**| Passion | Execution | Excellence |**

**Probe Agile –** **SOP**

**© Mahendra Next Wealth IT India Pvt Ltd | Mahendhirapuri | Mallasamudram (W) |**

**| Thiruchengode (TK) | Namakkal (DT) | Tamil Nadu – 637503 | Contact # 04288 – 288 509**

# Document History

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **Title-holder** | **Issue Version (Release)** | **Reviewed By** | **Approved By** | **Pages Revised** | **Reason for Revision** | **Revised Section & Description** |
| **Aug 12, 2024** | Surya S | 001 | Karupusamy N | Venkatesh D | ALL | Baseline | NA |

# Distribution

This document is classified as confidential and is only available to ***Probe Agile Team***. The head of MNW or its authorized representative shall authorize any special requests made for accessing this manual by other parties.

# Table of Contents

[I. Document History 2](#_Toc173234464)

[II. Distribution 2](#_Toc173234465)

[III. Table of Contents 3](#_Toc173234466)

[IV. Purpose 4](#_Toc173234467)

[V. Scope 4](#_Toc173234468)

[VI. Entry Criteria 4](#_Toc173234469)

[VII. Exit Criteria 4](#_Toc173234470)

[VIII. Deliverables 4](#_Toc173234471)

[IX. Workflow 5](#_Toc173234472)

[X. Process 6](#_Toc173234473)

[Work Instructions 6](#_Toc173234474)

[Requirement: 6](#_Toc173234475)

[Input data: 6](#_Toc173234476)

[1. Analysis 6](#_Toc173234477)

[2. Database Setup (MySQL and DBeaver) 7](#_Toc173234478)

[3. Scripting (MySQL and DBeaver) 7](#_Toc173234479)

[4. Script Testing and Bulk Data Extraction 9](#_Toc173234480)

[5. API Endpoint Setup (Django REST framework) 9](#_Toc173234481)

[6. API Testing 11](#_Toc173234482)

[7. Documentation 11](#_Toc173234483)

[8. End-to-End Testing 11](#_Toc173234484)

[9. Dashboard 11](#_Toc173234485)

[10. Deployment 11](#_Toc173234486)

[11. Maintenance 12](#_Toc173234487)

[XI. Check List 13](#_Toc173234488)

[XII. Responsibility 13](#_Toc173234489)

# Purpose

The Purpose of this document is to provide a handout/reading material for the new joiners of the ***Probe Agile Team.***

# Scope

The scope of this document is to define the workflow and to describe the work instructions.

# Entry Criteria

The input (Source) and the documentation shared by Client through Email to scrape the details available in the website.

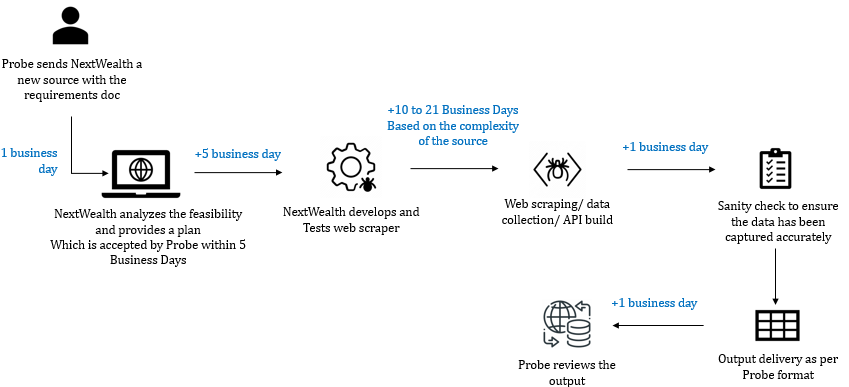
# Exit Criteria

The output (API) shared via Email and the codes are committed in GIT.

# Deliverables

Affiliates, who **completes** reading of this manuscript is expected to have a better idea about the ***Probe Agile*** Process.

# Workflow



# Process

## Work Instructions

Automation Web Scraping using Python

### Requirement:

The requirement of the project is to scrap details for the client required URL’s.

### Input data:

Source and the documentation will be shared by Probe team.

### 1. Analysis

**Objective Definition:** Define the goal of the web automation project.

Example: "Automate the extraction of data from client’s required website and store it in a MySQL database."

**Scope:** Outline the boundaries of the project.

Example: "Scrape historical data, incremental data and PDF files."

**Requirements Gathering:** List technical and non-technical requirements.

Technical: Python, Selenium, Beautiful Soup, MySQL, DBeaver, Django REST framework, HTML, CSS, JavaScript.

Non-technical: Excel bulk data merging.

**Feasibility Study:** Evaluate the feasibility of the project considering the website structure and potential obstacles like CAPTCHA or anti-scraping mechanisms.

**MNW team** need to analyze, evaluate the source and share the estimation time with the client along with the complexity level.

|  |  |  |  |
| --- | --- | --- | --- |
| **Complexity Type** | **Low** | **Medium** | **High** |
| Timeline for Development | 3 Weeks | 6 Weeks | 9 Weeks |

Share the Analysis and the complexity level to the client and get the approval.

### 2. Database Setup (MySQL and DBeaver)

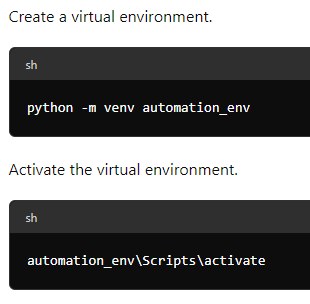
**Install MySQL:** Install and configure MySQL on our server or local machine.

**Create Database and Tables:** Use MySQL commands to set up the database and necessary tables.

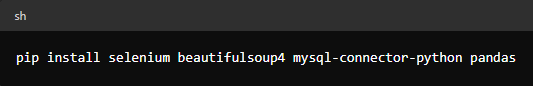
**Install DBeaver:** Use DBeaver for database management and query execution.

### 3. Scripting (MySQL and DBeaver)

**Set up Virtual Environment:** Create and activate a Python virtual environment.



**Install Necessary Libraries**: Install required Python libraries.



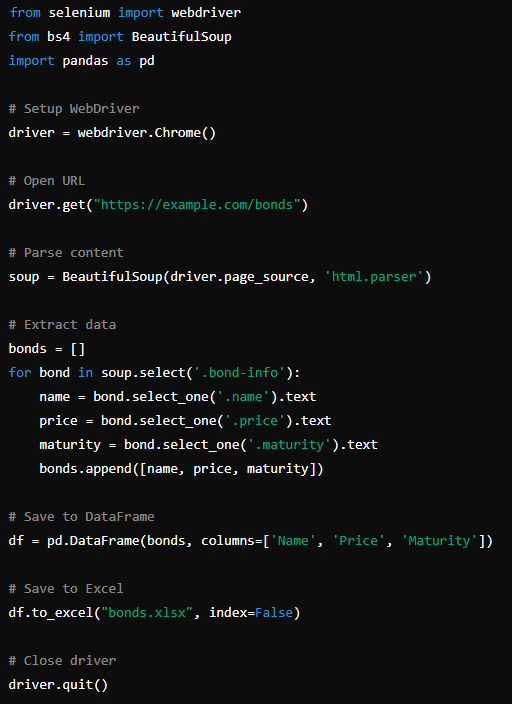
**Coding Standards:** Follow coding standards and best practices. Example: PEP 8 for Python.

**Proxy Handling:** Integrate proxy management to avoid IP bans.

**Timeout and Retry Logic**: Implement timeout and retry logic to handle transient issues.

**Error Handling**: Incorporate robust error handling and logging mechanisms.

**Scraping Logic:** Develop the core scraping logic.



### 4. Script Testing and Bulk Data Extraction

**Unit Testing:** Write unit tests for our script components.

**Example:** Test the data extraction and database insertion functions.

**Integration Testing:** Test the complete automation workflow.

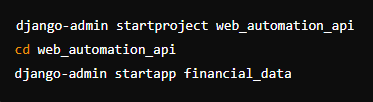
**Bulk Data Extraction:** Ensure the script can handle large volumes of data efficiently.

### 5. API Endpoint Setup (Django REST framework)

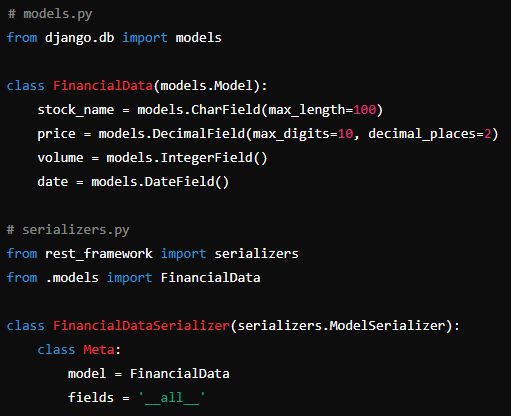
**Install Django and Django REST framework:**

Screenshot 2024-06-21 161601.png

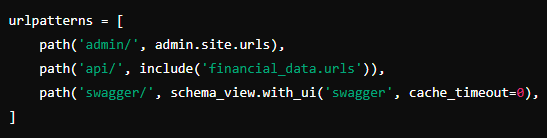
**Create Django Project:**



**Configure Models and Serializers:**



**Set Up API Views and URLs:**



### 6. API Testing

**Use Postman:** Test the API endpoints using Postman or a similar tool.

Go to **http://127.0.0.1:8000/api/your\_endpoint/** to interact with our API endpoints.

### 7. Documentation

Document the code to explain its functionality and external documentation such as user manuals and API documentation.

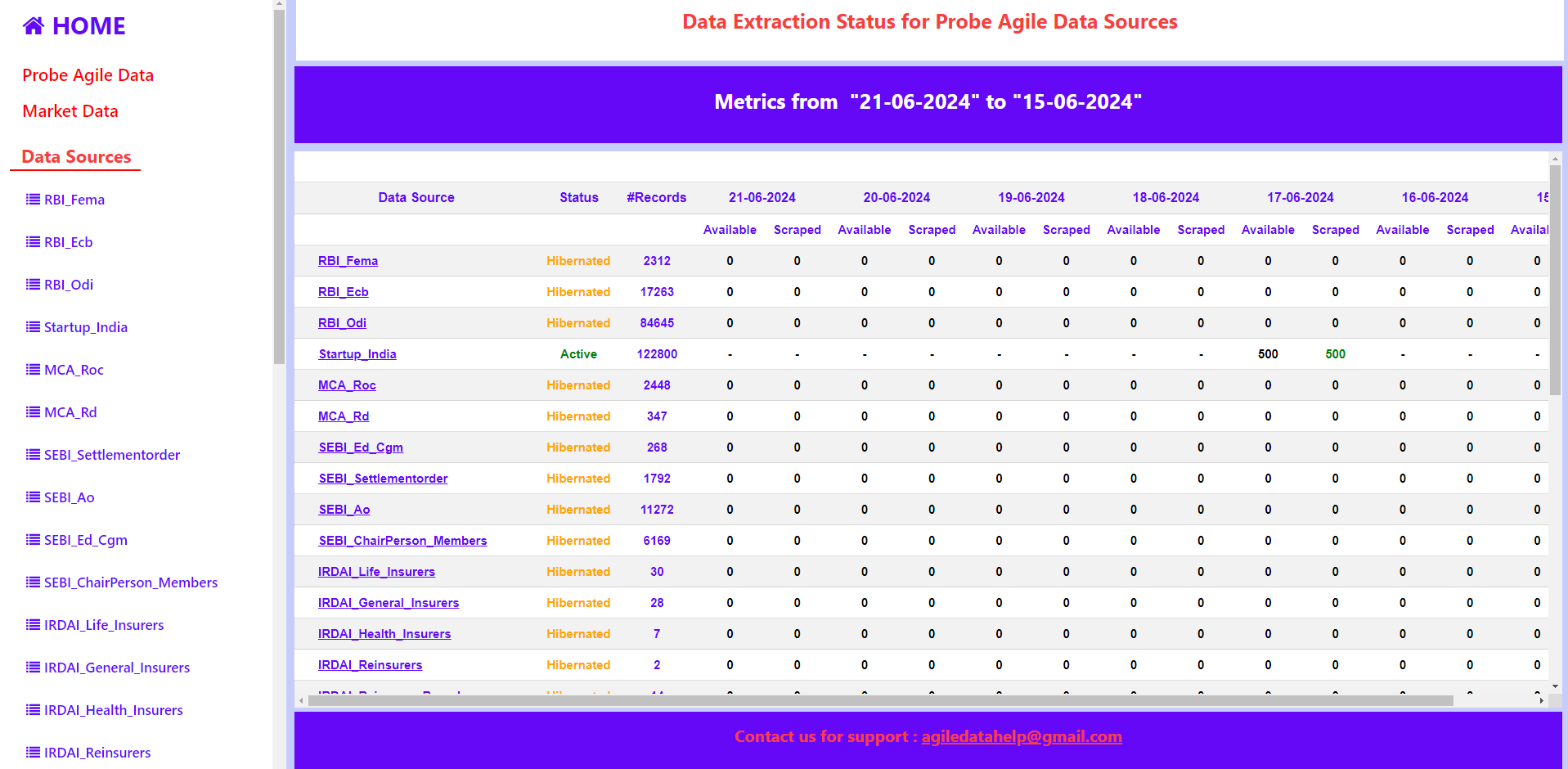
### 8. End-to-End Testing

**Complete Workflow Testing:** Test the entire process from data extraction to API endpoint usage.

**Load Testing:** Perform load testing to ensure the system can handle high traffic.

### 9. Dashboard

Display the number of records scrapped so far



### 10. Deployment

**Server Setup:** Set up the production server (e.g., using IIS).

After deploying the live environment, send the delivery note to the client. The delivery note include the API endpoint URL, API documentation, and Dashboard URL.

### 11. Maintenance

**Monitoring: Implement monitoring for the web scraping script and API endpoints.**

**Error Handling: Set up error handling.**

**Regular Updates: Update the script and API endpoints as necessary to handle changes in the website structure or new requirements.**

**Data Quality Checks: Perform regular data quality checks to ensure the accuracy and integrity of the extracted data.**

**Below Points noted under the Maintenance phase:**

**After sending the delivery note, we will run the deployed source for incremental purposes for one month. After the one-month period, the client will discuss the source can be either "Active" or "Hibernate."**

**If the source status is "Active," the client will specify the time to run the code and obtain incremental data. In case the source status is "Hibernate," we will not run the code until the client changes the source status to "Active."**

**We have two sources: one is "Market Data" and the other is "Startup India." The market data is a B-scope category source. We work on this source when the client needs data or requests any changes. For "Startup India," the client's decision is to run and obtain the incremental data twice a month.**

# Check List

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Probe Agile Checklist** | **Frequency** | **Owner** |
| 1 | Check the Scheduled scripts and log table if any issues occurs fix that | When Job is completed | Developer |
| 2 | Review the tasks and objectives for the day. | When new source come | Developer |
| 3 | work on script development (Historical script, incremental script and API endpoint) based on daily task | When new source come | Developer |
| 4 | Review and refactor existing scripts to improve efficiency and accuracy. | When new source come | Developer |
| 5 | Perform unit tests on new or updated scripts to verify functionality. | WER | Developer |
| 6 | Monitor the scraping process to ensure it completes without issues. | When Job is completed | Developer |
| 7 | Work on data processing from the database to Excel if the client needs data from a particular source. | When Job is completed | Developer |
| 8 | Compare the newly extracted data with previous datasets to identify any discrepancies or anomalies. | When Job is completed | Developer |
| 9 | Explore new tools and technologies that can enhance the scraping process. | When Job is completed | Developer |
| 10 | Reply the PEP comment | Weekly | Developer |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Probe Agile Checklist** | **Frequency** | **Owner** |
| 1 | Conduct Daily briefing-Standup call | Daily | TL |
| 2 | Check the task and allocate task to agents | Daily | TL |
| 3 | Check the debug raised by agents | Daily | TL |
| 4 | Ensure all client queries through live chat or mails are addressed on time | WER | TL |
| 5 | Clarify team members' doubt, also seek for client help if needed | WER | TL |
| 6 | Prepare weekly / monthly PPT | Weekly/Monthly | TL |
| 7 | Coordinate with client for team meetings | When Job is completed | TL |
| 8 | Cross check the delivery reports | When Job is completed | TL |
| 9 | Update the Standard Operating Procedure (SOP) if there are significant changes in the process. | WER | TL |

# Responsibility

|  |  |  |
| --- | --- | --- |
| Document / Procedure Responsibility | DEPARTMENT(S) / PROCESS (S) / FUNCTION (S) | DESIGNATION |
| PREPERATION | Operations | Team Leader |
| REVIEW | Operations | Deputy Manager |
| APPROVAL | Operations | Manager |