

WEB SCRAPER REPORT

PROJECT DESCRIPTION

The project automates data gathering through web scraping using Python. It is structured into three modules for better flexibility: -

Main Module – Connects to the web, scrapes data, and updates datasets.

Interface Module – Provides a user-friendly GUI built with Tkinter.

Processing Module – Handles data retrieval and processing tasks like fetching player stats, ranking top performers, and generating fantasy teams. Key functions include Update, Search, GetData, TopBat, TopBowl, and GetFantasy11.

The system uses urllib for making web requests and BeautifulSoup for parsing HTML. Extracted data is stored in CSV files and presented via the GUI.

ADVANTAGES, DISADVANTAGES, AND APPLICATIONS

Advantages: - Quick collection of real-time data. - High accuracy with user-friendly operation. - Easy development due to Python libraries.

Disadvantages: - Needs changes if website structure updates. - Limited to one website at a time.

Applications: - Price comparison, e-commerce, and real estate. - Academic and sports data analysis. - Market and competition monitoring.

RESULTS AND DISCUSSION

The scraper reduces manual data collection time from about 90 minutes to 2–3 minutes, saving effort and resources. The GUI makes it simple for users to interact with the system without coding knowledge.

Future Scope – Extend scraping to multiple sites, use NLP/OCR for scanned data, and offer customizable scraping options.

Conclusion – The project demonstrates that automated data scraping is reliable, efficient, and scalable, making it a practical solution for real-world data needs.