MINI PROJECT 2

- **Course Name: CMPUT 291 (Winter 2025)**
- **Group Members: Abubakar Shaikh (amshaik1) Ujjawal Pratap Singh (upsingh) Fatin Ahmed (fatin2) Stevin Santhosh (stevin1)**
- **Date: 8th April 2025**

---

2. General Overview of the System

Objective:
The goal of this phase was to implement a query interface in Python to interact with a MongoDB database. The system allows users to perform various analyses on product reviews stored in the 291db database. These analyses include retrieving product ratings, identifying top-rated products, listing active reviewers, tracking reviews over time, and flagging suspicious reviews.

Main Features:

- **Retrieve average rating and total reviews for a given ASIN.**
- **Find the top N highest-rated products by average rating.**
- **List the 10 most active reviewers based on the number of reviews submitted.**
- **Show how the number of reviews for a product changes over the years.**
- **Identify suspicious reviews based on low helpfulness and high ratings.**

Technologies Used:

- **Python: Programming language.**
- **PyMongo: Library to interact with MongoDB.**
- **MongoDB: NoSQL database for storing product reviews.**

---

3. User Guide

Program Interaction:
When the program is run, the user will interact with a menu-driven interface that supports the following options:

1. Get product rating by ASIN:
   The user provides the ASIN (Amazon Standard Identification Number) of a product, and the system retrieves the average rating and total reviews for that product.

2. Find top N products:
   The user can enter the number of top-rated products they wish to see. The system displays the top N products sorted by their average rating.

3. List most active reviewers:
   The program lists the 10 most active reviewers based on the number of reviews they've written.

4. Show reviews over time:
   The user inputs up to 5 years, and the system displays how the number of reviews for a product has changed over those years.

5. Flag suspicious reviews:
   The system flags reviews with high ratings (≥4.5) but low helpfulness (fewer than 10% of users found the review helpful).

6. Exit:
   Exits the program.

Usage Example:

bash

CopyEdit

```
$ python phase2_query.py 27017
```

After running the script, the user will be presented with a menu, allowing them to choose from the listed options.

---

4. Group Work Breakdown Strategy

Work Items and Responsibilities:
Each group member was assigned specific tasks to ensure efficient development and completion of the project.

- **Abubakar Shaikh:**
  - **Responsible for setting up the MongoDB connection and writing the functions to query the database.**
  - **Worked on implementing the get_product_rating**
- **Ujjawal:**
  - **Focused on implementing the logic for the Reviews Over Time and Suspicious Reviews queries.**
- **Fatin:**
  - **Took charge of testing the functionality, ensuring the queries returned accurate results.**
  - **Managed error handling and worked on refining the program's flow.**
- **Stevin**
  - **Handled the user interface portion and integrated all features into a menu-driven system.**
  - **get_top_products, and get_most_active_reviewers functions.**

Time Spent:

- **Abubakar: Estimated time spent: 12 hours.**
- **Ujjawal: Estimated time spent: 14 hours.**
- **Fatin: Estimated time spent: 10 hours.**
- **Stevin: Estimated time spent: 10 hours.**

5. Communication and Project Tracking

Communication Methods:

- **Slack: Primary communication tool for day-to-day discussions and progress tracking.**
- **GitHub: Used for code sharing and version control. Each member pushed code regularly and conducted pull requests for review.**
- **Google Docs: For collaborative documentation, particularly in planning and reporting.**

Progress Updates:
Weekly meetings were held via Zoom to review the progress of the project. During these meetings, we discussed any blockers, code review feedback, and next steps. Regular updates were shared through Slack to ensure continuous progress.

---

6. Decisions Made and Assumptions

Decisions:

- **JSON File Processing: We decided to process the JSON file in batches of 5000 records to optimize memory usage when loading data into MongoDB.**
- **MongoDB Aggregation: We used MongoDB's aggregation framework for most queries to handle large datasets efficiently, especially for tasks like calculating average ratings and identifying suspicious reviews.**

Assumptions:

- **The MongoDB server is running on localhost and accessible on the specified port.**
- **The provided data is in a consistent format and can be parsed without errors.**
- **No assumptions were made about missing data or edge cases beyond what is required by the project specification.**

---

7. Conclusion

In this phase of the project, a Python-based MongoDB query interface was developed to analyze product reviews. The system met the requirements of querying the database for ratings, active reviewers, reviews over time, and suspicious reviews. The project successfully implemented MongoDB aggregation queries to provide useful insights, and the team followed a structured approach to develop and document the system. Going forward, improvements could be made in expanding the types of analyses or implementing a graphical interface.