Assignment 2 (Final)

Total: 100 Marks

Solve and submit any 2 tasks from the following.

Task 1: Parallel Word Count (50 Marks)

Problem:
Given a large collection of .txt files, write a program that counts the frequency of each word across all files.

Hints:

- Use multiprocessing.Pool to process files in parallel.
- Merge word count results from each process.
- Compare execution time with a sequential implementation.

Bonus: Handle punctuation and case sensitivity correctly.

Task 2: Image Filter with Multithreading (50 Marks)

Problem:
Apply a grayscale or edge-detection filter to a directory of images using multithreading.

Hints:

- Use threading or ThreadPoolExecutor to process images concurrently.
- Use Pillow or OpenCV for image manipulation.
- Compare performance with a sequential version.

Bonus: Save processed images in a new directory with clear filenames.

Task 3: Simulating Weather Data Using Multiprocessing (50 Marks)

Problem:
Simulate weather readings (temperature, humidity, pressure) for 100 cities over 7 days in parallel.

Hints:

- Use multiprocessing.Process or Pool to simulate data generation.
- Write each city's data to a .csv file.
- Combine all data into a single weather_summary.csv.
- Use time.sleep() to simulate delays.

Bonus: Plot temperature trends for 5 cities using matplotlib.

Task 4: Web Scraping with Asyncio (50 Marks)

Problem:
Scrape titles or headings from a list of URLs asynchronously.

Hints:

- Use aiohttp and asyncio to fetch data concurrently.
- Extract <title> or <h1> tags.
- Compare execution time with a sequential approach.

Bonus: Gracefully handle failed requests or timeouts.

Submission Guidelines:

i. Submit a **Python script (.py file) or Colab script** for each task.
ii. Include a short report (PDF) summarizing results and performance comparisons.
iii. Provide screenshots of execution times.