZGIhdHJpY3NfQ29udmVyc2lVbI9QZWRpYXRyaWNzX05vbi1CcmFuZCUyNmdbGikJTnkYmFiMjNjYmMzMzdmMTcwZDBIMTU1YWY0M2JiZDYxOGUIMjZnY2xzcmlMIM2Qzcc5kcyUyNm1zY2xraWQIM2RiYWlyM2NiYzMzN2YxNzBkMGUxNTVhZjQzYmJkNjE4ZQ&rlid=bab23cbc337f170d0e155af43bbd618e	1
https://www.webmd.com/cancer/choose-pediatric-oncologist	1

New Search

Screenshot Saves using Selenium



Work Division

20%

Web Scraping & DB Setup

- Configure Selenium for automated web browsing in Flask.
- Set up functions to perform searches across search engines.



- Shradha
- Artjola

30%

HTML Parsing & Filtering

- Use BeautifulSoup to parse search result pages.
- Filter out duplicates and advertisements based on defined criteria



- Shradha
- Artjola

30%

Data Retrieval & Display

- Develop methods to fetch stored search results from the database.
- Implement functionality to display results on the GUI.



- Shradha
- Artjola

20%

Testing & Debugging

- Conduct thorough testing to ensure all functionalities work as expected.
- Debug any issues that arise during testing.



- Shradha
- Artjola

Summary

- Efficiently fetches relevant URLs while avoiding duplicates and ads.
- Ensures high-quality search results for user queries.

Benefits:

- Combines database constraints and HTML parsing for effective filtering.
- Delivers clean and relevant search results based on the count of frequency search terms on the GUI Browser.

Demo

