Magic Bricks – India Real Estate Website

Business Problem

People looking to buy property find it difficult to compare prices, locations, and property details because real-estate information is spread across many listings on different web pages. This makes decision-making slow and confusing.

Problem Solving / Solution Approach

This project solves the problem by using web scraping to automatically collect property details such as price, location, area, property type, and posted date from real-estate websites and store them in a structured CSV file. The collected data helps users easily compare properties and understand market trends.

Website Details (For Project Submission)

Primary Website

– Indai MagicBricks a Real Estate Listings
🔗 https://www.magicbricks.com/ **Website Used:**
MagicBricks (https://www.magicbricks.com/)

Data Source URL:
https://www.magicbricks.com/property-for-sale/residential-real-estate?cityName=Hyderabad

Description:
This page contains residential property listings in Hyderabad, including price, location, area, property type, and posted date, which were collected using web scraping techniques.

Tools, Software, and Libraries Used (With Explanation)

1. Python

- **What it is:** A programming language.
- **Why used:** Python is simple, powerful, and widely used for web scraping and data analysis.
- **Role in project:** Used to write the scraping code and process data.

Jupyter Notebook / Python Script

- **What it is:** An environment to run Python code.
- **Why used:**
  - Jupyter helps test code step by step
  - Python script (.py) helps run the full program at once
- **Role in project:** Used to write and execute the scraping program.

Google Chrome Browser

- **What it is:** A web browser.
- **Why used:** Selenium controls Chrome to load web pages like a real user.
- **Role in project:** Opens the Magic Bricks website and loads dynamic content

Selenium

- **What it is:** A browser automation library.
- **Why used:** Magic Bricks loads data using JavaScript, which normal requests cannot handle.
- **Role in project:**
  - Opens the website
  - Scrolls pages
  - Loads property listings dynamically

Chrome Driver

- **What it is:** A driver that connects Selenium with Chrome.
- **Why used:** Selenium needs ChromeDriver to control the browser.

- **Role in project:** Allows Selenium to interact with the Chrome browser.

WebDriver Manager

- **What it is:** A Python library that automatically downloads the correct Chrome Driver.
- **Why used:**
    - Avoids manual driver installation
    - Prevents version mismatch errors
- **Role in project:** Automatically manages Chrome Driver setup

Beautiful Soup

- **What it is:** An HTML parsing library.
- **Why used:** Extracts data from the web page source.
- **Role in project:**
    - Reads HTML content
    - Finds property details like price, location, area, etc.
    - **Pandas**
- **What it is:** A data analysis library.
- **Why used:** To store scraped data in table format.
- **Role in project:**
    - Creates Data Frame
    - Saves data into a CSV file

CSV File

- **What it is:** A structured data file format.
- **Why used:** Easy to open in Excel and analyze.
- **Role in project:** Final output file containing scraped property data.

Time Module

- **What it is:** A built-in Python module.
- **Why used:** Adds delays between actions.

- **Role in project:**
  - Prevents website blocking
  - Allows page content to load properly

Libraries Installed (Command Used)

```
!pip install selenium web driver-manager beautifulsoup4 pandas
```

Website Used

- **MagicBricks – India Real Estate Website**
- Used to collect residential property listings.

Data Collected

- Property price
- Location
- Property type
- Area (sq ft)
- Posted date

(Personal data like phone numbers or names were NOT collected for ethical reasons.)

One-Line Summary (Very Important for Viva)

This project uses Python, Selenium, Beautiful Soup, and Pandas to scrape real-estate property listings from Magic Bricks and store the data in a CSV file for analysis.