

# Action Plan

Tuesday, December 24, 2024 12:11 AM

## Scraping the real-estate listings from 99acres:

✓ 1. Set browser Options before instantiating the web driver

✓ 2. Visit the target URL: <https://www.99acres.com/>

✓ 3. Type Chennai in the search bar

✓ 3. Click on Search button

✓ 4. Adjust the Budget slider to 5cr INR

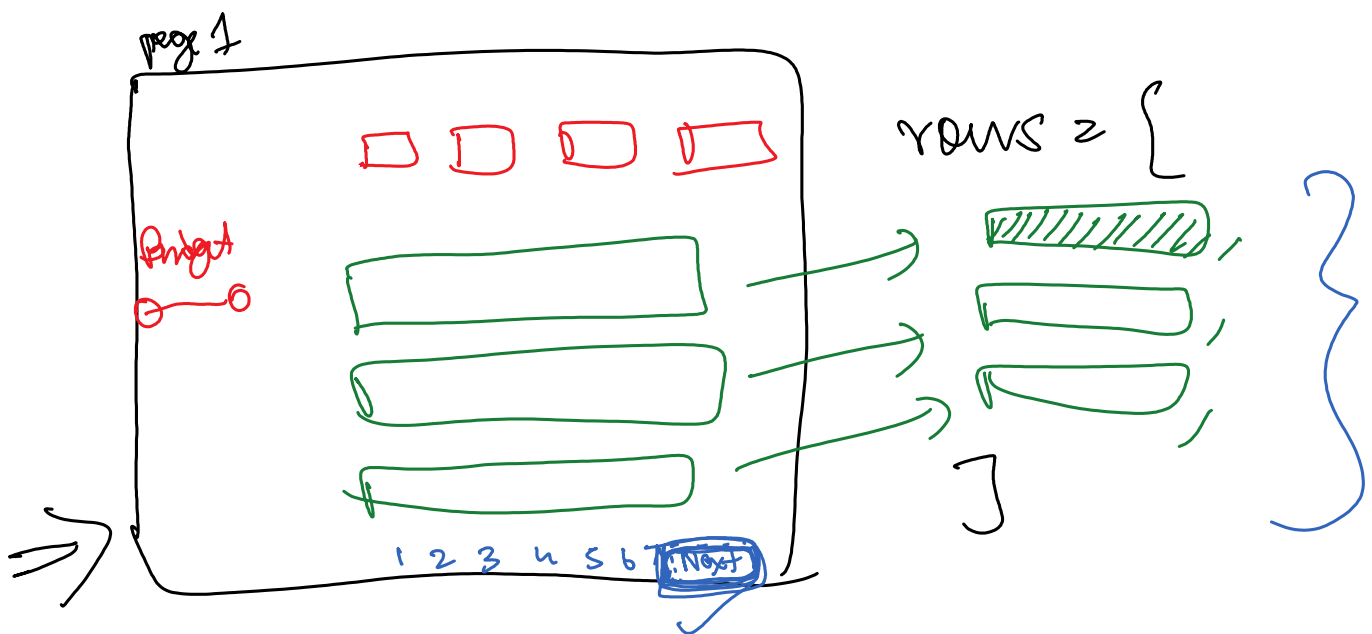
✓ 5. Click on the below options to filter the resulting data:

- Verified
- Ready To Move
- With Photos
- With Videos

✓ 6. Scrape the necessary data

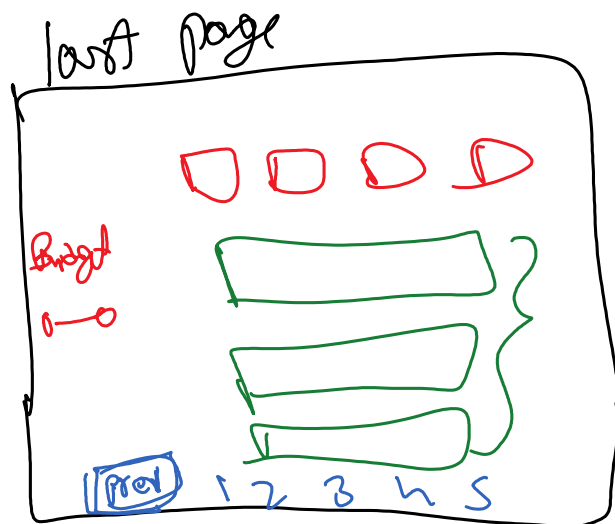
✓ 7. Clean the extracted data

✓ 8. Export cleaned data as an excel sheet



## Inside page: (2)

- 1st to 2nd last page
1. Identify "next page" button ✓
  2. Scroll down to it ✓
  3. Extract page data ✓
  4. Click on "next page" button ✓
  5. Wait 5s ✓

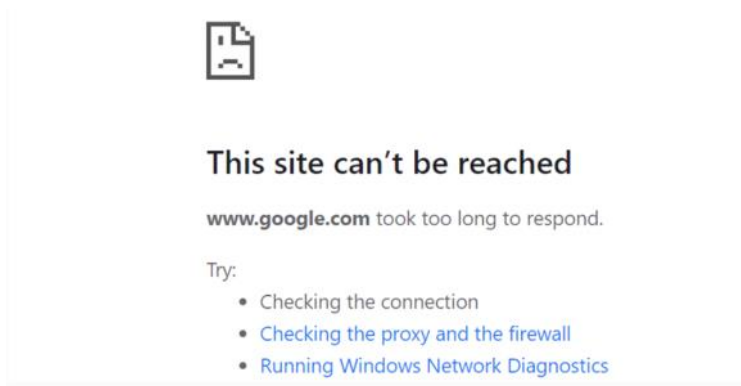


1. Try to identify "next page" btn
2. Break out of loop

# Prerequisites

Tuesday, December 24, 2024 10:46 PM

## 1. "This site can't be reached":



### • Causes:

- The ChromeDriver or WebDriver version might not match the installed browser version
- A proxy or VPN might interfere with the request
- Firewall or corporate network restrictions may block access
- Websites often restrict access to bots or automated tools like Selenium
- The website might have anti-scraping mechanisms in place
- The certificate or SSL/TLS version might not be supported or configured correctly

### • Solutions:

- Ensure your WebDriver matches the browser version
- Configure browser options in Selenium
- Force the browser to use HTTP/1.1
- Use a user-agent string to make requests appear like they are coming from a browser
- Run the browser in incognito mode

## 2. Selenium Options:

- In Selenium, **Options** is used to customize and configure browser settings and behavior when automating browsers
- Allows to specify preferences, enable or disable features, and set options that are specific to the browser you are automating
- Each browser driver has its own Options class to provide these configurations

### Common Usecases:

- Running browser in headless mode (without GUI)
- Disabling browser notifications
- Setting the default download directory

- Adding browser extensions

### **Key Methods:**

- **add\_argument(arg)**: Adds a command-line argument to the browser
- **add\_experimental\_option(name, value)**: Adds experimental options or preferences
- **set\_capability(name, value)**: Sets desired capabilities for the browser

### **Advantages:**

- Helps customize browser behavior according to your testing needs
- Suppress unnecessary UI elements like notifications or pop-ups
- Ensure the browser runs with specific settings each time, promoting consistency
- Improves efficiency as we can run browsers in headless mode to save resources during testing

### **3. getBoundingClientRect():**

- JavaScript function, used to manipulate and interact with elements on a webpage by executing JavaScript
- useful when standard Selenium methods can't achieve certain tasks directly, such as precise scrolling or positioning
- This approach ensures that Selenium scripts can interact with dynamically loaded or partially visible elements more reliably

### **Advantages:**

- Helps in cases where the elements are not immediately visible in the viewport
- Handles scenarios where the webpage layout changes, requiring precise adjustments
- Ensures elements are well-placed for user interaction or screenshots
- Works even when the standard Selenium methods (`scrollIntoView()`, `actions.move_to_element()`, etc.) don't work as intended

