SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution

SUMMER INTERNSHIP-II

(Innovation/Societal/Entrepreneurship-based Internship)

(Academic Batch 2021-25)

Submitted by

by

NAME **SUSHMITHA**
USN **4SF21IS114**
DEPARTMENT **INFORMATION SCIENCE & ENGINEERING**

Under the Mentorship of

Dr. Duddela Sai Prashanth

November 2023

CERTIFICATE

This is to certify that the **SUMMER INTERNSHIP-II (Innovation/Societal/Entrepreneurship)** has been carried out by Ms. Sushmitha bearing the USN 4SF21IS114, bonafide student of Department of Information Science &Engineering , Sahyadri College of Engineering & Management, Adyar, Mangaluru, during the Academic Year 2022-23.

The internship report is verified as per the requirements of the Academic Statute and is recommended for the award of the Academic Credit for the said course.

Mentor	**Head of the Department**
Mr. Vasudev Rao P V	Dr. Mustafa Basthikodi

Name of the Examiner	**Signature with Date**
1.....	.....
2.....	.....

ACKNOWLEDGEMENT

It is with great satisfaction and euphoria that I am submitting a report on internship carried out at "**Sahyadri College of Engineering & Management**" provided by **Coding key LLP** in partial fulfillment of the requirements for the IV Semester of Bachelor of Engineering in Information Science & Engineering. I have completed the project entitled "**Developing a search engine for organization**" during the internship.

I am immensely grateful to our guide, **Dr. Duddela Sai Prashanth**, for his invaluable guidance, timely advice, and unwavering encouragement. His mentorship has been instrumental in our journey, and I sincerely express my heartfelt gratitude.

I express our sincere gratitude to **Dr. Mustafa Basthikodi**, Head & Associate Professor, Department of Information Science & Engineering for his invaluable support and guidance.

I sincerely thank **Dr. Sidramappa Shivanna Injaganeri**, Principal, Sahyadri College of Engineering & Management and Dr. D. L. Prabhakara, Director, Sahyadri Educational Institutions,who have always been a great source of inspiration.

SUSHMITHA(4SF21IS114)

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Week 1

1.1 Introduction

* Understanding search engines is crucial in today's digital landscape. A search engine is tool that helps users find information on the internet by indexing and organizing vas amounts of web content. It employs algorithms to analyze and rank web pages, presenting users with relevant results based on their queries.

1.2 Day 1

* Clearly understanding the concept of search engine, how does it works and in which fields We can apply this and in what way it will be useful for us. And what is its application in Current situation.

1.3 Day 2

* Understanding the concepts of crawling where crawling is the process where search engines use automated bots to Navigate the web and gather information from web pages.

1.4 Day 3

* Clearly understood about the crawling process and applied that crawling Concept to collect the data from a particular website.

1.5 Day 4

* Crawlers, also known as spiders or bots, navigate from one page to another through links, collecting information for indexing. The bots crawl through the link which we provided along with the code and this link can be added through the html using the anchor tag.

1.6 Day 5

* Clear and logical URL structures can facilitate crawling and improve search engine optimization.

* It helps control the access of search engine bots to specific parts of a website.

1.7 Day 6

* Understood Canonicalization concept in crawling where Canonicalization is the process of selecting the preferred URL when multiple URLs point to the same or similar content. Canonical tags help search engines understand which version of a URL to prioritize.

1.8 Conclusion/ skill sets learnt

* web scraping and data extraction stands as a testament to our commitment to excellence. From the initial steps of manual website downloads to the intricacies of implementing a sophisticated web scraping code

Chapter 2

Week II

2.1 Introduction

* Data Collection - Establish mechanisms to collect and update data about organizations, including their names, industries, locations, contact details, leadership, financial data, and articles relevant news

2.2 Day 1

* Understanding the basics of html

Ex: Adding images, Adding URL of a particular website , How the Different tags in html works and understood about the how does we can Make use of html in the concept of search engine.

* HTML plays a crucial role in search engine optimization (SEO) by providing the structure and content of web pages.

2.3 Day 2

* Understanding the basics of CSS .

* Like, the way of styling html elements. Efficient use of CSS can contribute to faster page load times.

* CSS stylesheet provides a responsive and visually appearing layout for a web page ,including Styles for a header,loader,login link,container, search form, search box, search button etc.

2.4 Day 3

* Used a website to collect the data.

First we collected only single data items.

```
    import requests
import re
```

```
url = 'https://sahyadri.edu.in/'

try:
    response = requests.get(url)
    if response.status_code == 200:
        html_content = response.text

        phone_number_pattern = r'\(?\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}'


        phone_numbers = re.findall(phone_number_pattern, html_content)

        for phone_number in phone_numbers:
            print(phone_number)
    else:
        print(f'Failed to retrieve content. Status code:
{response.status_code}')

except requests.exceptions.RequestException as e:
    print(f'Error: {e}')
```

Fig.2.4.1 Code to crawl website and collect all the phone numbers.

2.5 Day 4

```
    import os
import re
import requests
from bs4 import BeautifulSoup
os.makedirs('email', exist_ok=True)
os.makedirs('contact', exist_ok=True)

url = 'https://sahyadri.edu.in/'

response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.text, 'html.parser')


    email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b'
    emails = re.findall(email_pattern, soup.get_text())


    contact_info_pattern = r'\b(?:\d{3}[-.\s]?)?\d{3}[-.\s]?\d{4}\b'  #
Example: Phone numbers
    contact_info = re.findall(contact_info_pattern, soup.get_text())
```

```
    with open('email/emails.txt', 'w') as email_file:
        for email in emails:
            email_file.write(email + '\n')


    with open('contact/contact_info.txt', 'w') as contact_file:
        for info in contact_info:
            contact_file.write(info + '\n')

    print("Email addresses and contact information scraped and saved
successfully.")
else:
    print(f"Error: {response.status_code}")
```

Fig:2.5.1

The following code crawl the given website through the URL specified and give the emails which are present in the website.

2.6 Day 5

```
  import csv
import re
from selenium import webdriver
driver = webdriver.Chrome( )


url = 'https://sahyadri.edu.in/'
driver.get(url)
page_source = driver.page_source

email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,7}\b'
phone_pattern = r'\b\d{3}[-.\s]?\d{3}[-.\s]?\d{4}\b'

emails = re.findall(email_pattern, page_source)
phones = re.findall(phone_pattern, page_source)

driver.quit()


csv_filename = 'info.txt'

with open(csv_filename, mode='w', newline='') as file:
    fieldnames = ['Email', 'Phone']
    writer = csv.DictWriter(file, fieldnames=fieldnames)
```

```
    writer.writeheader()

    for email in emails:
        writer.writerow({'Email': email})

    for phone in phones:
        writer.writerow({'Phone': phone})

print(f'Email addresses and phone numbers scraped and stored in
{csv_filename}')
```

Fig:2.6.1

2.8 Conclusion/ skill sets learnt

* Understanding the codes which will scan the website and gives us the required information.

Chapter 3

Week III

3.1 Introduction:

* Embarking on a week-long exploration, our focus intertwined the realms of thoughtful project design, dynamic user interfaces, and efficient database integration. Day 1 saw the conception of a project blueprint - a dual-page system catering to both user and admin interactions.

3.2 Day 1:

* create 2 pages- user page and admin page. User page is the one where the user can enter the url to be scraped. As soon as user clicks search button the informations(phone number, email, social media links) will be fetched directly from the database so that user no need to wait to get the info.

3.3 Day 2:

* Created Admin page is the one where admin will be able to add a new url . As soon as the admin adds a new url , that particular url has to be scraped and sould save the info in the database.

- Also admin can update the database and fix number of days for a update and if it is not updated it will automatically will give a signal.

3.3 Day 3:

* Created a User Interface in which that contains a enter url which is used to enter the new url and scrap button which will scrape the website and store all the scraped data's in the database. Whenever the user entered the url which is already scraped the bot will automatically give the data which is stored in the database.

3.4 Day 4:

* Integrating an admin panel for efficient project management. Admin panel seamlessly integrated with the Django database, emphasizing the importance of user-friendly interfaces.

3.5 Day 5:

* Enhanced admin side functionalities in which admin can update and Admin can view all the statistics which contains how many links have been scraped and how many users are there and how many data's has been scraped. And also admin can have the ability to update the database.

3.7 Conclusion/ skill sets learnt

- As the week concludes, our journey through project development reflects a harmonious convergence of design ingenuity and technical prowess.
- The successful integration of an admin panel, streamlined database operations
- The functionalities of the admin and the working of User Interface.

Chapter 4

Week IV

4.1 Introduction:

***** Django is a back-end server side web framework.Django is free, open source and written in Python. Django takes care of the difficult stuff so that you can concentrate on building your web applications.

- Django emphasizes reusability of components, also referred to and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

4.2 Day 1:

* In Django we created migration file which is used to manage database Schema , we have imported all the migration models. One list is defined and there we created three models Email, Phone number, sites and url Link. Also by using this migration file we can add and remove the fields.

4.3 Day 2:

* We have created another migration file which will be dependent upon the previous migration files which will specified in the dependencies. Here in this file a metadata model is created which contains two fields .

- Total count: An integer field representing some count or quantity.
- Name: Each record in the metadata table will be uniquely identified by its name.

4.4 Day 3:

* We imported Django.contrib model for configuring the admin interface. This admin model allows for managing the data in the models. Also we need to import all the models which is defined before and we need to register with the admin interface. After registering we can login to the admin interface and find this models listed for management. Now we can implement CRUD operations.

4.5 Day 4:

* We created all the models like email, phone number, links, sites Count etc.

4.6 Day 5:

* We created a login page which contains user information ensuring that Only authenticated users can access it. Here when user enters the URL it Will check if the site information is already is in the database and either retrieves the scraped data and saves new data.

4.7 Day 6:

* There is a another view which is a result view which handles database Operations related to the metadata model. It increments the totlacount field and save the changes.

4.7 Conclusions/skills learnt:

* Django provides a robust and powerful database system that simplifies the interaction with databases in web applications.

- Django models define the structure of the database tables. Models can include various Types of fields ,such as CharField, IntegerField, ForiegnKey, etc., which map to the Corresponding database column types. This abstractions allows developers to focus On the application's logic rather than database details.

REFERENCES

- https://www.geeksforgeeks.org/implementing-web-scraping-python-beautiful-soup/
- https://realpython.com/web-scraping-with-python/
- https://djangoforbeginners.com/database-setup/
- https://docs.djangoproject.com/en/stable/topics/migrations/