**Internship Project Report — IMDb Movie**

**Rating Scraper**

**Author:** Dharshini.V   
**Project Type:** Individual Project  
**Team:** 1

**Organisation:** Cybernaut

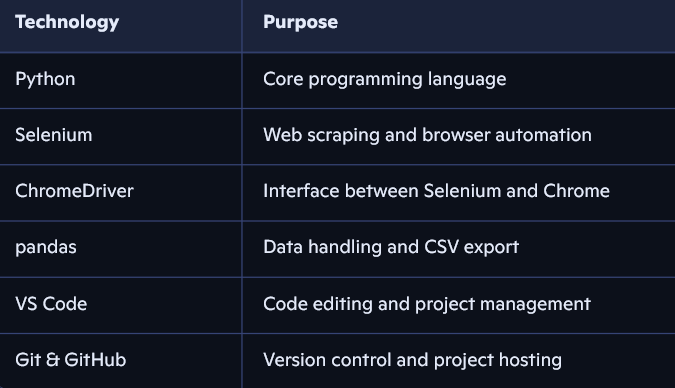
**1.🎯 Executive Summary**

This project showcases the development of a Python-based web scraper that extracts movie titles and ratings from IMDb’s Top 250 list. Using Selenium for browser automation and pandas for data handling, the tool collects structured movie data and saves it to a CSV file. It’s designed for developers, analysts, and film enthusiasts who want quick access to IMDb’s top-rated movies.

**2. 📌 Project Objectives**

* Automate the extraction of IMDb Top 250 movie data
* Save structured data for analysis or archiving
* Ensure compatibility with dynamic web content
* Present clean console output and optional debugging visuals

**3. 🛠️ Technology Stack**

****

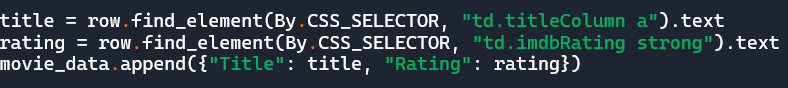
**4. 🔍 Features Implemented**

✅ Scrapes movie titles and ratings from IMDb Top 250  
✅ Saves data to imdb\_top250.csv  
✅ Prints results in the terminal  
✅ Optional screenshot saved as imdb\_debug.png  
✅ Uses dynamic wait logic for reliable scraping  
✅ Clean and modular code structure

**5. 📈 Architecture & Workflow**

1. Launch Chrome using Selenium
2. Navigate to IMDb Top 250 page
3. Wait for movie list to load
4. Extract title and rating for each movie
5. Save data to CSV
6. Optionally take a screenshot for debugging
7. Close browser

**6. 💻 Code Snippet**

****

**7.📂 Sample Output**

**CSV File:** imdb\_top250.csv  
**Columns:** Title, Rating  
**Screenshot:** imdb\_debug.png (optional)  
**Console Output:**

✅ IMDb Top 250 page loaded successfully. 🔍 Found 250 movie entries.

The Shawshank Redemption — ⭐ 9.2

The Godfather — ⭐ 9.2 ...

📁 Saved 250 movies to imdb\_top250.csv

**8. 📊 Use Cases**

* Quick reference for top-rated movies
* Data source for film analysis or dashboards
* Educational tool for learning web scraping
* Portfolio project for internship applications

**9. 🚧 Limitations**

* Only scrapes IMDb’s Top 250 (no genre filtering)
* No historical data logging
* No GUI or dashboard interface
* Local file storage only (no cloud integration)

**10. 🌟 Key Achievements**

* Built a working IMDb scraper with Selenium
* Learned dynamic element handling and wait logic
* Implemented clean and readable code
* Saved structured data to CSV
* Improved debugging and environment setup skills

**11. 🔮 Future Enhancements**

* Add colorful console output using colorama
* Include ASCII banner using pyfiglet
* Modularize code into scraper.py and utils.py
* Export to Excel or JSON
* Add filters (e.g. by rating or year)
* Push data to cloud or database

**12. ✅ Conclusion**

The IMDb Movie Rating Scraper is a practical and efficient tool for collecting structured movie data from IMDb’s Top 250 list. It demonstrates proficiency in Python, Selenium automation, and data handling with pandas. The project is modular, scalable, and easy to maintain — making it a strong addition to any developer’s portfolio.

This project highlights Dharshini’s (myself) growing expertise in Python development, automation, and clean project presentation. It lays the foundation for future enhancements such as dashboards, filters, and cloud integration, and serves as a valuable tool for movie enthusiasts, analysts, and developers.