**A**

**Project Report**

**On**

**Web Scraper**

**Submitted by**

***Shivam Yadav (2205101130097)***

***Shubham Panwar (2205101130099)***

**as**

**Partial fulfilment of Semester VI**

**of Bachelors of Computer Applications**

**for A.Y. 2024-2025**

**Under the Guidance of**

**Prof. Saswati Chatterjee**

**Submitted To**

**Parul Institute of Computer Application,**

**Faculty of IT & Computer Science**

**Parul University**



**Acknowledgement**

*The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.*

*We respect and thank Dr. Priya Swaminarayan, Dean, FITCS for providing us an opportunity to do the project work in BCA and giving us all support and guidance, which made us complete the project duly. We are extremely thankful to Mam for providing her support and guidance, although she had busy schedule managing the academic affairs.*

*We would not forget to remember Dr. Hina Chokshi, HOD, BCA department for her encouragement and more over for her timely support and guidance till the completion of our project work.*

*We owe our deep gratitude to our project guide Prof. Saswati Chatterjee, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.*

*We are thankful to and fortunate enough to get constant encouragement, support and guidance from our Parents, all Teaching staffs of BCA Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in laboratory for their timely support.*

***Shivam Yadav (2205101130097)***

***Shubham Panwar (2205101130099)***



PARUL INSTITUTE OF COMPUTER APPLICATION

**CERTIFICATE**

This is to certify that ***Shubham Panwar, Shivam Yadav*** the students of Parul Institute of Computer Application, has/have satisfactorily completed the project entitled “***Web Scraper****”* as a part of course curriculum in BCA semester-VI for the academic year 2024-2025 under guidance of ***Prof. Saswati Chatterjee.***

Enrollment Number: 2005101130097

Enrollment Number: 2005101130099

|  |  |  |
| --- | --- | --- |
| **Quality of work** | **Grade** | **Sign of Internal guide** |
| **Poor / Average /**  **Good /**  **Excellent** | **B /B+ / A / A+** |  |

Date of submission:

HOD, Principal,

**Dr. Hina Chokshi** **Dr. Priya Swaminarayan**

**INDEX**

|  |  |
| --- | --- |
| **Content** | **Page No.** |
| 1. Research | 1 |
| 1. Feasibility Studies | 2 |
| * 1. Technical Feasibility | 2 |
| * 1. Economic Feasibility | 3 |
| * 1. Operational Feasibility | 3 |
| 1. System Requirement Specification | 4 |
| * 1. Introduction to SRS | 4 |
| * 1. Abstract | 4 |
| * 1. System Users | 4 |
| * 1. Modules | 5 |
| * 1. Modules Description | 5 |
| * 1. Hardware / Software Requirement | 6 |
| * 1. Flow Chart | 7 |
| * 1. TimeLine Chart | 8 |
| 1. Technology Description | 9 |
| * 1. Features and Limitations of New System | 9 |
| 1. Data Flow Diagram | 10 |
| * 1. Context Level DFD’s | 10 |
| * 1. Level 1 DFD’s | 10 |
| * 1. Level 2 DFD’s | 11 |
| 1. Use Case Diagram | 12 |
| 1. System Architecture | 13 |
| 1. Activity Diagram | 14 |
| * 1. Description of Activity Diagram | 14 |
| 1. Sequence Diagram | 15 |
| * 1. Sequence Diagram | 15 |
| 1. Data Dictionary | 16 |
| * 1. Description of Data Dictionary | 16 |
| 1. Form Design (Screenshots Phase 1, 2, 3, 4) | 17 |
| 1. What is Testing? | 20 |
| * 1. Importance and Types of Testing | 21 |
| 1. Future Enhancement | 22 |
| 1. References and Bibliography | 23 |

1. **Research**
   1. **What is research?**

In a project, research refers to the systematic and systematic investigation of a specific topic or problem in order to gather information and understand the subject matter more fully. This research may be conducted in order to inform the development of a new product or service, to improve an existing one, or to solve a specific problem or challenge that the project team is facing. Research in a project context may involve a range of activities, including the collection and analysis of data, the review of existing literature or research on the topic, and the development of hypotheses or research questions that can be tested or explored further. The goal of research in a project is to inform and guide the decision-making and problem-solving processes of the project team, and to provide a strong foundation for the development of the project deliverable.

* 1. **Types of Research Methodology**
* Experimental research: This involves conducting controlled experiments in order to test hypotheses and gather data.
* Observational research: This involves observing and recording the behavior of subjects without manipulating variables.
* Survey research: This involves collecting data through surveys, either online or in person.
* Case study research: This involves in-depth analysis of a single case or a small number of cases in order to understand a particular phenomenon.
* Focus group research: This involves gathering a small, diverse group of people together to discuss a specific topic or product.
* Ethnographic research: This involves studying and observing people in their natural environments in order to understand their behaviors, beliefs, and values.
* Grounded theory: This involves collecting and analyzing data in order to develop a theory about a specific phenomenon.

1. **Feasibility Studies**

**What is Feasibility?**

A feasibility study is simply an assessment of the practicality of a proposed project plan or method. This is done by analysing technical, economic, legal, operational and time feasibility factors. Just as the name implies, you’re asking, “Is this feasible?” For example, do you have or can you create the technology to do what you propose? Do you have the people, tools and the resources necessary? And, will the project get you the ROI you expect?

* 1. **Technical Feasibility**

Technical feasibility evaluates the technical complexity of the expert system and often involves determining whether the expert system can be implemented with state-of-the-art techniques and tools.

* 1. **Economic Feasibility**

A feasibility study may also include an economic analysis of the project. The purpose of economic analysis is to determine whether there is an economic case for the investment decision. This assessment goes beyond the items typically included in a financial analysis.

* 1. **Operational Feasibility**

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

* 1. **Importance of Feasibility Studies**

Feasibility studies can identify the logistical, financial, and market challenges of a proposed project by evaluating:

* What the estimate would be to fund the project
* When the potential business will offer a return on investment
* The market for the proposed product or service
  1. **Feasibility Study of our Proposed System**
     1. **Technical Feasibility:**

The system is built on mature, open-source technologies (Python, Flask/Django, BeautifulSoup, Scrapy) and incorporates modern features such as a visual element selector, automated scheduling, and proxy support to handle anti-scraping measures. These factors ensure a robust and scalable technical foundation.

* + 1. **Economical Feasibility:**

Leveraging open-source components and scalable cloud infrastructure keeps initial costs low. The automation of scraping and data export processes reduces labor expenses and increases productivity, making the project economically viable even when considering future legal and maintenance updates.

* + 1. **Operational Feasibility:**

With an intuitive interface, drag-and-click element selection, automated task management, and secure user authentication, the system is designed for ease of use by both beginners and professionals. This user-centric design supports efficient daily operations and minimizes manual intervention.

1. **System Requirement Specification**
   1. **Introduction To SRS**
      1. **What is SRS?**

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

* + 1. **Need of SRS**

In order to fully understand one’s project, it is very important that they come up with a SRS listing out their requirements, how are they going to meet it and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.

* 1. **Abstract**

The proposed system is a Web Scraper—a powerful, user-friendly tool that enables data extraction from any website without requiring coding knowledge. Key features include:

* **Visual Element Selection:** A point-and-click interface for selecting webpage elements with real-time previews.
* **Automated Scheduling:** Users can schedule scraping tasks at custom intervals with email notifications.
* **Flexible Data Export:** Supports exporting data in multiple formats (CSV, JSON, Excel) and can integrate directly with databases.
* **Task Management:** Maintains a history of scraping tasks with features for task replication and result archiving.
* **Secure Access:** Implements robust authentication and encrypted data storage.

This tool is designed for both beginners and professionals, aiming to streamline data extraction while ensuring high performance and security.

* 1. **System User**
* Admin
* User
* **Description of User Role**

1. **Admin**

Admin can manage user accounts, review websites, configure parameters, update system settings, and view system logs.

1. **User**

Users can register, login, submit URLs for web scraper, view Scraper results, update profiles, and logout.

* 1. **Modules**
* Admin Module
* User Module
  1. **Modules Description**
* **Admin Module:**

This module enables system administrators to oversee and maintain the overall health and security of the Web Scraper system. Admins can manage users, review system activities, configure global settings, and monitor task performance.

* **Review Scraping Tasks:**

Allows the admin to review all scheduled, running, and completed scraping tasks. Admins can inspect task details, verify extraction accuracy, and address any errors or anomalies detected during scraping.

* **Manage User Accounts:**

Enables the admin to view, approve, update, or suspend user accounts. This submodule ensures that only authorized users can access the system and that account data is kept up to date.

* **Configure System Settings:**

Provides options to adjust global parameters such as scraping intervals, export formats, and proxy settings. Admins can update thresholds and rules to optimize performance and adapt to changing requirements.

* **Monitor System Logs:**

Offers a comprehensive view of system activities, including login attempts, task submissions, and error logs. This submodule helps ensure system security and facilitates performance tuning by providing detailed diagnostics.

* **User Module:**

This module empowers end users to interact with the Web Scraper for data extraction purposes without requiring coding expertise. Users can schedule tasks, visually select webpage elements, and access the extracted data.

* **User Registration:**

Allows new users to register and existing users to log in securely. This submodule ensures that access to the scraping tools is restricted to authorized individuals.

* **Create and Configure Scraping Tasks:**

Enables users to input target website URLs, use a drag-and-click interface to select webpage elements, and set scheduling options (e.g., one-time or recurring tasks).

* **View and Download Scraping Results:**

Provides a dashboard for users to review the results of their scraping tasks. Users can view task histories, inspect extracted data, and download results in multiple formats (CSV, JSON, Excel).

* 1. **Hardware / Software Requirements :-**

|  |  |
| --- | --- |
| **Name of Components** | **Specification** |
| Processor | Intel core I3, I5 |
| RAM | 4GB/\*GB |
| Hard Disk | 512GB/1TB |

* **Hardware Requirement:**

|  |  |
| --- | --- |
| **Name of Components** | **Specification** |
| Operating System | Window XP, Window 7and above |
| Software development Kit | Google Chrome and any suitable browser |
| Tools & languages | Python and HTML |

* **Software Requirement:**
  1. **Flow Chart**

Figure 3.7.1. Flow Chart



* 1. **Time Line Chart**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Development phase** | ***75 Days*** | | | | | | **Duration**  **N**  **(days)** |
| 0to10 days | 11to20  days | 21to30 days | 31to40 days | 41to50 days | 51to75  days |
| **Requirement**  **Gathering** |  |  |  |  |  |  | 07 |
| **Analysis** |  |  |  |  |  |  | 09 |
| **Design** |  |  |  |  |  |  | 10 |
| **Development Phase 1** |  |  |  |  |  |  | 13 |
| **Development Phase 2** |  |  |  |  |  |  | 13 |
| **Development Phase 3** |  |  |  |  |  |  | 13 |
| **Documentation** |  |  |  |  |  |  | 10 |
| **Total time**  **(Days)** |  | | | | | | **75** |

Figure 3.8. Time Line Chart

1. **Technology Description**
   1. **Features and Limitations of New System**

|  |  |
| --- | --- |
| **Existing System** | **New System** |
| More manual hours are needed for Scraper the data. | Avoids more manual hours that need to Scraper the data. |
| Need lots of man power and time | Not need much man power and time |
| There is compulsory need in physical presence in the time to copy and paste data on File. | The User are allowed to scraper online at any time and they can even export the data in any type by sitting at home. |

* 1. **Challenges in Web Scraping**
     1. **Technical Challenges:**
        1. **Handling Dynamic Content** – Many websites load data dynamically using JavaScript, requiring tools like Selenium or Puppeteer.
        2. **CAPTCHA and Anti-Scraping Measur**es – Websites implement CAPTCHAs, IP blocking, and bot detection to prevent automated access.
        3. **Website Structure Variabili**ty – Different sites have unique layouts, requiring customized scraping logic.
        4. **Rate Limits and Throttling** – Frequent requests can trigger IP bans, necessitating request delays or proxy rotation.
        5. **Data Cleaning and Processing** – Extracted data may be unstructured or contain inconsistencies that require preprocessing.
     2. **Ethical & Legal Concerns:**
        1. **Data Ownership and Privacy** – Scraping personal or proprietary data without permission can raise ethical and legal issues.
        2. **Website Terms of Service Compliance** – Many sites prohibit scraping, making it essential to review their policies before proceeding.
        3. **Fair Use and Copyright Laws** – Some data may be protected by copyright or fair use restrictions.
        4. **Impact on Website Performance** – High-frequency scraping can strain servers, potentially disrupting a website’s normal operation.

1. **Data Flow Diagram**
   1. **Context Level DFD’s**

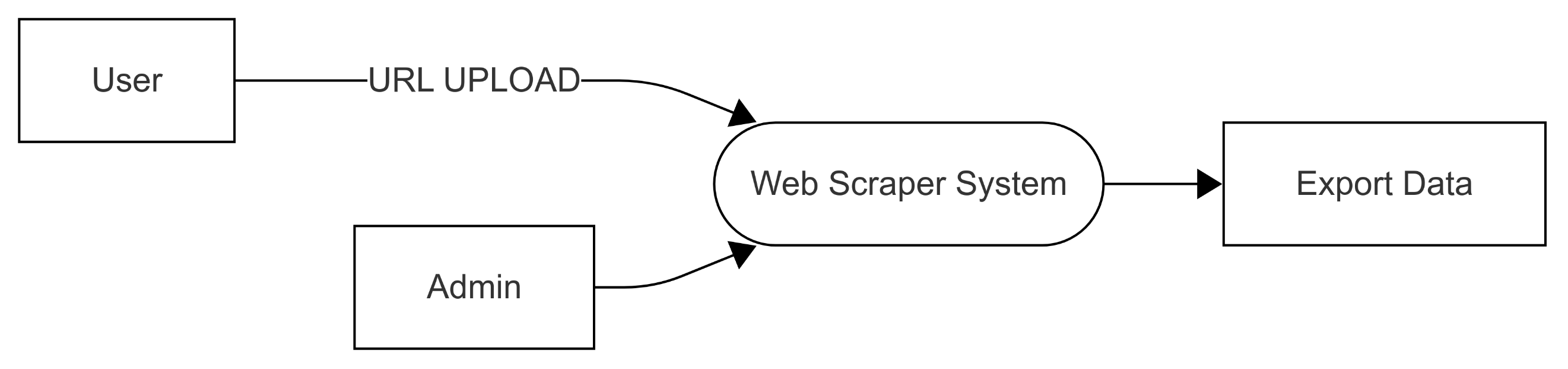


Figure 5.1.1. Context Level DFD: 0 Level

5.1.2. **Description of Context Level DFD: 0 Level**

At the highest level, the Web Scraper System interacts with two external entities: the User (who creates and monitors scraping tasks), the Admin (who manages accounts, settings, and logs).

* 1. **Level 1 DFD’s:**
* **1 level DFD : -**

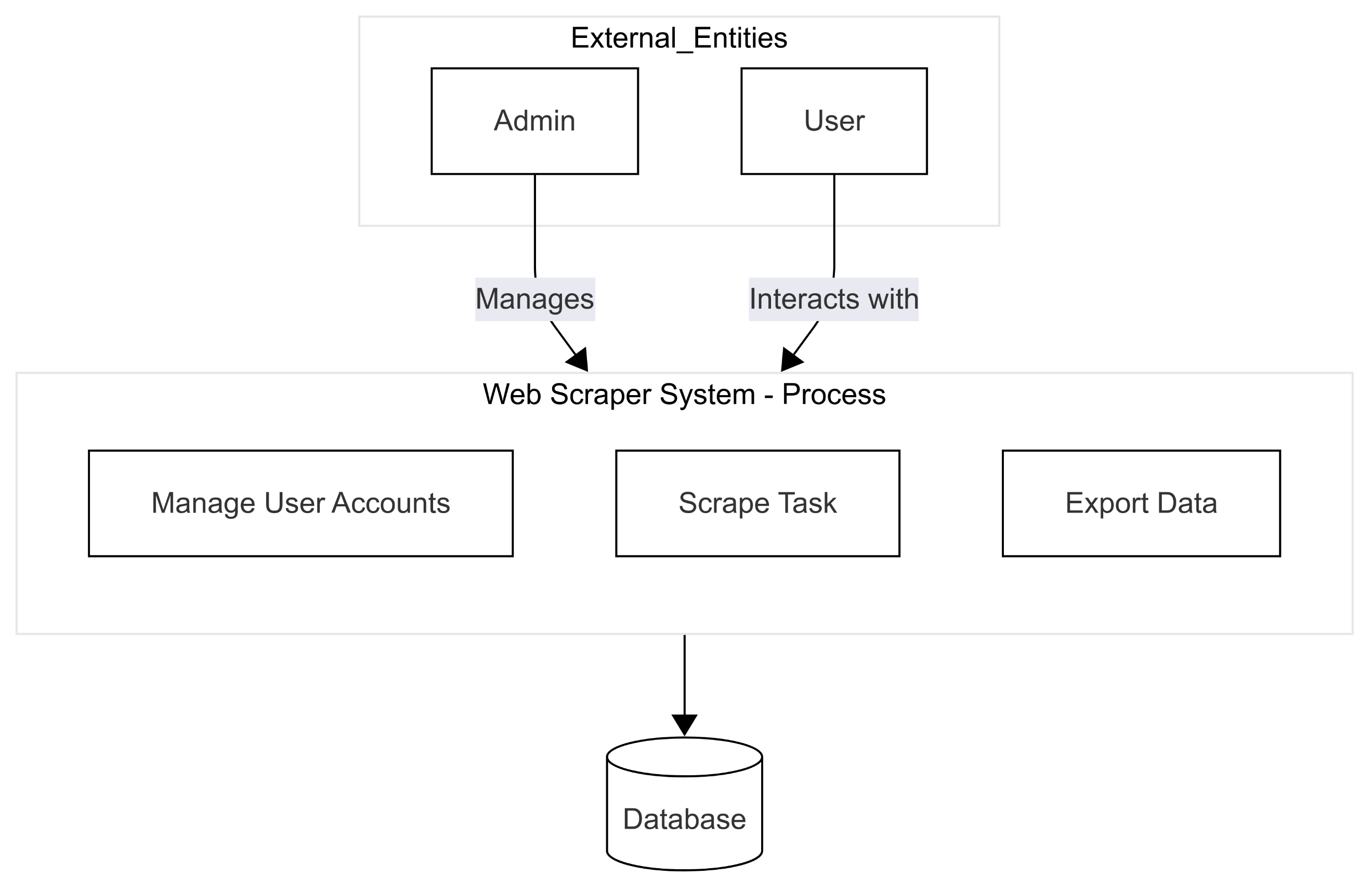


Figure 5.2.1 1st level DFD

**5.2. Description of Context Level DFD: 1 Level**

This diagram decomposes the system (Process 0) into three main sub-processes:

* + Manage User Accounts (handled primarily by Admins),
  + Scrape Task (core data extraction process),
  + Export Data (formatting and delivering results).  
    All these processes interact with the database.
  1. **Level 2 DFD’s**
* **2nd Level DFD : -**

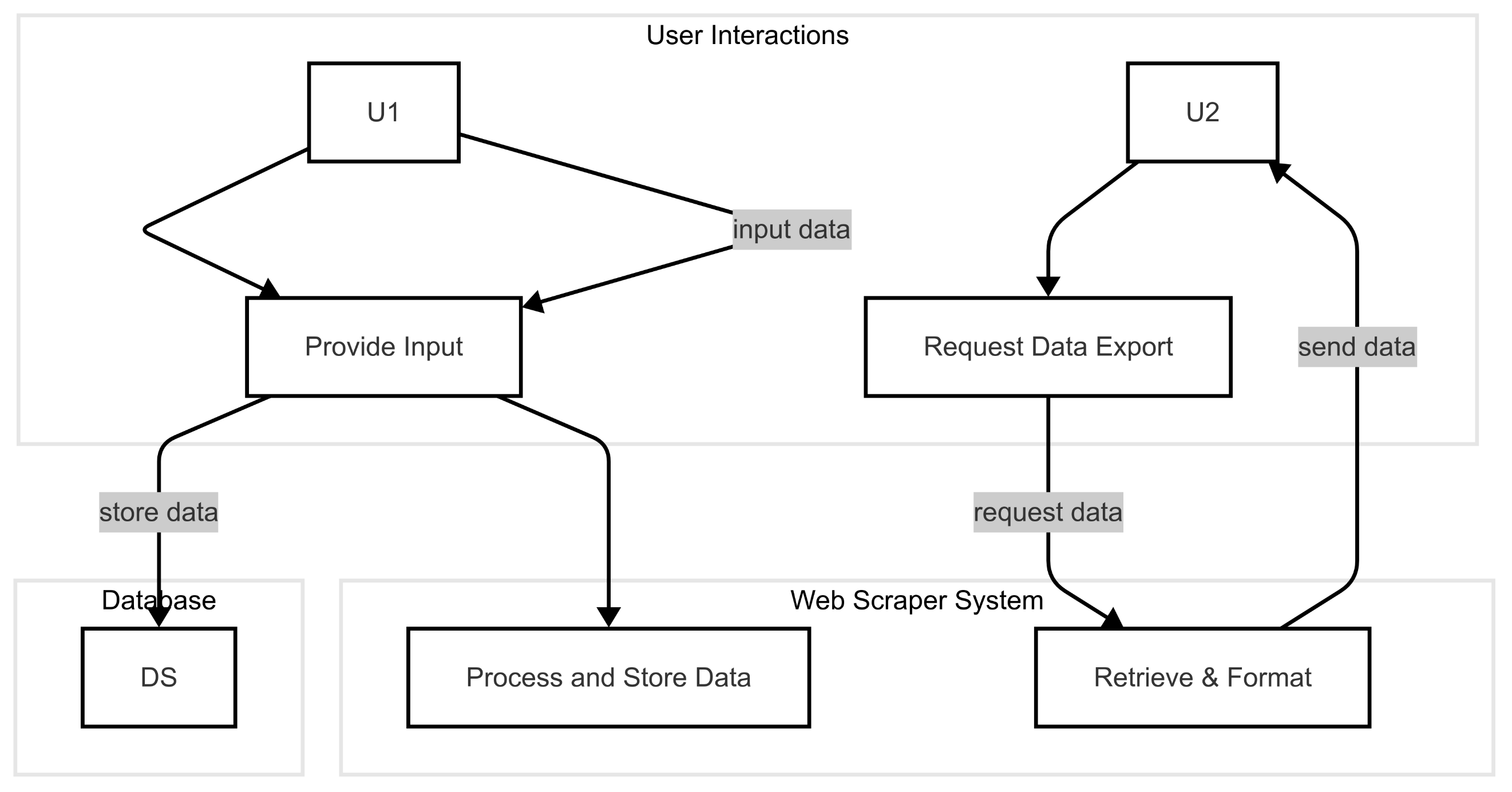
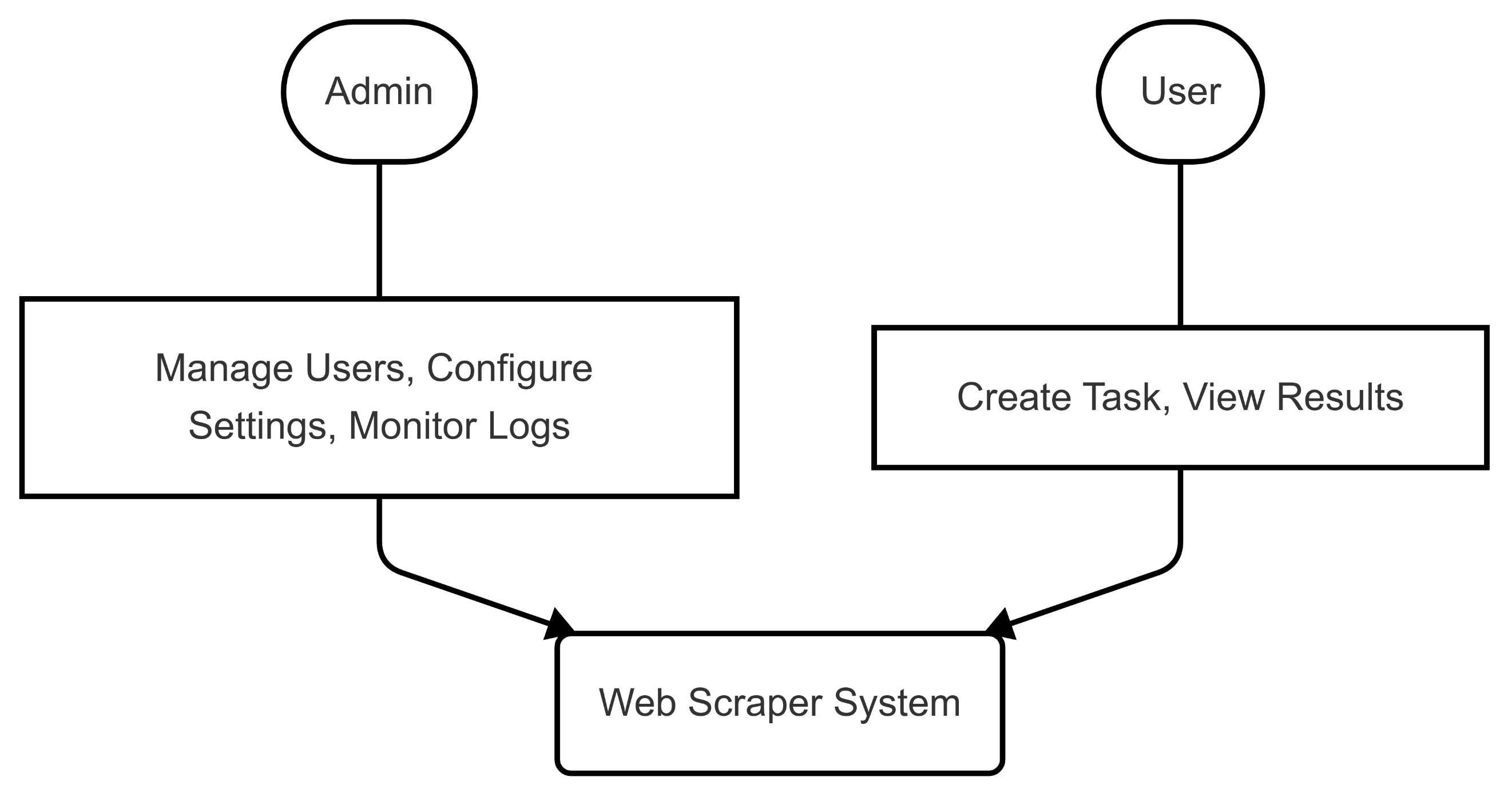


Figure 5.3.1 2nd level DFD

1. **Use Case Diagram**



1. Form Design (Screenshots Phase 1, 2, 3, 4 and Validation’s Screenshots)

Figure 6. Use Case Diagram

* 1. **Description of Use Case Diagram**

This diagram shows the primary actors and their interactions with the system:

* Admin: Manages users, configures settings, and monitors logs.
* User: Creates and manages scraping tasks and views results.

1. **System Architecture**

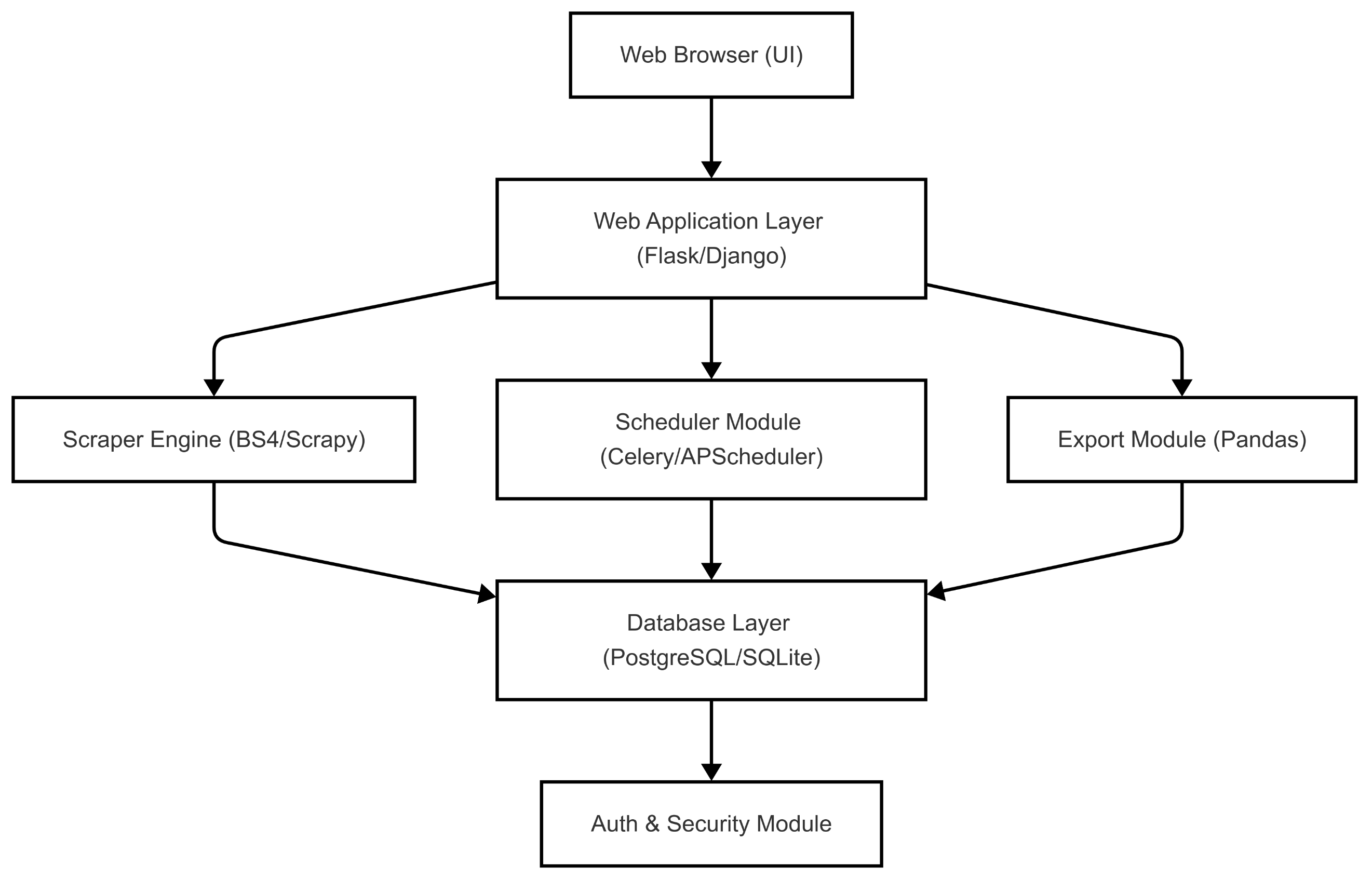


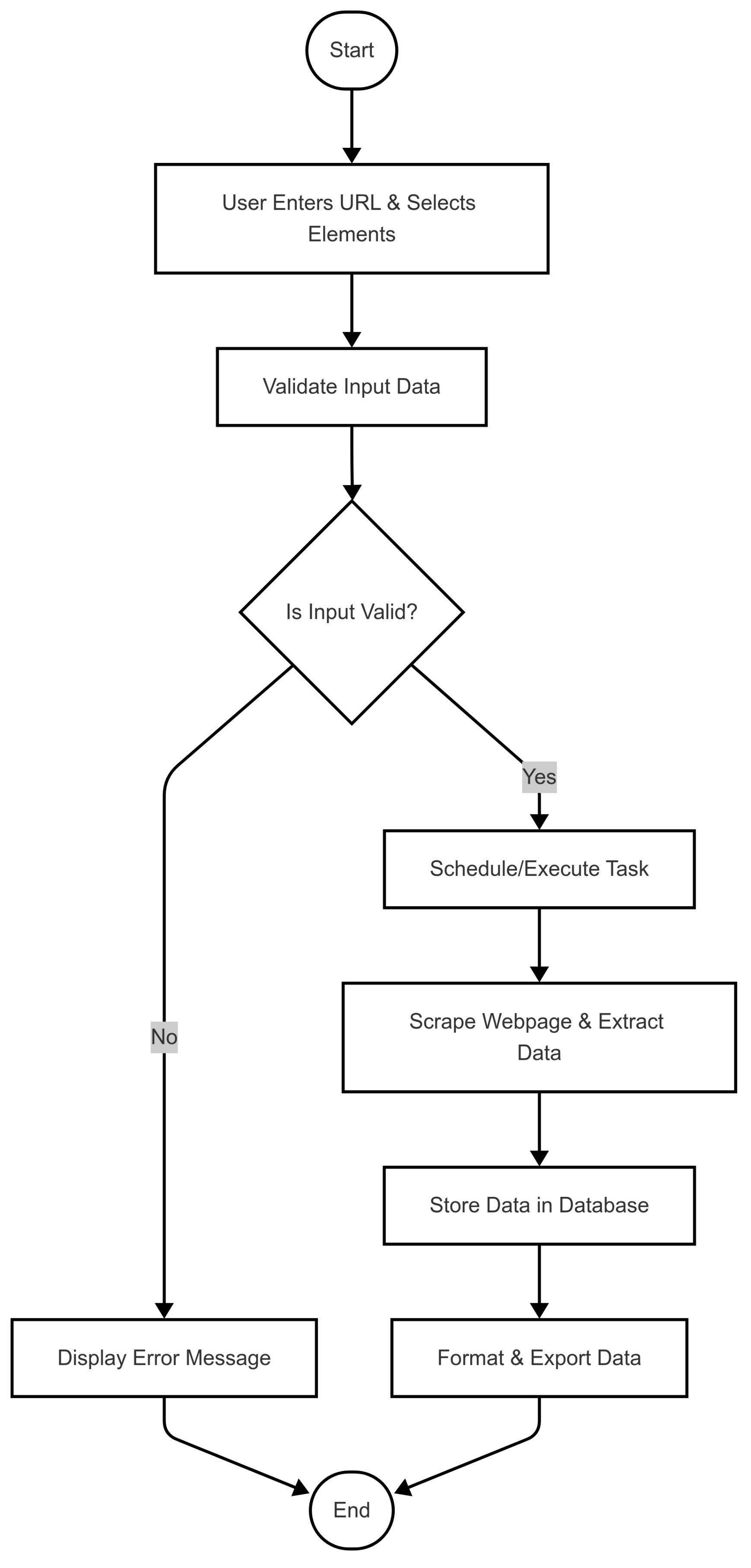
Figure 7.1. System Architecture

* 1. **Description of System Architecture**

This layered architecture diagram outlines the major components:

* **User Interface:** Delivered via a web browser.
* **Web Application Layer:** Built on frameworks like Flask or Django.
* **Core Services:** Include the Scraper Engine, Scheduler, and Export Module.
* **Database:** Persistent storage for tasks and results.
* **Security:** Authentication and protection functions.

1. **Activity Diagram**



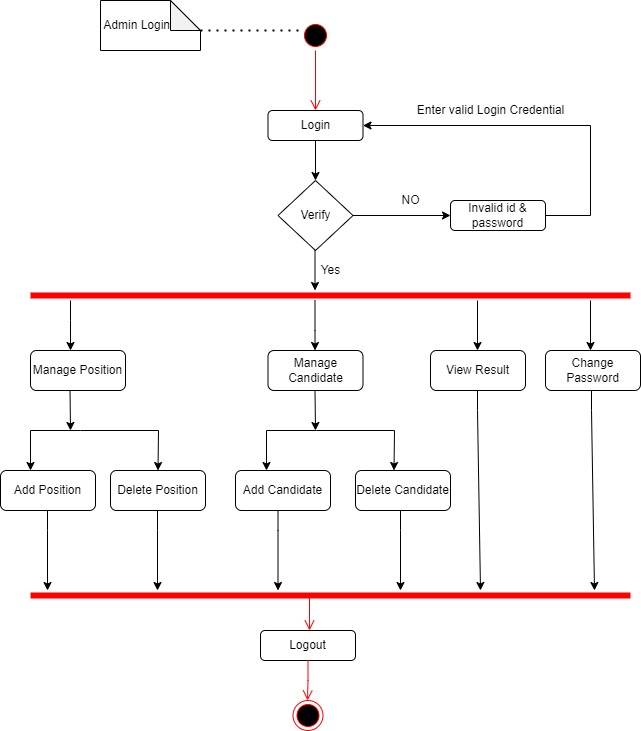


Figure 8 Activity Diagram

* 1. **Description of Activity Diagram**

This activity diagram captures the detailed flow for a scraping task—from user input and validation, through scraping, data storage, and export.

An activity diagram is a **behavioural diagram** i.e., it depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

1. **Sequence Diagram**

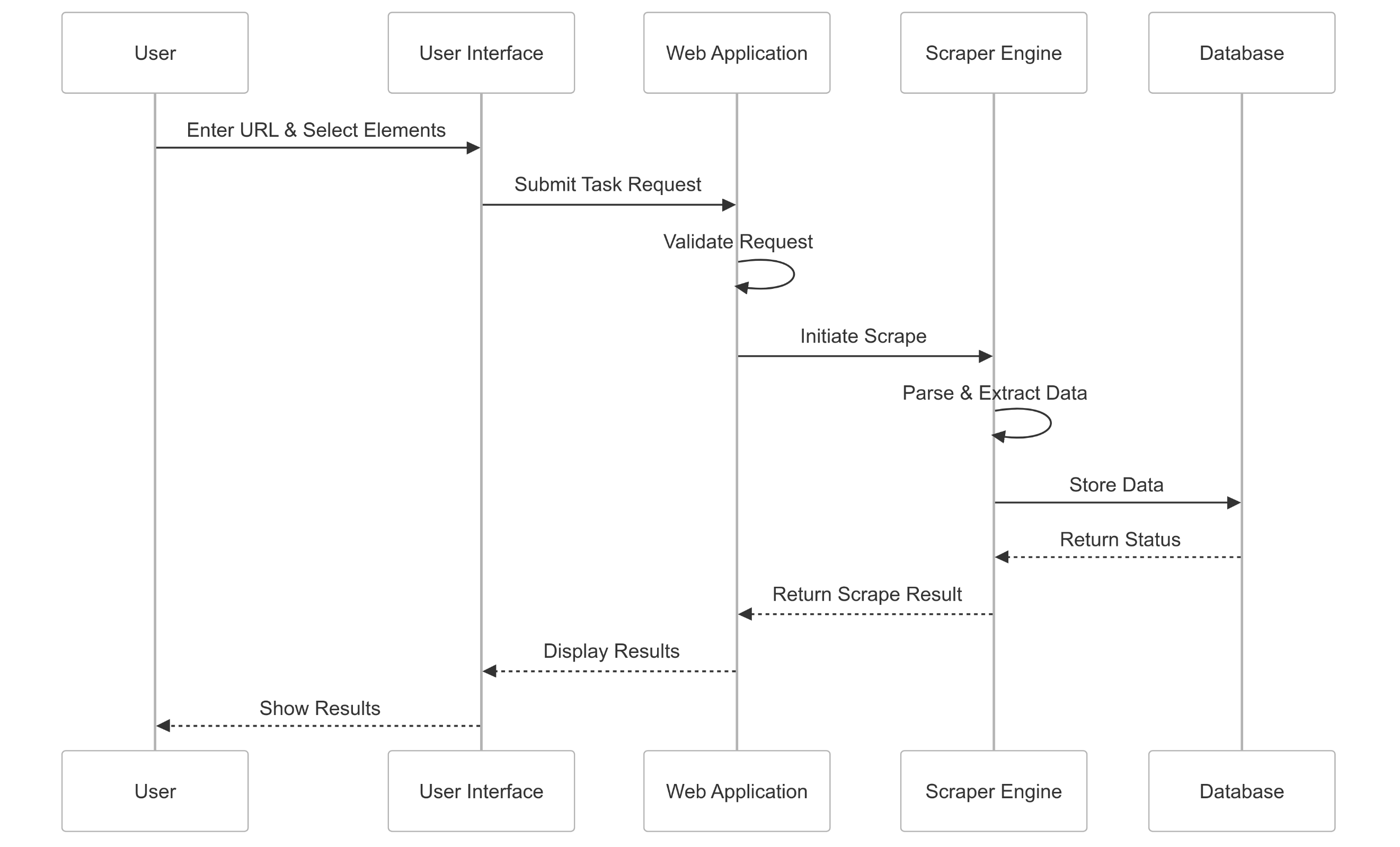


Figure 9 Sequence Diagram

* 1. **Description of Sequence Diagram:**

Unified Modelling Language (UML) is a modeling language in the field of software engineering that aims to set standard ways to visualize the design of a system. UML guides the creation of multiple types of diagrams such as interaction, structure, and behavior diagrams. A sequence diagram is the most commonly used interaction diagram.

This sequence diagram details the interactions when a user initiates a scraping task:

1. The **User** enters the URL and selects elements.
2. The **UI** sends the task request to the **Web App**.
3. The **Web App** validates the request and triggers the **Scraper Engine**.
4. The **Scraper Engine** parses the webpage, extracts data, and stores results in the **Database**.
5. Finally, results are returned to and displayed by the UI.
6. **Data Dictionary**
   1. **Admin Table:**

Below table show that the detail information about admin like admin name, admin id, password, etc. which contains different field name, data type, constrains and many more.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Field Name** | **Datatype** | **Size** | **Description** | **Constraint** | **Example** |
| 1 | Email | varchar | 45 | NOT NULL | Admin email address | admin@gmail.com |
| 2 | Password | varchar | 15 | NOT NULL | Admin password | Admin123 |

Table 1. Admin Table

* 1. **User Table:**

Below table show that the detail information about User like User name, User id, password, etc. which contains different field name, data type, constrains and many more.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraint** | **Description** | **Example** |
| 1 | Email | varchar | 45 | NOT NULL | User email address | user@gmail.com |
| 2 | Password | varchar | 15 | NULL | User password | user123 |

Table 2. User Table

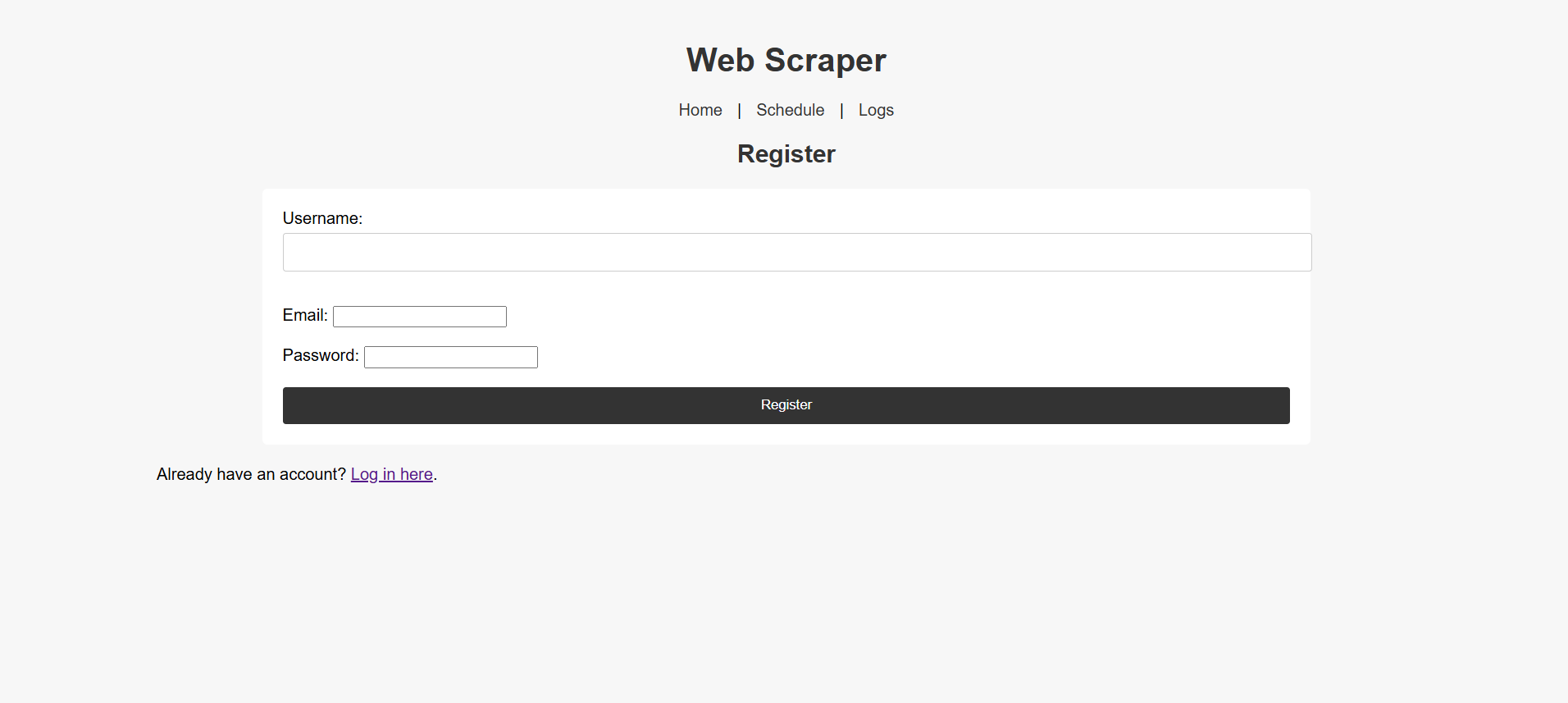
1. **Form Design**

**(Screenshots Phase 1, 2, 3)**

**11.1 Development Phase -1**

* **Login Page Design:-**

Below screenshot shows the login page which contain username/email, password and there is login button which perform after correctauthentication entered by User.



* **Code :-**

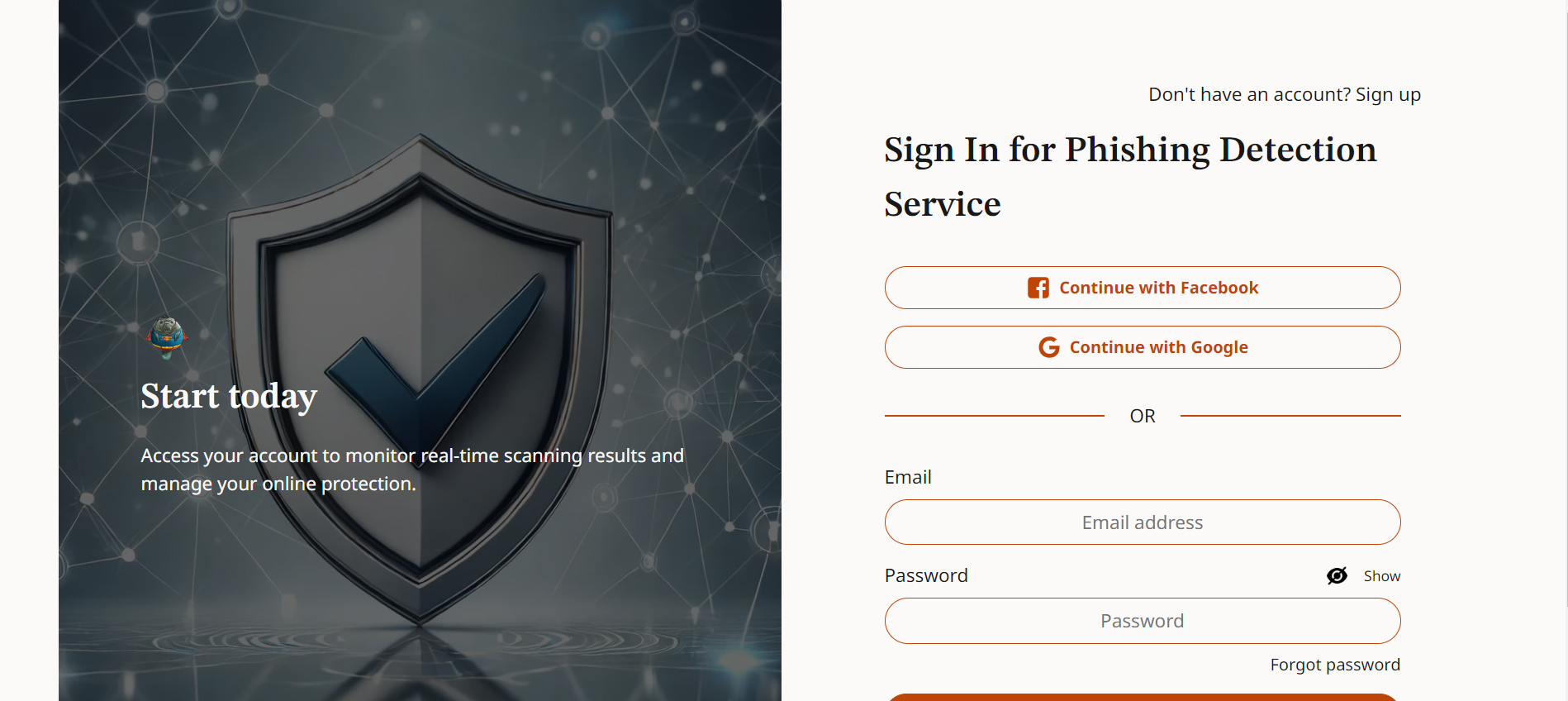
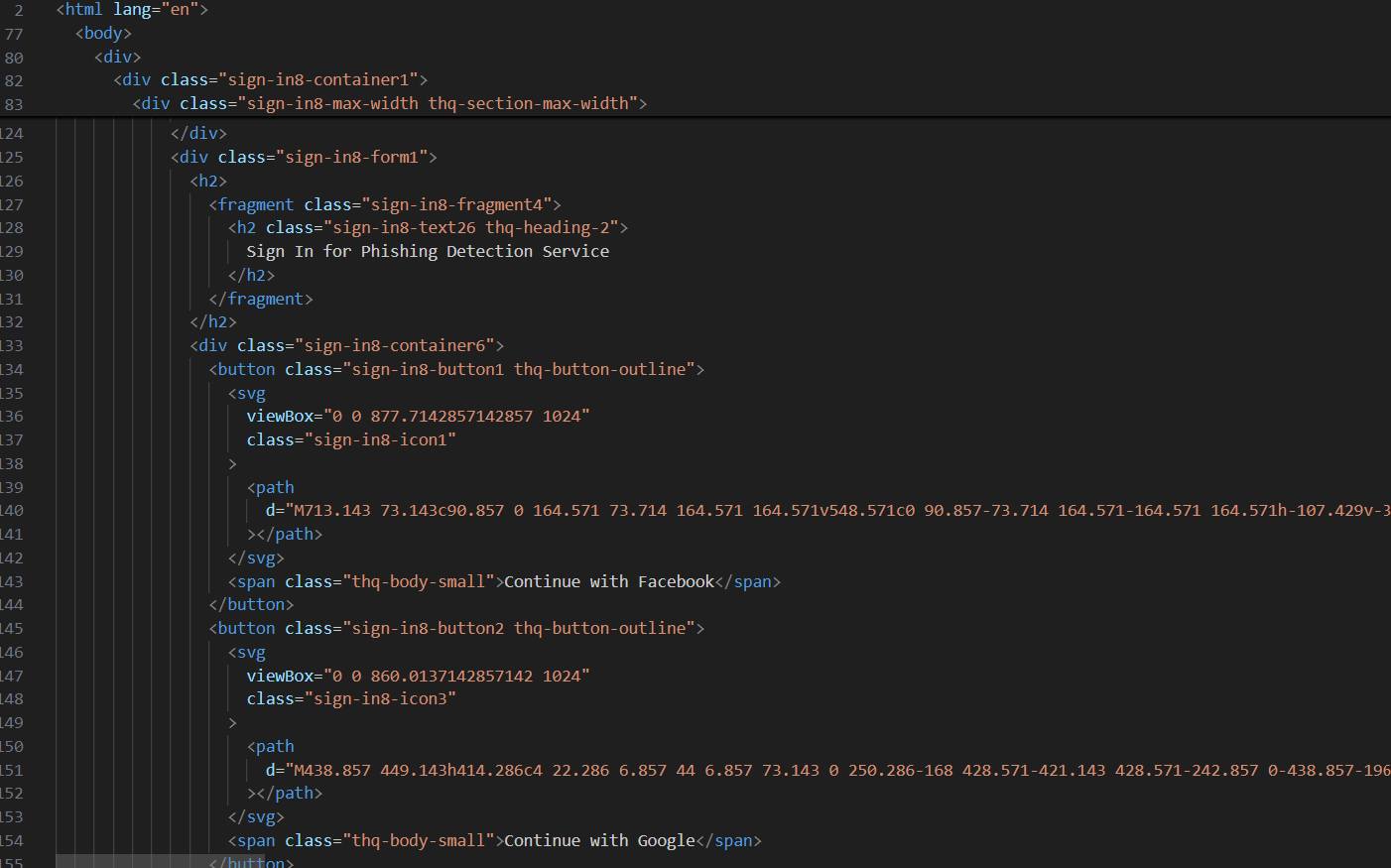
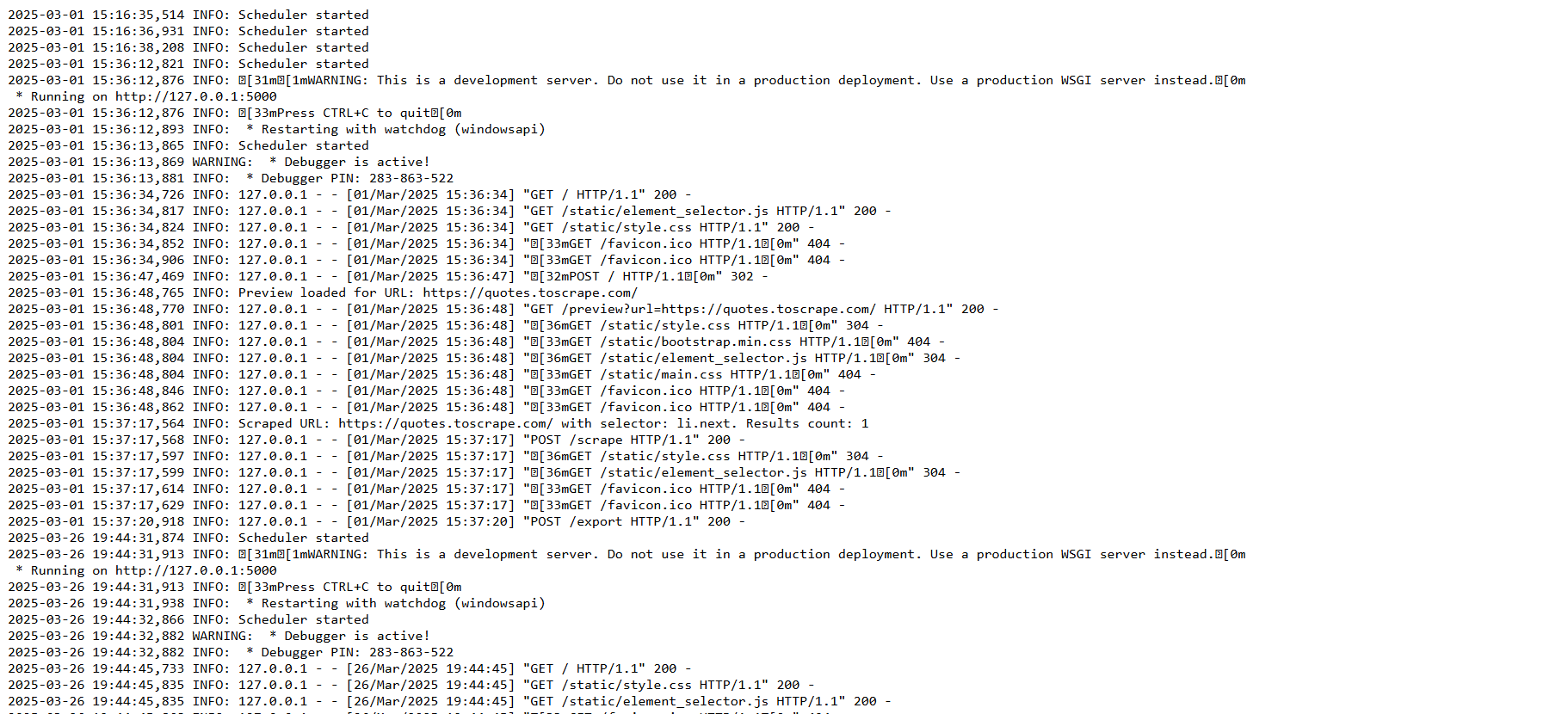


Figure 11.1 Login Page Design

**11.2 Development Phase -2**

* **Logs Design:-**

Below screenshot shows the logs page which contain username and the logs page to display log data directly in a timestamp format with **Username, Action, Timestamp, and Details**, along real-time updates. here is logs data entered by User.



* **Code :-**

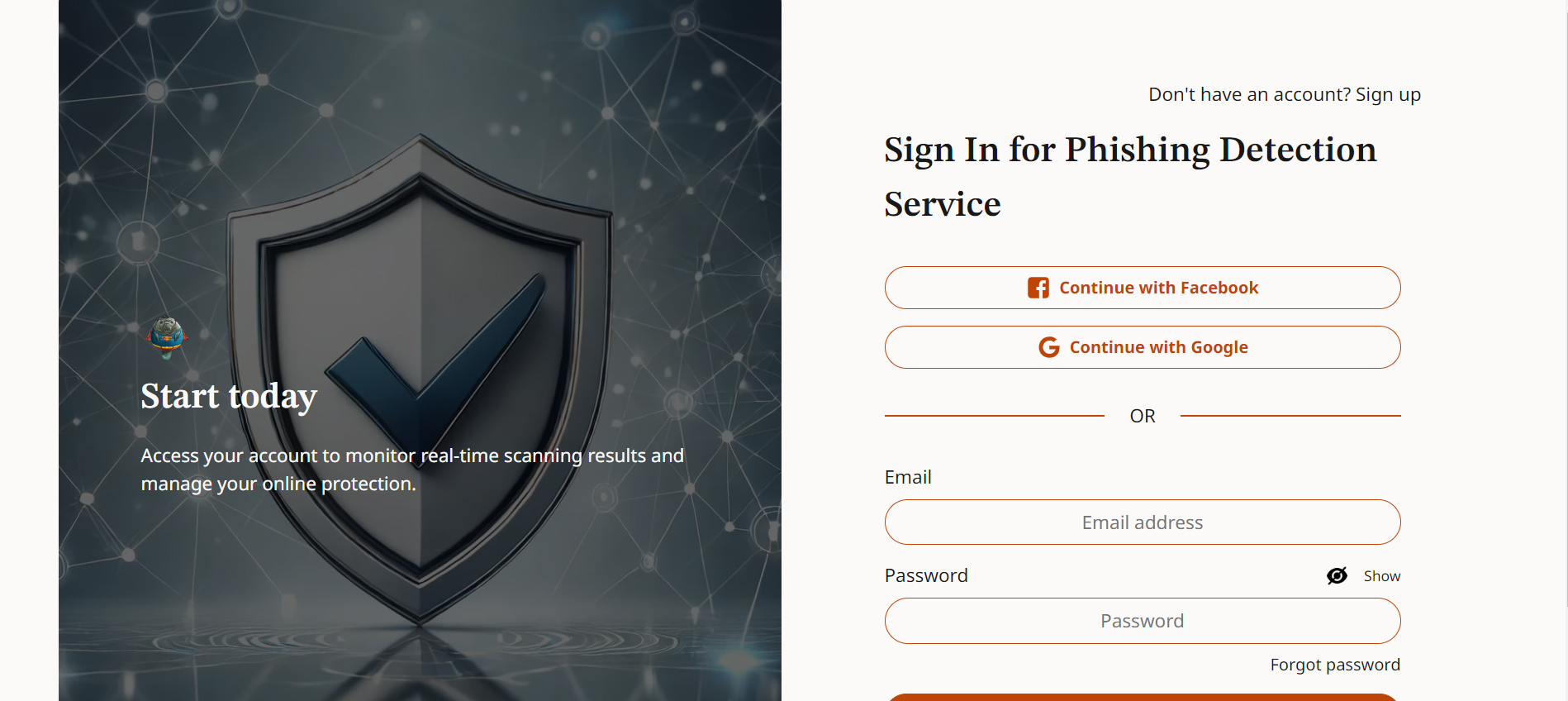
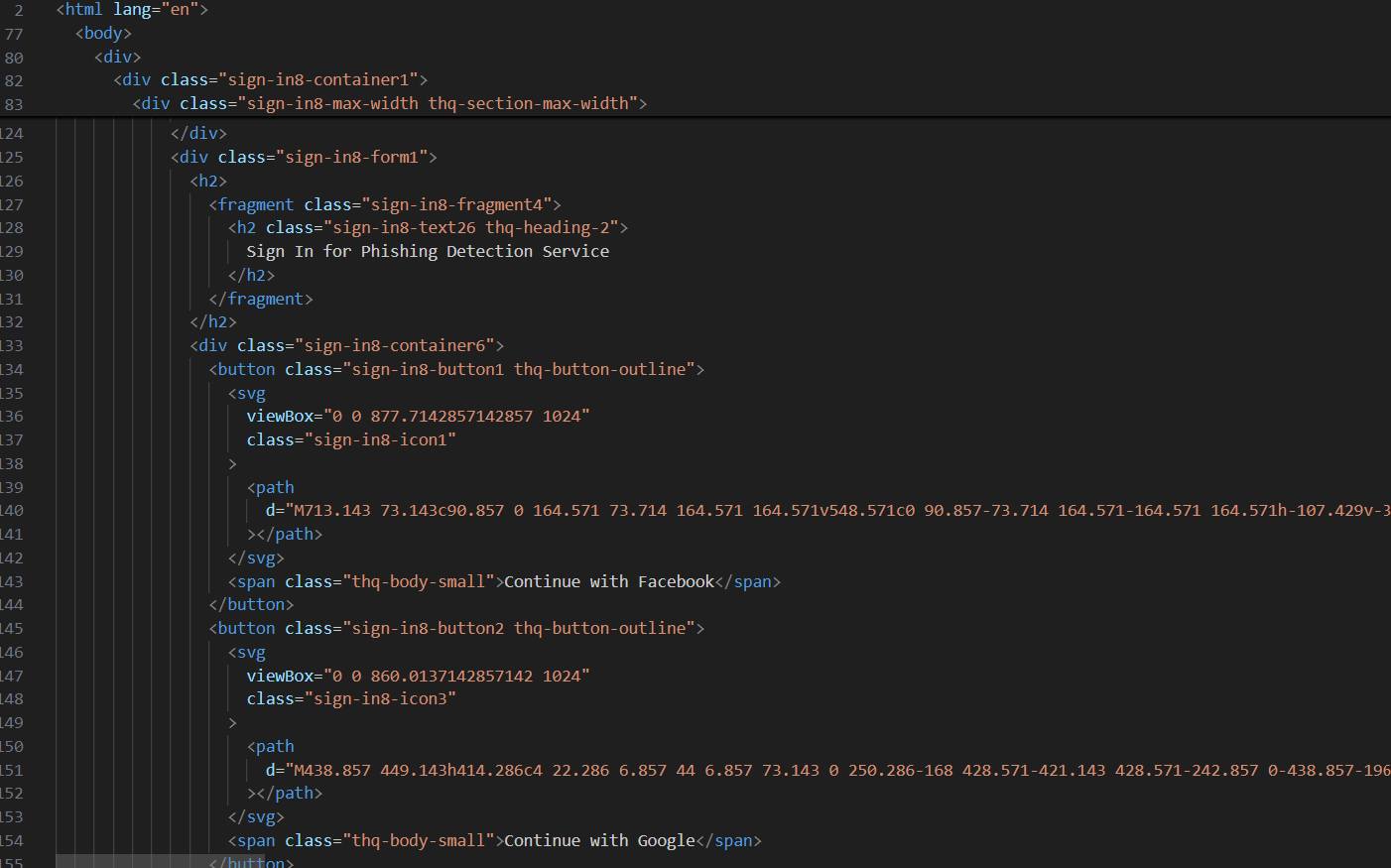
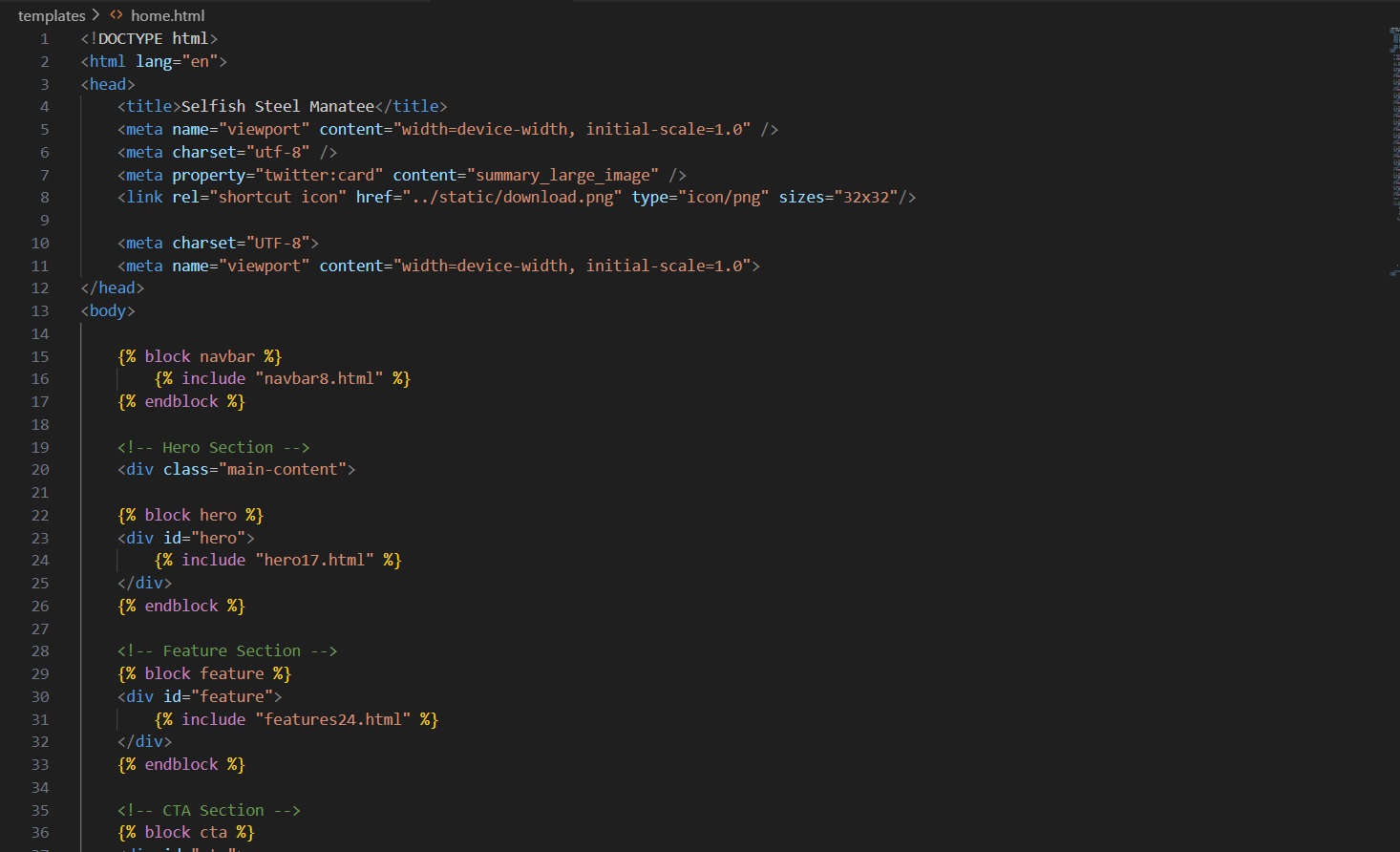


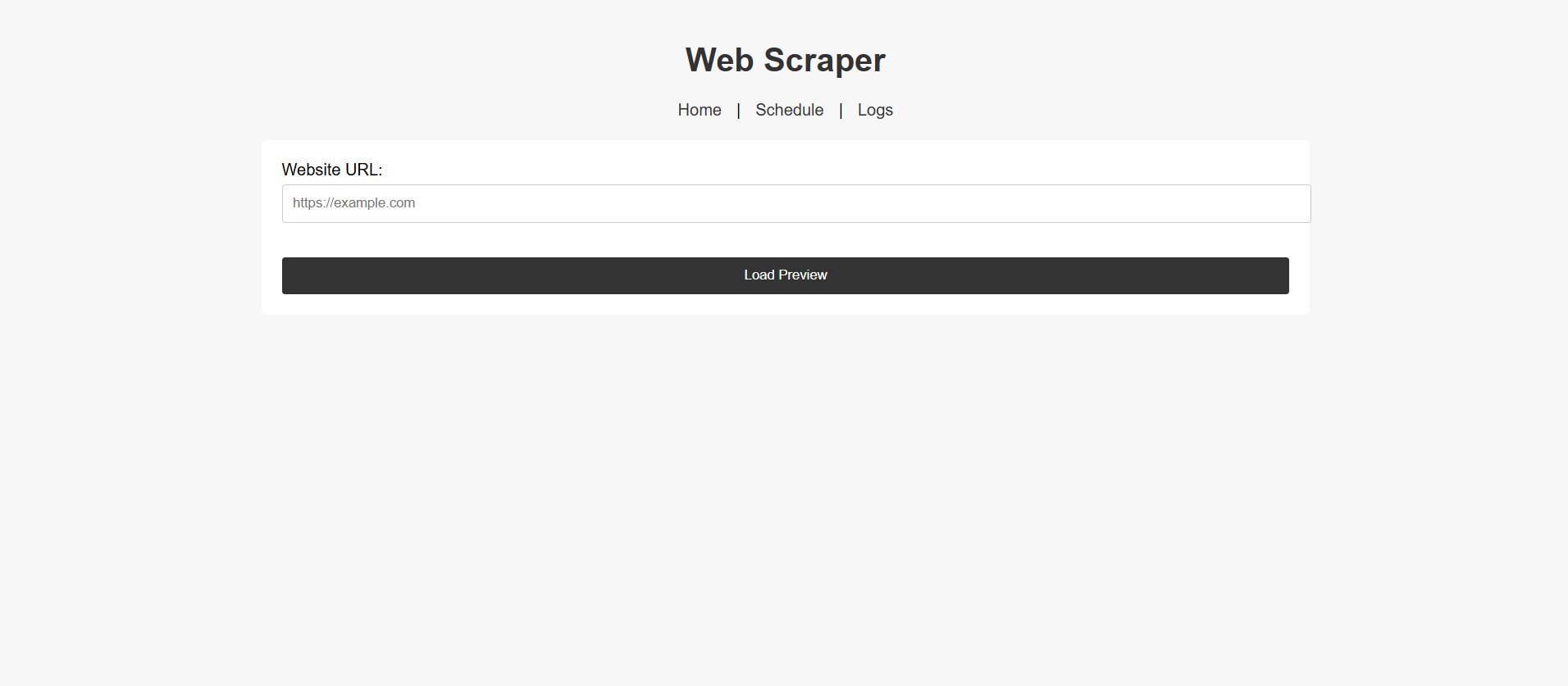
Figure 11.2

Figure 11.3 Logs Design

**11.3 Development Phase -3**

* **Web Scraper Design:-**

Below screenshot shows the Web Scraper page which contain input field where the user can enter data and there is button which perform after correctauthentication entered by User.



* **Code :-**

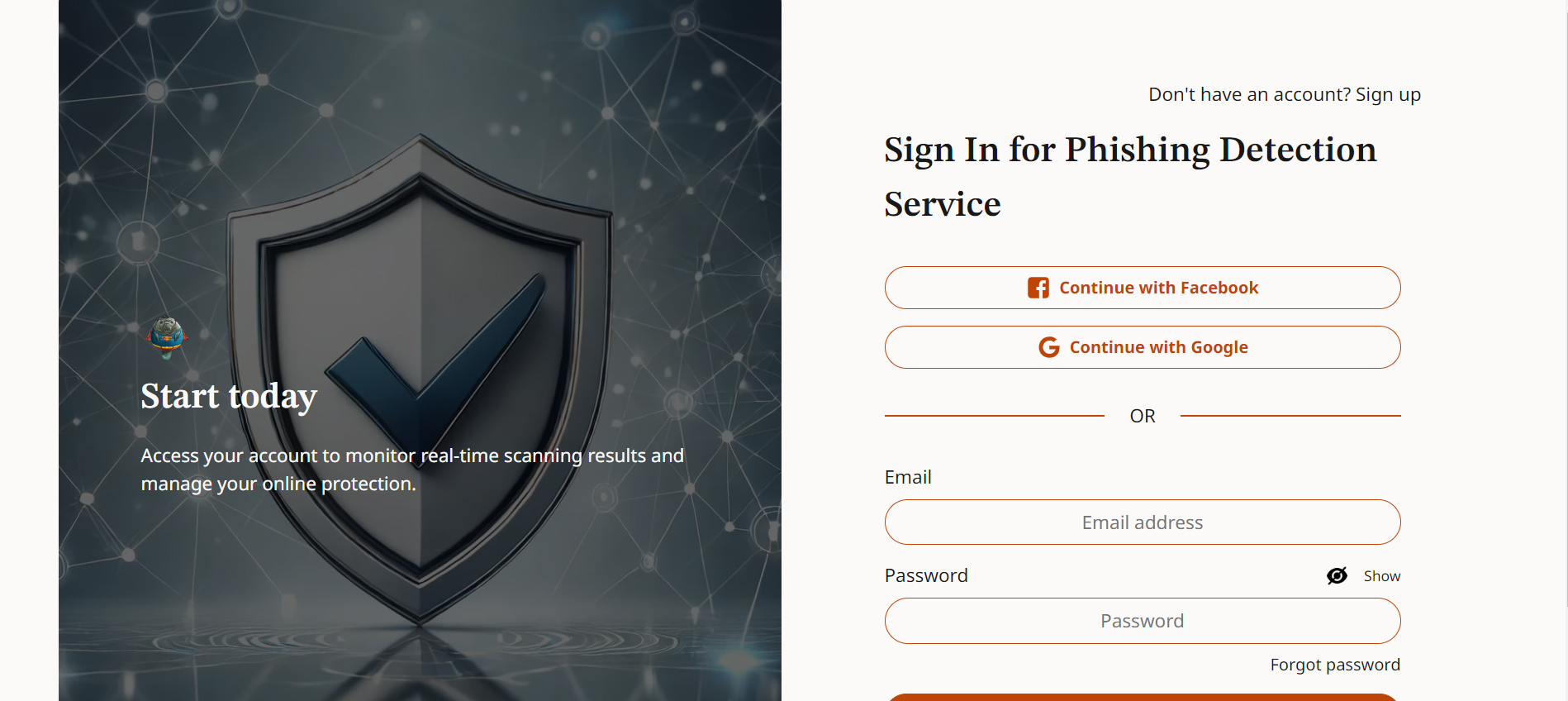
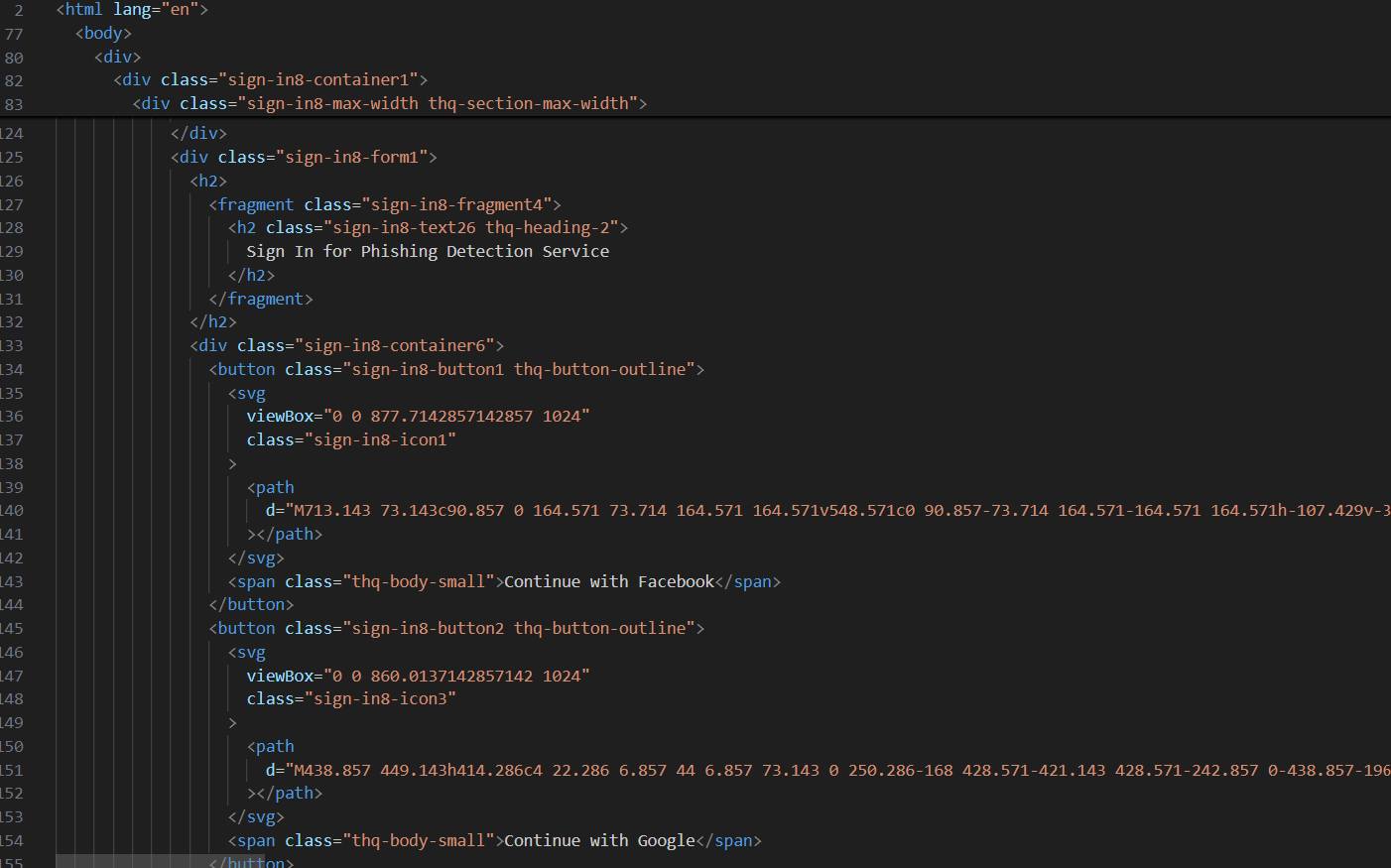
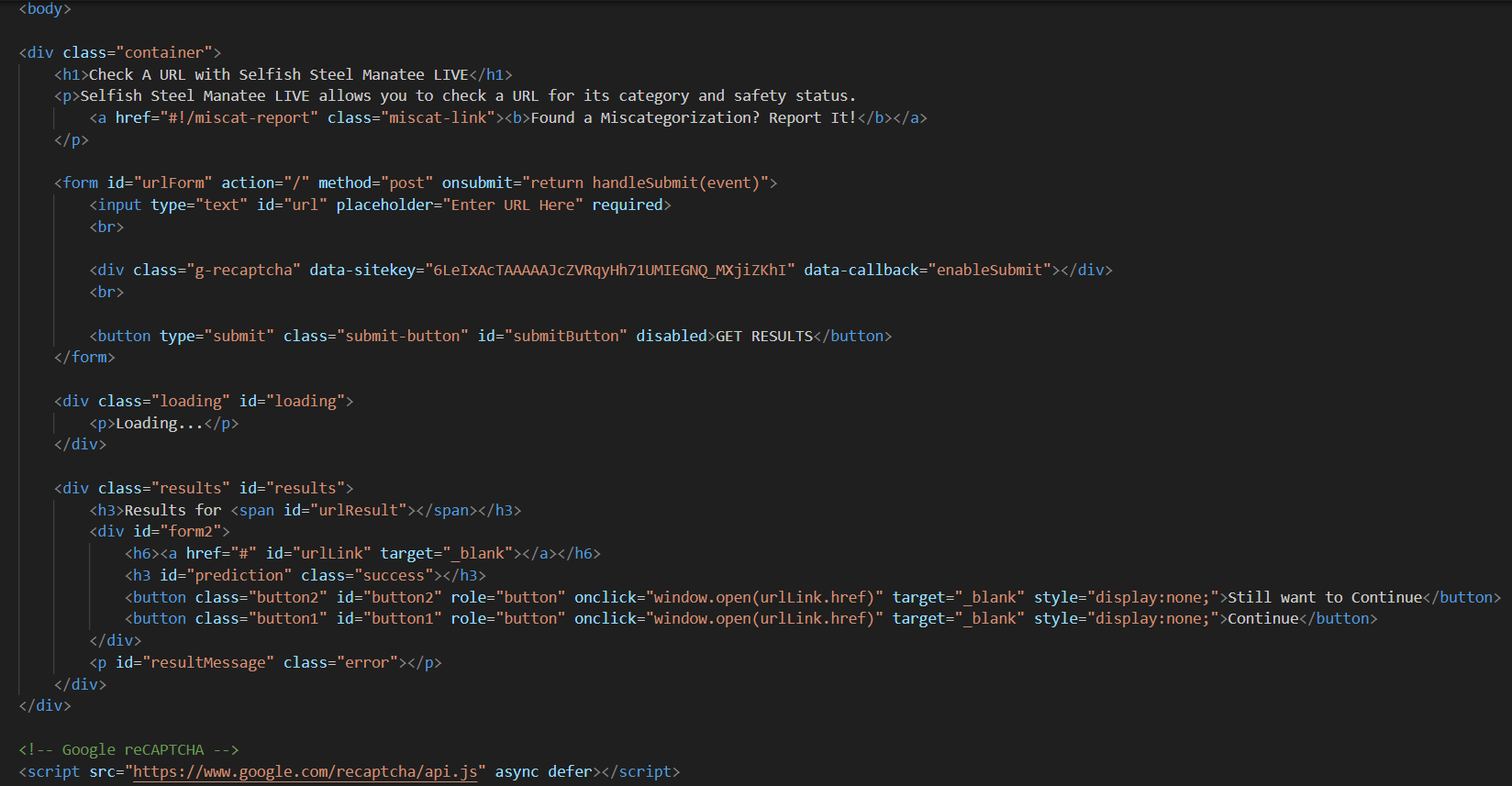


Figure 11.2

Figure 11.3 Web Scraper Design

**12. What is testing?**

**Web Testing,** or website testing is checking your web application or website for potential bugs before its made live and is accessible to general public. Web Testing checks for functionality, usability, security, compatibility, performance of the web application or website.

During this stage issues such as that of web application security, the functioning of the site, its access to handicapped as well as regular users and its ability to handle traffic is checked.

**12.1. Importance of testing**

* **Product quality** - Products always serve users, so it is very important that they deliver the values they promise. Hence, they should work properly to ensure a great customer experience. Following product requirements is imperative because it helps you get the wanted results.
* **Security** - there are a bunch of situations in which the information and details of the users are stolen. Therefore, people look for all-tested and reliable products. Testing will ensure that the user gets a trustworthy product, keep the user's personal information and data safe, and provide vulnerability-free products.
* **Customer satisfaction** - the first impression is important, and if you fail at it, users are going to find another product that will accomplish all their requirements. One reason why apps should be tested is to provide the best user experience possible. Being the best product in the market will help you gain trust-worthy clients, which will have great long-term effects.
* **Cost effectiveness** - including testing in your project can save money in the long run. Software development consists of many stages, and if bugs are caught in the earlier phases, it costs much less to fix them. That is why it is important to start with testing as soon as possible.

**12.2. Types of testing**

* **Unit tests** - Unit tests are very low level and close to the source of an application. They consist in testing individual methods and functions of the classes, components, or modules used by your software. Unit tests are generally quite cheap to automate and can run very quickly by a continuous integration server.
* **Integration tests** - Integration tests verify that different modules or services used by your application work well together. For example, it can be testing the interaction with the database or making sure that microservices work together as expected.
* **Functional tests**- Functional tests focus on the business requirements of an application. They only verify the output of an action and do not check the intermediate states of the system when performing that action.
* **End to end tests**- End-to-end testing replicates a user behaviour with the software in a complete application environment.
* **Acceptance testing-** Acceptance tests are formal tests that verify if a system satisfies business requirements. They require the entire application to be running while testing and focus on replicating user behaviours.
* **Performance testing-** Performance tests evaluate how a system performs under a particular workload. These tests help to measure the reliability, speed, scalability, and responsiveness of an application.

**13. Future Enhancement**

The Web Scraper can be enhanced in the future by implementing the following methods:

* **REST API Integration:**

Develop a comprehensive RESTful API to allow programmatic control of the scraper. This would enable third-party applications or developers to trigger scraping tasks, retrieve results, and integrate the scraper into larger workflows.

* **Cloud-Based Distributed Scraping:**

Transition to a cloud-based, distributed architecture to improve scalability and performance. This would involve leveraging cloud resources (e.g., AWS, Azure, or Google Cloud) to run concurrent scraping tasks across multiple nodes, thereby handling higher loads and reducing task execution times.

* **Advanced Analytics Dashboard:**

Build an interactive dashboard that presents detailed analytics on scraping performance, error rates, data trends, and user activity. This dashboard can help both administrators and users make data-driven decisions and monitor system health in real time.

* **Custom Scraping Rules Engine:**

Implement a flexible, rule-based engine that allows users to define custom rules and logic for scraping. This could include advanced filtering, conditional extraction based on dynamic criteria, and the ability to adapt to changes in website structures automatically**.**

* **Browser Extension Integration:**

Create a browser extension to facilitate an even more user-friendly experience. With a browser extension, users could select elements directly on any web page with a single click, streamlining the setup process for scraping tasks.

Enhanced Reporting and Analytics:

Introduce an advanced reporting module that provides detailed, real-time insights into the scraping operations. This module would feature interactive dashboards displaying key performance metrics, data trends, error rates, and historical comparisons. Users and administrators can leverage these insights to monitor system health, optimize scraping configurations, and make data-driven decisions for continuous improvement.

**14. References & Bibliography**

Book:

1. **Flask Web Development** - Miguel Grinberg
2. **Mastering Flask** - Jack Stouffer
3. **Flask Framework Cookbook** - Shalabh Aggarwal
4. **Flask By Example** - Gareth Dwyer
5. **Web Scraping with Python** - Ryan Mitchell (Covers BeautifulSoup and Scrapy)

Website:

1. <https://www.w3schools.com>
2. <https://www.tutorialspoint.com>
3. <https://www.geeksforgeeks.org>
4. https://pypi.org/project/beautifulsoup4

Other resources:

1. <https://www.youtube.com/c/YahooBaba>
2. <https://www.youtube.com/c/CodeWithHarry>
3. <https://www.youtube.com/c/ApnaCollegeOfficial>
4. <https://www.youtube.com/user/GeekyShow1>