AUTOMATED WEB SCRAPER

Introduction

An Automated Web Scraper is a software tool designed to extract specific information from websites without manual effort. In today's data-driven world, businesses and developers rely heavily on up-to-date online data such as product prices, news articles, job listings, and research data. Collecting such information manually is time-consuming and inefficient.
This project focuses on building an automated system that fetches, processes, and stores web data in a structured format, reducing human effort and improving accuracy.

Abstract

The Automated Web Scraper is a software tool designed to extract structured information from websites without manual interaction. As businesses increasingly rely on online data for decision-making, collecting information manually becomes time-consuming, repetitive, and error-prone. This project aims to build an automated system that fetches web data, processes it, and stores it in a usable format, enabling faster insights and reducing human effort.

Problem Statement

- Many websites contain large amounts of valuable information, but extracting that data manually is slow, repetitive, and prone

to errors. Organizations often require real-time or frequent updates, which makes manual collection impractical.
There is a need for an automated solution that can:

- ◈ Fetch data consistently and accurately
- ◈ Handle frequent updates
- ◈ Extract only the required information

Features

- ◈✓ Automatic Data Extraction
- ◈Collects data from targeted websites automatically using libraries such as BeautifulSoup, Requests, or Selenium.
- ◈✓ Selective Scraping
- ◈Allows choosing specific elements like:
- ◈Titles
- ◈Prices
- ◈Links
- ◈Images

Scope

✅ Data Extraction from Web Pages

- The system will automatically extract data from selected web pages based on predefined rules (e.g., product details, news headlines, prices, blog content).

✅ Support for Multiple Websites (Configurable)

- User can configure multiple target websites/URLs.
- Each website can have its own scraping rules/selectors (like CSS selectors/XPaths), stored in a configuration file or database.

✅ Scheduling & Automation

- The scraper can be scheduled to run at specific intervals (e.g., hourly, daily, weekly).
- Once scheduled, it runs without manual intervention and collects new/updated data.