Selenium-Based Automation for E-Paper PDF Collection

1. Introduction

The project involves automation for downloading PDF files from an online newspaper using a web automation framework, Selenium. The main objective of this project is to simplify the process of accessing and saving the daily e-paper without manual intervention. Selenium, paired with browsers WebDriver, allows for easy interaction with web elements that change frequently like buttons on links for downloading PDFs.

2.Overview

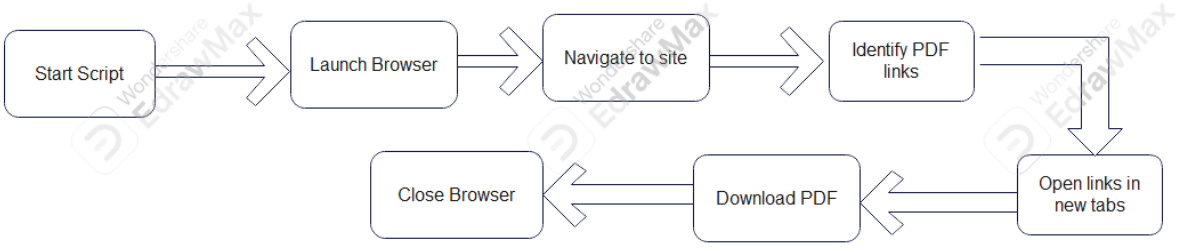
The project is designed to visit Gorkhapatra Online e paper website, identify all available PDF files on page, download them to a specified folder. This process involves accessing Gorkhapatra website, identifying PDF links using xpath expressions and automating the browsers action to save PDF files directly on the disk.

3.Methodology

The methodology of this project is divided into several stages:

- Setting up environment
  Installing libraries such as Selenium and configuring the Chrome WebDriver.
- Launching the browser
  A chrome instance is launched using Selenium's WebDriver, with configurations to automatically handle PDF downloads.
- Navigating to the website
  The script navigates to the target website, ie Gorkhapatra Online e-paper.
- Locating the PDF links
  The program waits until the PDF links are available on the page and identifies them using Xpath.
- Downloading the PDFs
  Each PDF link is opened in a new tab, triggering a download into the specified directory.
- Error handling
  The program includes error handling for common issues like timeouts or missing elements.

4.Block Diagram



4.1.Workflow

- Start the Script
  Initialize the environment and browser settings.

- Launch Browser
  Use Selenium WebDriver to launch the browser with custom options for PDF handling.

- Navigate to Website
  Go to Gorkhapatra Online e-paper website.

- Identify PDF link
  Wait for the PDF elements to load and identify them using Xpath.

- Open Links in new Tabs
  Open each link in new tab and trigger downloads.

- Download PDF
  Automatically download the PDFs into the specified directory.

- Close the Browser
  Close the tabs and end script.

5.Requirements

5.1 Software

- Python 3.9 or higher
- Selenium: For automating web actions.

- Chrome Driver: To control the Chrome browser.

5.2 Libraries

- Selenium was installed using (pip install selenium).

5.3 Hardware:

- Any modern computer with an internet connection to run the script.
- Sufficient storage to save downloaded PDFs.

5.4 Environment setup

1. Install the necessary Python libraries.
2. Download and set up ChromeDriver to match running Chrome browser version.
3. Set the default download directory in script where PDF will be saved.

6.Conclusion

The project automates the tedious task of downloading PDFs from news website using Selenium, making the process faster and eliminating the need for manual intervention saving a lot of time. Automation can also be adapted for other website and application that involve downloading resources or interacting with web elements programmatically.