Report — Spotify Playlist Scraper Automation

(Task No. 1)

Project Title: Spotify Featured Playlist Data
Author: Rohan Jadhav
Date: 22, July 2025

Objective:

This project automates the process of extracting metadata and track-level details from any public Spotify playlist. The extracted data is saved to a formatted Excel workbook for further analysis or archival.

Tools/Technologies used:

Requirement Type	**Name / Tool**	**Version (Recommended)**	**Purpose**
Programming Language	Python	3.9 or above	Core language for scripting and automation
Browser Driver	ChromeDriver	Compatible with Chrome	Required for Selenium to control the Chrome browser
Web Browser	Google Chrome	Latest (curr. version 138.0.7204.158)	Used for loading and scraping playlist content
Automation Library	Selenium	4.x	For web scraping and DOM interaction
Excel Library	openpyxl	3.x	To create and modify Excel (.xlsx) workbooks
Text Editor / IDE	VSCode / PyCharm / Sublime	Latest	Recommended for editing and running the Python scripts

Table 1: Software Requirements

Methodology:

- The script takes a Spotify playlist URL from the user via a command-line interface.
- It uses Selenium WebDriver to open the playlist page in a Chrome browser and simulate user scrolling to load all track data.
- Playlist metadata and detailed track information (track name, artist, album, duration, date added) are extracted by targeting specific DOM elements.
- The openpyxl library is used to export this data into a structured Excel workbook with formatted sheets for both metadata and tracks.
- Within Excel, simple visualizations like pivot tables, bar charts, and duration bins are applied for basic data analysis.
- The entire process is automated and modular, enabling reuse for any public Spotify playlist URL with minimal changes.

Summarized Approach:

- Selenium WebDriver is used to automate browser interaction and simulate scrolling to load tracks.
- Playlist metadata (title, description, saves, etc.) and track-level information (track name, artists, album, date added, duration) are parsed from the DOM.
- Extracted data is organized and exported using openpyxl into an Excel workbook with bold headers and separate sheets.

Project Directory Structure:

The project is modular and organized into three main scripts:

project-1/

├── main.py

├── scraper.py

├── save_excel.py

├── outputs/

│ └── Spotify_Playlist_Export.xlsx

└── README.pdf

1. *scraper.py:* Handles browser setup and scraping logic
2. *save_excel.py:* Responsible for Excel export using openpyxl
3. *main.py:* CLI runner script that connects components
4. *outputs/ :* Auto-generated folder for exported Excel files
5. *README.pdf* : Report Document

Execution:

The script is executed via the CLI using the command:

python main.py "url"

e.g. python main.py “https://open.spotify.com/playlist/37i9dQZF1DWUAOn5dYbrDa”

It opens the playlist, extracts relevant data, and saves it to path - *outputs/Spotify_Playlist_Export.xlsx.*

Screenshots:

```python
def get_track_data(driver, wait):
    scroll_to_load(driver)
    # Find the first 21 rows of tracks
    rows = driver.find_elements(By.XPATH, "//div[@role='row' and .//div[@aria-colindex='2']]"
    data = []
    for idx, row in enumerate(rows):
        try:
            #track and artist
            track_col = row.find_element(By.XPATH, ".//div[@aria-colindex='2']")
            track_name = track_col.find_element(By.XPATH, ".//div[@dir='auto' and contains(@c
            artist_links = track_col.find_elements(By.XPATH, ".//span[contains(@class, 'UudGC
            artists = ", ".join([a.text.strip() for a in artist_links])
            #album and date added, duration
            album_elem = row.find_element(By.XPATH, ".//div[@aria-colindex='3']//a")
            album_name = album_elem.text.strip()
            date_added = row.find_element(By.XPATH, ".//div[@aria-colindex='4']//span").text.
            duration = row.find_element(By.XPATH, ".//div[@aria-colindex='5']").text.strip()
            # Append the data to the list
            data.append({
                "Track Name": track_name,
                "Artist(s)": artists,
                "Album Name": album_name,
                "Date Added": date_added,
                "Duration (mm:ss)": duration
            })
        except Exception as e:
            print(f"[Row {idx}] Skipped due to error: {e}")
    return data
```

Fig 1. Code Snippet

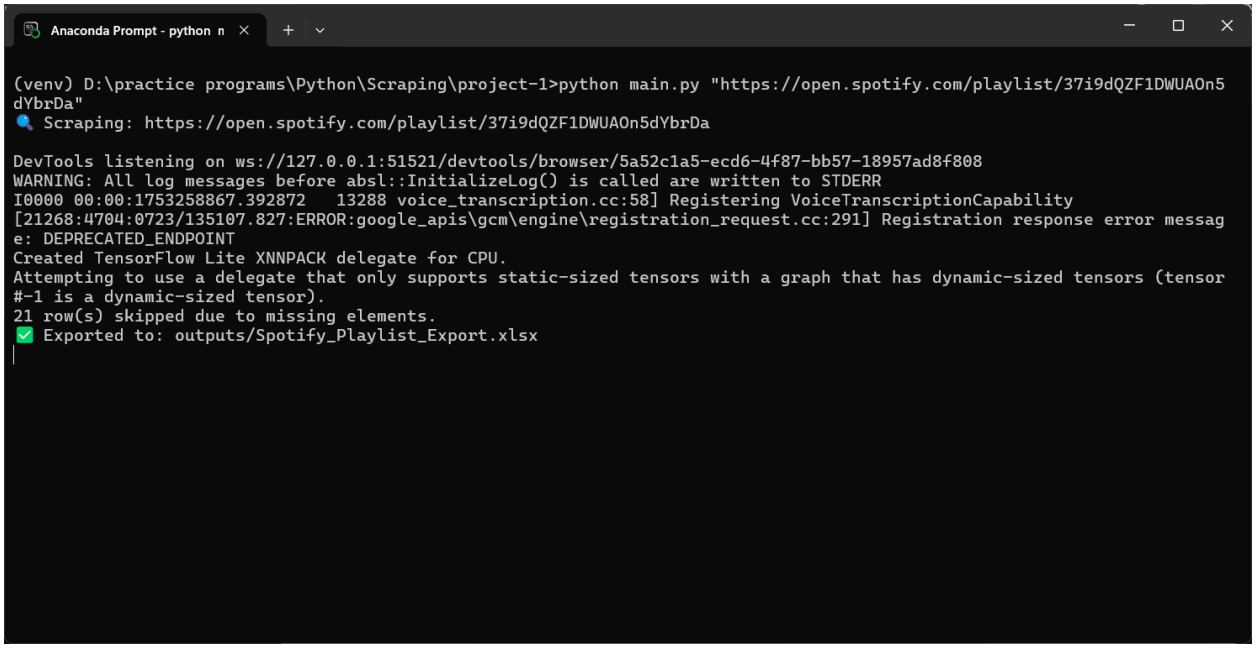
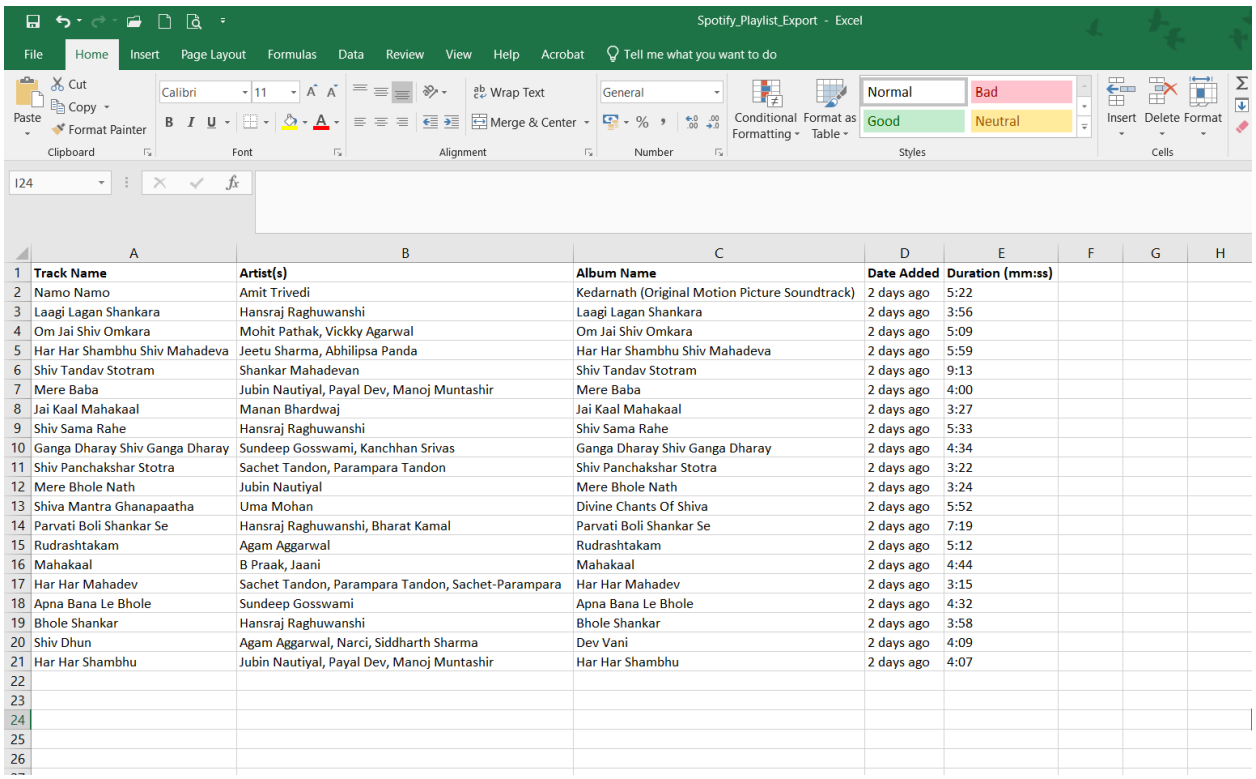


Fig 2. Terminal Output



	A	B	C	D	E	F	G	H
1	**Track Name**	**Artist(s)**	**Album Name**	**Date Added**	**Duration (mm:ss)**			
2	Namo Namo	Amit Trivedi	Kedarnath (Original Motion Picture Soundtrack)	2 days ago	5:22			
3	Laagi Lagan Shankara	Hansraj Raghuwanshi	Laagi Lagan Shankara	2 days ago	3:56			
4	Om Jai Shiv Omkara	Mohit Pathak, Vickky Agarwal	Om Jai Shiv Omkara	2 days ago	5:09			
5	Har Har Shambhu Shiv Mahadeva	Jeetu Sharma, Abhilipsa Panda	Har Har Shambhu Shiv Mahadeva	2 days ago	5:59			
6	Shiv Tandav Stotram	Shankar Mahadevan	Shiv Tandav Stotram	2 days ago	9:13			
7	Mere Baba	Jubin Nautiyal, Payal Dev, Manoj Muntashir	Mere Baba	2 days ago	4:00			
8	Jai Kaal Mahakaal	Manan Bhardwaj	Jai Kaal Mahakaal	2 days ago	3:27			
9	Shiv Sama Rahe	Hansraj Raghuwanshi	Shiv Sama Rahe	2 days ago	5:33			
10	Ganga Dharay Shiv Ganga Dharay	Sundeep Gosswami, Kanchhan Srivas	Ganga Dharay Shiv Ganga Dharay	2 days ago	4:34			
11	Shiv Panchakshar Stotra	Sachet Tandon, Parampara Tandon	Shiv Panchakshar Stotra	2 days ago	3:22			
12	Mere Bhole Nath	Jubin Nautiyal	Mere Bhole Nath	2 days ago	3:24			
13	Shiva Mantra Ghanapaatha	Uma Mohan	Divine Chants Of Shiva	2 days ago	5:52			
14	Parvati Boli Shankar Se	Hansraj Raghuwanshi, Bharat Kamal	Parvati Boli Shankar Se	2 days ago	7:19			
15	Rudrashtakam	Agam Aggarwal	Rudrashtakam	2 days ago	5:12			
16	Mahakaal	B Praak, Jaani	Mahakaal	2 days ago	4:44			
17	Har Har Mahadev	Sachet Tandon, Parampara Tandon, Sachet-Parampara	Har Har Mahadev	2 days ago	3:15			
18	Apna Bana Le Bhole	Sundeep Gosswami	Apna Bana Le Bhole	2 days ago	4:32			
19	Bhole Shankar	Hansraj Raghuwanshi	Bhole Shankar	2 days ago	3:58			
20	Shiv Dhun	Agam Aggarwal, Narci, Siddharth Sharma	Dev Vani	2 days ago	4:09			
21	Har Har Shambhu	Jubin Nautiyal, Payal Dev, Manoj Muntashir	Har Har Shambhu	2 days ago	4:07			
22								
23								
24								
25								
26								

Fig 3. Collected Data Sample

Insights:

1. Track durations revealed that most songs in the playlist fall between 3–4 minutes, aligning with commercial music norms and listener preferences.
2. By extracting the "Date Added" field, we observed how the playlist evolved over time — showing when most tracks were added (e.g., in bursts or steadily).
3. Sorting tracks by album name or artist allowed us to spot repeated contributors, helping highlight if the playlist heavily features a specific album or artist.

Notes:

- Headless execution is supported via config
- Currently limited to top 21 tracks (can be customized)
- Playlist must be public