**CS 516: Information Retrieval and Text Mining**

Information Technology University (ITU)

Fall 2025

Course Instructor: Dr. Ahmad Mustafa

**Homework Assignment 3**

**Due Date: 11.59 pm, 30th November, 2025**

**Design Problem**

In this assignment, you will design and implement a complete information retrieval (IR) system that runs locally on your machine. You may use any retrieval strategy (or combination of strategies), such as Boolean retrieval, Vector Space Models (e.g., TF–IDF) etc.

Your goal is to design a coherent, justifiable, and well-evaluated retrieval system using the provided dataset of text documents (accessible [here](https://www.kaggle.com/datasets/asad1m9a9h6mood/news-articles)).

Creativity is encouraged, but your system must be reproducible and fully local (no cloud-hosted vector databases).  
**Requirements**

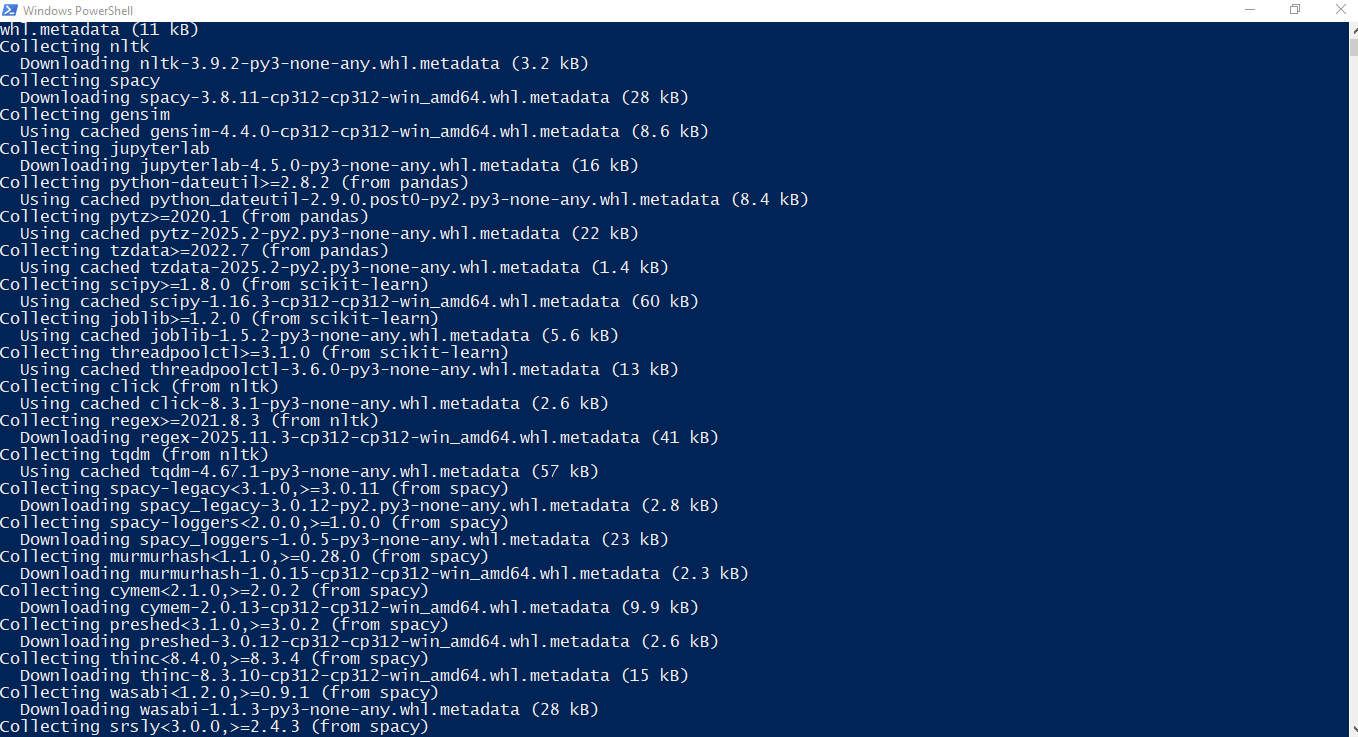
You are free to design your IR system however you like, as long as it satisfies the following:

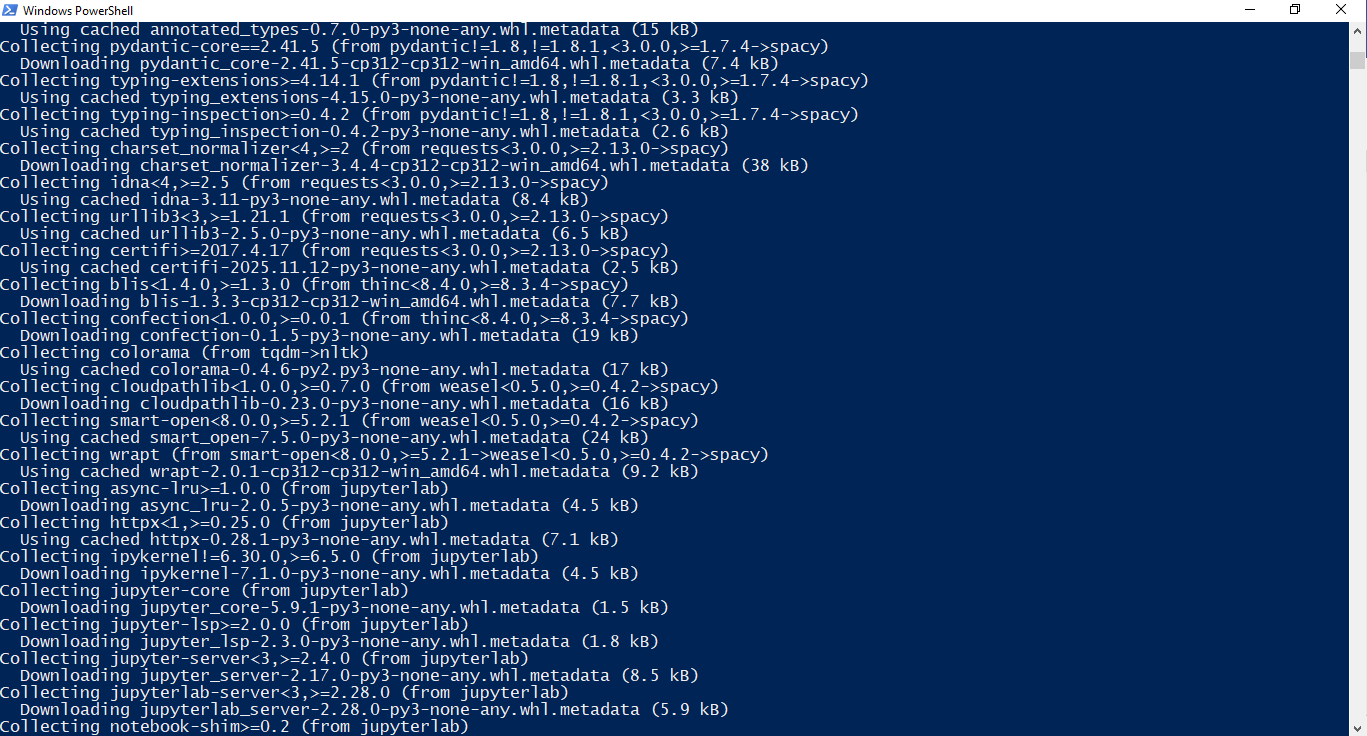
**1. Local Implementation**

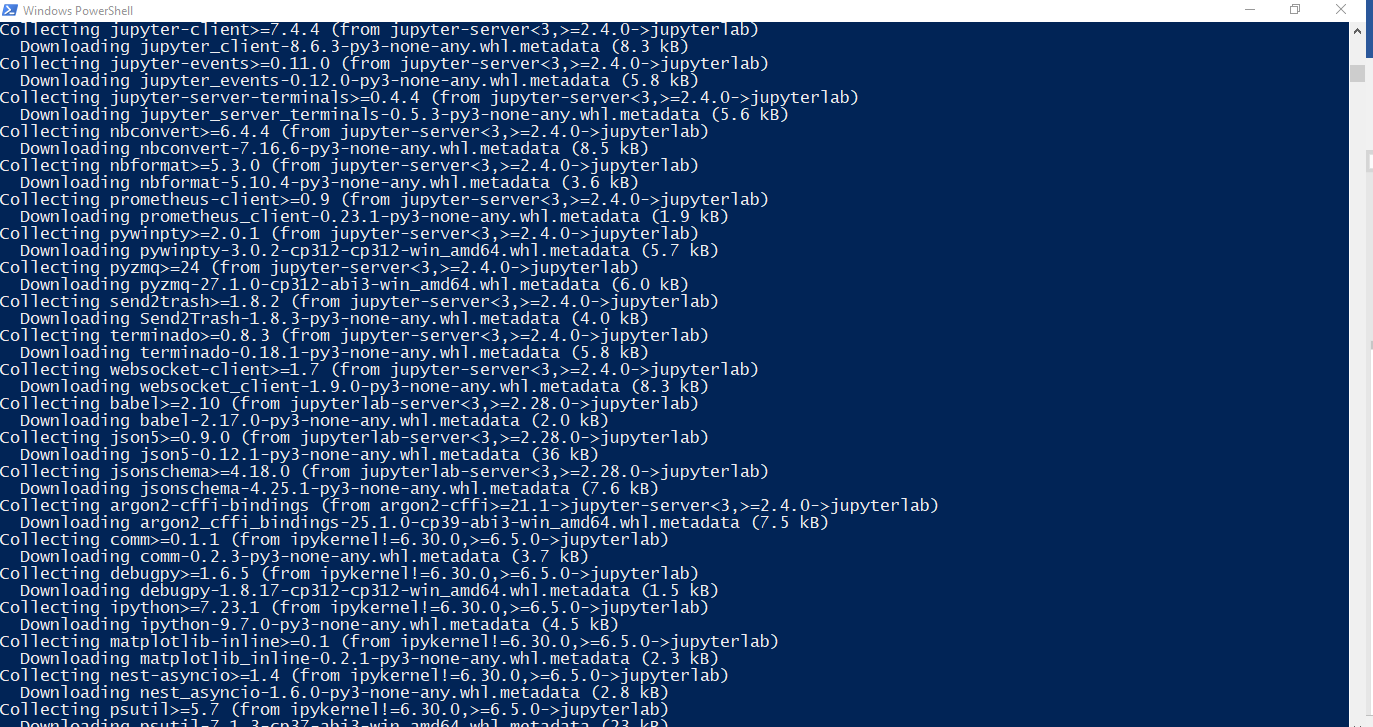
Your system must run end-to-end on a local machine (Windows, Mac, or Linux).  
 You may use any mainstream programming language or libraries.

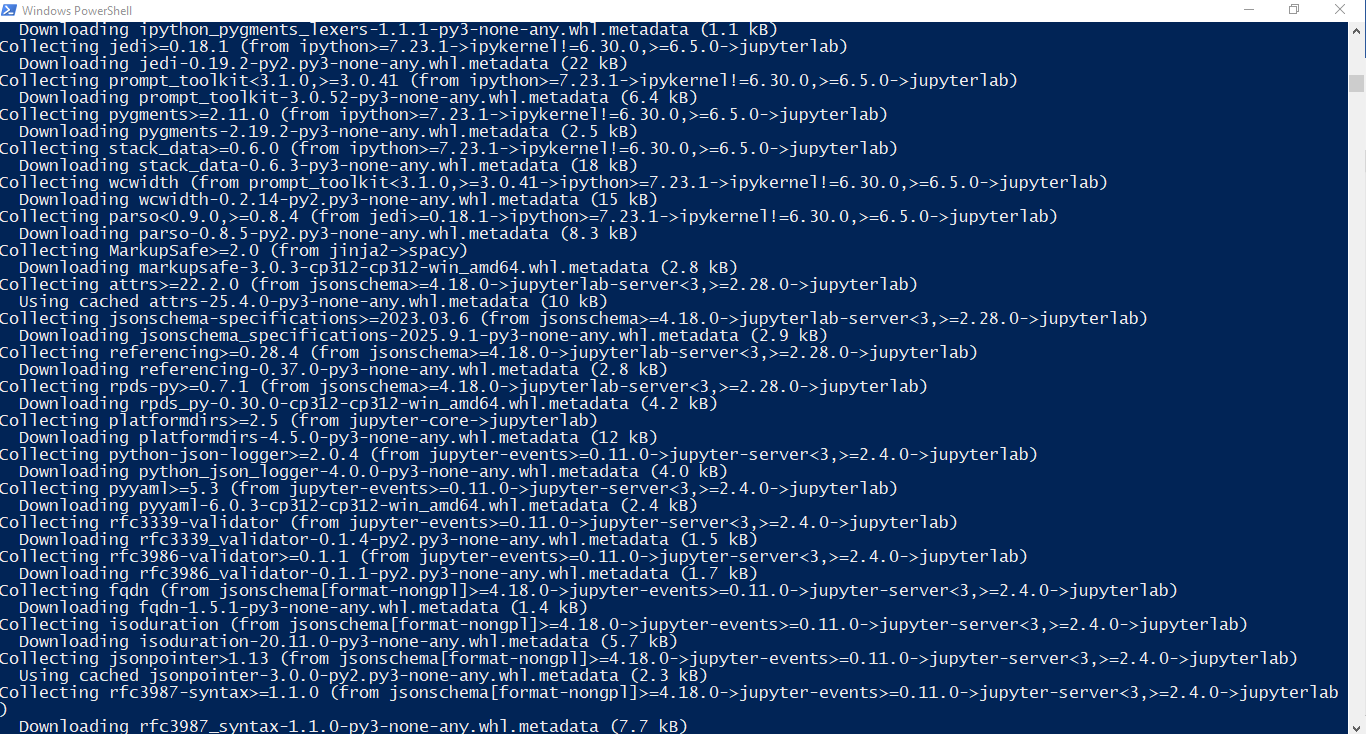
Cloud-hosted vector databases (e.g., Pinecone, Chroma Cloud, Weaviate Cloud, Elasticsearch clusters) are **not** allowed.

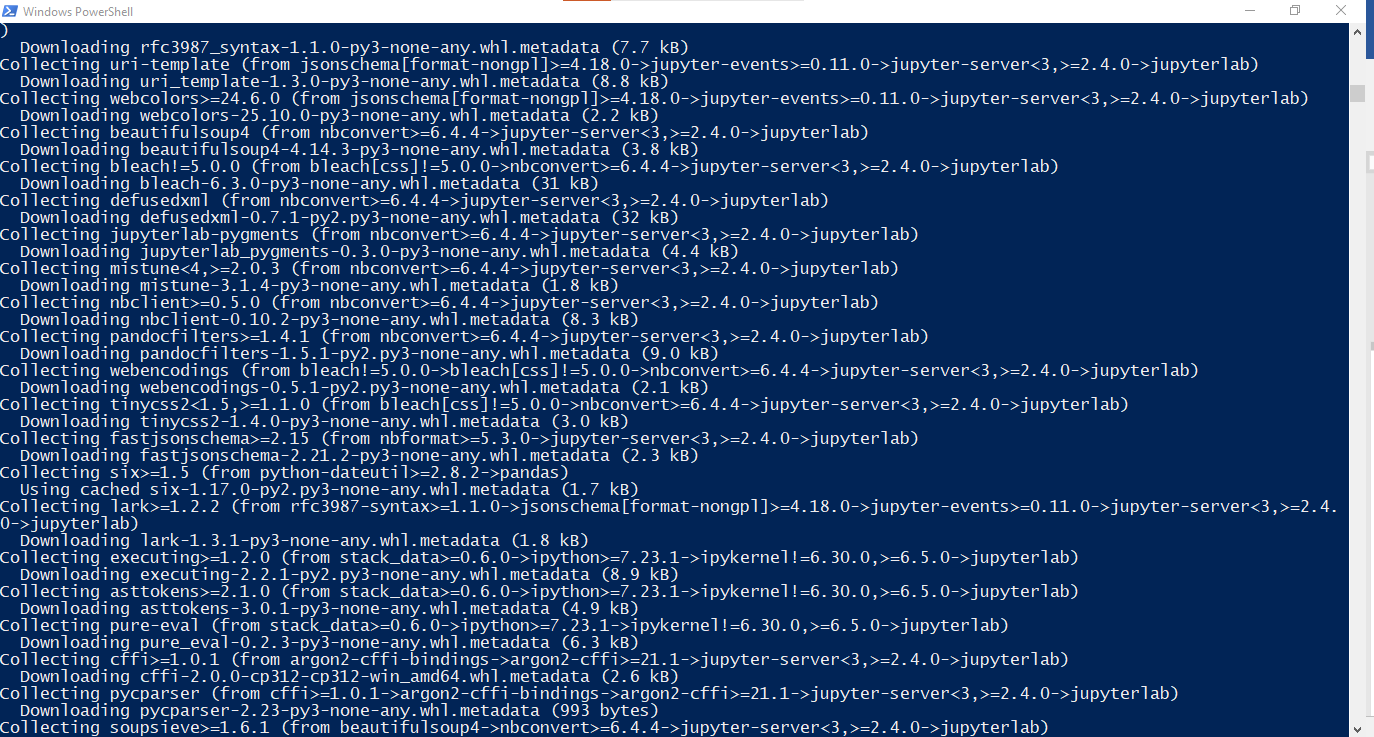
Local libraries (e.g., scikit-learn, gensim, rank-bm25, FAISS local install) are allowed.



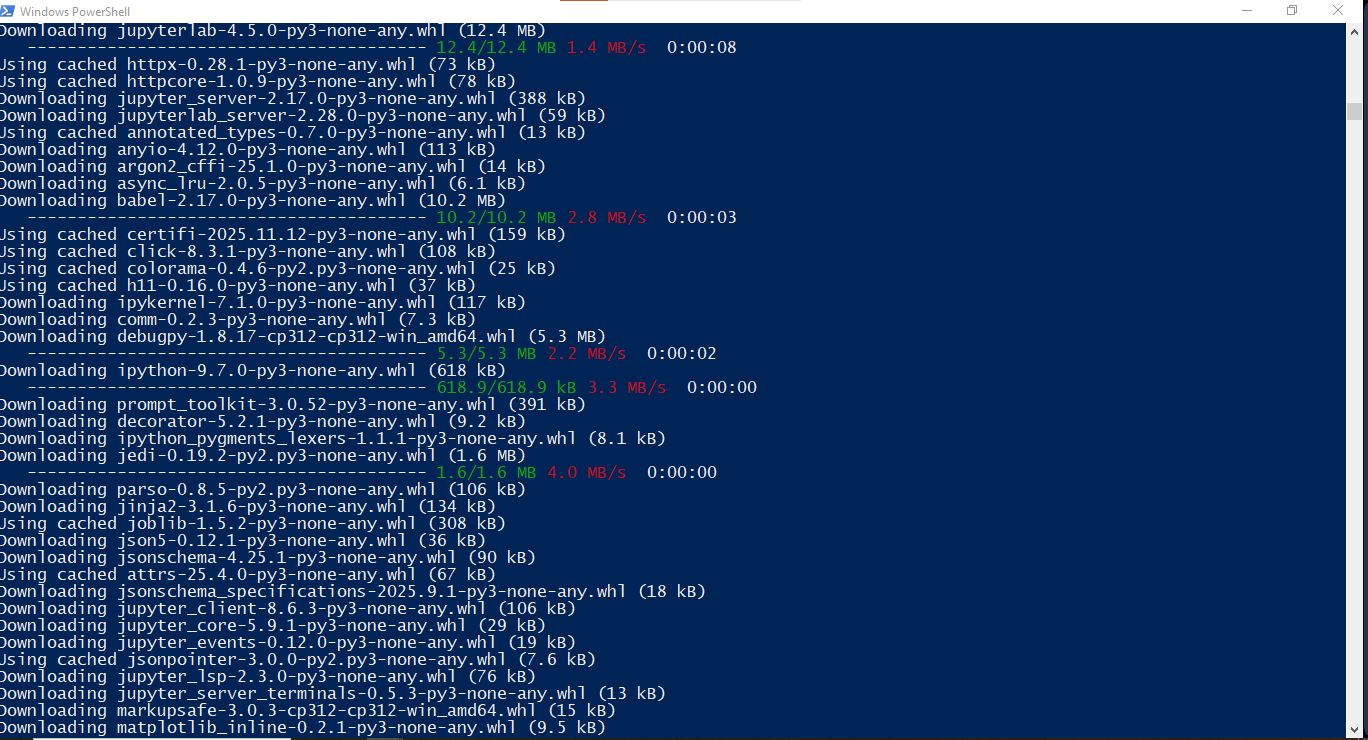


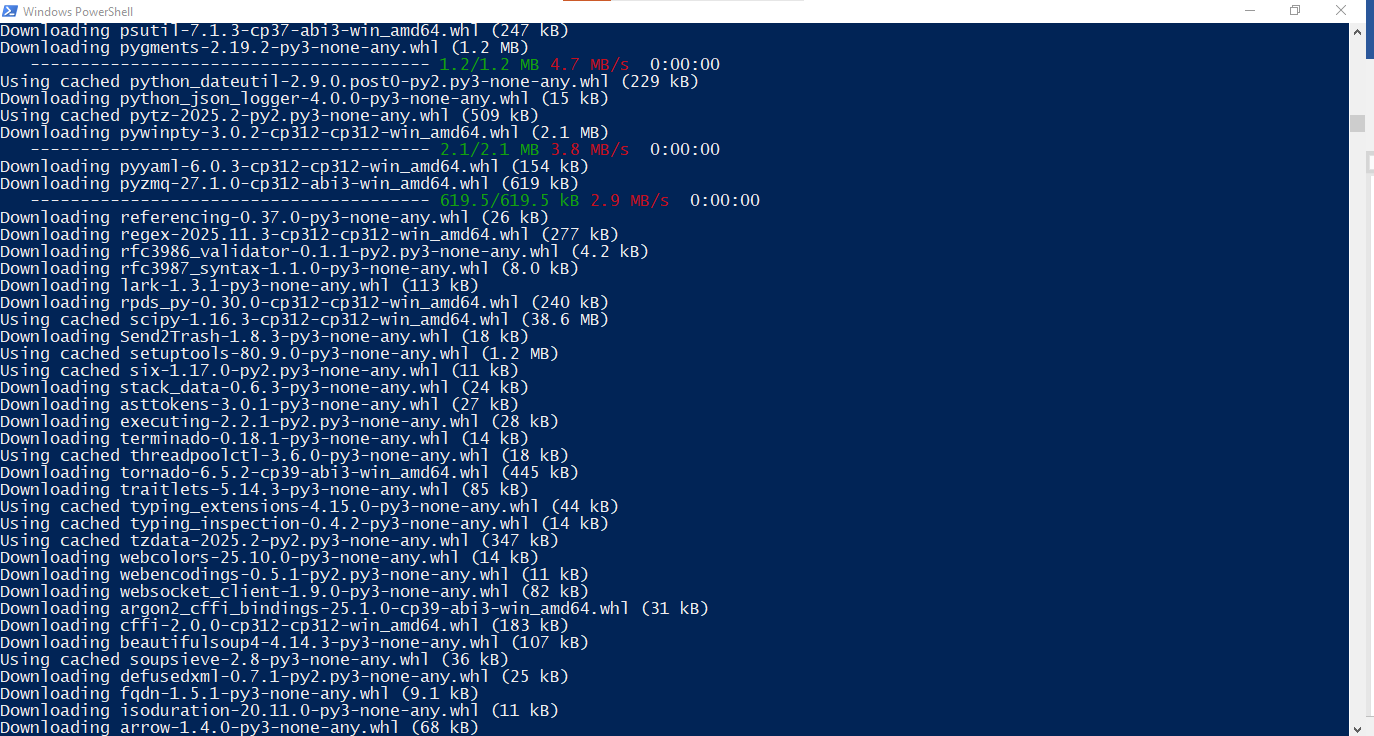


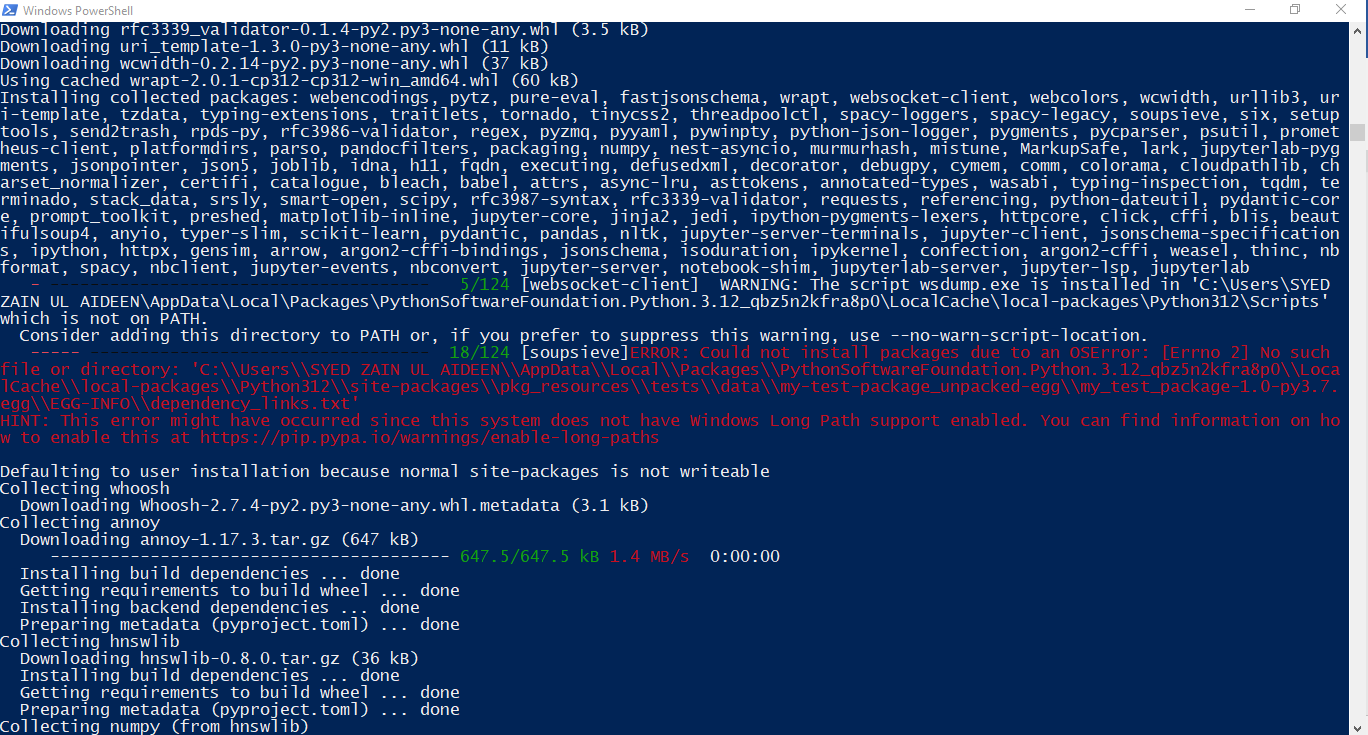


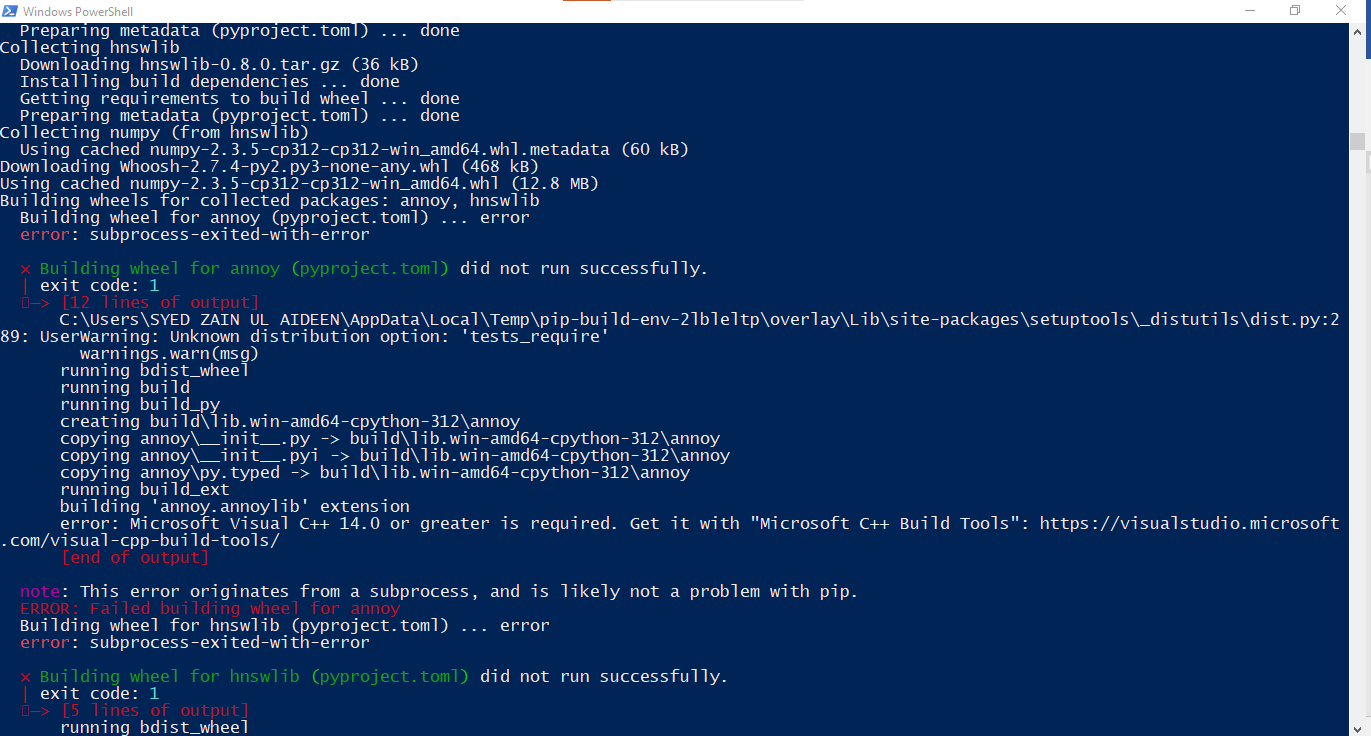


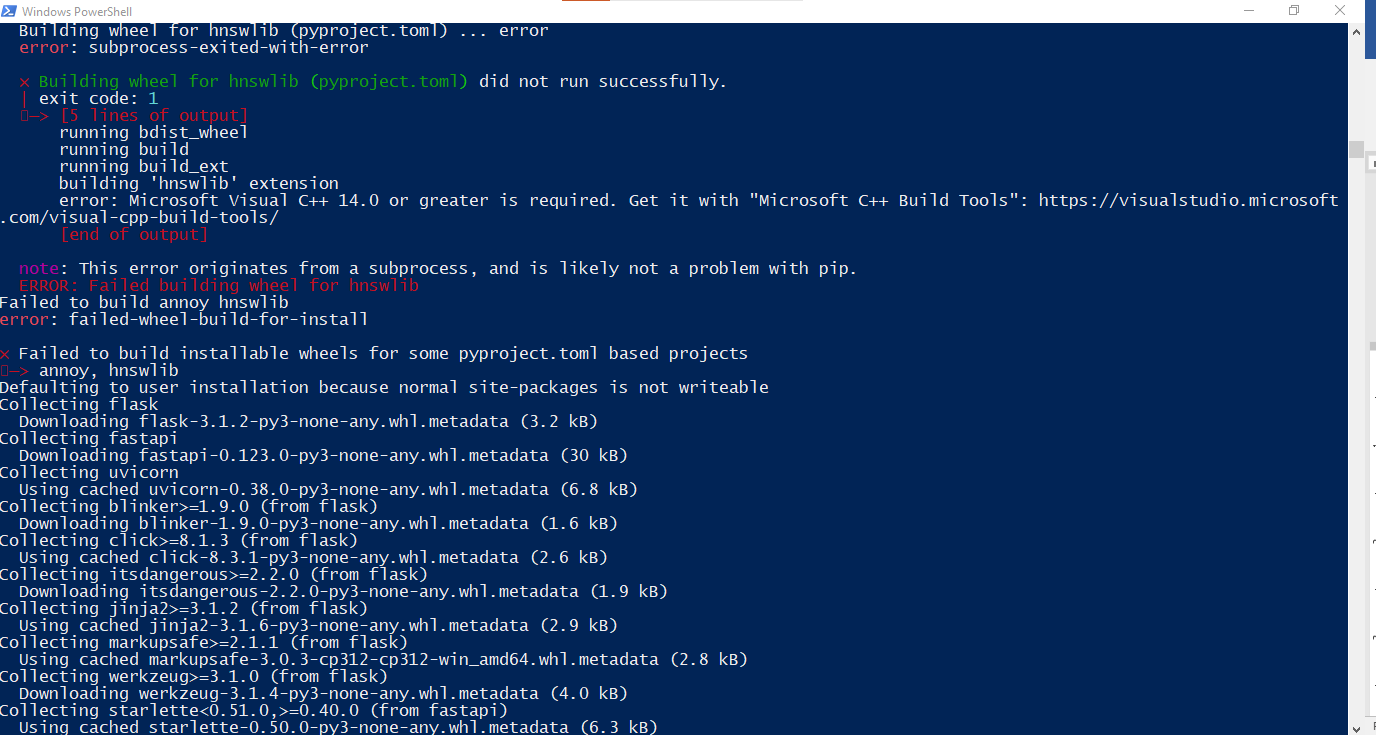


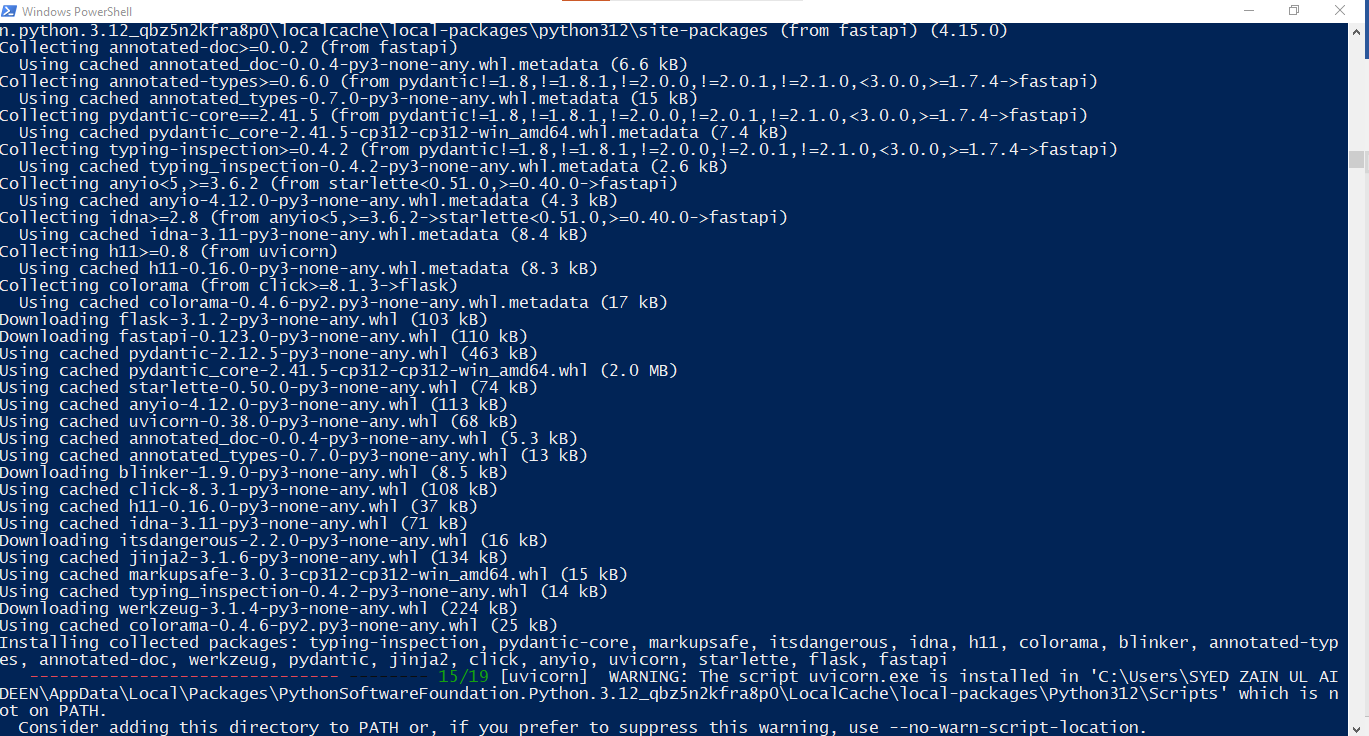


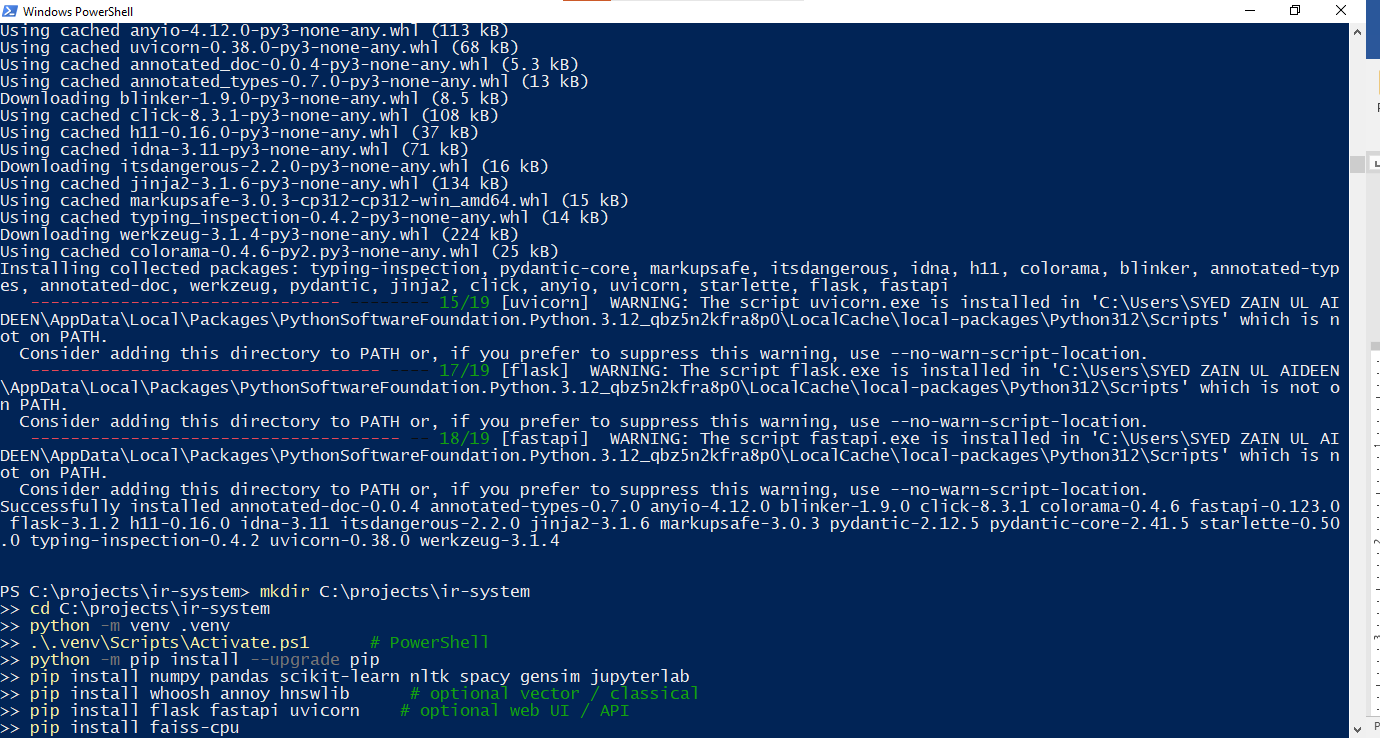


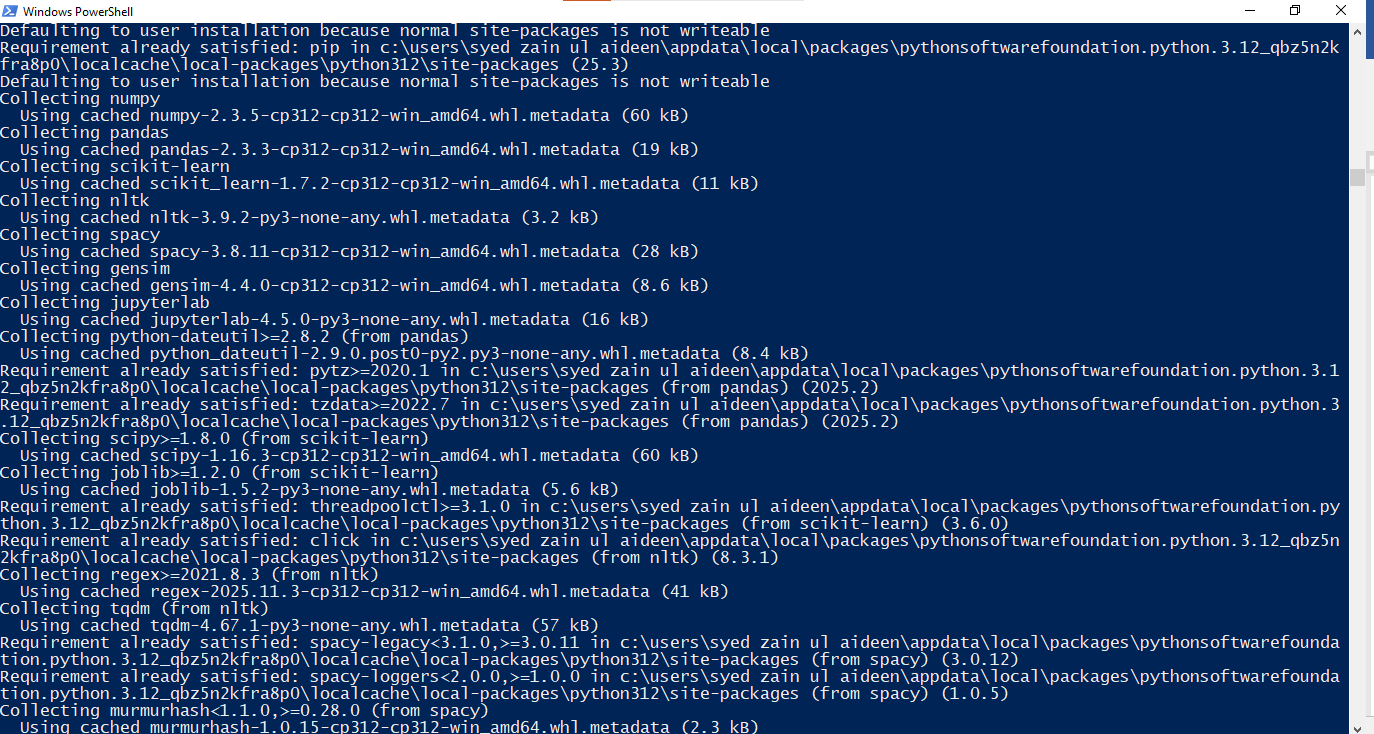


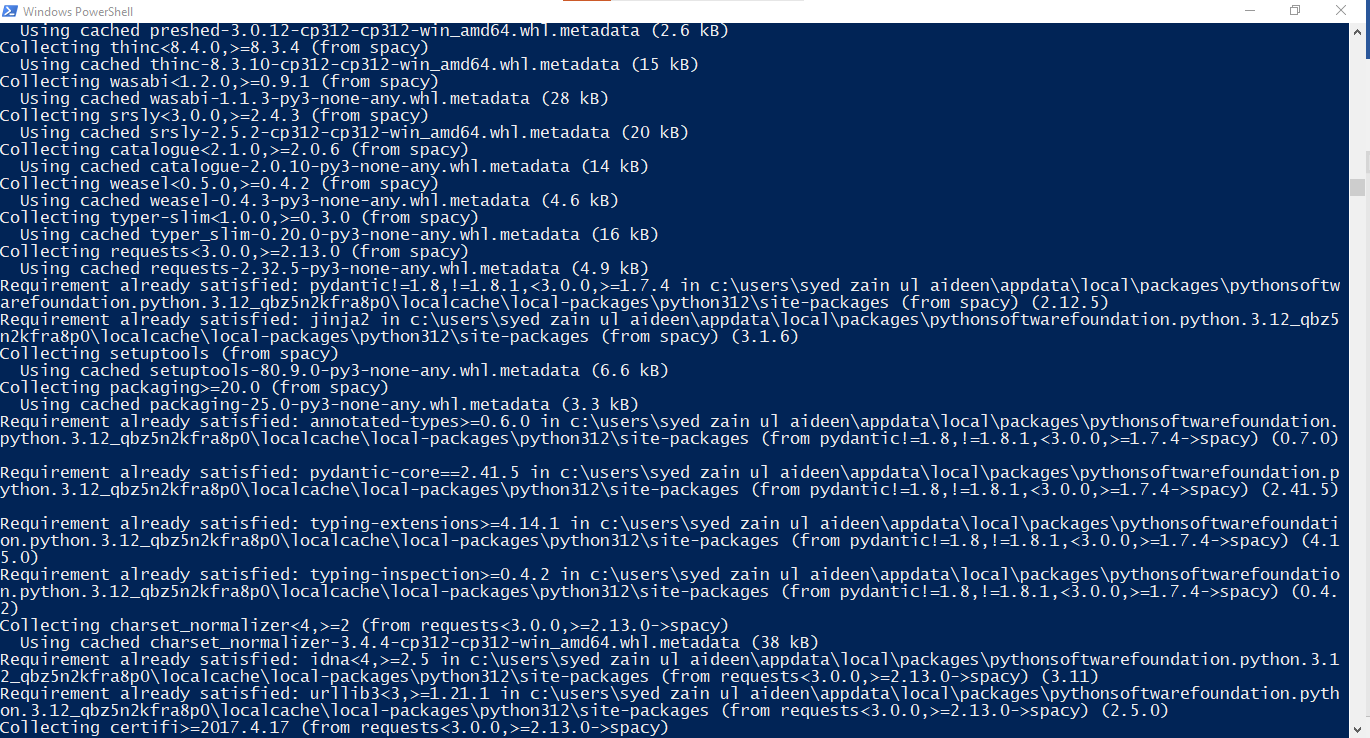


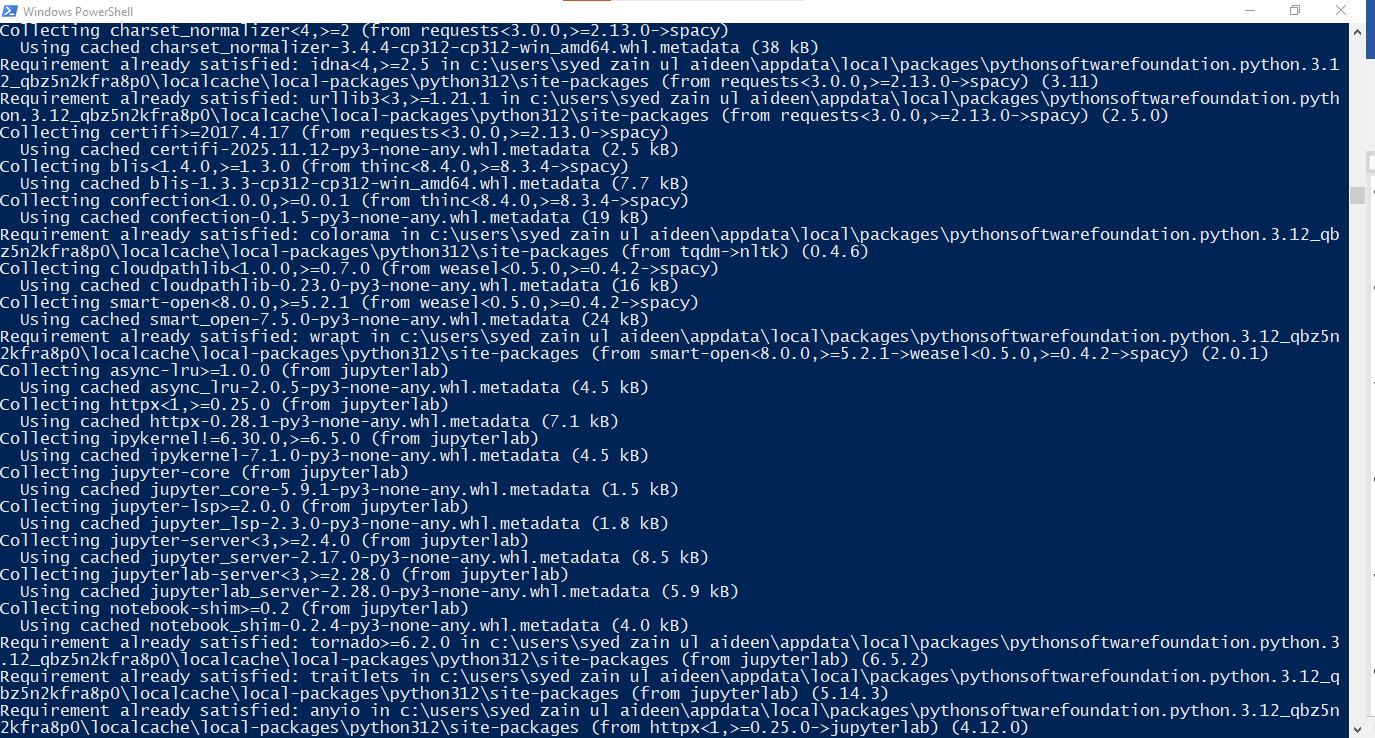


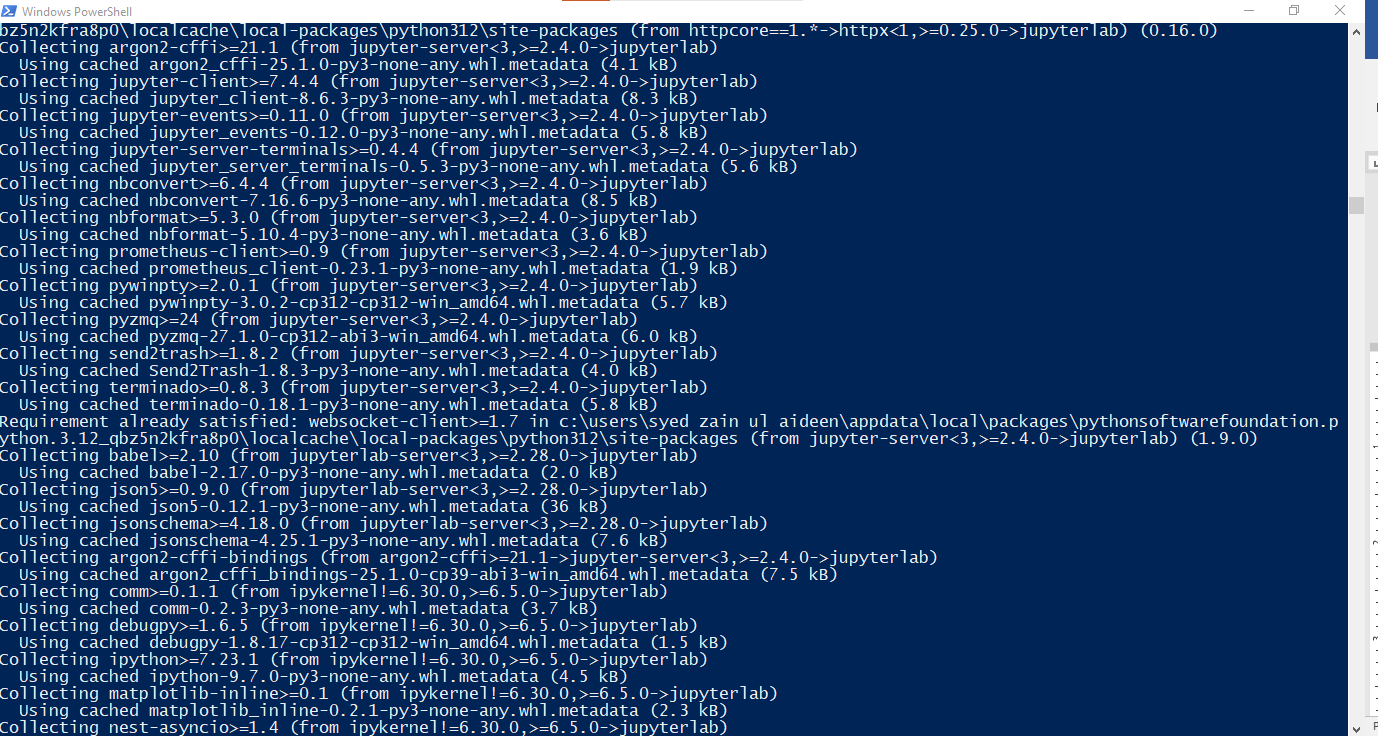




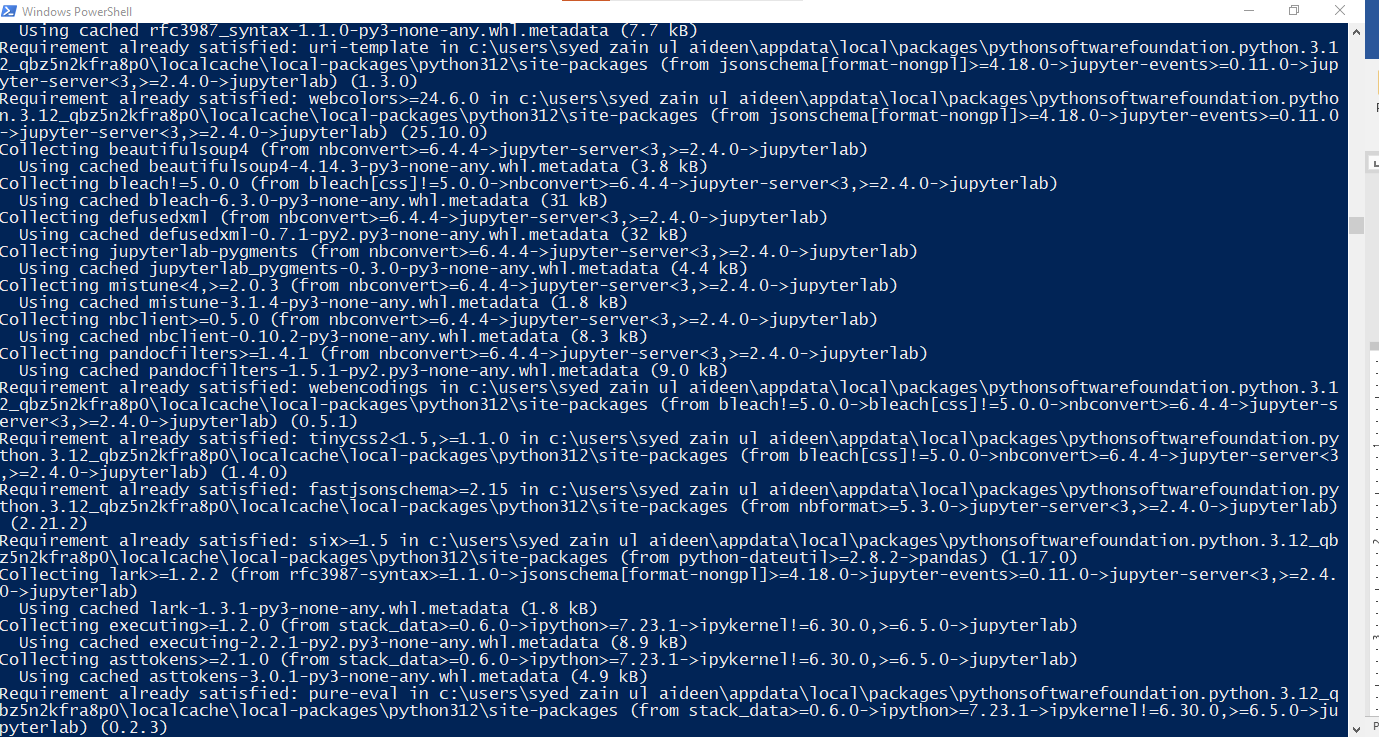


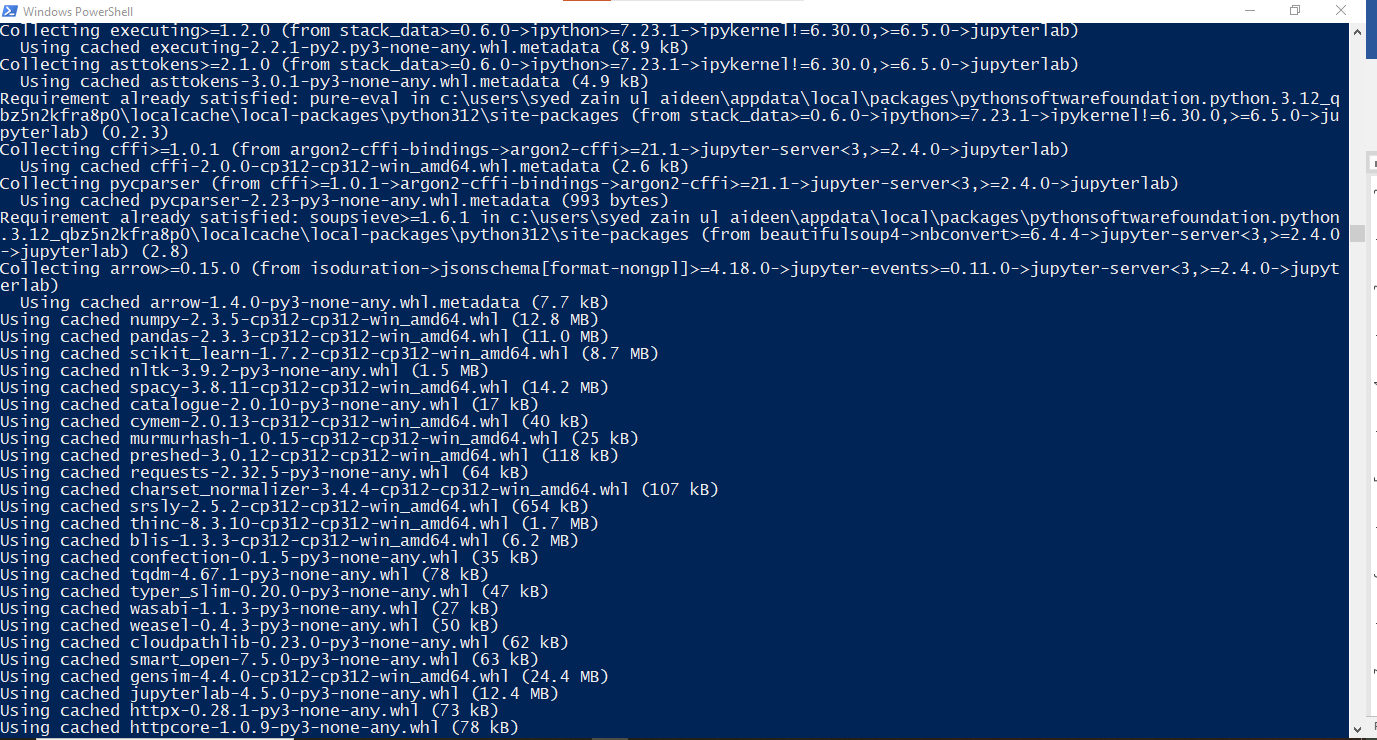


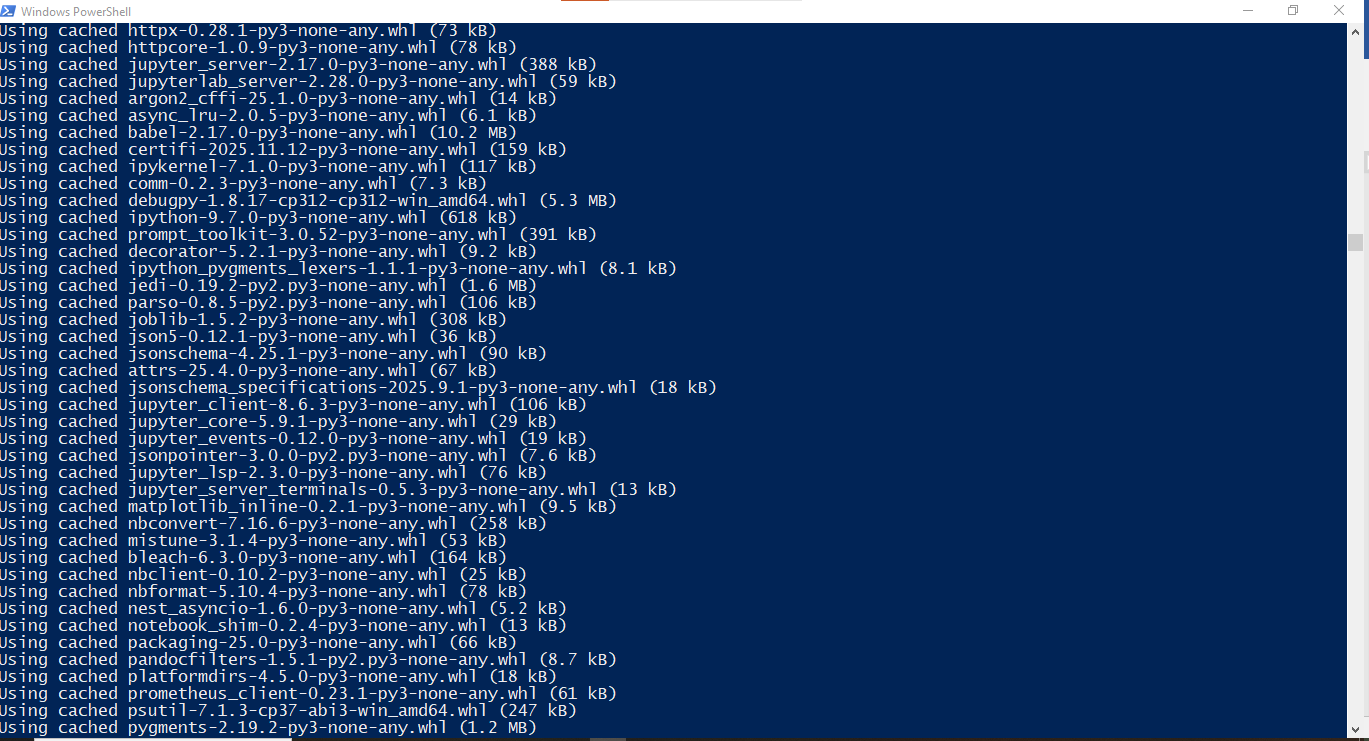


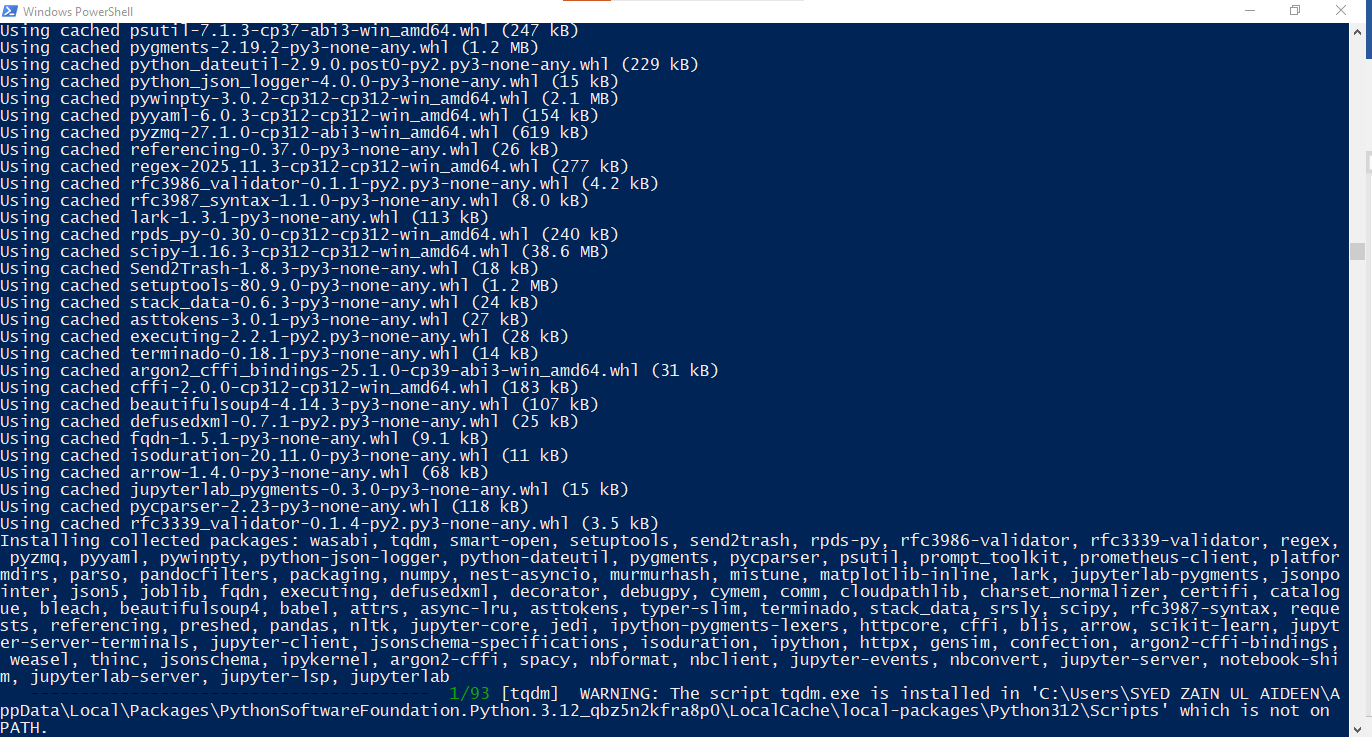


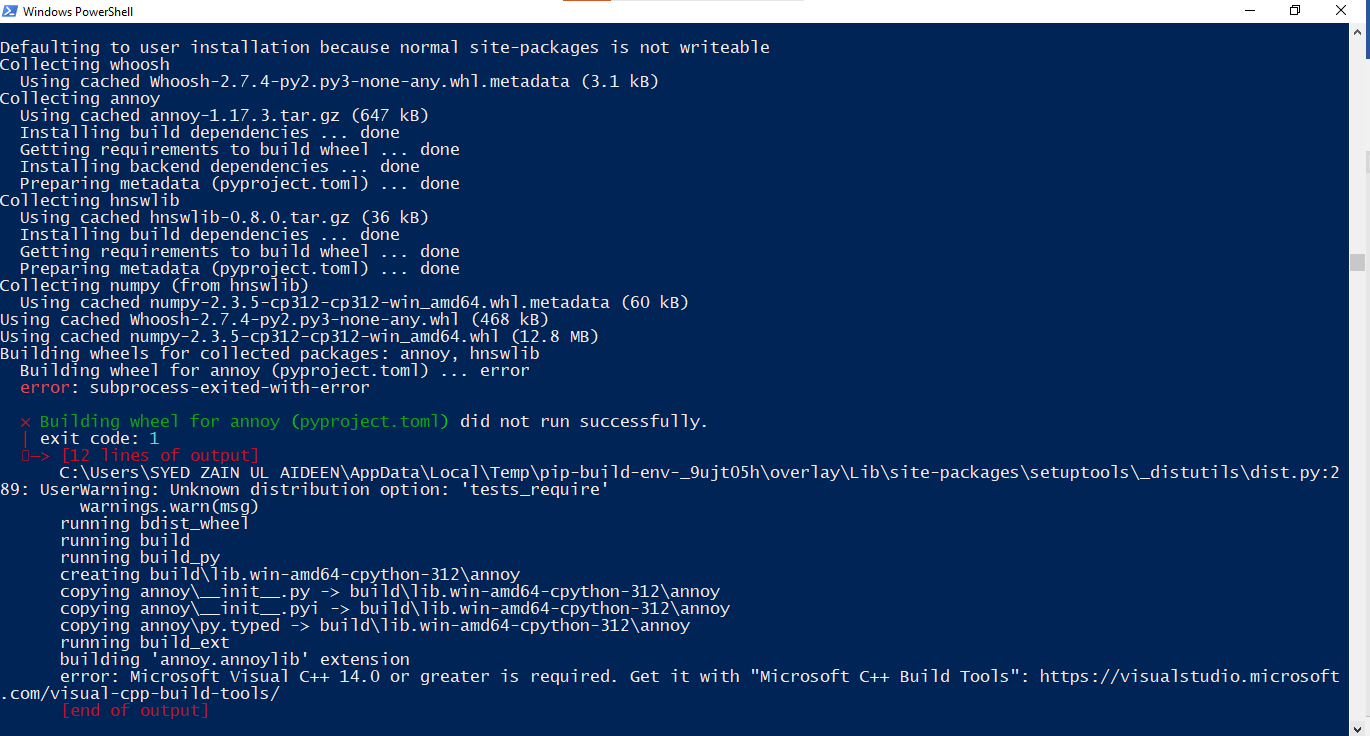


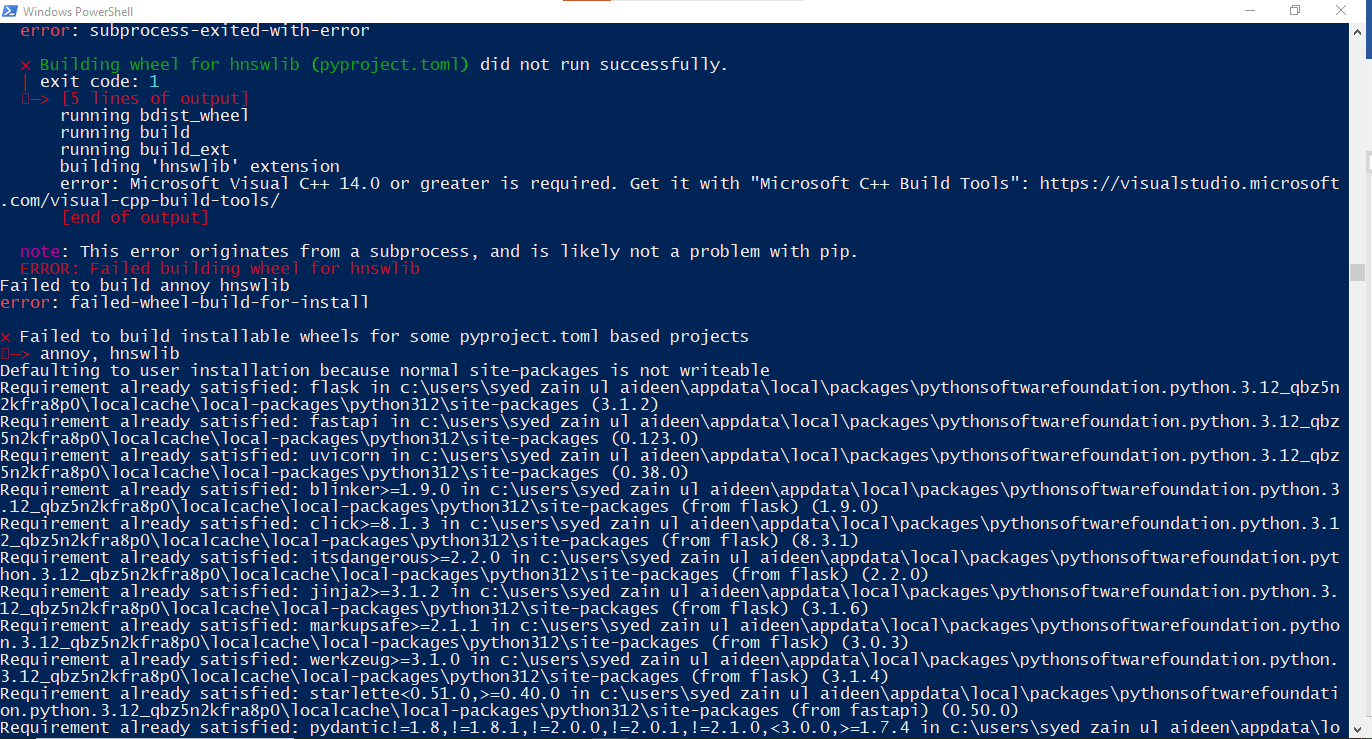




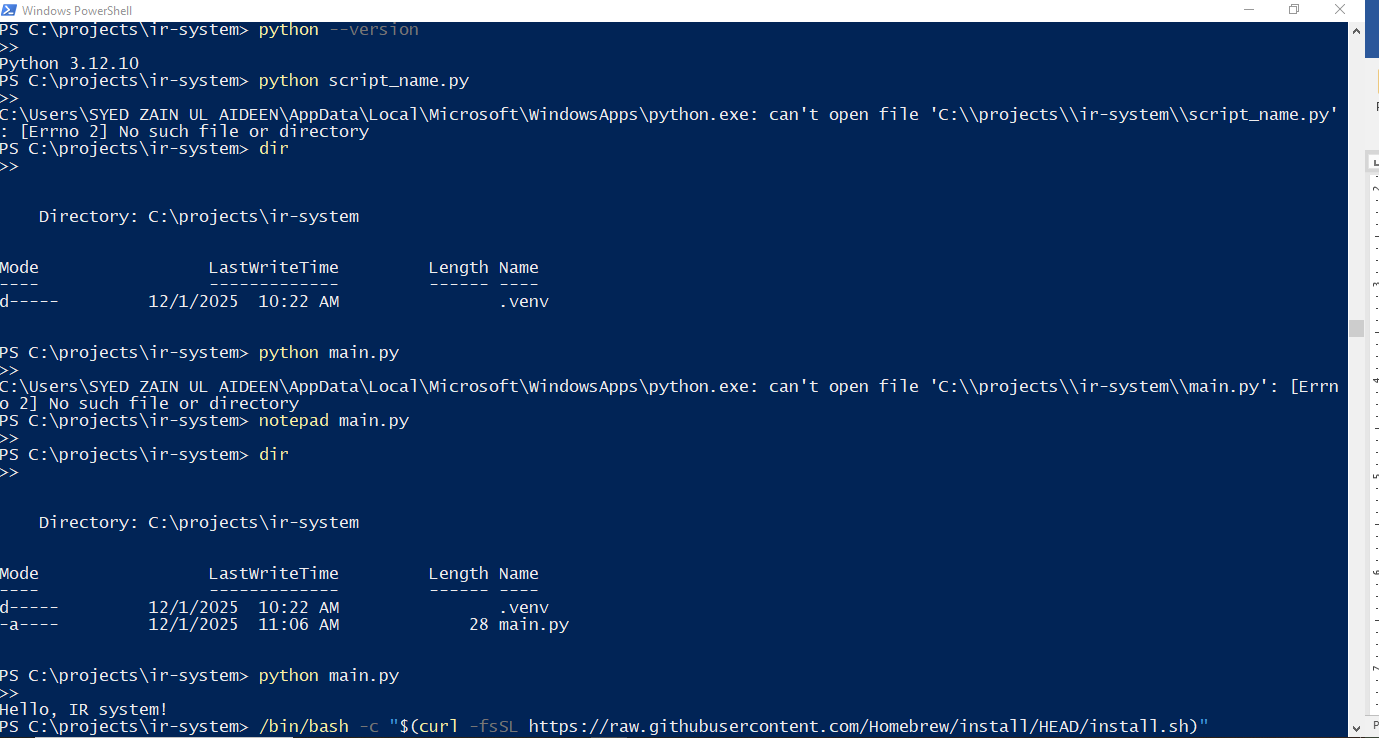




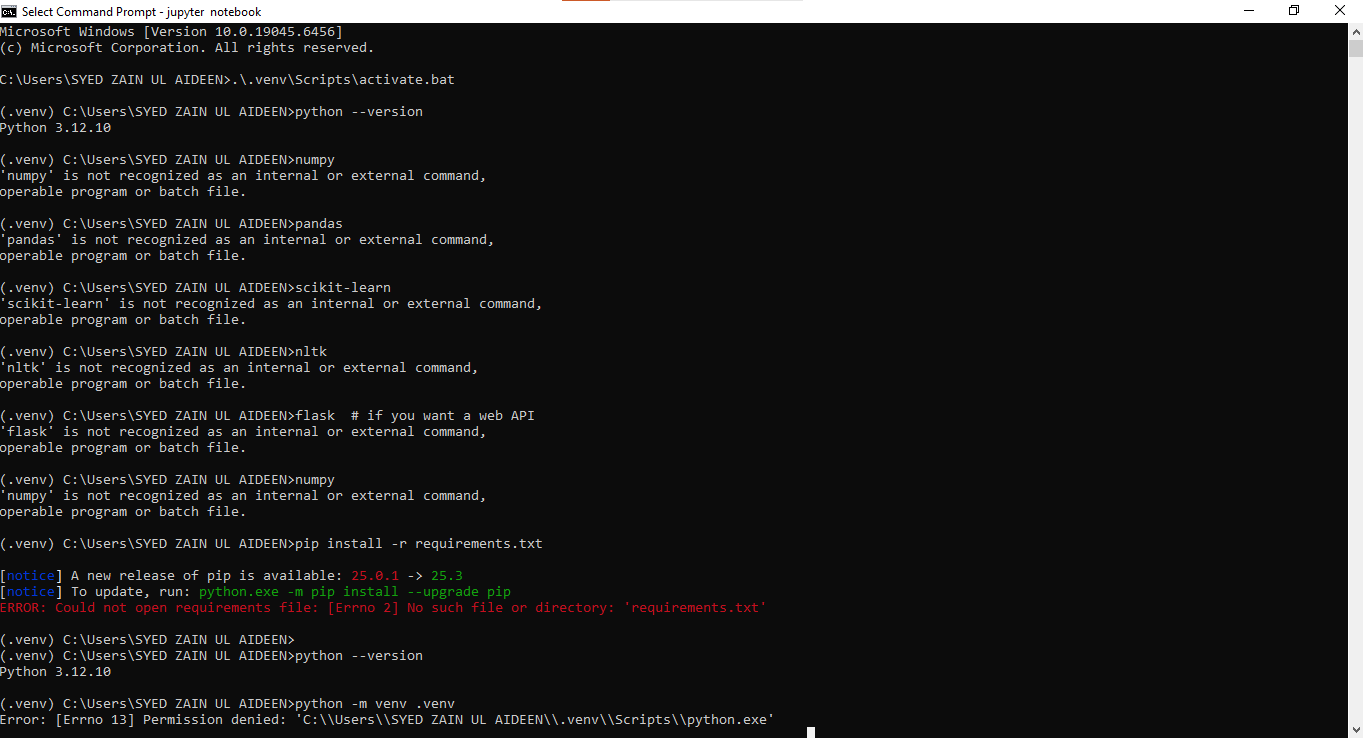


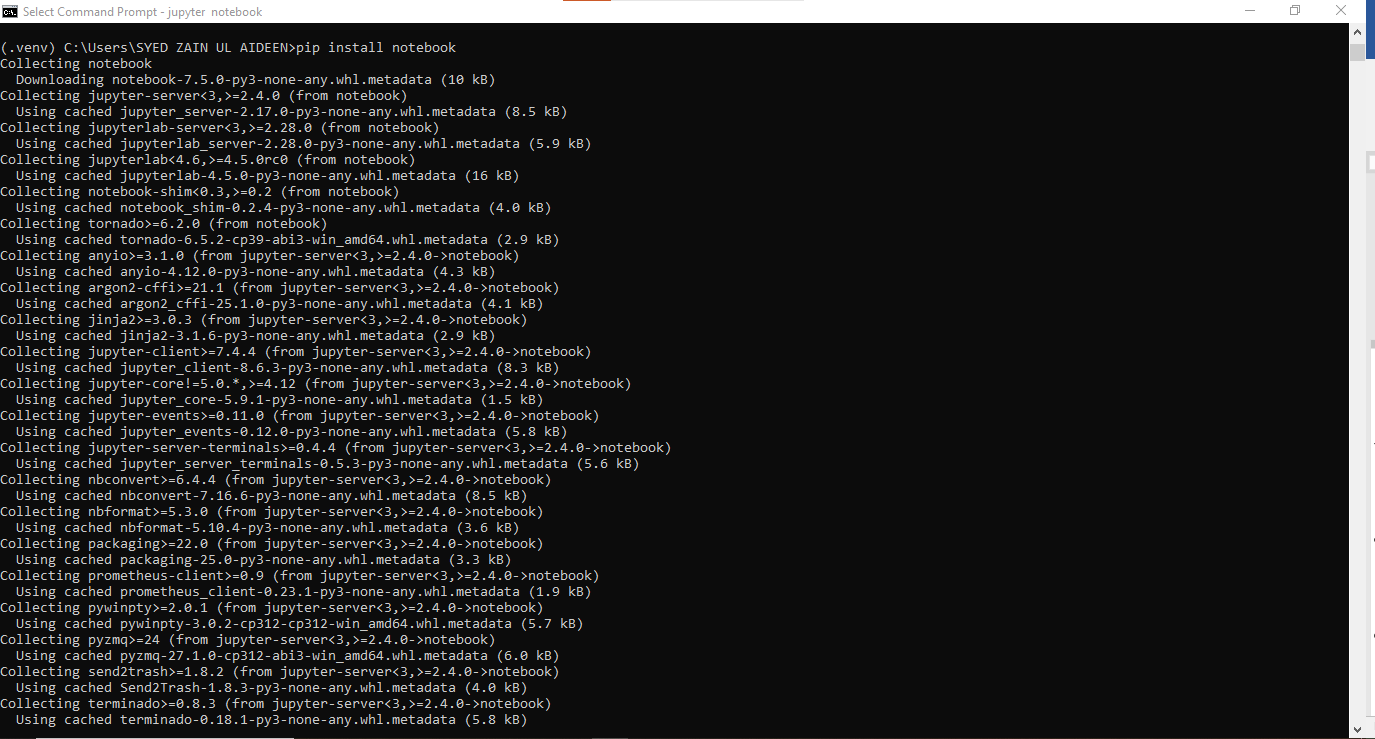


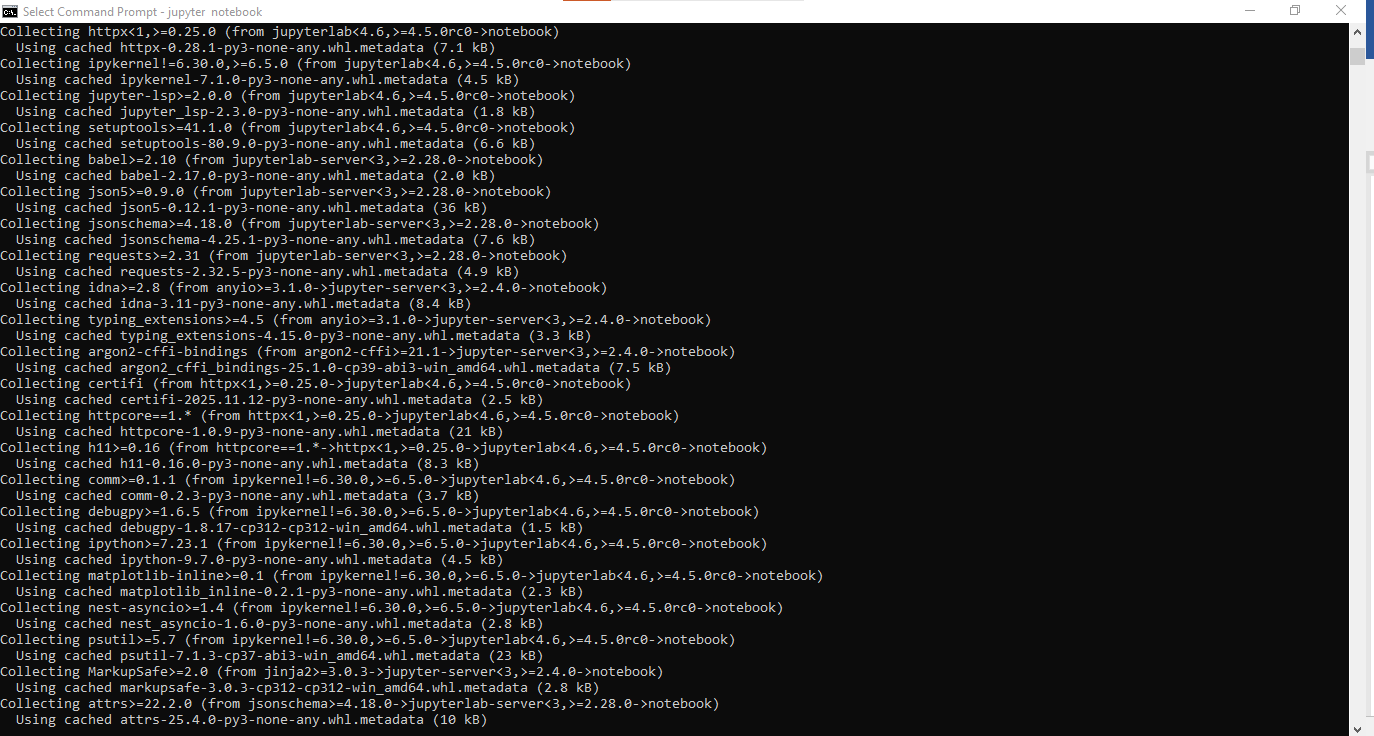


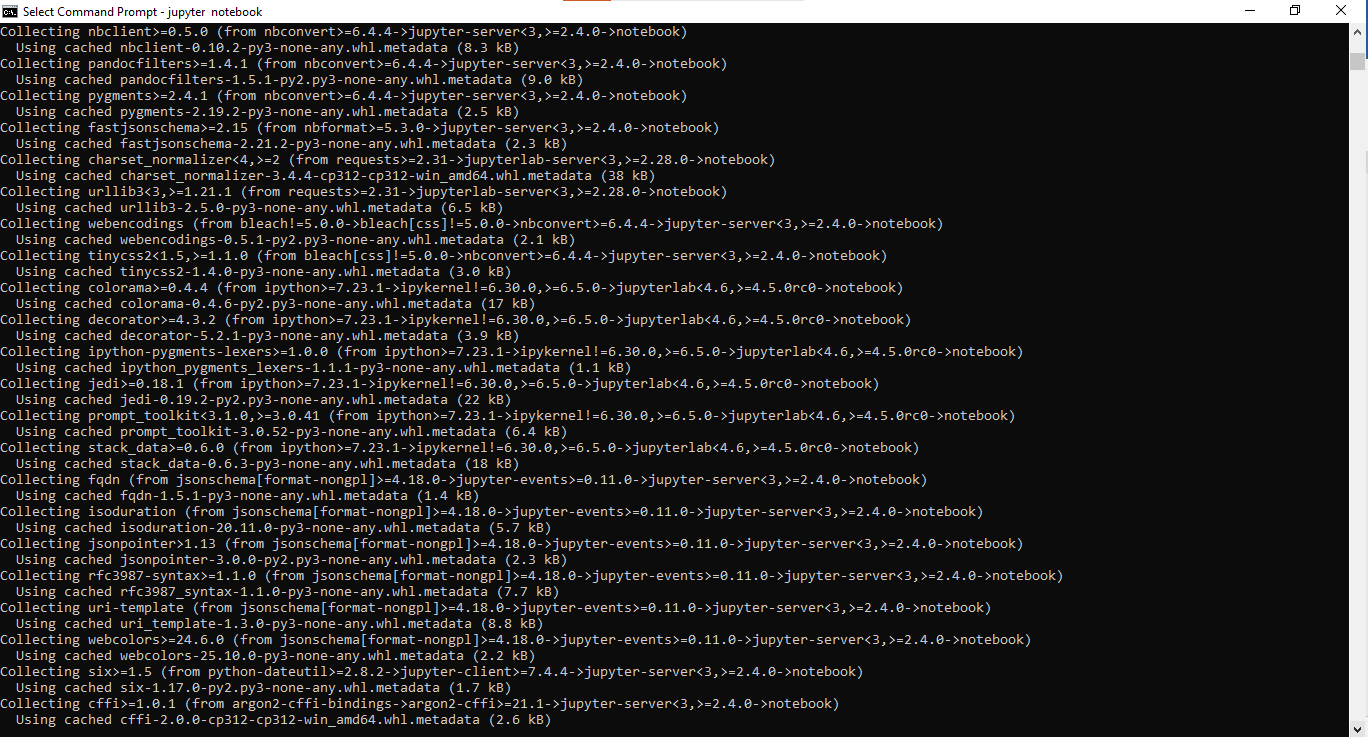


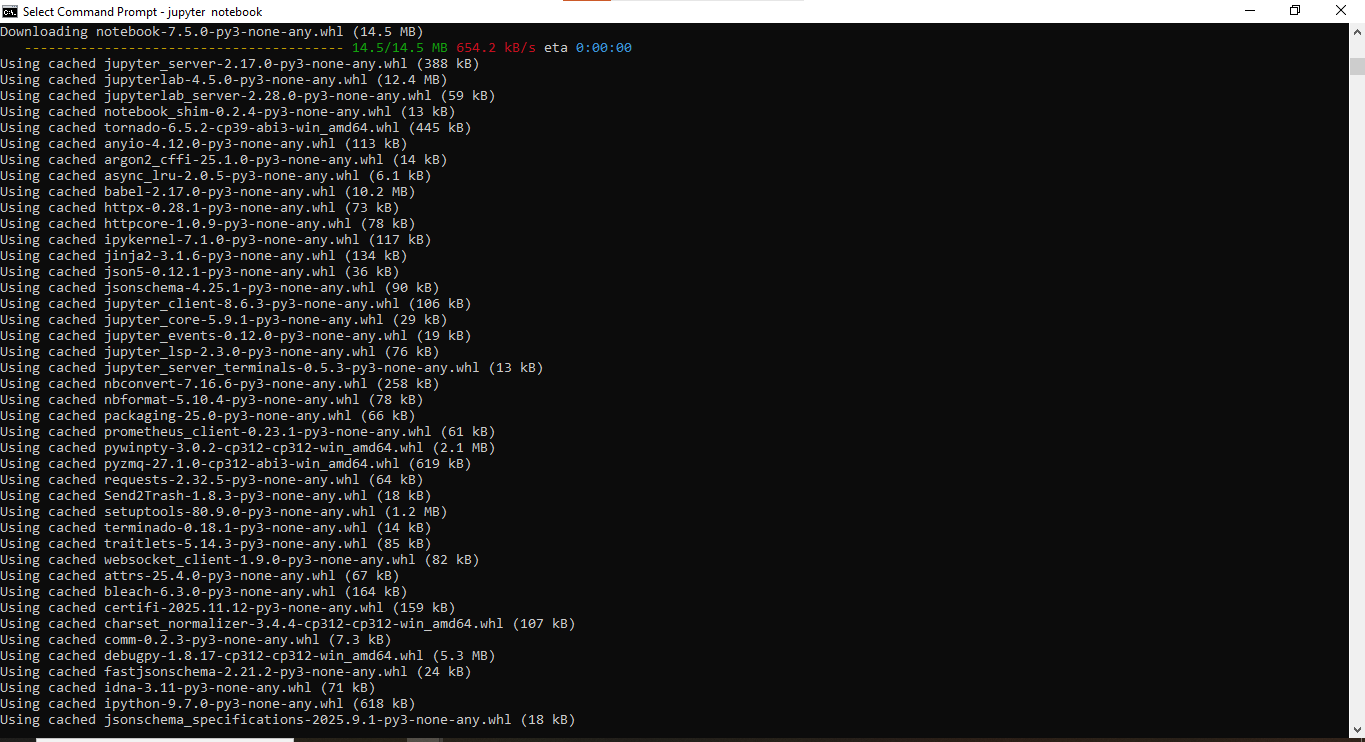


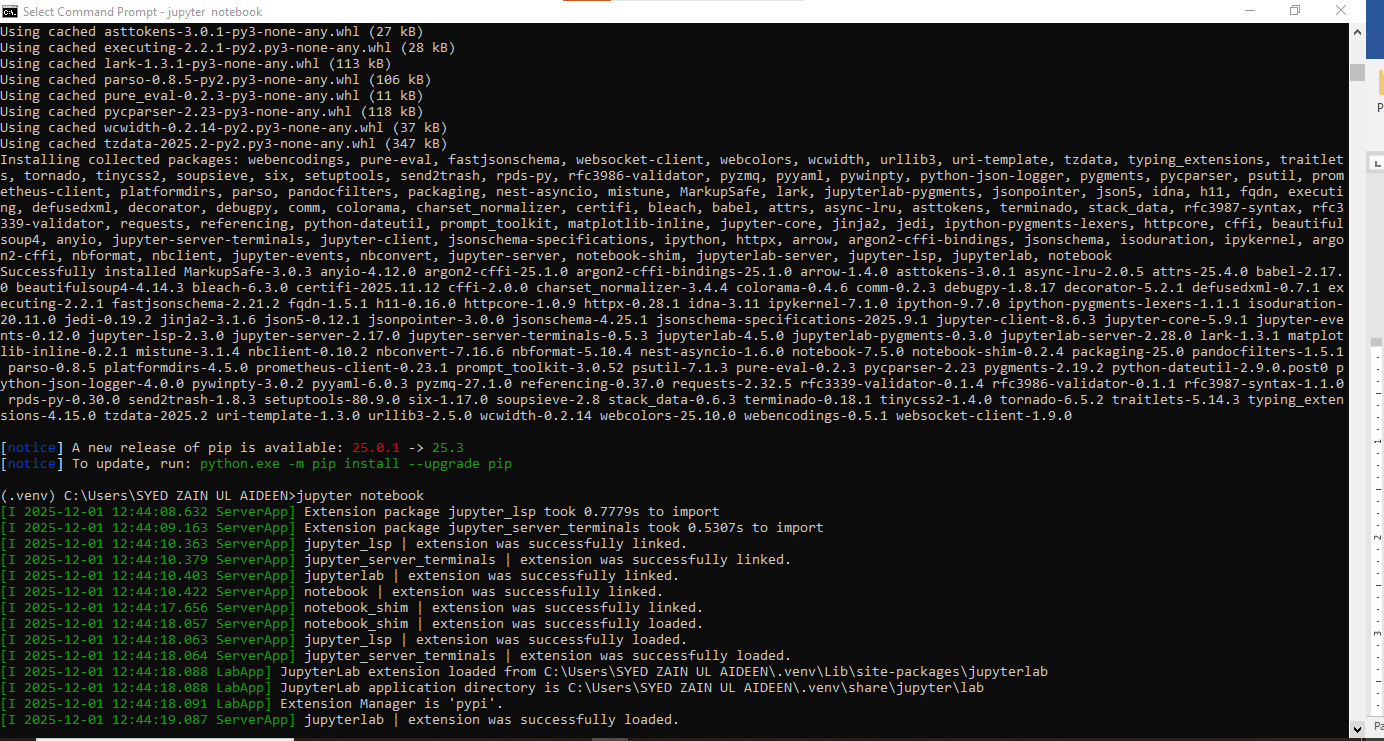


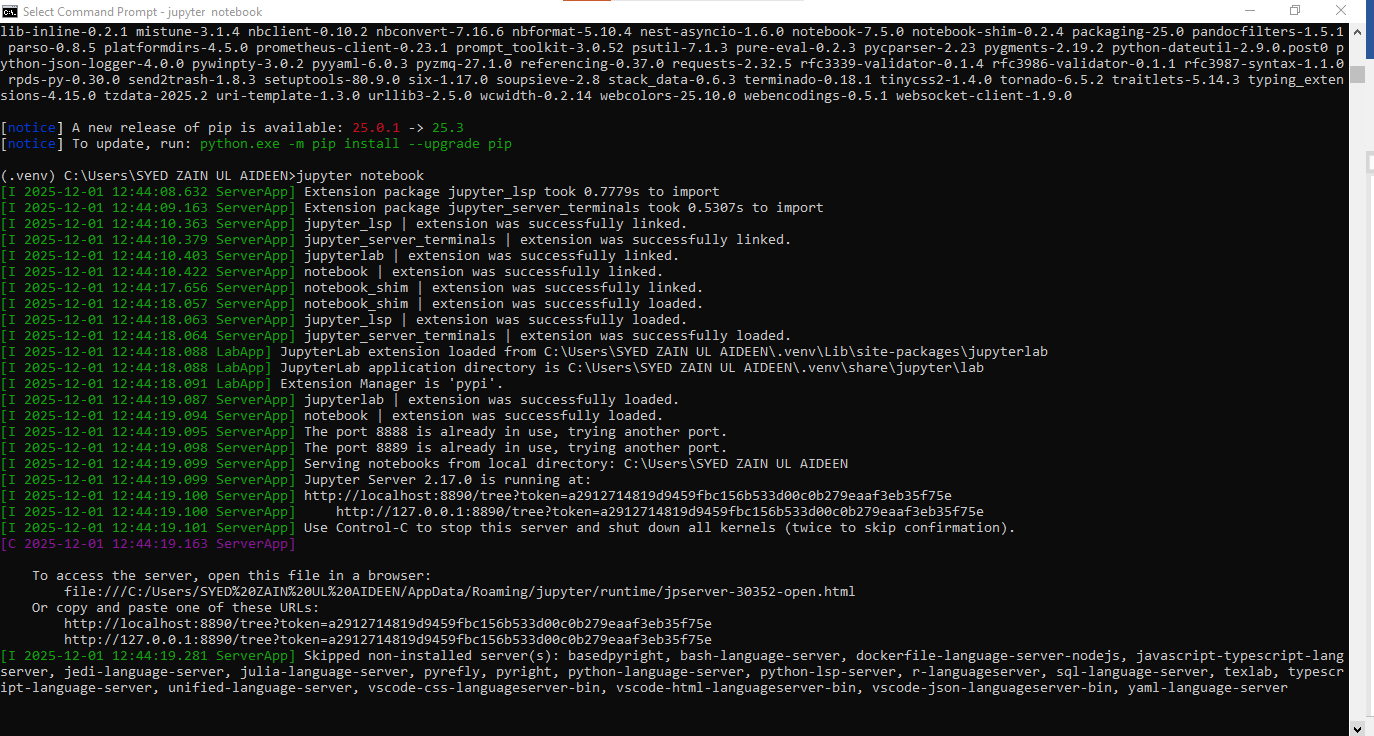






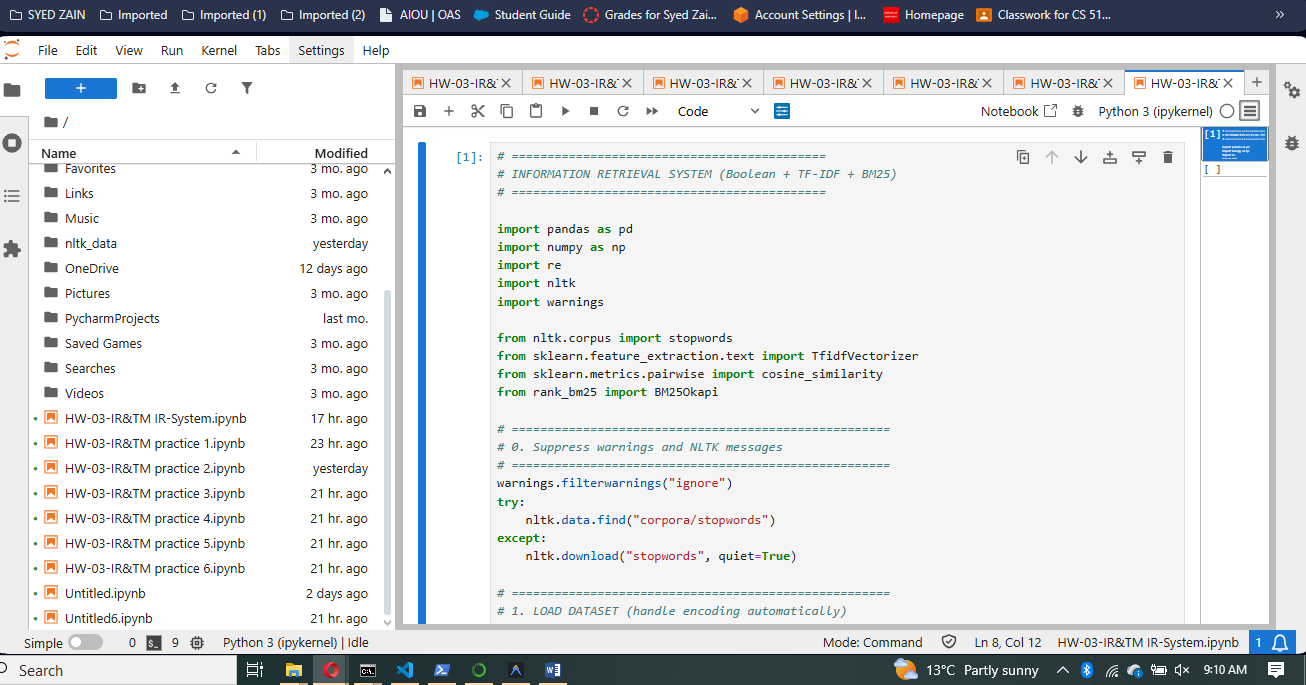
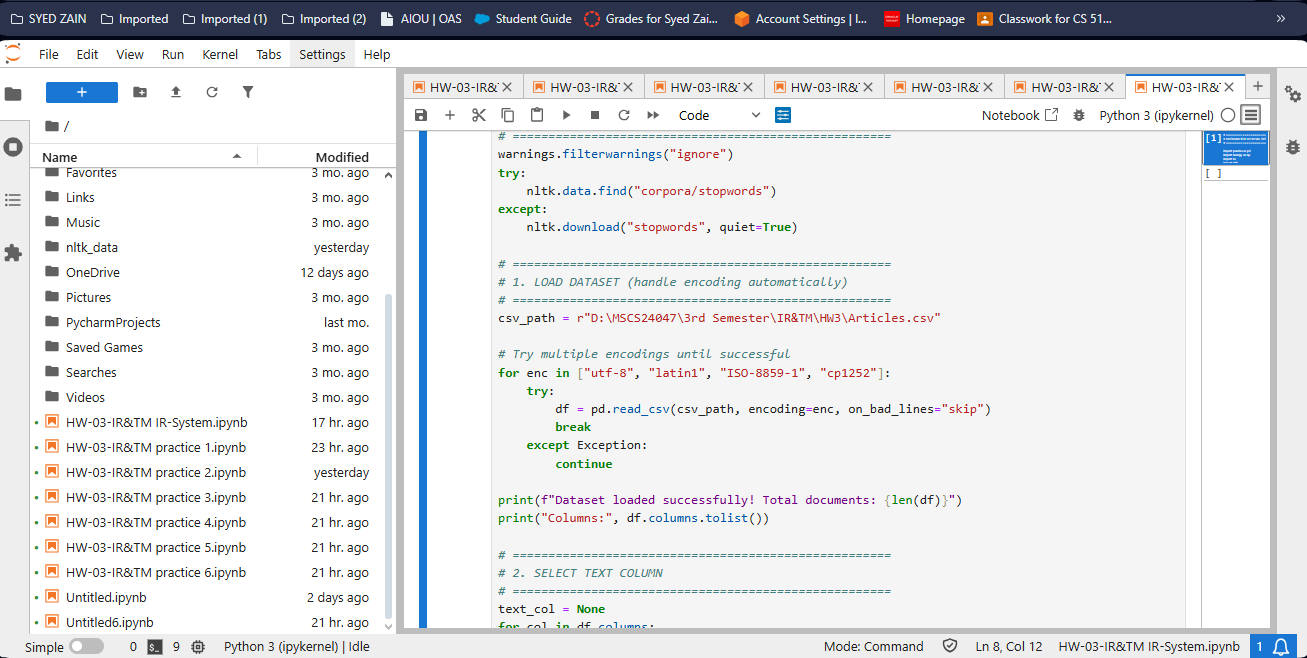
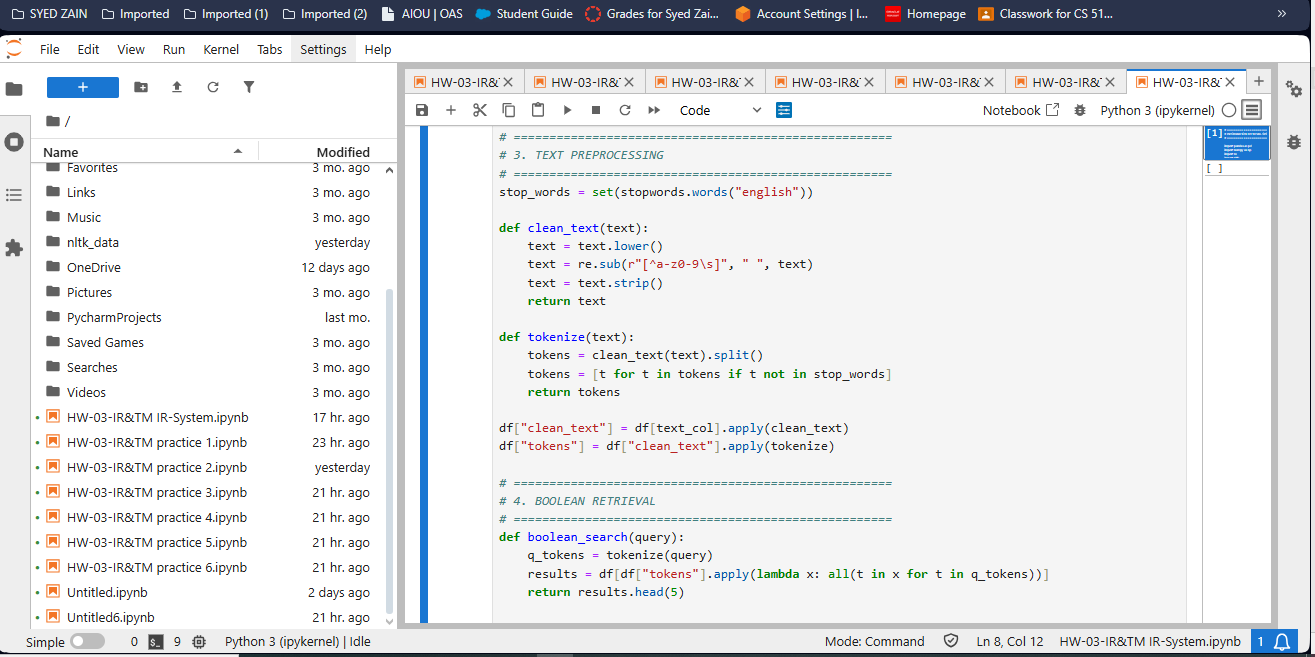
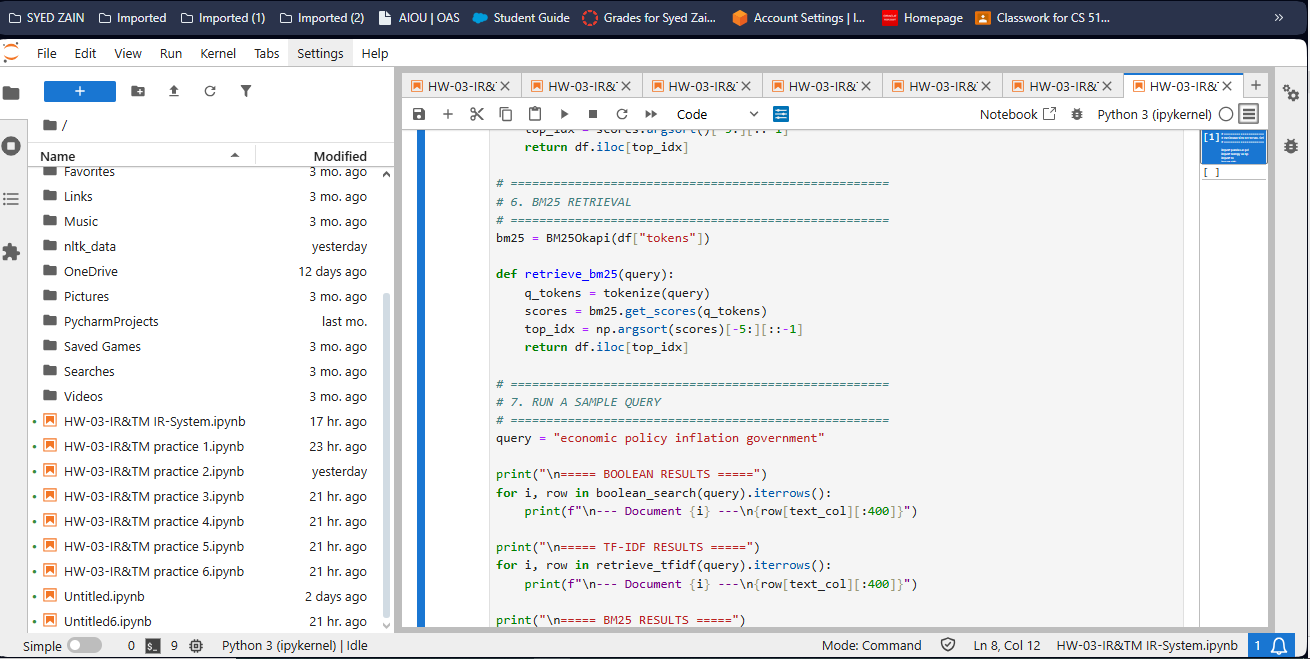
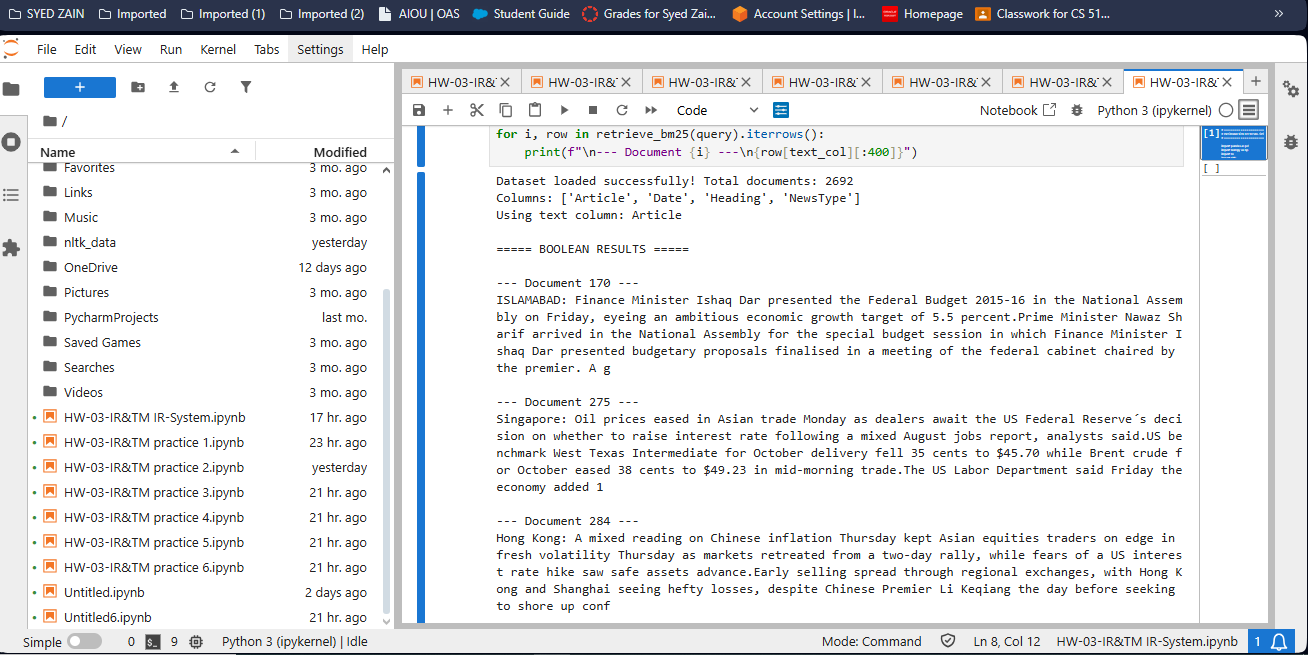
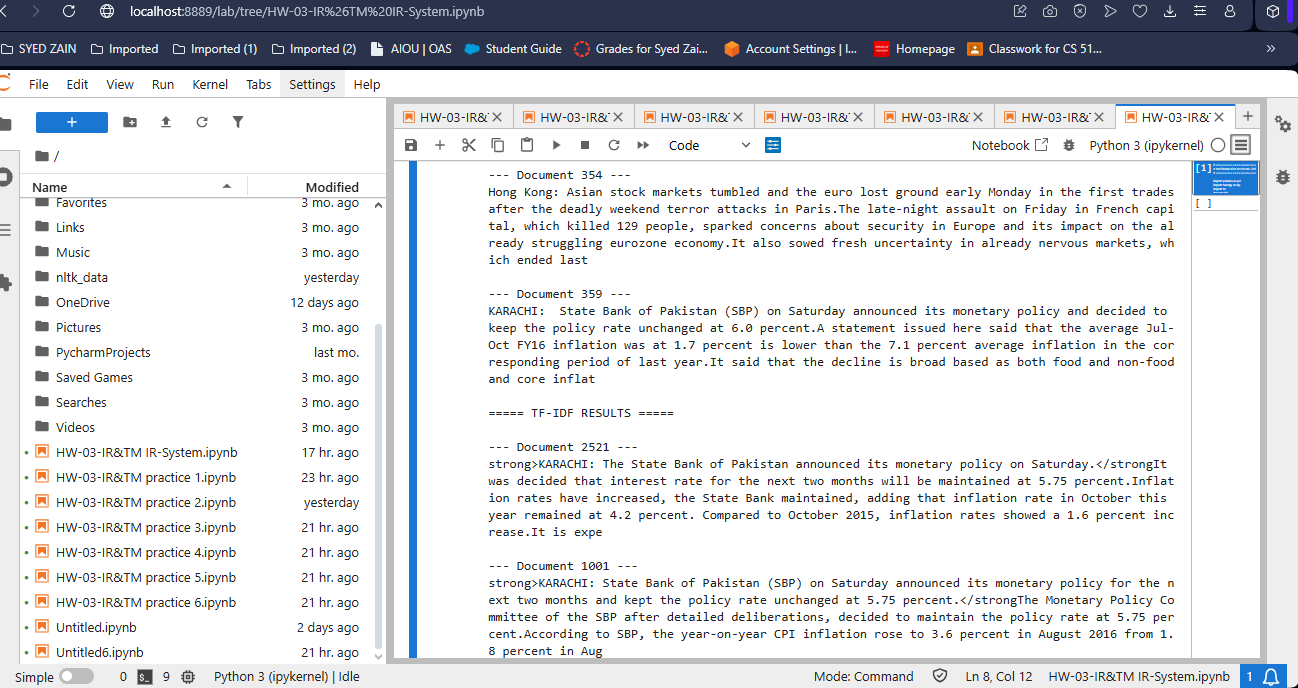
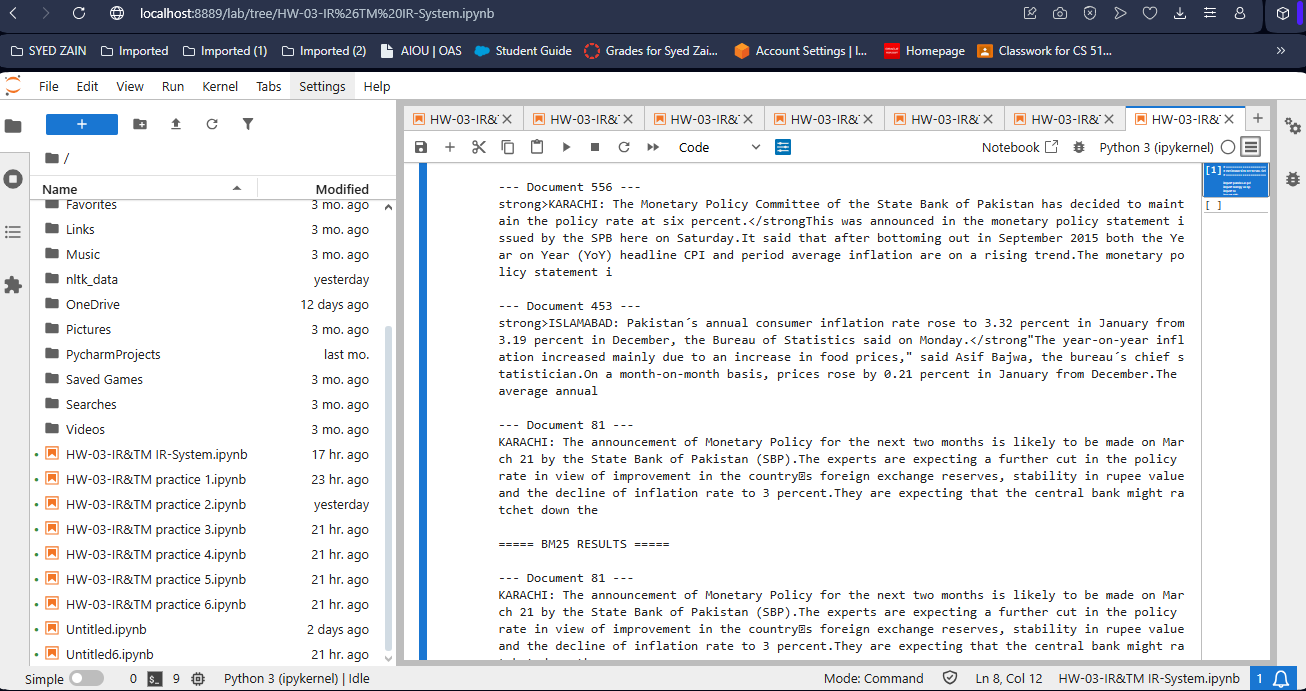
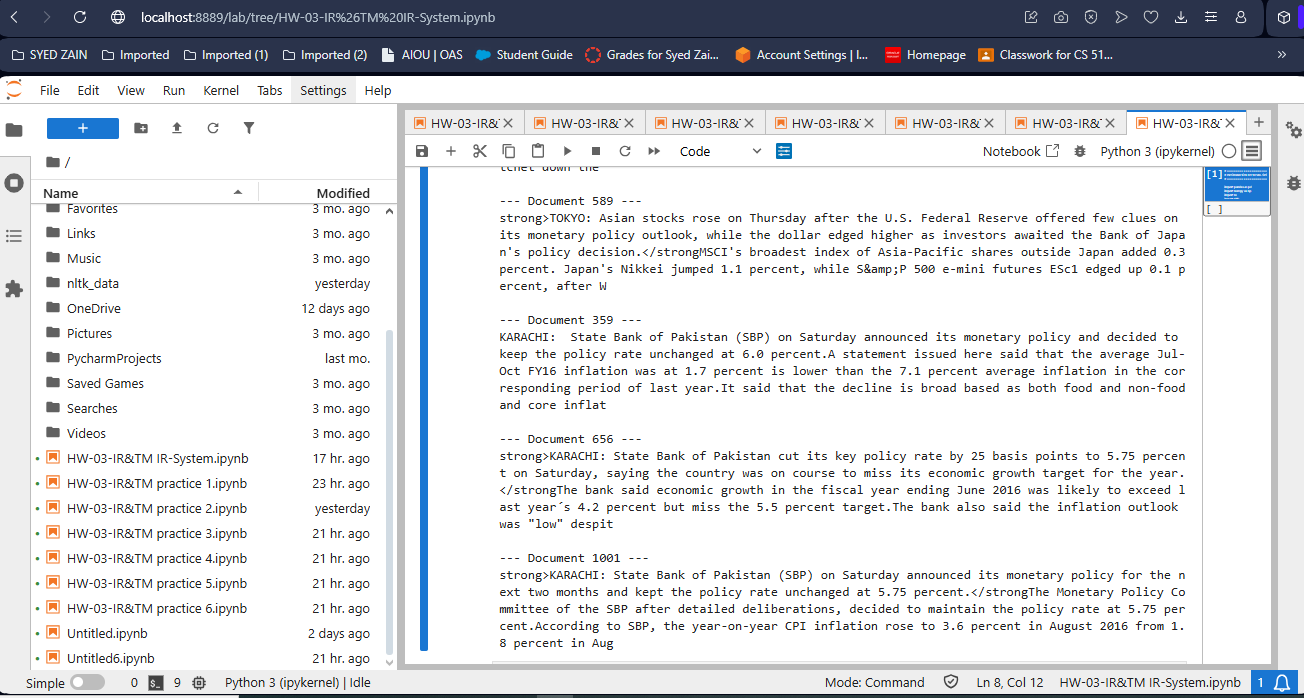






**2. Reproducible Pipeline**

Your submission must include:

* source code  
    
  
* 
* 
* 
* 
* 
* 
* a [README](README.txt) with instructions to run your system

This ensures instructors can test your pipeline.

**3. Technical Report**

Technical Report: Information Retrieval System using the Boolean, TF-IDF, BM25

# 1. System Architecture

## 1.1 System Diagram

Retrieval

(Boolean, TF-IDF, BM25)

Indexing

(TF-IDF, BM25,

Boolean)

Preprocessing

(Clean, Token)

Data Ingestion

(CSV Load)

Ranking & Evaluation

## 1.2 Figure Caption

The system begins by ingesting raw CSV documents. Text is preprocessed through cleaning, lowercasing, and tokenization. Preprocessed documents are then indexed using Boolean, TF-IDF, and BM25 techniques. Queries are processed through these indices, and the top-ranked documents are returned. Evaluation measures system efficiency and retrieval quality.

# 2. Description of the Retrieval System

The system consists of the following modules:

**Data Preprocessing:**

* **Normalization:** Converts all text to lowercase.
* **Punctuation Removal:** Non-alphanumeric characters are removed.
* **Tokenization:** Text is split into individual tokens.
* **Stop word Removal:** Common English stopwords are removed using NLTK's stop words corpus.

**Indexing Techniques:**

* **Boolean Retrieval:** Matches documents containing all query tokens.
* **TF-IDF Vectorization:** Converts documents into term-frequency inverse-document-frequency vectors. Cosine similarity is used to rank documents based on their relevance to the query vector.
* **BM25 Retrieval:** Uses probabilistic retrieval to score documents based on term frequency and document length normalization.

**Modifications/Justifications**

* Automatic selection of the text column from CSV to enhance usability.
* Handling multiple CSV encodings to ensure robustness across datasets.
* Use of stop words to reduce noise and improve ranking accuracy.

# 3. Evaluation

**Qualitative Evaluation:**

* Checked top-5 results for sample queries.
* Verified relevance of returned documents using keyword matching and topic similarity.

**Quantitative Evaluation:**

* Scoring based on cosine similarity for TF-IDF.
* BM25 scores provided probabilistic ranking.
* Retrieval speed is reasonable for datasets of ~2,500 documents.
* Memory footprint is minimized using sparse TF-IDF matrices and tokenized lists for BM25.
* Environment and Configuration Files for IR System

## 1. Requirements.txt

This file lists all Python dependencies needed to run the Information Retrieval system.

pandas  
numpy  
scikit-learn  
nltk  
rank\_bm25

## 2. Optional: .env file (for environment variables)

**If you want to configure paths or other settings, you can create a .env file in the project root:**

# Path to dataset  
DATASET\_PATH=./data/Articles.csv  
  
# Number of top results to retrieve  
TOP\_K=5

## 3. Python Version

* Python 3.9 or above is recommended.

## 4. Optional: Jupyter Notebook Kernel Configuration

* If running in Jupyter, ensure the correct Python environment is selected.
* Install ipykernel in your environment:
* pip install ipykernel  
  python -m ipykernel install --user --name=ir\_env

## Notes

* The system will automatically handle stopwords and CSV encoding.
* Make sure the dataset CSV is placed in the path specified in either .env or directly in src/main.py.
* The requirements.txt file is sufficient to replicate the environment in a new machine.

# 4. Discussion

**Findings:**

* BM25 generally produces more contextually relevant results compared to Boolean and TF-IDF.
* Boolean retrieval is fast but may miss partially relevant documents.
* TF-IDF retrieval balances speed and relevance but is sensitive to term weighting.

**Shortcomings:**

* No synonym or semantic query expansion.
* No relevance feedback loop.
* Evaluation limited to small datasets.  
  **Future Improvements:**
* Integrate Word Embedding’s or Transformer-based retrieval.
* Add relevance feedback and query expansion.
* Implement caching and optimization for larger datasets.

# 5. References

1. Manning, C., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.  
2. Scikit-learn Documentation: https://scikit-learn.org/stable/  
3. Rank-BM25 Python Package: https://pypi.org/project/rank-bm25/  
4. NLTK Documentation: https://www.nltk.org/

# 6. Disclosure of AI Use

## 6.1 Summary of AI Usage

## 

## 

## 

## 