Automated News Headline

TECHNOLOGY

Web Automation & Web Scraping:

Web Automation involves using scripts or programs to perform tasks on the internet automatically. This can range from filling out forms, interacting with web applications, or extracting data from websites. On the other hand, Web Scraping is a specific form of automation focused on extracting structured information from web pages. It involves parsing HTML and navigating the Document Object Model (DOM) to locate and extract relevant data. Popular tools for web scraping include Selenium, which automates browser actions, and libraries like BeautifulSoup and Scrapy for parsing HTML content.

XPath:

XPath (XML Path Language) is a powerful and versatile language for navigating and querying XML and HTML documents. In the context of web automation, XPath plays a crucial role in precisely locating and extracting specific elements on a webpage. It utilizes a path notation, similar to navigating a file system, to define the location of elements within the document's hierarchical structure. XPath expressions can be used to identify elements based on their attributes, tag names, or relationships with other elements. In web automation, Selenium, a widely used tool for browser automation, integrates XPath for locating elements on web pages. Web developers and automation engineers use XPath to create robust and reliable locators, ensuring accurate interaction with the target elements during automated testing or data extraction tasks. XPath's flexibility makes it a valuable asset in web automation, providing a standardized and efficient way to navigate the intricate structure of HTML documents.

Selenium Library:

Selenium is employed for web automation, allowing scripted interactions with web browsers. In this project, Selenium facilitates the navigation of the Chrome browser to the Hindustan Times website, the identification of specific HTML elements using XPath, and the extraction of news headlines and links. Its versatile capabilities make it an essential tool for automating browser-based tasks.

Pandas Library:

Pandas is utilized for data manipulation and analysis. In this project, once the news headlines and links are extracted, Pandas is instrumental in organizing this data into a structured format. The extracted information is structured into a Pandas DataFrame, providing a convenient way to handle and export the data to a CSV file. Pandas simplifies tasks related to data handling, making it an efficient choice for this project.

PyInstaller Library:

PyInstaller is used to convert the Python script into a standalone executable. This step is crucial for distributing the automation script as an independent application. PyInstaller packages the script along with its dependencies, enabling users to run the automation task without needing to install Python separately. This is particularly valuable for ensuring the script's portability and usability across different environments.

ChromeDriver:

ChromeDriver serves as the WebDriver interface for Selenium to control the Chrome browser. It acts as a bridge between Selenium commands in the script and the Chrome browser, enabling automated browser interactions. In this project, ChromeDriver is essential for initiating and configuring the Chrome browser, navigating to the website, and locating specific HTML elements. Its integration with Selenium is pivotal for the success of the web automation task.

CODING

```
# Import necessary libraries
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
import pandas as pd
from datetime import datetime
import os
import sys

# Get the path of the directory where the Python executable is located
application_path = os.path.dirname(sys.executable)
# The executable will be in the same folder when created

# Get the current date and format it as 'ddmmyyyy'
now = datetime.now()
month_day_year = now.strftime("%d%m%Y")
# Use strftime.org to set the format of date and time
# Target website for web scraping
website = "https://www.hindustantimes.com/"

# Path to the ChromeDriver executable
path = "C:/Users/ankit/Downloads/Compressed/chromedriver-win64/chromedriver-win64/chromedriver.exe"

# Set up Chrome options for a headless browser and initialize the ChromeDriver service
options = Options()
options.headless = True
service = Service(executable_path=path)

# Create a WebDriver instance using Chrome, configured with options and service
driver= webdriver.Chrome(service=service, options=options)
# Open the specified website in the Chrome browser
```

```
driver.get(website)

# Locate all <h3> elements with the class "hdg3" using XPath
containers = driver.find_elements(by="xpath" , value='//h3[@class="hdg3"]')

# Initialize empty lists to store extracted titles and links
titles = []
links = []

# Loop through each container and extract title and link information
for container in containers:
    title = container.find_element(by="xpath" , value='./a').text
    titles.append(title)
    link = container.find_element(by="xpath" , value='./a').get_attribute("href")
    links.append(link)
    # (//h3[@class="hdg3"] == .) - XPath expression used for locating elements
# Create a dictionary with titles and links
my_dic = {'titles': titles, 'links': links}

# Create a Pandas DataFrame from the dictionary
df_headlines = pd.DataFrame(my_dic)

# Define the filename for the CSV file with the current date
file_name = f'headline-{month_day_year}.csv'

# Combine the application path and the filename to create the final file path
final_path = os.path.join(application_path, file_name)

# Export the DataFrame to a CSV file at the specified path
df_headlines.to_csv(final_path)

# Close the Chrome browser
driver.quit()
```

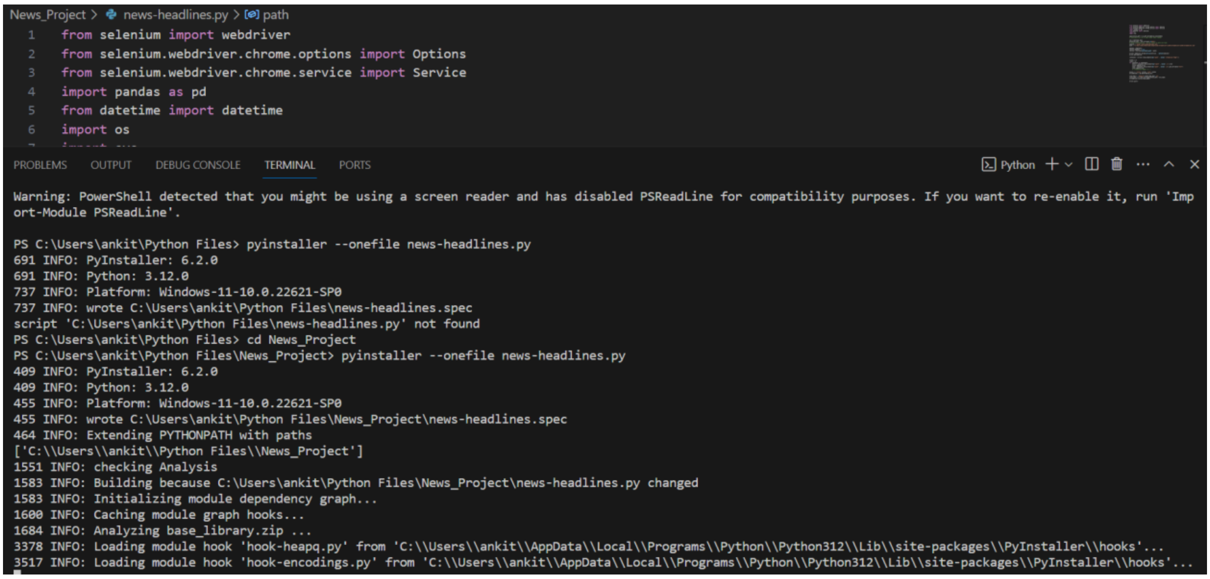
IMPLEMENTATION

1. Python Script to Executable (.exe):

Convert the Python script to an executable format (.exe).

Ensure the script file is in the same directory when running this conversion in the terminal.

Explanation: The initial step involves converting the Python script into an executable file (.exe) for ease of distribution and execution.



```python
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
import pandas as pd
from datetime import datetime
import os
```

```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Imp
ort-Module PSReadLine'.

PS C:\Users\ankit\Python Files> pyinstaller --onefile news-headlines.py
691 INFO: PyInstaller: 6.2.0
691 INFO: Python: 3.12.0
737 INFO: Platform: Windows-11-10.0.22621-SP0
737 INFO: wrote C:\Users\ankit\Python Files\news-headlines.spec
script 'C:\Users\ankit\Python Files\news-headlines.py' not found
PS C:\Users\ankit\Python Files> cd News_Project
PS C:\Users\ankit\Python Files\News_Project> pyinstaller --onefile news-headlines.py
409 INFO: PyInstaller: 6.2.0
409 INFO: Python: 3.12.0
455 INFO: Platform: Windows-11-10.0.22621-SP0
455 INFO: wrote C:\Users\ankit\Python Files\News_Project\news-headlines.spec
464 INFO: Extending PYTHONPATH with paths
['C:\\Users\\ankit\\Python Files\\News_Project']
1551 INFO: checking Analysis
1583 INFO: Building because C:\Users\ankit\Python Files\News_Project\news-headlines.py changed
1583 INFO: Initializing module dependency graph...
1600 INFO: Caching module graph hooks...
1684 INFO: Analyzing base_library.zip ...
3378 INFO: Loading module hook 'hook-heapq.py' from 'C:\\Users\\ankit\\AppData\\Local\\Programs\\Python\\Python312\\Lib\\site-packages\\PyInstaller\\hooks'...
3517 INFO: Loading module hook 'hook-encodings.py' from 'C:\\Users\\ankit\\AppData\\Local\\Programs\\Python\\Python312\\Lib\\site-packages\\PyInstaller\\hooks'...
```

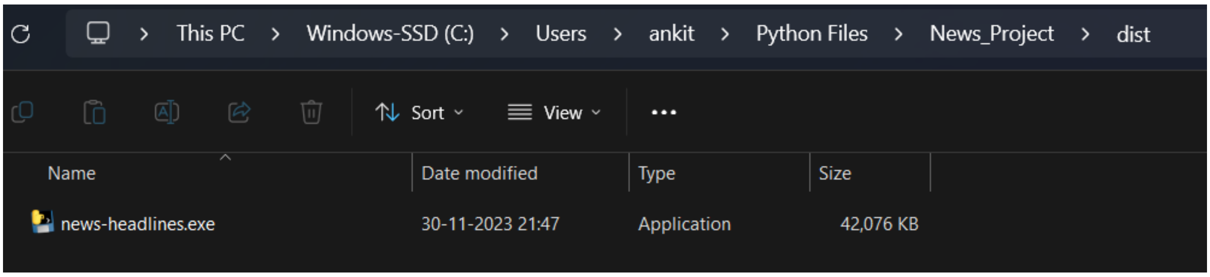
Figure 1: Python Script Conversion to Executable Format

2. Creation of 'dist' Folder:

Upon successful compilation, a 'dist' folder is created in the same directory.

Inside 'dist,' the compiled executable file, named 'news-headlines.exe,' is located.

Explanation: The compilation process generates a 'dist' folder housing the executable file, streamlining the organization of the project files.



This PC > Windows-SSD (C:) > Users > ankit > Python Files > News_Project > dist
Sort
View
•••

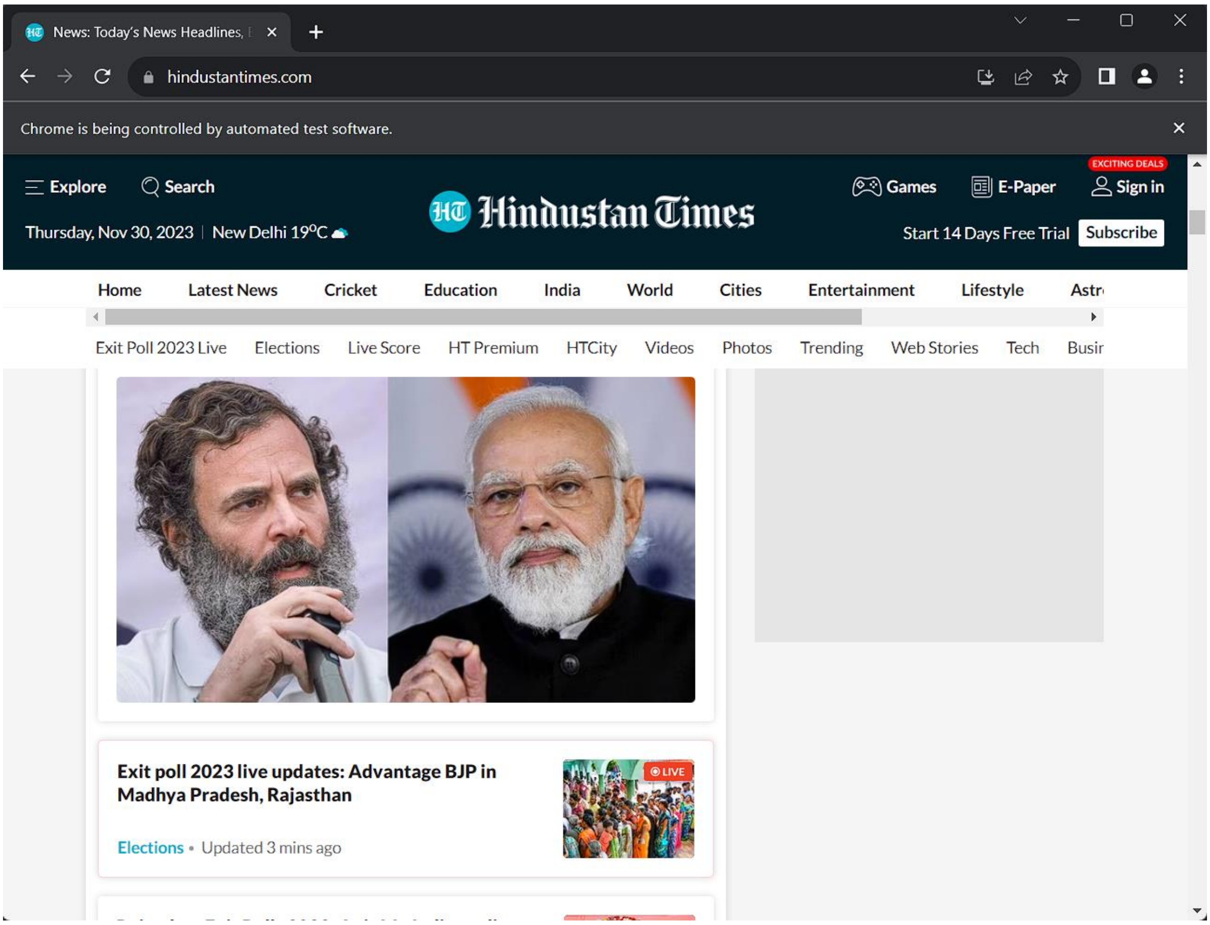
Name	Date modified	Type	Size
news-headlines.exe	30-11-2023 21:47	Application	42,076 KB

Figure 2: Successful Creation of Executable File

3. Automatic Website Interaction:

Execute 'news-headlines.exe' to automatically open and close the Hindustan Times website within a few seconds.

Explanation: The executable initiates the web automation, navigating to the specified website, and concludes the task swiftly.



Chrome is being controlled by automated test software.

Thursday, Nov 30, 2023 | New Delhi 19ºC

Exit poll 2023 live updates: Advantage BJP in Madhya Pradesh, Rajasthan

Elections • Updated 3 mins ago

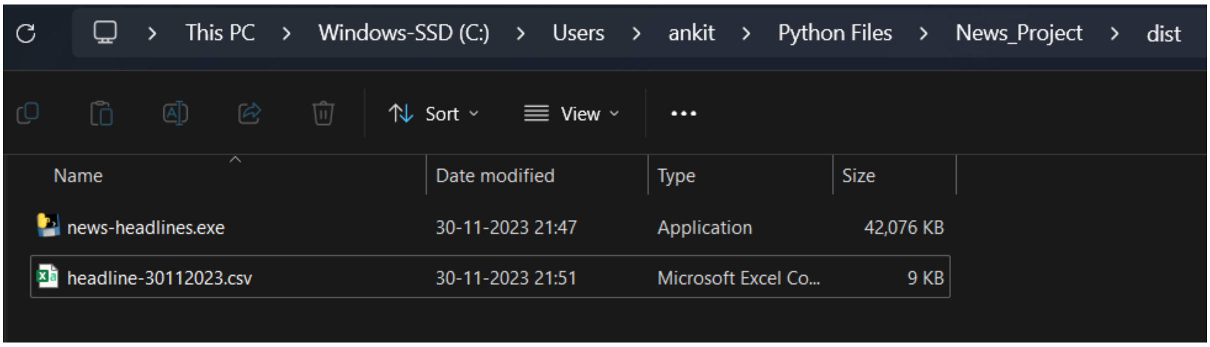
Figure 3: Automated Opening of Hindustan Times Website

4. CSV File Generation:

Post-execution, a new CSV file named 'headline-30112023.csv' is generated in the 'dist' directory.

The file name reflects the date in the format 'ddmmyyyy' when it was created.

Explanation: The automated process results in the creation of a CSV file containing extracted news headlines and links, uniquely named based on the execution date.



This PC > Windows-SSD (C:) > Users > ankit > Python Files > News_Project > dist

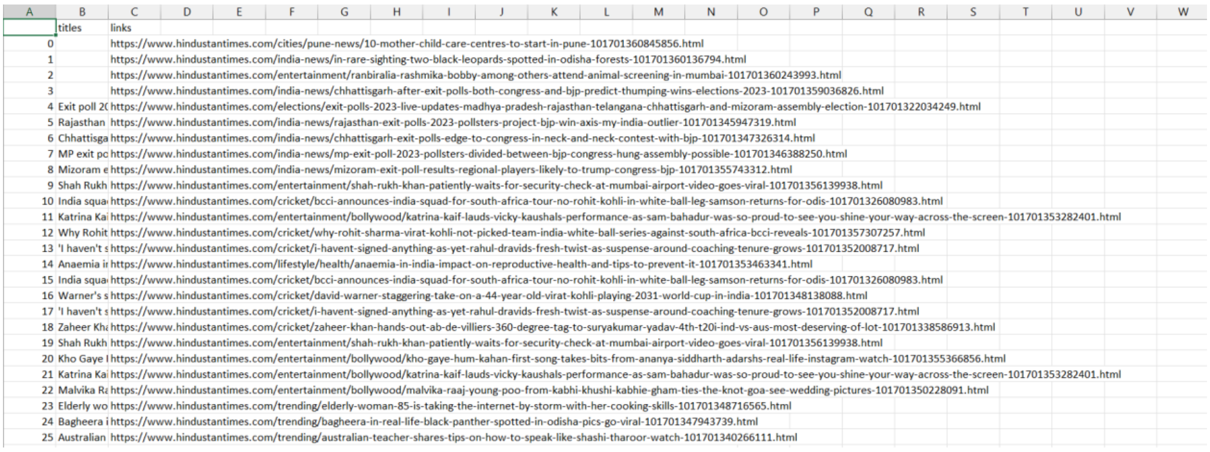
Name	Date modified	Type	Size
news-headlines.exe	30-11-2023 21:47	Application	42,076 KB
headline-30112023.csv	30-11-2023 21:51	Microsoft Excel Co...	9 KB

Figure 4: Generation of CSV File

5. Structured Data Storage:

Inside the generated CSV file, all headline titles with their corresponding webpage links are stored sequentially.

Explanation: The CSV file organizes the extracted data, presenting news headlines and links in a structured and easily accessible manner.



A	B	C
	titles	links
0		https://www.hindustantimes.com/cities/pune-news/10-mother-child-care-centres-to-start-in-pune-101701360845856.html
1		https://www.hindustantimes.com/india-news/in-rare-sighting-two-black-leopards-spotted-in-odisha-forests-101701360136794.html
2		https://www.hindustantimes.com/entertainment/ranbiralia-rashmika-bobby-among-others-attend-animal-screening-in-mumbai-101701360243993.html
3		https://www.hindustantimes.com/india-news/chhattisgarh-after-exit-polls-both-congress-and-bjp-predict-thumping-wins-elections-2023-101701359036826.html
4	Exit poll 2(	https://www.hindustantimes.com/elections/exit-polls-2023-live-updates-madhya-pradesh-rajasthan-telangana-chhattisgarh-and-mizoram-assembly-election-101701322034249.html
5	Rajasthan	https://www.hindustantimes.com/india-news/rajasthan-exit-polls-2023-pollsters-project-bjp-win-axis-my-india-outlier-101701345947319.html
6	Chhattisga	https://www.hindustantimes.com/india-news/chhattisgarh-exit-polls-edge-to-congress-in-neck-and-neck-contest-with-bjp-101701347326314.html
7	MP exit pc	https://www.hindustantimes.com/india-news/mp-exit-poll-2023-pollsters-divided-between-bjp-congress-hung-assembly-possible-101701346388250.html
8	Mizoram e	https://www.hindustantimes.com/india-news/mizoram-exit-poll-results-regional-players-likely-to-trump-congress-bjp-101701355743312.html
9	Shah Rukh	https://www.hindustantimes.com/entertainment/shah-rukh-khan-patiently-waits-for-security-check-at-mumbai-airport-video-goes-viral-101701356139938.html
10	India squa	https://www.hindustantimes.com/cricket/bcci-announces-india-squad-for-south-africa-tour-no-rohit-kohli-in-white-ball-leg-samson-returns-for-odis-101701326080983.html
11	Katrina Kai	https://www.hindustantimes.com/entertainment/bollywood/katrina-kaif-lauds-vicky-kaushals-performance-as-sam-bahadur-was-so-proud-to-see-you-shine-your-way-across-the-screen-101701353282401.html
12	Why Rohit	https://www.hindustantimes.com/cricket/why-rohit-sharma-virat-kohli-not-picked-team-india-white-ball-series-against-south-africa-bcci-reveals-101701357307257.html
13	'I haven't s	https://www.hindustantimes.com/cricket/i-havent-signed-anything-as-yet-rahul-dravids-fresh-twist-as-suspense-around-coaching-tenure-grows-101701352008717.html
14	Anaemia ir	https://www.hindustantimes.com/lifestyle/health/anaemia-in-india-impact-on-reproductive-health-and-tips-to-prevent-it-101701353463341.html
15	India squa	https://www.hindustantimes.com/cricket/bcci-announces-india-squad-for-south-africa-tour-no-rohit-kohli-in-white-ball-leg-samson-returns-for-odis-101701326080983.html
16	Warner's s	https://www.hindustantimes.com/cricket/david-warner-staggering-take-on-a-44-year-old-virat-kohli-playing-2031-world-cup-in-india-101701348138088.html
17	'I haven't s	https://www.hindustantimes.com/cricket/i-havent-signed-anything-as-yet-rahul-dravids-fresh-twist-as-suspense-around-coaching-tenure-grows-101701352008717.html
18	Zaheer Khi	https://www.hindustantimes.com/cricket/zaheer-khan-hands-out-ab-de-villiers-360-degree-tag-to-suryakumar-yadav-4th-t20i-ind-vs-aus-most-deserving-of-lot-101701338586913.html
19	Shah Rukh	https://www.hindustantimes.com/entertainment/shah-rukh-khan-patiently-waits-for-security-check-at-mumbai-airport-video-goes-viral-101701356139938.html
20	Kho Gaye I	https://www.hindustantimes.com/entertainment/bollywood/kho-gaye-hum-kahan-first-song-takes-bits-from-ananya-siddharth-adarshs-real-life-instagram-watch-101701355366856.html
21	Katrina Kai	https://www.hindustantimes.com/entertainment/bollywood/katrina-kaif-lauds-vicky-kaushals-performance-as-sam-bahadur-was-so-proud-to-see-you-shine-your-way-across-the-screen-101701353282401.html
22	Malvika Ri	https://www.hindustantimes.com/entertainment/bollywood/malvika-raaj-young-poo-from-kabhi-khushi-kabhie-gham-ties-the-knot-goa-see-wedding-pictures-101701350228091.html
23	Elderly wo	https://www.hindustantimes.com/trending/elderly-woman-85-is-taking-the-internet-by-storm-with-her-cooking-skills-101701348716565.html
24	Bagheera i	https://www.hindustantimes.com/trending/bagheera-in-real-life-black-panther-spotted-in-odisha-pics-go-viral-101701347943739.html
25	Australian	https://www.hindustantimes.com/trending/australian-teacher-shares-tips-on-how-to-speak-like-shashi-tharoor-watch-101701340266111.html

Figure 5 : Organized Data Stored in CSV File

Project Overview:

This Python script represents a pragmatic application of Web Automation and Web Scraping. Its primary objective is the automated extraction of daily news headlines and associated links from the Hindustan Times website. Powered by the Selenium library, the script efficiently navigates the website, pinpointing specific HTML elements through XPath, and extracting pertinent information. Notably, the use of a headless browser ensures the discreet execution of these tasks without a visible browser window. Subsequently, the script employs the Pandas library to structure the extracted data into an organized format, which is then exported to a CSV file bearing the current date as its name. In summary, this project eliminates the need for manual news collection, streamlining the entire process and delivering a daily summary of news updates in a structured and easily accessible format.

Break down of the project step by step:

1. Initialization:

The script begins by setting up essential components, including the path to the ChromeDriver executable, the current date, and the target website (Hindustan Times in this case).

Python code:

```
application_path = os.path.dirname(sys.executable)

now = datetime.now()

month_day_year = now.strftime("%d%m%Y")

website = "https://www.hindustantimes.com/"

path="C:/Users/ankit/Downloads/Compressed/chromedriver-win64/chromedriver-win64/chromedriver.exe"
```

2. Configuring Selenium:

Selenium is configured with options for a headless browser (invisible to the user) and the ChromeDriver service.

Python code:

```
options = Options()

options.headless = True

service = Service(executable_path=path)

driver = webdriver.Chrome(service=service, options=options)
```

3. Navigating to the Website:

The script then directs the Chrome browser to the specified website.

Python code

```
driver.get(website)
```

4. Locating HTML Elements:

Using XPath, the script identifies all <h3> elements with the class "hdg3" on the webpage.

Python code

```
containers = driver.find_elements(by="xpath", value='//h3[@class="hdg3"]')
```

5. Extracting Data:

Iterating through the identified elements, the script extracts the text content (news headlines) and links associated with each headline..

Python code

```
titles = []

links = []
```

```
for container in containers:
    title = container.find_element(by="xpath", value='./a').text
    titles.append(title)
    link = container.find_element(by="xpath", value='./a').get_attribute("href")
    links.append(link)
```

6. Organizing Data with Pandas:

The extracted data is organized into a Pandas DataFrame, facilitating structured storage and manipulation.

Python code

```
my_dic = {'titles': titles, 'links': links}
df_headlines = pd.DataFrame(my_dic)
```

7. Exporting to CSV:

The DataFrame is then exported to a CSV file named with the current date.

Python code

```
file_name = f'headline-{month_day_year}.csv'
final_path = os.path.join(application_path, file_name)
df_headlines.to_csv(final_path)
```

8. Closing the WebDriver:

Finally, the Chrome browser is closed, completing the automated process.

Python code

```
driver.quit()
```

In summary, the project uses Selenium and Python to automate the collection of news headlines and links from a website. It navigates the site, extracts data from specific HTML elements, organizes it into a structured format, and saves it as a CSV file. This automation streamlines the daily task of obtaining news updates, saving time and ensuring consistency.

Conclusion:

The web automation and scraping project utilizing Selenium, Pandas, PyInstaller, and ChromeDriver successfully automates the extraction of news headlines and links from the Hindustan Times website. The script efficiently navigates the web, extracts relevant data, and organizes it into a structured CSV file. By leveraging Selenium's capabilities, the project streamlines the process of obtaining daily news updates, saving time and effort. The use of Pandas enhances the data handling, providing a structured format for easy analysis or further processing. PyInstaller ensures the script's portability by converting it into a standalone executable, making it accessible across various environments without the need for a separate Python installation. The combination of these technologies creates a robust solution for automating web interactions and data extraction.

Future Enhancements:

1. Dynamic Content Handling: Enhance the script to handle dynamic content loading, ensuring that all relevant headlines, even those loaded asynchronously, are captured.

2. User Configuration: Implement user-configurable options, allowing users to specify the website, target elements, or extraction criteria, making the script adaptable to various news websites.

3. Error Handling and Logging: Integrate a comprehensive error handling mechanism and logging system to capture and report any issues during automation, enhancing script robustness.

4. Scheduled Execution: Implement a scheduling mechanism, such as using a task scheduler or a cloud-based service, to enable automatic execution of the script at predefined intervals without manual intervention.

5. Data Visualization: Incorporate data visualization tools or frameworks to generate graphical representations of the extracted data, providing users with insightful and visually appealing summaries of news trends.

6. Browser Compatibility: Extend compatibility to other browsers by incorporating drivers for Firefox, Edge, or other popular browsers, ensuring broader applicability.

7. User Interface (UI): Develop a simple graphical user interface (GUI) to make the script more user-friendly, allowing users to configure settings and initiate the automation process without delving into the script itself.

8. Advanced Filtering: Implement advanced filtering options to allow users to extract news based on categories, keywords, or specific criteria, adding flexibility to the scraping process.