TECH JOB PORTAL BY SCRAPING INDEED AND GLASSDOOR

-Krishna Gowsalya M

INTRODUCTION

In the modern employment landscape, countless job listings are posted daily across various websites. Navigating through these listings to find relevant tech jobs can be time-consuming. Traditional job portals often include roles from all industries, making it difficult for developers and IT professionals to locate opportunities that match their skills.

The Tech Job Portal addresses this issue by focusing exclusively on technology-based jobs scraped directly from Indeed. Using automated scripts, it extracts job titles, company names, locations, and application links. The data is then displayed in a clean, user-friendly web interface built with Flask.

The portal provides an efficient and organized way to explore tech-related opportunities without the distractions of unrelated job listings.

OBJECTIVES

- To create a web portal dedicated to technology-related job listings.
- To automatically extract job postings from Indeed using web scraping.
- To develop a simple user system with login and registration features.
- To use a database (SQLite) for storing and retrieving job data.
- To display job details (title, company, location, and apply link) dynamically.
- To provide an easy-to-navigate interface for users to explore jobs.
- To simulate real-time job fetching for demonstration purposes.

LIBRARIES USED

The Tech Job Portal project utilizes several important Python libraries to perform web scraping, data storage, and web application development efficiently. Each library plays a specific role in the functioning of the system.

1.Flask

Flask is a lightweight and flexible web framework used to build the backend of the project. It handles routing, template rendering, and user session management. Flask makes it easy to connect the web interface with the Python logic and database.

2.Requests

The Requests library is used to send HTTP requests to the Indeed website. It allows the scraper to fetch the HTML content of job listing pages easily and reliably.

3.Beautifulsoup

BeautifulSoup, part of the bs4 module, is used for parsing and extracting job-related data from the HTML content fetched from Indeed. It helps identify specific HTML tags containing job titles, companies, locations, and links.

SYSTEM REQUIREMENTS

Hardware Requirements

Processor: Intel Core i3 or higher RAM:

Minimum 4 GB Hard Disk: Minimum

500 MB free space Display: Standard 15-

inch or higher

Software Requirements

- Programming Language: Python

Software Requirements

Programming Language: Python

Framework: Flask

Libraries Used: BeautifulSoup, Requests,SQLite3

Tools: Visual Studio Code

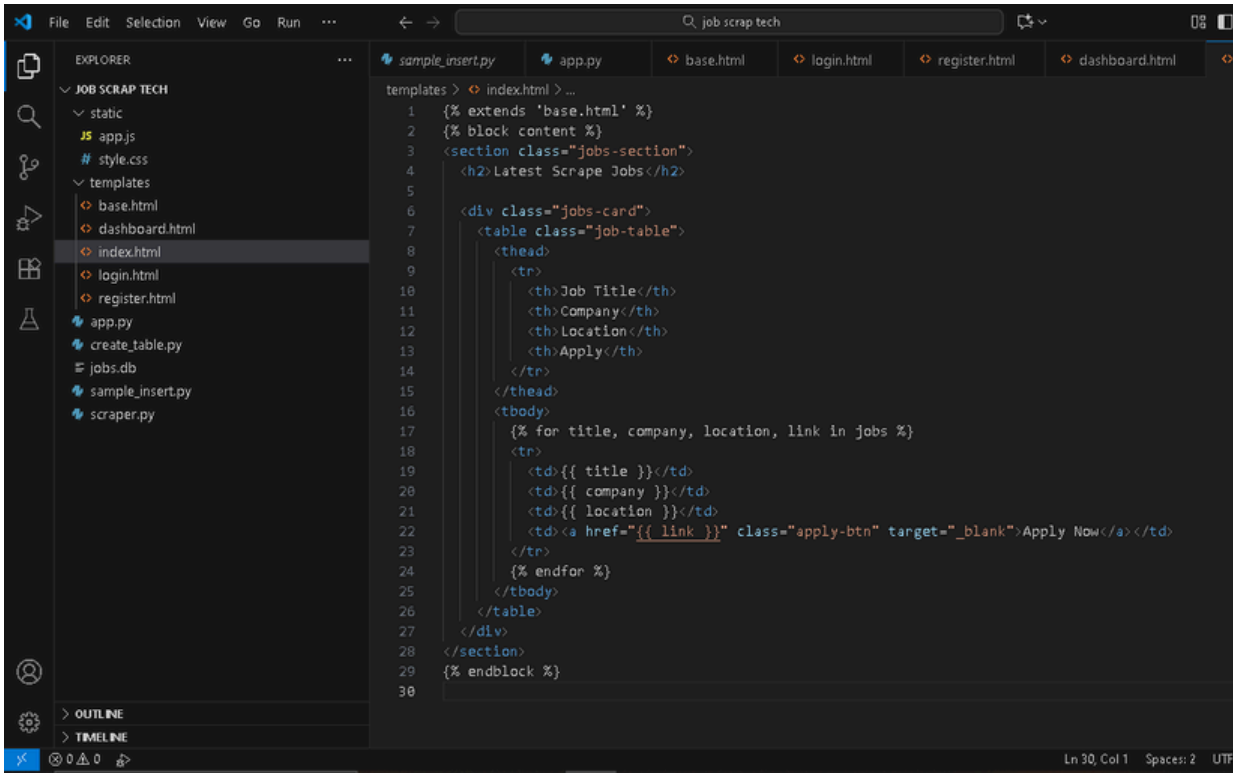
SYSTEM DESIGN

The project is structured using a three-tier architecture:

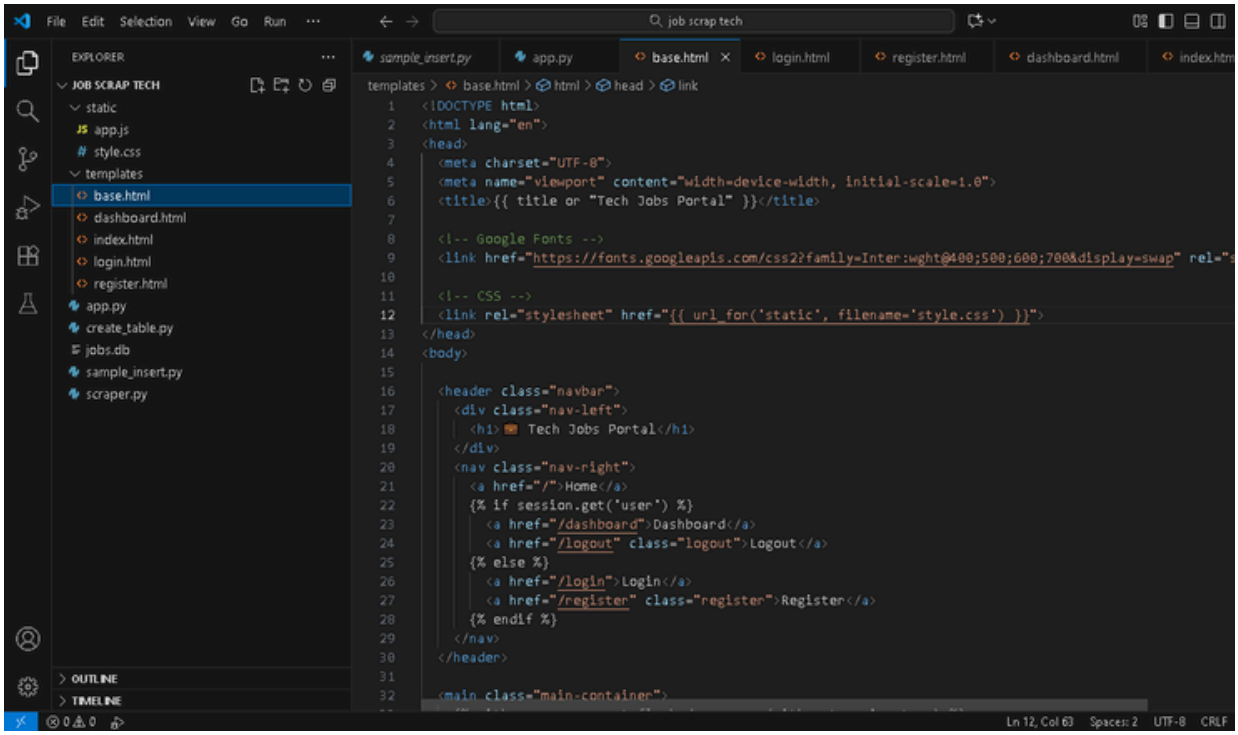
Frontend:

Handles user interaction using HTML, CSS, and JavaScript. Provides forms for login, registration, and job search.

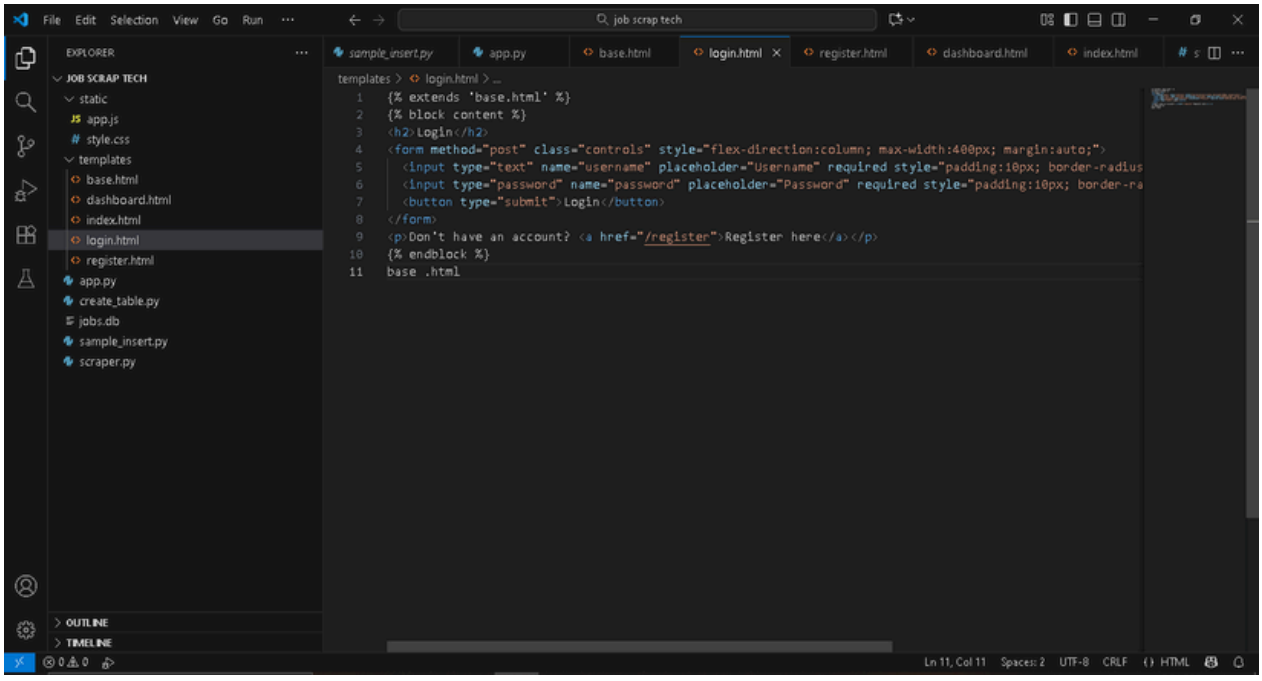
Index.html



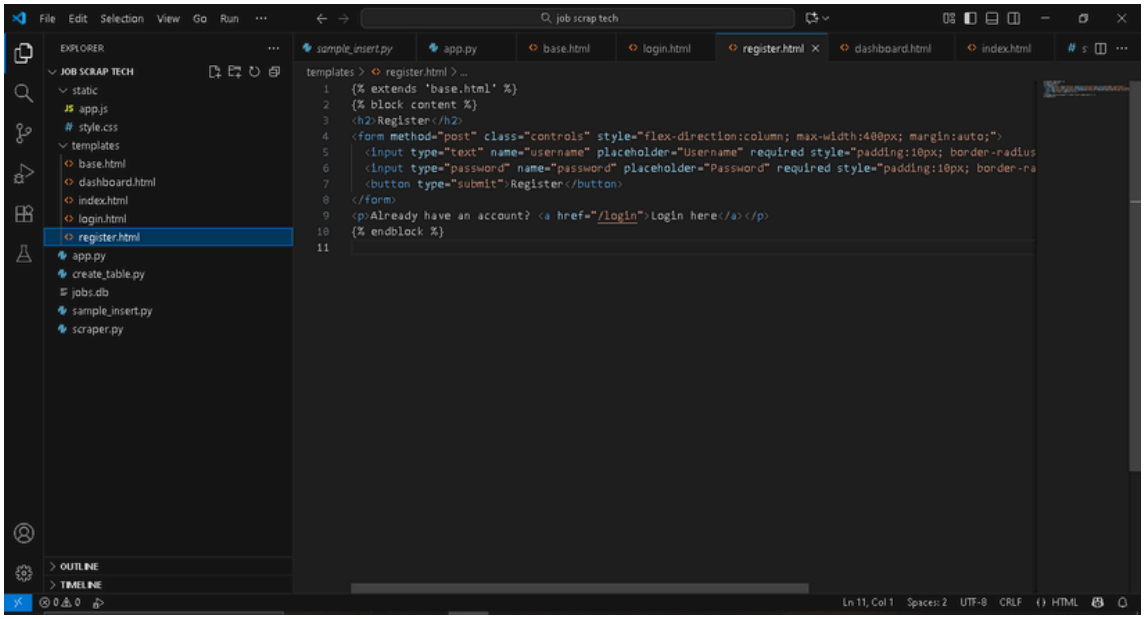
base .html



login.html



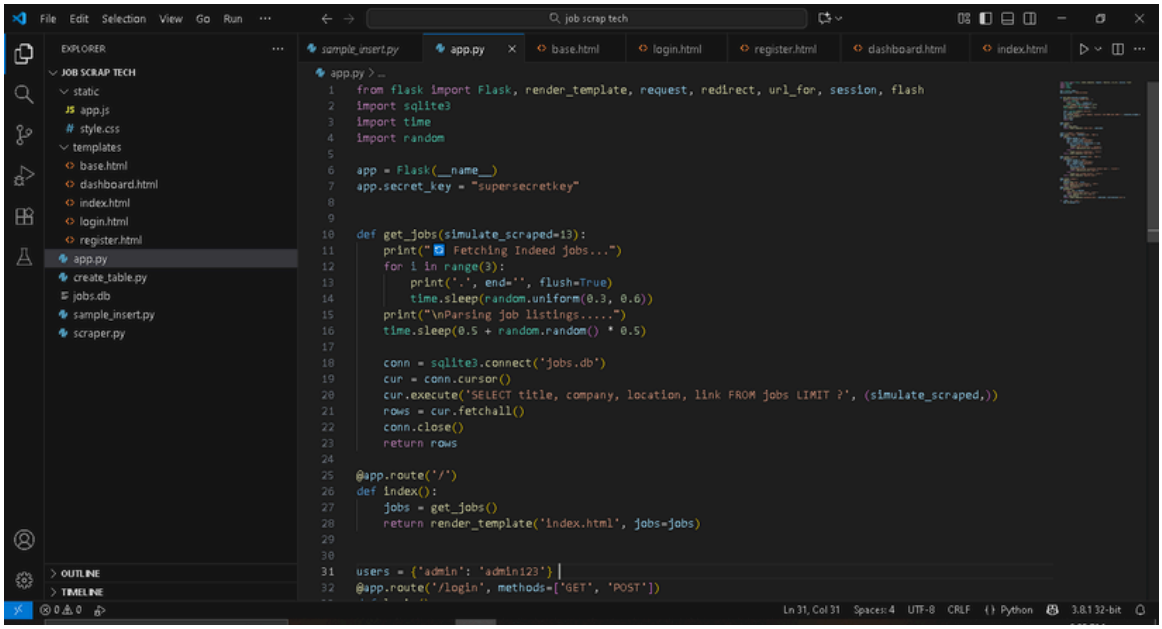
Register.html



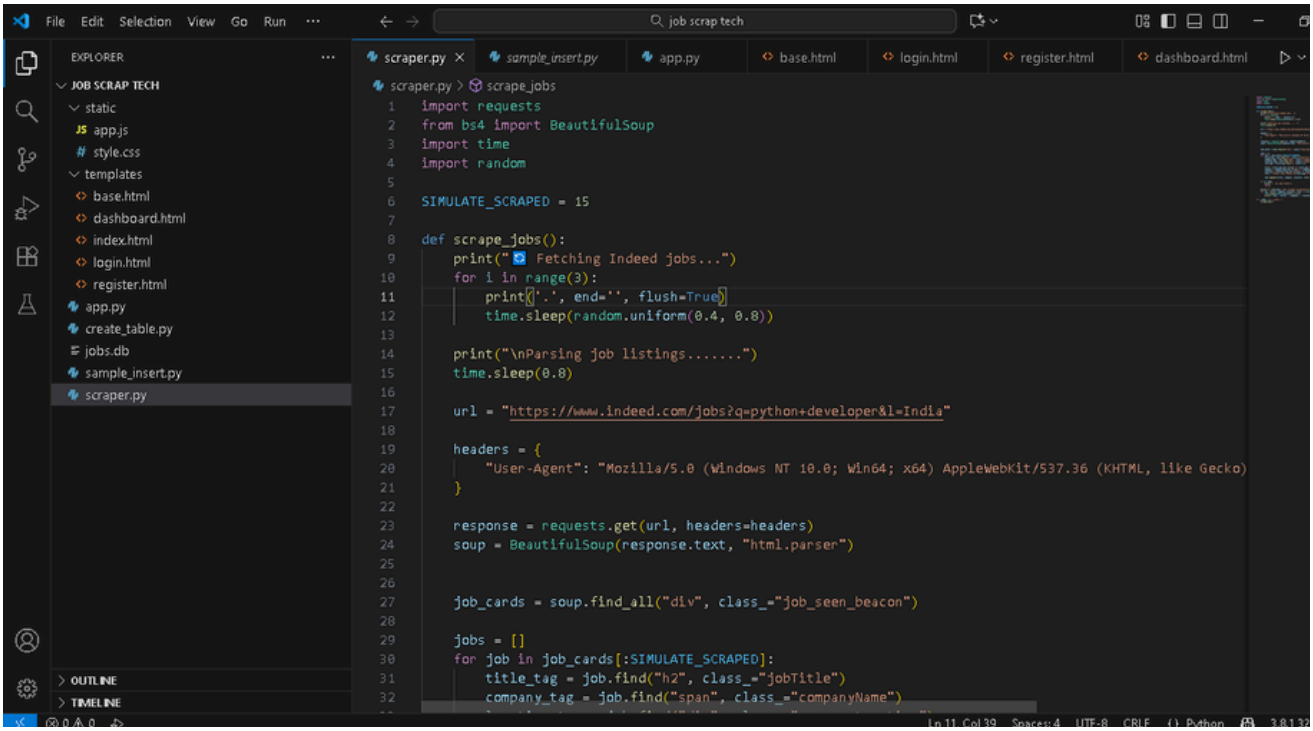
Backend

Manages routes, server-side logic, and data processing. Handles web scraping requests, job search operations, and communication with the database.

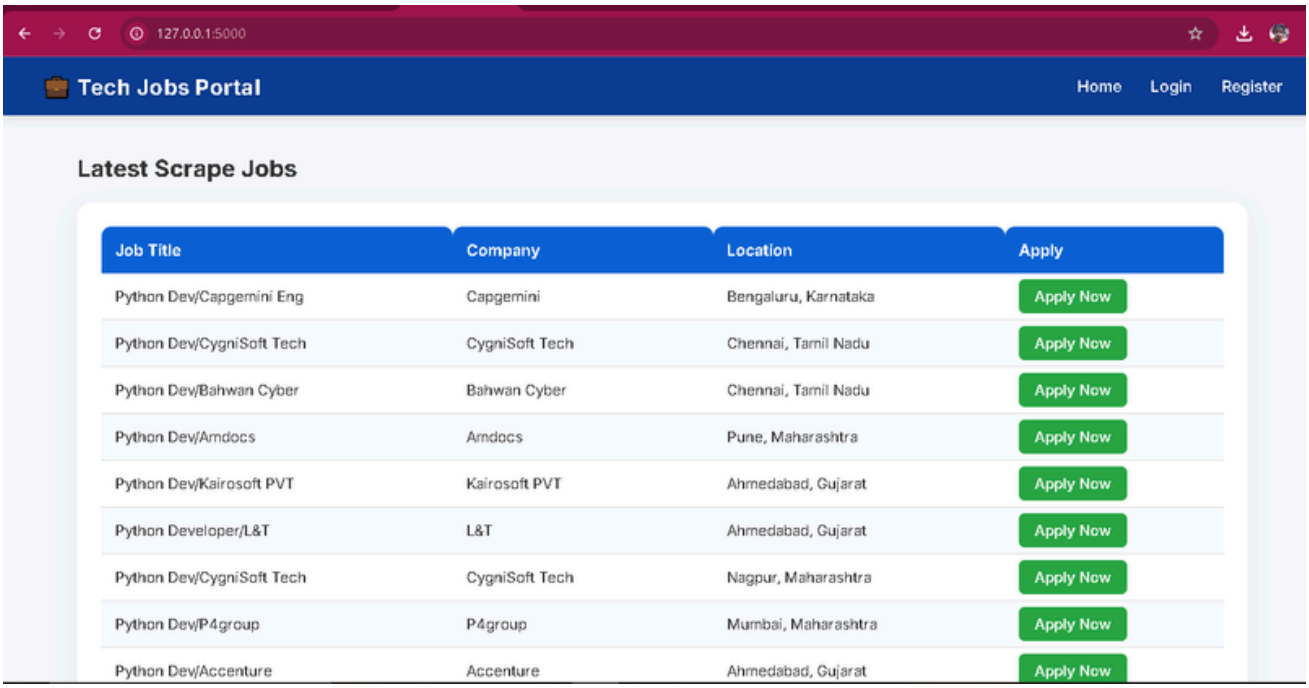
app.py



Scraper.py



output



IMPLEMENTATION

The project was implemented in several stages:

1. Environment Setup:

   Installed Python, Flask, and required libraries. The project structure includes separate folders for templates, static files, and scripts.

2. Web Scraping:

   The scraper script uses BeautifulSoup to extract job details such as title, company, location, and link from Indeed.

3. Database Integration:

   SQLite was used to store and retrieve job listings efficiently.

4. User Interface:

   The frontend pages were built using HTML and CSS, providing a simple and intuitive layout. Job listings were displayed in a card-like format, making them easy to browse.

CONCLUSION

The Tech Job Portal by Scraping Indeed effectively demonstrates how web scraping and Flask can be combined to create an automated job search platform. It reduces manual effort, ensures data relevance, and presents job listings in a user-friendly interface.
This project achieves its goal of helping users access the latest tech job opportunities efficiently, using automation and lightweight technologies like Python and SQLite.

FUTURE ENHANCEMENTS

To improve and expand the platform further, the following enhancements can be implemented:

- Add job filtering options (by title, location, or company).
- Include email alerts for new job updates.
- Implement resume upload and profile matching.
- Introduce an admin panel for job management.
- Expand scraping to include other platforms like LinkedIn or Glassdoor.
- Create a mobile version for easier access.

REFERENCES

https://www.indeed.com

https://www.glassdoor.com

https://flask.palletsprojects.com

https://docs.python.org

https://beautiful-soup-4.readthedocs.io