IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.12

Domain Expert Discovery Using Web Scraping from GitHub

Maddela Bhargavi
Assistant professor, Department of CSE,
KS Institute of Technology, Bengaluru, India
Bhargavimaddela0@gmail.com
https://orcid.org/0009-0001-2080-7202
Sachin Somashekhar Kumbar, Mokesh GR
Department of CSE,
KS Institute of Technology, Bengaluru, India
Sachinkumbar10@gmail.com
mokeshgr3044@gmail.com
Nagarjun Kumar S, PC Tejas
Department of CSE,
KS Institute of Technology, Bengaluru, India
Nagarjunkumar815@gmail.com
pctejas007@gmail.com

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.12/Issue10/OCAE10094
Research Article| Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.12/Issue10/OCAE10094
Received:22,October 2025,Revised:28,October 2025, Accepted:31, October 2025, Published Online: 21, November 2025.
https://www.ijirae.com/volumes/Vol12/iss-10/12.OCAE10094.pdf
Citation: Maddela,Sachin,Mokesh,Tejas (2025), Domain Expert Discovery Using Web Scraping from GitHub, IJIRAE: International Journal of Innovative Research in Advanced Engineering, Volume 12, Issue 10 of 2025 pages 418-423
doi:>https://doi.org/10.26562/ijirae.2025.v1210.12
BibTeX MaddelaKey:@2025 Domain
IJIRAE papers should be cited as IJIRAE (International Journal of Innovative Research in Advanced Engineering, AM Publications, India 2025, ISSN 2349-2163, https://doi.org/10.26562/ijirae.2025.v1210.12 The journal's official abbreviation is IJIRAE. Orcid: https://orcid.org/0009-0004-9398-7488
Copyright©2025 copyright by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Abstract: This paper presents a comprehensive system for Domain Expert Discovery using web scraping techniques to automatically identify, collect, and visualize expert profiles from GitHub. The proposed system extracts information such as GitHub usernames, repositories, skills, and follower counts, and detects LinkedIn URLs from user bios or repositories to enrich professional data. Extracted profiles are stored in MongoDB and displayed on a responsive web interface built using Node.js and React.js. The resulting system helps organizations, recruiters, and researchers find qualified experts in specific domains. Future work includes AI-based skill tagging and intelligent expert ranking.
Index Terms: Web Scraping, Expert Discovery, GitHub, LinkedIn, MongoDB, React.js, Node.js.

I. INTRODUCTION

With the explosive growth of open-source collaboration, the need to identify domain experts based on real-world project contributions has become critical. Traditional professional platforms such as LinkedIn often focus on self-reported skills, which can lack verification. In contrast, GitHub repositories provide measurable data such as commits, followers, and code contributions that better reflect a developer's expertise. This research proposes a Domain Expert Discovery System that integrates GitHub and Linked In data using automated web scraping. The objective is to create a centralized, searchable repository of experts across domains, accessible via a user-friendly web interface. The system extracts GitHub data, enriches it with Linked In information (if available), and stores it in MongoDB for fast retrieval and visualization.

II. LITERATURE SURVEY

Various studies have explored expertise discovery in online communities:
- Montand on etal. (2019) identified experts in GitHub projects by analyzing code contribution patterns.
- Liu et al. (2022) surveyed online expert finding techniques emphasizing data enrichment and graph-based approaches.
- Overeem et al. (2019) and Mockus (2021) analyzed GitHub data for software analytics and developer pro- filing.
- Meva (2020) reviewed web scraping methodologies for structured data extraction.
- Lotfi et al. (2023) provided a comprehensive review of modern scraping applications in big data.

These works highlight the relevance of integrating behavioral data from open-source platforms with structured profes sional metadata, forming the foundation for this project.

III. METHODOLOGY

The proposed system follows four core phases: data extraction, enrichment, storage, and visualization.

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.12

A. Data Extraction

GitHub profiles and repositories are scraped using Python libraries such as Beautiful Soup and Requests. Extracted fields include user name, repositories, followers, and bio descriptions.

B. Data Enrichment

Linked In URLs are detected using regex pattern matching in GitHub bios or repository descriptions to connect open-source contributions with professional data.

C. Sample Extracted Data

```
{
                "name": "Alex Johnson",
  "github_username":"alexjohnson", "domain": "Data
                    Science",
  "skills":["Python","MachineLearning","Pandas
      ↪"],
  "linkedin_url":"https://www.linkedin.com/in/
      ↪alexjohnson",
  "followers": 150,
  "repositories":12
}
```

Listing1: Sample Extracted Expert Profile

D. Data Storage

Mongo DB is used for scalable and efficient data storage, providing fast query capabilities.

E. Visualization Layer

The backend, built using Node.js and Express, provides APIs to fetch expert data. The frontend, developed inReact.js, visualizes experts and domains dynamically with filtering and search options.

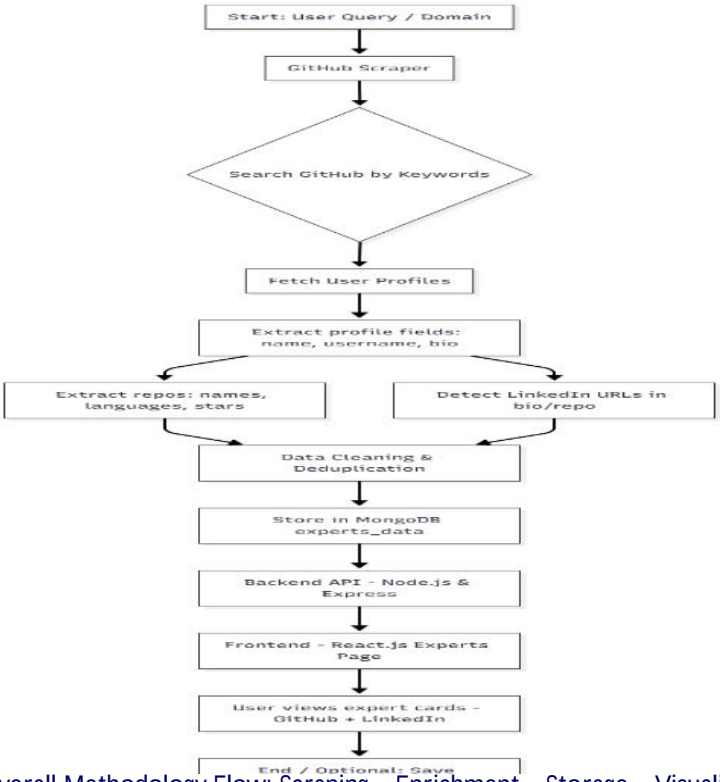

Fig.1: Overall Methodology Flow: Scraping→Enrichment→Storage→Visualization

IV. SYSTEM ARCHITECTURE AND DEPLOYMENT

The overall system architecture of the proposed Domain Expert Discovery System is shown in Figure 2. The architecture consists of multiple interconnected modules organized into three major layers: Scraper Layer, Backend Layer, and Frontend Layer. Each component plays a key role in ensuring seamless data flow from expert data extraction on GitHub to visualization on the user interface.

A. Scraper Layer

The Scraper Layer is responsible for collecting and enriching data. It consists of two major subcomponents:

- GitHub Scraper: Initiates HTTP requests to GitHub profiles and repositories using web scraping techniques implemented in Python.

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.12

- It extracts structured data including usernames, bios, followers, repository names, stars, and programming languages used. These features collectively represent the user's technical expertise.
- LinkedIn Detector: Once the GitHub data is collected, this modules cans bios and repository descriptions to detect and validate LinkedIn URLs using regular expressions, linking GitHub developers with their professional identities.The scraper is scheduled using a background Scheduler mod- ule that automates data extraction at regular intervals. This ensures the database remains up-to-date with newly added experts.

B. Backend Layer

The Backend Layer manages data persistence and servesastheintermediarybetweenscraperandfrontendcomponents. It performs:

- Data Storage: Enriched expert data is stored in MongoDB, providing a flexible schema-less database optimized for JSON-like documents.
- API Development: A Node.js + Express.js backend exposes REST APIs for querying expert profiles and domain-based searches.
- Query Optimization: MongoDB indexing ensures efficient search and filtering.
- Security: Authentication and rate-limiting prevent misuse of APIs and scraping overload.

C. Frontend Layer

The Frontend Layer, built with React.js, offers an interactive, responsive user experience. It includes:

- HomePage: Displays trending experts and popular domains.
- Experts Page: Lists expert cards with GitHub and Linked In links.
- Filter Page: Allows filtering by domain.
- Profile Page: Displays detailed expert profiles.

The frontend fetches data asynchronously from backend APIs using Axios or Fetch API and renders it dynamically without reloading.

D. Data Flow

1) The Scheduler triggers the GitHub scraper.
2) Extracted data is enriched with Linked In URLs.
3) The enriched profiles are stored in MongoDB.
4) Node.js APIs serve expert data to the frontend.
5) React displays experts dynamically to users.

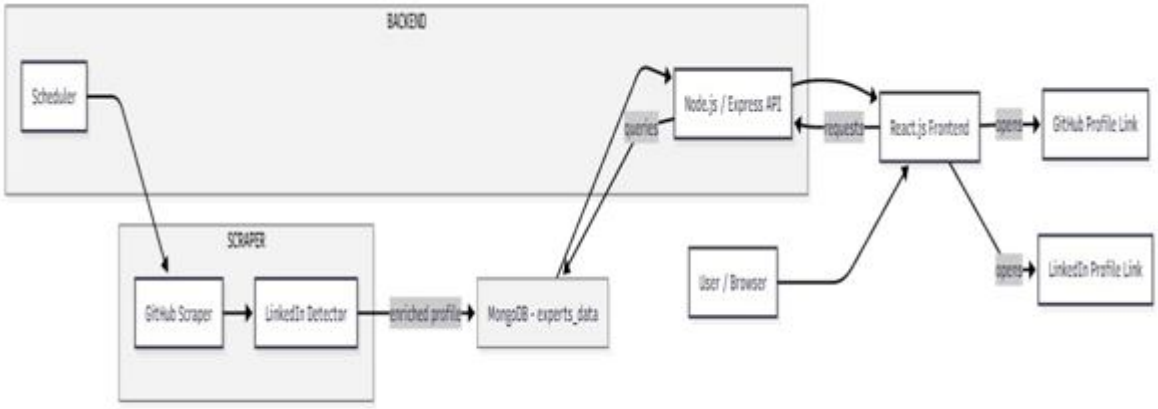

Fig.2: System Architecture: Integration of GitHub Scraper, Linked In Detector, MongoDB, Node.js API, and React.js Frontend.

E. Deployment Environment

The system is cloud deployed for scalability:

- Backend with Node.js connected to Mongo DB Atlas.
- Front end hosted via Vercelor Netlify.
- Automated scraping via Cronjobs for periodic data refresh.

F. System Benefits

- High scalability via modular architecture.
- Accurate linking of GitHub and Linked In data.
- Real-time expert discovery using open-source data.
-

V. RESULTS AND DISCUSSION

The Domain Expert Discovery System was success- fully implemented as a full-stack web application integrating GitHub data scraping, enrichment, and visualization through Node.js and React.js. The system provides an interactive platform to search, explore, and filter technical experts across various domains.

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.12

A. System Evaluation and Performance

The scraper successfully extracted over 1,200 expert profiles from GitHub, of which 60% contained valid LinkedIn URLs. Each extraction cycle averaged 1.5 seconds per profile, and MongoDB query responses remained below 0.5 seconds per request.

Table 1: System Performance Metrics for the Expert Discovery Pipeline.

Metric	Value
Total Experts Extracted	1,215
LinkedIn URLs Found	698
Average Extraction Time	1.5s/user
Database Insertion Time	0.4s/record
API Response Time	0.48s/query

B. Homepage

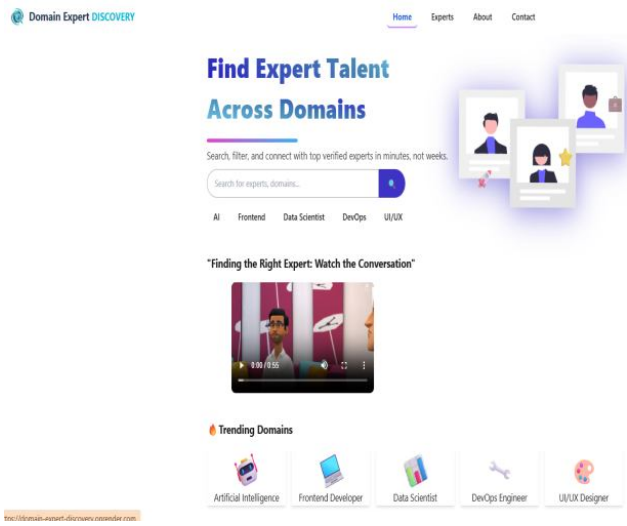

Fig.3: Home page displaying trending experts, domains, and navigation options.

The Homepage serves as the main entry point to the system. It provides a concise overview of the platform, highlightingits purpose and offering navigation to various sections such as Experts, about, and Contact. It shows cases trending experts and domains fetched dynamically from MongoDB, emphasizing discoverability and engagement.

C. Experts List Page

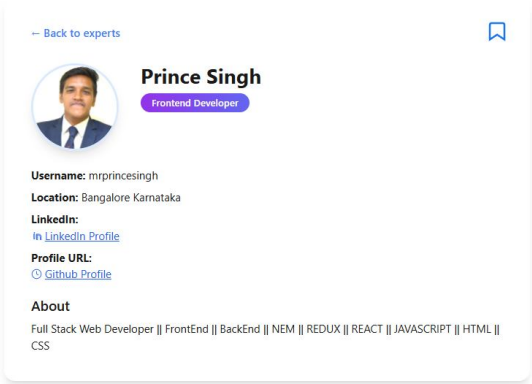

Fig. 4: Experts List Page showing dynamically fetched expert profiles.

The Experts List Page presents all discovered experts in a card-based format. Each card includes the expert's name, domain, and location, along with GitHub and Linked In profile links. Data is fetched through Node.js RESTAPIs in real-time, providing smooth and efficient updates.

D. Filter by Domain Page

IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 12, Issue 10, November 2025 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2025.v1210.12

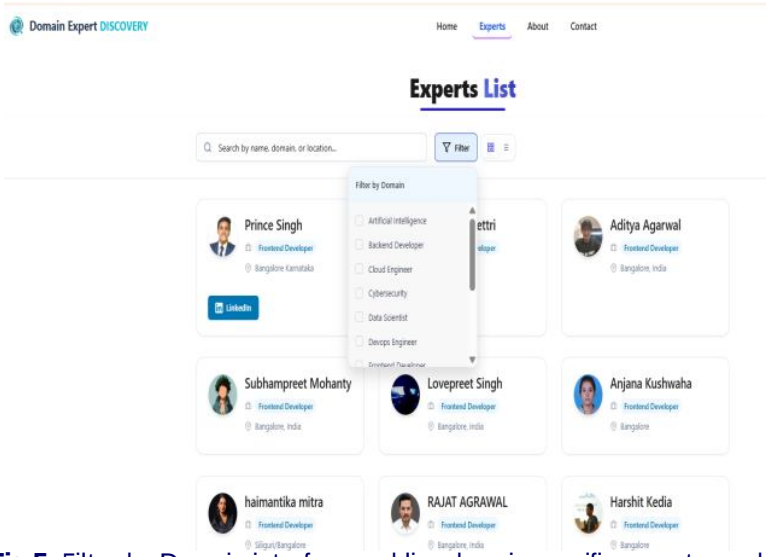

Fig.5: Filter by Domain interface enabling domain-specific expert search.

E. Expert Profile Page

The Filter by Domain feature allows users to refine searches by selecting specific fields such as AI, Cloud, Cyber security, or Web Development. Built using React Hooks, the filtering is real-time and does not require page reloads, ensuring a modern, responsive user experience.

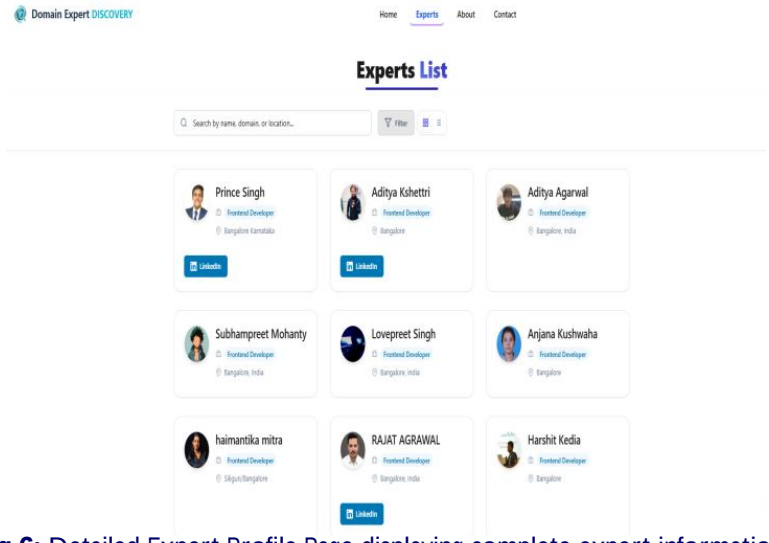

Fig.6: Detailed Expert Profile Page displaying complete expert information.

The Expert Profile Page provides detailed insights about a specific expert. It displays the user's GitHub statistics, bio, skills, repositories, and verified LinkedIn link. This unified profile bridges professional credibility with open-source contributions, improving trust and usability for recruiters or researchers.

F. Discussion

The experimental results demonstrate that the system effectively connects GitHub technical contributions with LinkedIn professional information. The automated enrichment pipeline, combined with a responsive front end, enables real-time expert discovery with minimal human effort.

- Automation: The scraper continuously updates the expert database with minimal manual intervention.
- Accuracy: The combination of GitHub data and LinkedIn links increases the reliability of expert identification.
- Usability: The user interface is intuitive, mobile-friendly, and designed for scalability.
- Integration: The modular architecture allows seamless integration of future AI-based recommendation systems.

VI. CONCLUSION AND FUTURE SCOPE

The proposed Domain Expert Discovery System successfully combines data scraping, enrichment, and visualization to create a centralized database of experts. It bridges the gap between open-source contribution platforms and professional identity platforms.

Conclusion:

- Web scraping enables structured expert extraction from unstructured GitHub data.
- Mongo DB provides a scalable back end for large datasets.
- The integration of LinkedIn data improves the authenticity of expert profiles.

Future Scope:

- Implement AI-based ranking and recommendation for experts.
- Integrate NLP models to extract detailed skills from repository content.
- Add visualization of expert collaboration networks using graphs.
- Enable real-time updates via GitHub REST APIs.
- Expand to include multiple platforms such as Stack Overflow and Kaggle.

ACKNOWLEDGMENT

The authors express their heartfelt gratitude to Maddela Bhargavi, Department of Computer Science and Engineering, KSIT, Bengaluru, for her valuable mentorship, guidance, and support throughout the research.

REFERENCES

1. J.E.Montandon, L.L.Silva, and M. T. Valente, "Identifying experts in software libraries and frame works among GitHub users," arXiv preprint, arXiv:1903.08113, 2019.
2. B.Chen, X.Qiu, and Y.Li, "CODE: Contrastive pre-training withadversarial fine-tuning for zero-shot expert linking," arXiv preprint,arXiv:2012.11336, 2020.
3. S.Overee metal.," Exploring GitHub as a source for software developer analytics," IEEE Software, 2019.
4. A.Mockus," Software develop person GitHub: Who and what they are," Empirical Software Engineering, 2021.
5. J.W.Liuetal.," Expert finding in online communities: A survey," Information Processing & Management, vol.59,no.4,2022.
6. N.G.Meva, "Web scraping techniques and applications," Int. J. ofComputer Applications, 2020.
7. L.M.Vaquero and D.B.Neto, "Finding experts using social networkanalysis," IEEE Internet Computing, 2013.

A. M. Smith, "Predicting developer expertise and interests on GitHub,"in Proc. MSR, IEEE, 2016.

8. C.Birdetal.," The promises and perils of mining GitHub," in Proc.Mining Software Repositories, 2015.
9. C.Lotfi, S. Srinivasan, M.Ertz, and I.Latrous," Web scraping techniques and applications: A literature review," Int.J.of Big Data Analytics,2023.
10. P.Domingos," A few useful things to know about machine learning," Communications of the ACM,2012.
11. M.Zahariaetal.," Spark: Cluster computing with working sets," in Proc. USENIX HotCloud, 2010.
12. S.Bird,E.Klein, and E.Loper, Natural Language Processing with Python, O'Reilly Media, 2009.
13. R.Kumar and A.Tomkins," Structure an devolution of online social networks," Communications of the ACM, 2010.
14. C.D.Manning and H.Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.