LaSalle College
Montréal

Final Project: 30%

Course Identification

Name of program(s) – Code(s):	COMPUTER SCIENCE TECHNOLOGY–PROGRAMMING (420.BP) INFORMATION TECHNOLOGY PROGRAMMER-ANALYST (LEA.3Q)
Course title:	**SCRIPTING LANGUAGE**
Course number:	420-LS3-AS
Teacher's name:	Marcela Gonçalves dos Santos
Semester:	Winter 2025

Instructions

- **For this assignment, students can have groups of 3 for each.**

Mark Breakdown

- Code Quality & Implementation 30
  - Clean, modular, and well-documented Python code.
- Database & Security Measures 20
  - Proper data storage and security best practices.
- Functionality & Features 20
  - Application meets the proposed objectives with working features.
- Testing & Debugging 15
  - Functional test plan, debugging process, and error handling.
- Presentation & Documentation 15
  - Clear project report and engaging final presentation.

TOTAL: 100 POINTS

LaSalle College
Montréal

Final Project: 30%

Project Proposal: Basic Web Scraper

1. Project Overview

The **Basic Web Scraper** is a Python-based application that extracts useful data from websites and presents it in a structured format. The system will allow users to scrape text, links, and images from specified web pages and save the extracted data in different formats such as CSV, JSON, or a database. The application will feature a **graphical user interface (GUI)** for ease of use and include filtering and search functionalities.

2. Objectives

The main objectives of the project are:

- To develop an **automated data extraction tool**.
- To scrape **text, links, and images** from web pages.
- To provide an **option for storing data in different formats** (CSV, JSON, SQLite).
- To enable users to **filter and search scraped content**.
- To implement **error handling and debugging mechanisms**.

3. Key Features

1. **Website Scraping:**
    - Extract text, images, and hyperlinks from web pages.
    - Support dynamic page handling.
2. **Data Storage & Export:**
    - Store extracted data in **CSV, JSON, or SQLite**.
    - Allow users to choose their preferred format.
3. **Filtering & Searching:**
    - Implement keyword-based filtering.
    - Provide search functionality for extracted data.
4. **Graphical User Interface (GUI):**
    - Allow users to enter URLs and configure scraping options.
    - Display extracted data in a user-friendly format.
5. **Error Handling & Debugging:**
    - Handle issues such as missing data, slow responses, and CAPTCHA restrictions.
    - Implement logging for debugging purposes.

LaSalle College
Montréal

Final Project: 30%

4. System Design & Technologies

4.1 Technology Stack

- **Programming Language:** Python
- **Web Scraping Libraries:** BeautifulSoup, Requests, Selenium (for dynamic content)
- **Data Storage:** CSV, JSON, SQLite
- **GUI Framework:** Tkinter or PyQt
- **Logging & Error Handling:** Python's logging module

5. Milestones & Deliverables

Week	Milestone	Deliverable
Week 1-2	Setup Development Environment & Project Initialization	Initialize Git repository, set up virtual environment, install dependencies
Week 3-4	Implement Basic Web Scraping Functions	Develop script to extract text, images, and links from web pages
Week 5-6	Implement Data Storage & Export Options	Store scraped data in CSV, JSON, or SQLite formats
Week 7-8	Develop GUI for User Interaction	Implement GUI elements for input and display of extracted data
Week 9	Implement Filtering & Search Functionalities	Add keyword-based filtering and search features

LaSalle College
Montréal

Final Project: 30%

Week 10	Security Enhancements & Error Handling	Implement logging, error handling, and CAPTCHA bypass strategies
Week 11	Testing & Debugging	Conduct functional testing, log errors, and apply necessary fixes
Week 12	Final Submission & Presentation	Submit final project, code repository, and project report

6. Expected Outcome

Upon successful completion of the project, the **Basic Web Scraper** will allow users to automate data extraction from web pages and store structured data efficiently. The project will serve as a practical demonstration of **web scraping, data handling, and GUI development**.

7. Conclusion

This project will provide hands-on experience in **automated web scraping and data processing** while introducing students to **real-world programming challenges**. By working on this project, students will gain practical skills in **web automation, data extraction, GUI design, and debugging**, which are essential for modern software development roles.

Project Team

- Student 1 (Lead Developer)
- Student 2 (Scraping & Data Handling Specialist)
- Student 3 (GUI & User Experience Designer)