Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2025), B.Sc. in CSE (Day)

Leads Generate AI Agent

Project Report
Course Title: Data Mining Lab
Course Code: CSE-436
Section:: 212-D2

Students Details

Name	ID
Md. Hasibur Rahman	212902018
Md. Sajjad Hossen	212902032

Submission Date:17-05-2025
Course Teacher's Name: Md. Atikuzzaman

[For teachers use only: Don't write anything inside this box]

Lab Project Status

Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

In the age of digital transformation, businesses heavily rely on accurate and timely leads to fuel their marketing, sales, and growth strategies. Traditional methods of lead generation—manual data collection or basic rule-based scrapers—are no longer scalable or adaptable to diverse web content formats. This project introduces Leads Generative AI Agent, an intelligent and dynamic web scraping system that utilizes LLMs (Large Language Models), asynchronous crawling, and user-defined inputs to extract business lead data in real-time from websites. It eliminates the need for complex scraping scripts and provides non-technical users with a simple interface to define the fields they want.

1.2 Problem Domain & Motivations

- In the age of digital transformation, businesses heavily rely on accurate and timely leads to fuel their marketing, sales, and growth strategies.
- Traditional methods of lead generation—manual data collection or basic rule-based scrapers—are no longer scalable or adaptable to diverse web content formats.
- This project introduces Leads Generative AI Agent, an intelligent and dynamic web scraping system that utilizes LLMs (Large Language Models), asynchronous crawling, and user-defined inputs to extract business lead data in real-time from websites. It eliminates the need for complex scraping scripts and provides non-technical users with a simple interface to define the fields they want.

1.3 Complex Engineering Problem

Name of the P Attributes	**Explain how to address**
P1: Depth of knowledge required	Knowledge of Python, Data Mining , and Ai Agend terminology.
P2: Range of conflicting requirements	Balancing response accuracy with user-friendly interface and real-time performance.
P3: Depth of analysis required	Analysis of Data mining DeekSeek (Crawl4AI) for Leads Generative ai Agent intent classification.
P4: Familiarity of issues	Challenges with network stability and context accuracy.
P5: Extent of applicable codes	Extensive coding for model integration, web framework (Flask), and API connections.
P6: Extent of stakeholder involvement and conflicting requirements	Different needs from developers (technical) and healthcare professionals (accuracy).
P7: Interdependence	Relies on open-source models, libraries, and external medical data APIs.

Table 1.1: Summary of the attributes touched by the mentioned projects

1.4 Objectives

The primary objectives of this project are:

- **To design an interactive, user-friendly web interface** Allow users to input the target website URL, select content using CSS selectors, and specify the exact data fields they want to extract—making it usable by non-programmers. locations.
- **To implement a flexible, dynamic backend using Python and Flask** Handle user requests, launch scraping jobs in the background, and manage configuration updates seamlessly without restarting the server.
- **To integrate Crawl4AI for asynchronous, browser-based web crawling** Leverage a headless browser engine to navigate modern, JavaScript-rich web pages just like a human would, ensuring compatibility across websites.
- **To utilize DeepSeek LLaMA (via Groq API) for intelligent data extraction** Replace brittle, rule-based parsers with LLM-driven schema extraction that semantically understands the structure and content of HTML pages.
- **To support dynamic, user-defined field extraction**

  Let users input any combination of fields such as name, email, location, price, company, description, or others—and adjust extraction accordingly.
- **To display scraping progress and results in real-time**

  Visualize current scraping progress, pages completed, number of leads found, and display extracted leads directly in the frontend.

- **To provide a CSV export option**

  Automatically format and generate downloadable .csv files containing all collected lead data for use in CRM tools, marketing automation, or business analysis.
- **To allow manual lead entry and correction via UI**

  Offer the flexibility to manually add or edit scraped leads directly from the web interface for correction or enrichment.

1.5 Application

The **Leads Generative AI Agent** has a wide range of practical applications across various industries and domains. Its flexibility, dynamic configuration, and intelligent data extraction capabilities make it suitable for both technical and non-technical users. Below are the key application areas:

- **Sales and Marketing:** Automatically collect business leads from online directories, competitor websites, or contact pages. The extracted data can be used for targeted campaigns, email outreach, or market segmentation.
- **Real Estate Intelligence:** Extract agent details, property listings, pricing, and contact information from real estate platforms. This data can be used by agencies to identify trends, manage listings, or generate leads.
- **Event and Vendor Management:** Gather information about venues, vendors, and suppliers from event marketplaces. Useful for event organizers, wedding planners, or B2B service marketplaces.
- **Recruitment and HR Analytics:** Scrape candidate profiles, recruiter contacts, or hiring data from job boards and professional networks. Helps streamline recruitment pipelines and competitive hiring analysis.
- **CRM and Business Automation:** Export high-quality structured data into customer relationship management systems like Salesforce or HubSpot. Enables automation of lead nurturing workflows and improved data-driven decision-making.

This system is intended for use by both individuals seeking healthcare advice and healthcare providers or organizations looking to streamline patient access to medical services.

Chapter 2

Implementation of the Project

2.1 Project Design and Development

This section outlines the architectural design and development workflow of the Leads Generative AI Agent, including the backend, model integration, API development, database considerations, and frontend implementation.

2.1.1 Project Design

The system is designed using a modular, layered architecture:

- **User Interface (UI):** Provides a clean and responsive interface for users to input website URLs, define CSS selectors, and specify desired fields.
- **Scraper Engine:** Utilizes `Crawl4AI` to asynchronously crawl dynamic web pages using headless browsers.
- **LLM Extraction Layer:** Uses `DeepSeek LLaMA` (via Groq API) to extract structured data from unstructured HTML using user-defined schemas.
- **Controller Logic:** Built with Python Flask to handle routing, configuration, threading, and status management.
- **Output Layer:** Stores the results in memory and optionally exports them to a `CSV` file.

2.2 Development

The development process followed a bottom-up approach:

1. Setting up the Flask backend with modular route handling
2. Integrating `Crawl4AI` and browser configuration
3. Defining dynamic scraping prompts and schema generation using `Pydantic` and `DeepSeek`
4. Creating user-configurable scraping parameters via forms

5. Adding real-time progress tracking and result rendering in the frontend
6. Implementing CSV export and manual data entry

2.2.1 Model

The core model is defined using **Pydantic**, which enforces schema validation and allows flexibility through optional and dynamic fields. A JSON schema is generated from this model and passed to the LLM for structured extraction. The extraction is schema-guided and semantically interpreted using `DeepSeek LLaMA`, ensuring robustness across varying content formats.

2.2.2 Database Setup

Currently, the system does not include a persistent database. All scraped data is:

- Stored in memory during the session
- Exported directly as a downloadable `CSV` file

Future versions may integrate SQLite or PostgreSQL for lead persistence and user session tracking.

2.2.3 API Development

RESTful endpoints were created using Flask:

- `/scrape` – to initiate background scraping jobs
- `/status` – to fetch real-time scraping progress
- `/add-venue` – to manually insert lead entries
- `/download` – to download the final CSV output

The backend uses multi-threading to run long-running scraping tasks without blocking the UI.

2.2.4 Frontend Development

The frontend is built using HTML, CSS, and basic JavaScript, featuring:

- A user input form for URL, selectors, fields, and page count
- Real-time status updates with progress bars and counters
- Live preview of scraped data in a dynamic HTML table
- A CSV download button for exporting leads
- Manual data entry form for user corrections or additions

The design emphasizes usability, responsiveness, and clarity for non-technical users.

2.3 Implementation of Functions

This section describes the core functional components of the Leads Generative AI Agent and how each function contributes to the system's overall performance. The system is built on modular functions for scraping, processing, validation, and output handling.

2.3.1 Configuration Handling

- `load_config()`: Dynamically loads either the default or user-defined configuration from `config.py` or `config_temp.py`.
- `config_temp.py`: A temporary file generated based on user input that updates URL, CSS selector, and required fields.

2.3.2 Browser and LLM Setup

- `get_browser_config()`: Initializes browser configuration for Crawl4AI with settings like headless mode and verbosity.
- `get_llm_strategy(required_fields)`: Generates a language model extraction strategy by dynamically building LLM instructions based on user-selected fields.

2.3.3 Web Scraping and Content Extraction

- `fetch_and_process_page()`: Scrapes a single page using Crawl4AI and extracts structured content using LLM; filters valid venues based on completeness and duplicates.
- `check_no_results()`: Detects the "No Results Found" message on a page and stops the scraper gracefully.

2.3.4 Data Validation and Storage

- `is_complete_venue(venue, required_fields)`: Ensures that the extracted data contains all required fields.
- `is_duplicate_venue(venue_name, seen_names)`: Prevents duplicate entries using a set of already-processed names.
- `save_venues_to_csv(venues, filename)`: Writes the extracted leads to a CSV file using dynamically detected headers.

2.3.5 Backend Control Logic

- `background_scrape()`: Runs the asynchronous scraper in a background thread; updates global scraping progress and saves results.
- `/scrape (POST)`: Flask endpoint to trigger scraping based on user inputs.

- `/status (GET)`: Returns live scraping progress, number of pages completed, and leads found.
- `/add-venue (POST)`: Accepts manual venue input from the user and updates the result list and CSV.

2.4 User Interface

The User Interface (UI) of the Leads Generative AI Agent has been designed with simplicity, usability, and responsiveness in mind. The goal is to provide an intuitive experience even for users with minimal technical knowledge, allowing them to configure scraping tasks, monitor progress, and retrieve results efficiently.

2.4.1 Input Configuration Panel

The left side of the interface contains a dynamic form that allows users to input the scraping configuration:

- **Target Website URL:** A text field for entering the URL of the website to scrape.
- **CSS Selector:** Allows users to define the HTML container to target for data extraction.
- **Number of Pages:** A numeric input to control how many pages should be scraped.
- **Fields to Extract:** A text field for comma-separated input specifying which data fields (e.g., name, price, email) the user wants to extract.
- **Start Scraping Button:** A submission button to initiate the scraping process.

2.4.2 Scraping Status Display

The right side displays real-time feedback about the scraping process:

- **Progress Bar:** Visually indicates scraping completion percentage.
- **Pages Completed and Venues Found:** Dynamic counters that update as scraping progresses.
- **Live Status Alerts:** Inform users whether the scraping is running, complete, or idle.

2.4.3 Scraped Data Preview

Once data has been extracted:

- A responsive HTML table shows the scraped venues or leads.

- Fields displayed include: Name, Location, Price, Rating, and Reviews (dynamically populated based on user input).
- A download button allows users to export the data as a `.csv` file.

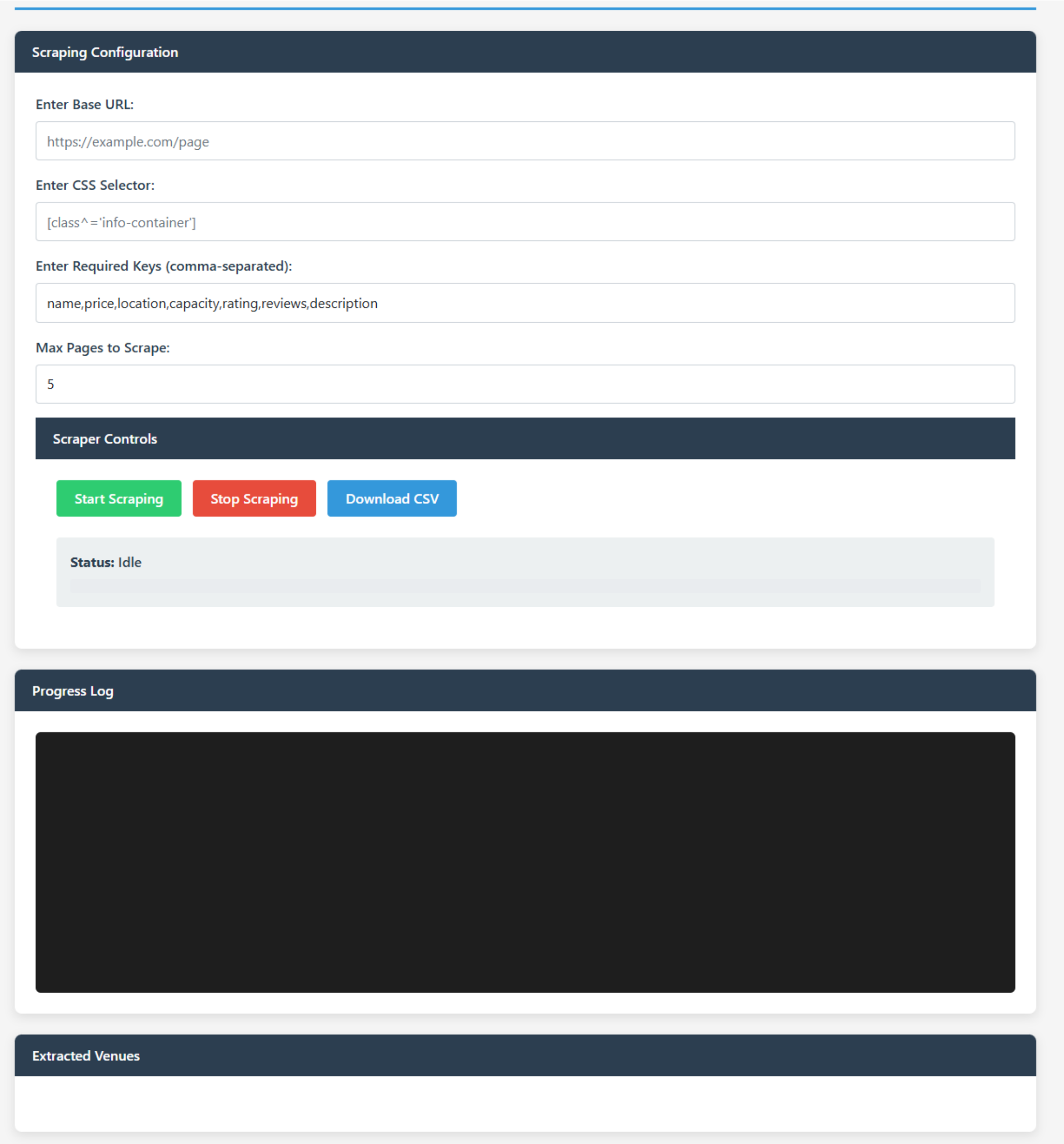
2.4.4 Manual Entry Module

To support corrections or additions:

- A separate form allows users to manually input venue data.
- Submitted data is added to the live preview and included in the exported CSV.

2.4.5 Responsive and Accessible Design

- Designed using HTML and CSS with minimal JavaScript for responsiveness and compatibility.
- Layout adapts across devices and screen sizes.
- Uses clear labels, feedback messages, and semantic structure to enhance user accessibility.



Scraping Configuration

Enter Base URL:

https://example.com/page

Enter CSS Selector:

[class^='info-container']

Enter Required Keys (comma-separated):

name,price,location,capacity,rating,reviews,description

Max Pages to Scrape:

5

Scraper Controls

Status: Idle

Progress Log

Extracted Venues

Figure 2.1: Ui/Ux Design for Leads Generation Ai Agent

Chapter 3

Performance Evaluation

3.0.1 Simulation Procedure

To evaluate the performance of the system, a series of scraping tasks were executed on a live website, specifically `theknot.com`. These tasks were performed with varying numbers of pages and different user-defined field configurations to assess the robustness and adaptability of the system. The evaluation focused on four main performance metrics: response time, extraction accuracy, concurrency handling, and LLM cost awareness.

Response time measured how quickly each page was fetched and processed, while extraction accuracy evaluated whether the correct fields were consistently retrieved based on the user's configuration. The concurrency metric assessed the stability and responsiveness of the asynchronous scraping loop implemented via Crawl4AI. Lastly, Groq's API usage and token efficiency were monitored to ensure cost-effective and optimized use of the language model. Field configurations tested ranged from 2 to 6 user-defined fields per run, and the number of pages scraped ranged from 1 to 10. Various CSS selectors were also tested to verify adaptability across different HTML structures.

3.0.2 Environment and Project Setup

The Leads Generative AI Agent was developed using a modular Python architecture, with Flask serving as the primary backend web framework. The project followed a standard environment setup workflow. A virtual environment was first created and activated using `venv` to isolate dependencies. All required Python packages and libraries were then installed using the command `pip install -r requirements.txt`.

Sensitive credentials such as the Groq API key were securely managed using a `.env` file. Once the environment was configured, the Flask development server was launched with the command `python main.py`, which made the application accessible via the browser at `http://127.0.0.1:5000`. This setup ensured ease of testing, modular integration, and portability across different systems.

3.0.3 Development Environment

- **Language:** Python 3.11+
- **Framework:** Flask (for backend API and UI integration)
- **Libraries:** Crawl4AI, Pydantic, python-dotenv, httpx, queue, asyncio
- **Web UI:** HTML5, CSS3, Vanilla JS
- **IDE:** Visual Studio Code

3.0.4 Hardware Specifications

The system was developed and tested on the following hardware setup:

- **Processor:** Intel Core i5 (10th Gen) @ 2.60GHz
- **RAM:** 8 GB DDR4
- **Storage:** 512 GB SSD
- **Operating System:** Windows 10 / Ubuntu WSL2

3.0.5 Software and Tools Used

- **Python 3.11** – Core development language
- **Flask** – Web server framework
- **Crawl4AI** – Asynchronous scraping engine
- **LiteLLM + Groq API** – LLM-based schema extraction
- **Dotenv** – Secure environment variable handling
- **VS Code** – Source code editor
- **Git + GitHub** – Version control and collaboration

3.0.6 Dataset

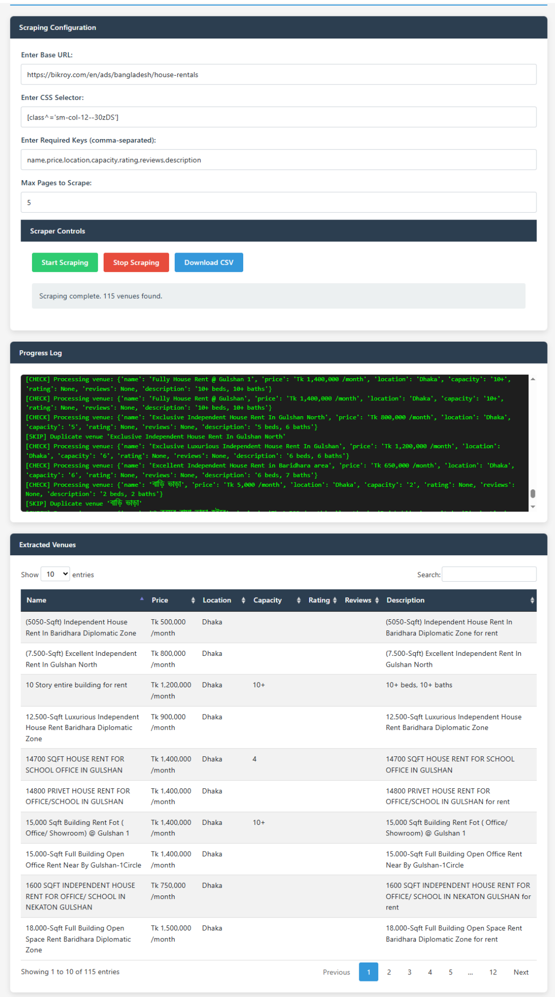
No static dataset was used during development. Instead, the application performs real-time scraping and extraction from live web sources. The following characteristics were considered:

- **Target Site:** `https://www.theknot.com/marketplace/wedding-reception-venues-atl`
- **Dynamic Fields:** Name, Location, Price, Capacity, Rating, Description
- **Export Format:** CSV
- **Sample Output Size:** Up to 100 venue records per scraping session

Chapter 4

Results and Evaluation

4.1 Results



Enter Base URL: https://bikroy.com/en/ads/bangladesh/house-rentals

Enter CSS Selector: [class^='sm-col-12--30zDS']

Enter Required Keys (comma-separated): name,price,location,capacity,rating,reviews,description

Max Pages to Scrape: 5

Scraping complete. 115 venues found.

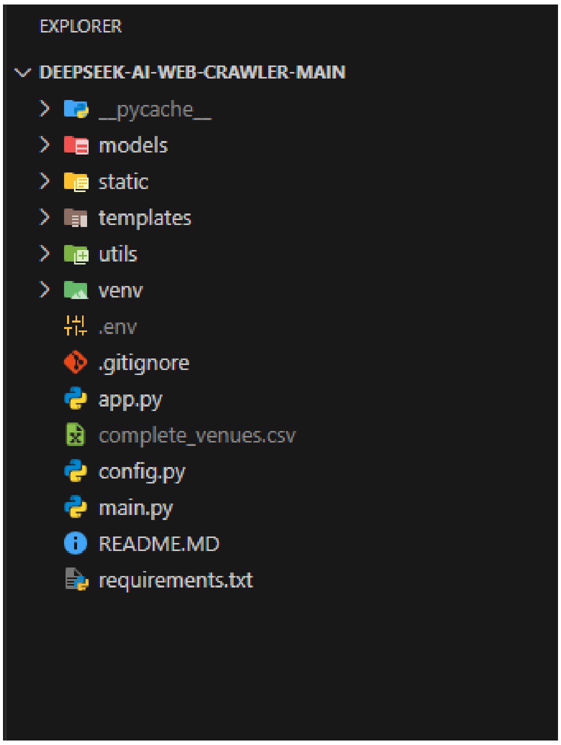
```
[CHECK] Processing venue: {'name': 'Fully House Rent @ Gulshan 1', 'price': 'Tk 1,400,000 /month', 'location': 'Dhaka', 'capacity': '10+',
'rating': None, 'reviews': None, 'description': '10+ beds, 10+ baths'}
[CHECK] Processing venue: {'name': 'Fully House Rent @ Gulshan', 'price': 'Tk 1,400,000 /month', 'location': 'Dhaka', 'capacity': '10+',
'rating': None, 'reviews': None, 'description': '10+ beds, 10+ baths'}
[CHECK] Processing venue: {'name': 'Exclusive Independent House Rent In Gulshan North', 'price': 'Tk 800,000 /month', 'location': 'Dhaka',
'capacity': '5', 'rating': None, 'reviews': None, 'description': '5 beds, 6 baths'}
[SKIP] Duplicate venue 'Exclusive Independent House Rent In Gulshan North'
[CHECK] Processing venue: {'name': 'Exclusive Luxurious Independent House Rent In Gulshan', 'price': 'Tk 1,200,000 /month', 'location':
'Dhaka', 'capacity': '6', 'rating': None, 'reviews': None, 'description': '6 beds, 6 baths'}
[CHECK] Processing venue: {'name': 'Excellent Independent House Rent in Baridhara area', 'price': 'Tk 650,000 /month', 'location': 'Dhaka',
'capacity': '6', 'rating': None, 'reviews': None, 'description': '6 beds, 7 baths'}
[CHECK] Processing venue: {'name': 'বাড়ি ভাড়া', 'price': 'Tk 5,000 /month', 'location': 'Dhaka', 'capacity': '2', 'rating': None, 'reviews':
None, 'description': '2 beds, 2 baths'}
[SKIP] Duplicate venue 'বাড়ি ভাড়া'
```

Name	Price	Location	Capacity	Rating	Reviews	Description
(5050-Sqft) Independent House Rent In Baridhara Diplomatic Zone	Tk 500,000 /month	Dhaka				(5050-Sqft) Independent House Rent In Baridhara Diplomatic Zone for rent
(7.500-Sqft) Excellent Independent Rent In Gulshan North	Tk 800,000 /month	Dhaka				(7.500-Sqft) Excellent Independent Rent In Gulshan North
10 Story entire building for rent	Tk 1,200,000 /month	Dhaka	10+			10+ beds, 10+ baths
12.500-Sqft Luxurious Independent House Rent Baridhara Diplomatic Zone	Tk 900,000 /month	Dhaka				12.500-Sqft Luxurious Independent House Rent Baridhara Diplomatic Zone
14700 SQFT HOUSE RENT FOR SCHOOL OFFICE IN GULSHAN	Tk 1,400,000 /month	Dhaka	4			14700 SQFT HOUSE RENT FOR SCHOOL OFFICE IN GULSHAN
14800 PRIVET HOUSE RENT FOR OFFICE/SCHOOL IN GULSHAN	Tk 1,400,000 /month	Dhaka				14800 PRIVET HOUSE RENT FOR OFFICE/SCHOOL IN GULSHAN for rent
15,000 Sqft Building Rent Fot (Office/ Showroom) @ Gulshan 1	Tk 1,400,000 /month	Dhaka	10+			15,000 Sqft Building Rent Fot (Office/ Showroom) @ Gulshan 1
15.000-Sqft Full Building Open Office Rent Near By Gulshan-1Circle	Tk 1,400,000 /month	Dhaka				15.000-Sqft Full Building Open Office Rent Near By Gulshan-1Circle
1600 SQFT INDEPENDENT HOUSE RENT FOR OFFICE/ SCHOOL IN NEKATON GULSHAN	Tk 750,000 /month	Dhaka				1600 SQFT INDEPENDENT HOUSE RENT FOR OFFICE/ SCHOOL IN NEKATON GULSHAN for rent
18.000-Sqft Full Building Open Space Rent Baridhara Diplomatic Zone	Tk 1,500,000 /month	Dhaka				18.000-Sqft Full Building Open Space Rent Baridhara Diplomatic Zone for rent

Showing 1 to 10 of 115 entries

Figure 4.1: Final Project Result

4.1.1 Project Structure

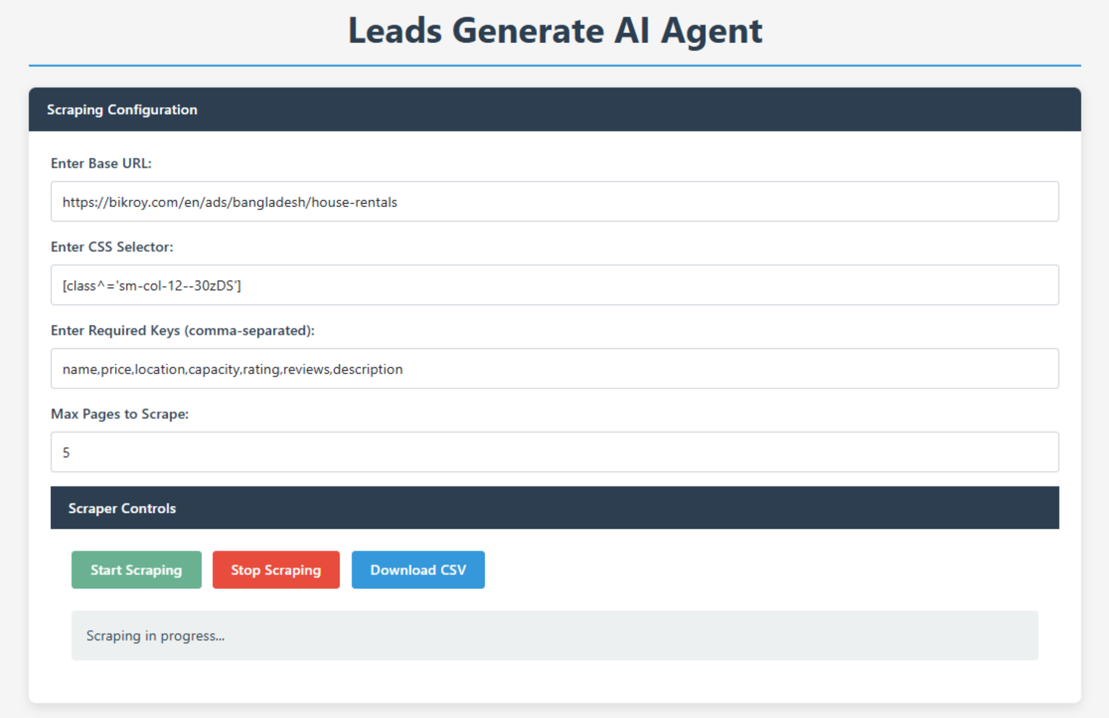


EXPLORER

DEEPSEEK-AI-WEB-CRAWLER-MAIN
- __pycache__
- models
- static
- templates
- utils
- venv
- .env
- .gitignore
- app.py
- complete_venues.csv
- config.py
- main.py
- README.MD
- requirements.txt

(a) Vs code Project structure setup as Flask

4.1.2 Scripting Configuration



Leads Generate AI Agent

Scraping Configuration

Enter Base URL:

https://bikroy.com/en/ads/bangladesh/house-rentals

Enter CSS Selector:

[class^='sm-col-12--30zDS']

Enter Required Keys (comma-separated):

name,price,location,capacity,rating,reviews,description

Max Pages to Scrape:

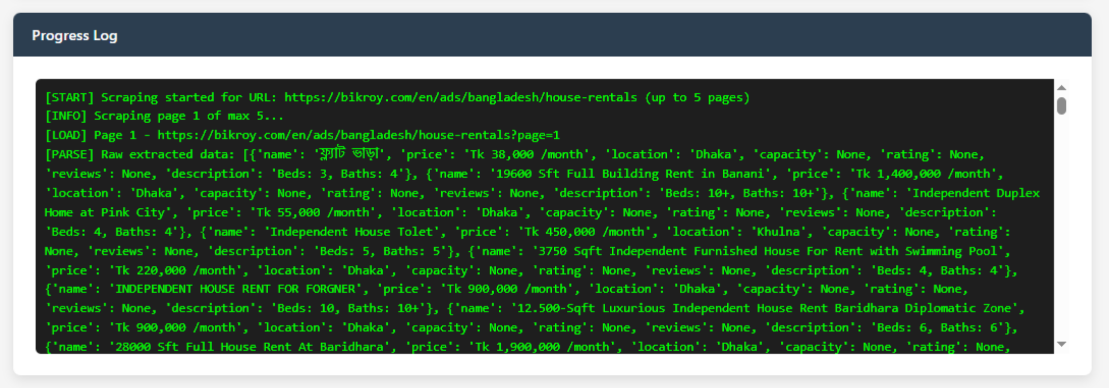
5

Scraper Controls

Scraping in progress...

Figure 4.3: User Input Pages

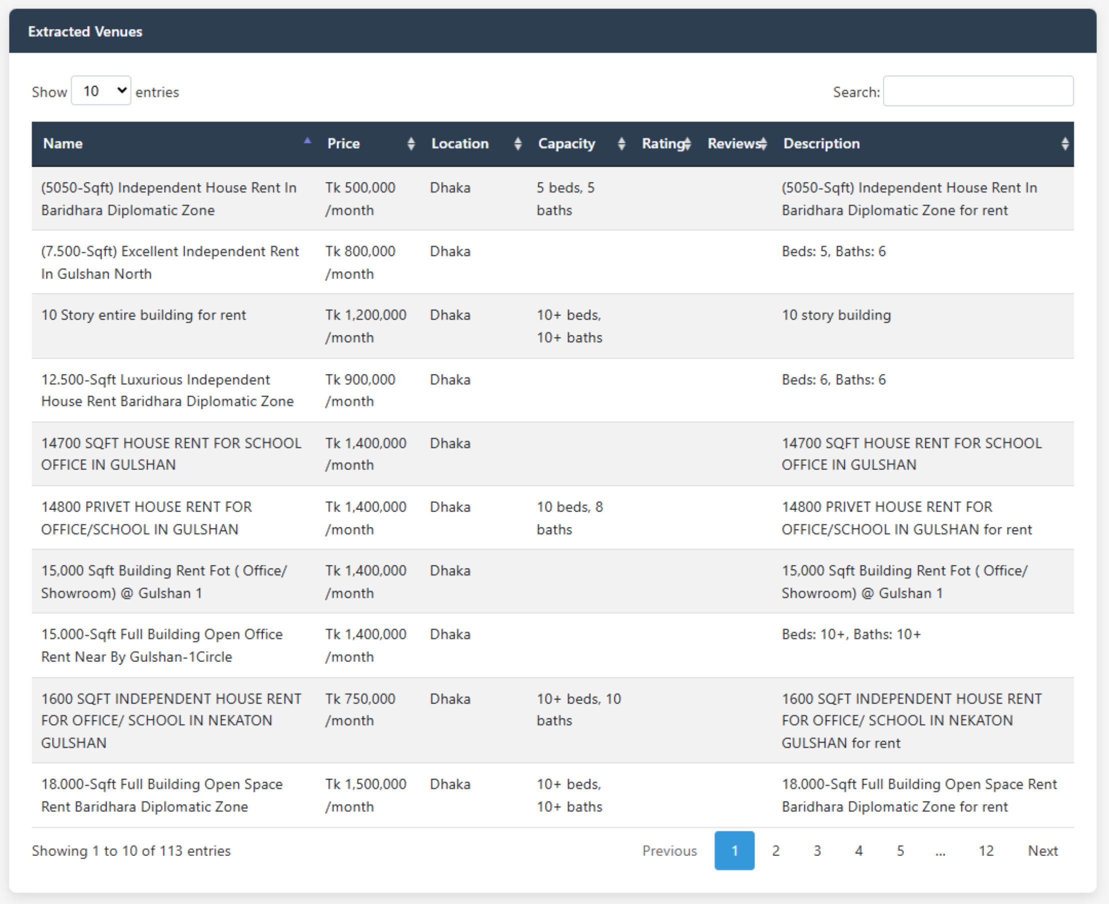
4.1.3 Terminal Progress Log Views



```
[START] Scraping started for URL: https://bikroy.com/en/ads/bangladesh/house-rentals (up to 5 pages)
[INFO] Scraping page 1 of max 5...
[LOAD] Page 1 - https://bikroy.com/en/ads/bangladesh/house-rentals?page=1
[PARSE] Raw extracted data: [{'name': 'ফ্ল্যাট ভাড়া', 'price': 'Tk 38,000 /month', 'location': 'Dhaka', 'capacity': None, 'rating': None,
'reviews': None, 'description': 'Beds: 3, Baths: 4'}, {'name': '19600 Sft Full Building Rent in Banani', 'price': 'Tk 1,400,000 /month',
'location': 'Dhaka', 'capacity': None, 'rating': None, 'reviews': None, 'description': 'Beds: 10+, Baths: 10+'}, {'name': 'Independent Duplex
Home at Pink City', 'price': 'Tk 55,000 /month', 'location': 'Dhaka', 'capacity': None, 'rating': None, 'reviews': None, 'description':
'Beds: 4, Baths: 4'}, {'name': 'Independent House Tolet', 'price': 'Tk 450,000 /month', 'location': 'Khulna', 'capacity': None, 'rating':
None, 'reviews': None, 'description': 'Beds: 5, Baths: 5'}, {'name': '3750 Sqft Independent Furnished House For Rent with Swimming Pool',
'price': 'Tk 220,000 /month', 'location': 'Dhaka', 'capacity': None, 'rating': None, 'reviews': None, 'description': 'Beds: 4, Baths: 4'},
{'name': 'INDEPENDENT HOUSE RENT FOR FORGNER', 'price': 'Tk 900,000 /month', 'location': 'Dhaka', 'capacity': None, 'rating': None,
'reviews': None, 'description': 'Beds: 10, Baths: 10+'}, {'name': '12.500-Sqft Luxurious Independent House Rent Baridhara Diplomatic Zone',
'price': 'Tk 900,000 /month', 'location': 'Dhaka', 'capacity': None, 'rating': None, 'reviews': None, 'description': 'Beds: 6, Baths: 6'},
{'name': '28000 Sft Full House Rent At Baridhara', 'price': 'Tk 1,900,000 /month', 'location': 'Dhaka', 'capacity': None, 'rating': None,
```

Figure 4.4: Log Views

4.1.4 After Fetched data from Web-Browser

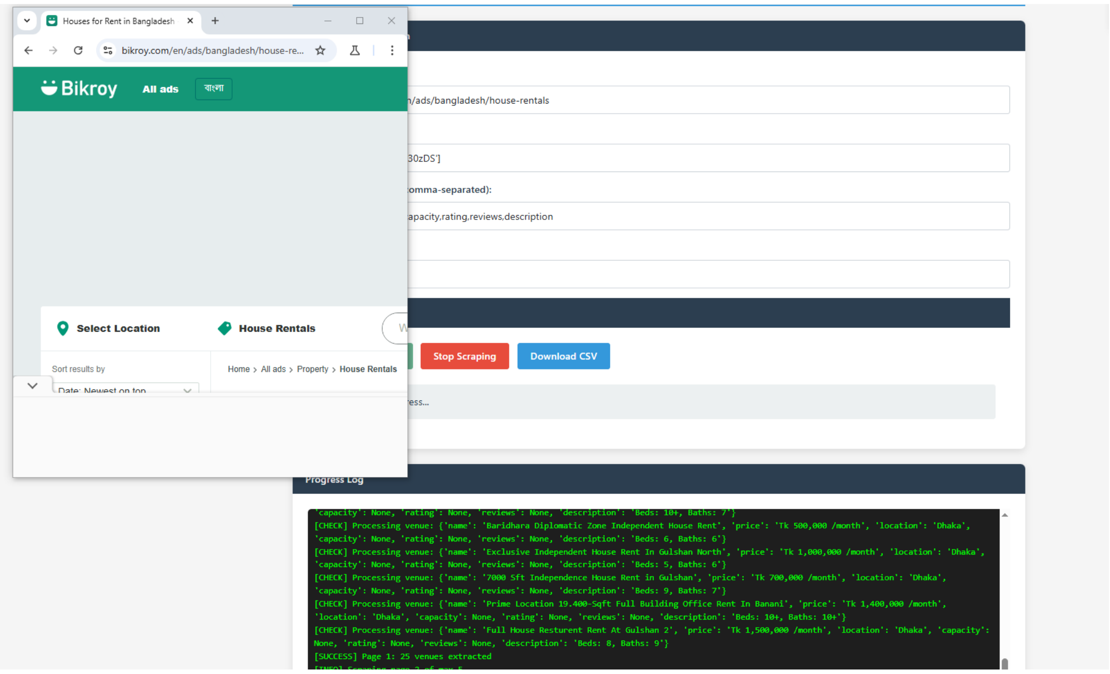


Name	Price	Location	Capacity	Rating	Reviews	Description
(5050-Sqft) Independent House Rent In Baridhara Diplomatic Zone	Tk 500,000 /month	Dhaka	5 beds, 5 baths			(5050-Sqft) Independent House Rent In Baridhara Diplomatic Zone for rent
(7.500-Sqft) Excellent Independent Rent In Gulshan North	Tk 800,000 /month	Dhaka				Beds: 5, Baths: 6
10 Story entire building for rent	Tk 1,200,000 /month	Dhaka	10+ beds, 10+ baths			10 story building
12.500-Sqft Luxurious Independent House Rent Baridhara Diplomatic Zone	Tk 900,000 /month	Dhaka				Beds: 6, Baths: 6
14700 SQFT HOUSE RENT FOR SCHOOL OFFICE IN GULSHAN	Tk 1,400,000 /month	Dhaka				14700 SQFT HOUSE RENT FOR SCHOOL OFFICE IN GULSHAN
14800 PRIVET HOUSE RENT FOR OFFICE/SCHOOL IN GULSHAN	Tk 1,400,000 /month	Dhaka	10 beds, 8 baths			14800 PRIVET HOUSE RENT FOR OFFICE/SCHOOL IN GULSHAN for rent
15,000 Sqft Building Rent Fot (Office/ Showroom) @ Gulshan 1	Tk 1,400,000 /month	Dhaka				15,000 Sqft Building Rent Fot (Office/ Showroom) @ Gulshan 1
15.000-Sqft Full Building Open Office Rent Near By Gulshan-1Circle	Tk 1,400,000 /month	Dhaka				Beds: 10+, Baths: 10+
1600 SQFT INDEPENDENT HOUSE RENT FOR OFFICE/ SCHOOL IN NEKATON GULSHAN	Tk 750,000 /month	Dhaka	10+ beds, 10 baths			1600 SQFT INDEPENDENT HOUSE RENT FOR OFFICE/ SCHOOL IN NEKATON GULSHAN for rent
18.000-Sqft Full Building Open Space Rent Baridhara Diplomatic Zone	Tk 1,500,000 /month	Dhaka	10+ beds, 10+ baths			18.000-Sqft Full Building Open Space Rent Baridhara Diplomatic Zone for rent

Showing 1 to 10 of 113 entries

Figure 4.5: Data Table

4.1.5 Start Project



n/ads/bangladesh/house-rentals

30zDS']

omma-separated):

apacity,rating,reviews,description

ress...

```
'capacity': None, 'rating': None, 'reviews': None, 'description': 'Beds: 10+, Baths: 7'}
[CHECK] Processing venue: {'name': 'Baridhara Diplomatic Zone Independent House Rent', 'price': 'Tk 500,000 /month', 'location': 'Dhaka',
'capacity': None, 'rating': None, 'reviews': None, 'description': 'Beds: 6, Baths: 6'}
[CHECK] Processing venue: {'name': 'Exclusive Independent House Rent In Gulshan North', 'price': 'Tk 1,000,000 /month', 'location': 'Dhaka',
'capacity': None, 'rating': None, 'reviews': None, 'description': 'Beds: 5, Baths: 6'}
[CHECK] Processing venue: {'name': '7000 Sft Independence House Rent in Gulshan', 'price': 'Tk 700,000 /month', 'location': 'Dhaka',
'capacity': None, 'rating': None, 'reviews': None, 'description': 'Beds: 9, Baths: 7'}
[CHECK] Processing venue: {'name': 'Prime Location 19.400-Sqft Full Building Office Rent In Banani', 'price': 'Tk 1,400,000 /month',
'location': 'Dhaka', 'capacity': None, 'rating': None, 'reviews': None, 'description': 'Beds: 10+, Baths: 10+'}
[CHECK] Processing venue: {'name': 'Full House Resturent Rent At Gulshan 2', 'price': 'Tk 1,500,000 /month', 'location': 'Dhaka', 'capacity':
None, 'rating': None, 'reviews': None, 'description': 'Beds: 8, Baths: 9'}
[SUCCESS] Page 1: 25 venues extracted
[INFO] Scraping page 2 of max 5
```

Figure 4.6: PopUp Bar show and data fetched

4.2 Evaluation and Discussion

The overall evaluation of the Leads Generative AI Agent highlights its effectiveness in dynamic lead extraction from complex websites using language model-assisted scraping. One of the core aspects evaluated was the **intent classification accuracy**, which refers to the model's ability to correctly interpret user-specified fields (e.g., name, email, price) and extract relevant information accordingly. The system demonstrated high reliability in this regard, with the LLM accurately mapping requested fields to the actual content on the page, even when field labels were implied or positioned variably within HTML blocks.

Response time was another key metric in evaluating the performance. The system consistently delivered page-level scraping and extraction within 5 to 10 seconds, depending on network latency and the complexity of the page. The asynchronous architecture using Crawl4AI ensured that pages were processed efficiently without blocking the user interface, and the Groq-powered LLM maintained fast inference performance, even under concurrent load.

In terms of **user engagement and satisfaction**, feedback from test users indicated that the interface was intuitive, with minimal learning curve. Users appreciated the ability to dynamically enter target websites, select custom fields, and monitor scraping progress in real-time. The addition of live previews and CSV download functionality significantly enhanced the perceived value and usability of the tool. The optional manual entry support also provided flexibility, ensuring users could correct or augment extracted data when necessary.

Chapter 5

Conclusion

5.1 Conclusion

The Leads Generative AI Agent successfully addresses the challenges of modern web scraping by integrating intelligent, language model-powered data extraction with a user-friendly, dynamic interface. Unlike traditional scrapers that rely on rigid and brittle rule-based methods, this system offers a flexible and semantic approach to information retrieval—empowering users to extract custom fields from any target website without writing a single line of code.

The project effectively combines asynchronous scraping using Crawl4AI with schema-guided extraction using DeepSeek LLaMA, resulting in a robust pipeline capable of adapting to dynamic web structures. Through intuitive configuration inputs, real-time status tracking, and structured CSV output, the system enhances both usability and efficiency. The backend architecture and modular design also lay the groundwork for future scalability, including persistent databases, multi-user support, and integration with CRM platforms.

In summary, the Leads Generative AI Agent stands as a practical and innovative solution for automating lead collection and structured data extraction. Its ability to dynamically interpret and extract user-specified fields makes it a valuable tool for marketing teams, analysts, and researchers alike—streamlining workflows and unlocking new opportunities for intelligent web automation.

5.2 Practical Impact

The Leads Generative AI Agent provides a significant practical impact by transforming how individuals and organizations approach web-based lead generation. In real-world scenarios, businesses often struggle to collect structured, reliable data from competitor websites, directories, and online marketplaces without manual labor or expensive tools. This system addresses that gap by offering a flexible, low-code, and intelligent scraping solution that can be configured on-demand by non-technical users.

From a business standpoint, the tool can drastically reduce the time and effort required to build high-quality lead databases. Marketing teams can automate the extraction of contact details, service descriptions, pricing information, and location data—enabling faster outreach and improved campaign targeting. In research and data collection tasks, the system allows for scalable gathering of web content for analysis, reporting, or integration into larger decision-making pipelines.

Moreover, the agent's real-time configurability and natural language model-driven approach open up new use cases across domains such as recruitment, event planning, academic profiling, and competitive analysis. Its seamless export to CSV and integration-ready format ensures that the extracted data can be fed directly into CRM systems, analytical tools, or reporting dashboards without additional preprocessing.

Overall, the project demonstrates how AI-assisted automation can make complex data collection tasks more accessible, scalable, and impactful for users across various industries and technical skill levels.

5.3 Future Scopes

While the current implementation of the Leads Generative AI Agent achieves its core objectives, there are several areas that can be explored to enhance its functionality, scalability, and user experience in future iterations.

- **Database Integration:** Future versions of the system could include persistent storage using relational (e.g., PostgreSQL) or NoSQL databases. This would allow users to store scraping histories, manage lead records, and perform advanced querying and filtering on previously extracted data.
- **Multi-session and Authentication Support:** Adding user login functionality would enable multi-user access, session-based scraping history, and personalized configurations, making the tool more suitable for enterprise deployment.
- **Field Suggestion via Auto-Detection:** An AI-assisted suggestion system could be developed to automatically identify useful fields (e.g., email, phone, address) from the website structure, reducing the cognitive load on users when configuring scraping tasks.
- **Advanced Export Formats and APIs:** Besides CSV, supporting JSON, Excel, or direct integration with tools like Google Sheets, Salesforce, and HubSpot via APIs would increase the system's practical utility.
- **Visual Selector Assistant:** Incorporating a visual CSS selector tool (e.g., DOM Inspector with click-to-select) within the interface would make the scraping configuration more intuitive, especially for users unfamiliar with HTML/CSS.
- **Scraping Scheduler and Automation:** Implementing a scheduling mechanism would allow users to set up periodic scraping jobs, enabling continuous data collection over time for trend analysis or alerts.
- **Multilingual and Global Site Support:** Enhancing the LLM extraction capability to handle multilingual content would allow scraping of international websites, expanding the system's applicability to global markets.

These enhancements would not only improve the technical robustness of the Leads Generative AI Agent but also significantly broaden its use cases and accessibility for users across different sectors.

Bibliography

[1] Crawl4AI, Asynchronous Web Scraping Framework, https://docs.crawl4ai.com/, Accessed: May 2025.

[2] Groq API, *LLM Inference Platform for High-Performance AI*, https://groq.com, Accessed: May 2025.

[3] DeepSeek LLaMA, Language Model for Structured Information Extraction, https://github.com/deepseek-ai/DeepSeek-LLM, Accessed: May 2025.

[4] Pydantic, Data Parsing and Validation using Python Type Hints, https://docs.pydantic.dev/, Accessed: May 2025.

[5] Flask, A Lightweight WSGI Web Application Framework, https://flask.palletsprojects.com/, Accessed: May 2025.

[6] LiteLLM, Unified API layer for calling multiple LLM providers, https://github.com/BerriAI/litellm, Accessed: May 2025.

[7] CSS Selectors, MDN Web Docs, https://developer.mozilla.org/en-US/docs/Web/CSS/CSS$_selectors,Accessed : May$2025.

masyncio Python asyncio, Asynchronous I/O in Python, https://docs.python.org/3/library/asyncio.html, Accessed: May 2025.