I. System Overview:

This project is a two-phase system designed to process and analyze product review data stored in a MongoDB database.

- Phase 1 (load-json.py): This script loads JSON data into the MongoDB database.
- Phase 2 (phase2_query.py): This script provides various querying functionalities for retrieving insights from the review data.

The system is designed to:

- Store and manage review data in MongoDB.
- Compute average product ratings.
- Identify top-rated products.
- Find the most active reviewers.
- Track review trends over time.
- Detect suspicious reviews.
- Exit from the program

II. User guide:

- Ensure that Python 3.0 or above version and MongoDB is installed in the system.
- Install pymongo (if not already installed):

  Run the following command in your terminal.

  ```
  Pip3 install pymongo
  ```
- Create the database directory:

  Use the command mkdir to create a directory for MongoDB to store its data:

  ```
  Mkdir db_folder
  ```
- Start the MongoDB server:

  Run the following command to start the MongoDB server with the specified port and database path:

  ```
  Mongod – port 27017 –dbpath db_folder
  ```

  Leave this terminal tab/window open to keep the server running.
- Run your Python script for phase-1 with the input file and port number as input:

  Open a new terminal tab/window and run your python script with the port number

  ```
  Python3 load-json.py 10000.json 27017
  ```

  ```
  Enter the Batch No: <Enter a number>
  ```

  Ensure your Python script correctly handles the port number as a command-line argument using sys.srgv.
- verify the results for phase-1:

  Output would be:

  ```
  Data loading complete. Total documents inserted: xx
  ```

  Observe the output of your script. Perform the required operations and verify that MongoDB is being accessed correctly.
- Run your Python script for phase-2 with port number as input:

  ```
  Python3 phase2_query.py 27017
  ```

  Ensure your Python script correctly handles the port number as a command-line argument using sys.srgv.

- verify the results for phase-2:
  - Output would be the menu:

    ```
    Connected to the database...
    ```

Menu:
Get average rating of a product
Find top N highest-rated products
List most active reviewers
Reviews over time
Flag suspicious reviews
Exit
Enter choice: 1/2/3/4/5/6

Observe the output of your script. Perform the required operations by traversing the friendly user-friendly menu and verify that MongoDB is being accessed correctly.

- Terminate the MongoDB server:

  Once you're done, stop the MongoDB server by pressing Ctrl +C in the terminal window where it is running.

III. <u>Detailed functionality:</u>

a. <u>Phase 1 (load-json.py):</u> The Python script contains the following function:

- <u>load_data_to_mongo(json_file, port, verbose):</u> Loads Amazon reviews from a JSON file into a MongoDB collection.
  - Connects to a MongoDB server using the specified port.
  - Creates or resets the reviews collection in the 291db database.
  - Reads and parses the JSON file line by line.
  - Inserts reviews into MongoDB in batches of the specified size.
  - Handles JSON decoding errors and skips invalid entries.
  - Reports the number of successfully inserted documents.
- main():
  - Calls the appropriate function based on user input.
- Catches and handles errors related to database connection, file access, and unexpected exceptions gracefully.

The script also includes an if __name__ == "__main__": block to handle command-line argument parsing and execute load_data_to_mongo() when run as a script.

b. <u>Phase 2 (phase2_query.py):</u> The provided Python script contains the following functions:

- connect_to_db(port)

  Establishes a connection to a MongoDB instance running on the specified port.
  Returns the database object for 291db.
- get_average_rating(db, asin)

  Retrieves and calculates the average rating of a product given its ASIN (Amazon Standard Identification Number).
  Prints the average rating or an error message if no reviews are found.
- get_top_n_products(db, n)

  Finds the top n highest-rated products based on their average rating.
  Prints the ASIN and average rating of each product.
- get_most_active_reviewers(db)

  Identifies the top 10 reviewers who have written the most reviews.
  Prints the reviewer ID and the number of reviews they have written.
- reviews_over_time(db, asin, years)

  Retrieves the number of reviews for a given ASIN over specified years.
  Prints the number of reviews per year or a message if no reviews are found.

- flag_suspicious_reviews(db)
  Identifies suspicious reviews based on a low helpfulness ratio (less than 10%) and a high rating (≥4.5).
  Prints details of flagged reviews or a message if none are found. (upto 10 / or specified)
- main():
  - Handles user interaction through a menu-driven interface.
  - Calls the appropriate function based on user input.
  - Provides options to get product ratings, find top-rated products, list active reviewers, check reviews over time, and flag suspicious reviews.
- Catches and handles errors related to database connection, file access, and unexpected exceptions gracefully.
  The script is executed via the if __name__ == "__main__": block, which runs main().

IV. Group work break-down strategy:

Work Module	Assigned Member	Function involved	Time Spent	Progress Made	Coordination
Project Setup/Github loading and commits	Vikasinisenthil/ Saachi07	NA	-	Complete	Google Shared document. Chats, emails, zoom meetings, Communication, Github, online meetings
UI Development and planning	All	NA	1 day	Complete	
Functionality/ Database connectivity & loading data	Gskakar	Phase -1: load_data_to_mongo (json_file, port, verbose):	2 -3 – Ave hs/wk	Complete	

Functionality Phase 2	Saachi07	connect_to_db(port)
		get_average_rating(db, asin)
			2 -3 – Ave hs/wk	Complete	

Functionality Phase 2	vikasinisenthil	get_top_n_products(db, n)
		get_most_active_reviewers(db)
			2 -3 – Ave hs/wk	Complete	

Functionality Phase 2	eshaankrishna	reviews_over_time(db, asin, years)
		flag_suspicious_reviews(db)
			2 -3 – Ave hs/wk	Complete	

Reports	All	NA	Ad Hoc	Complete	
Refinement	All	NA	Ad Hoc	Complete	
Testing and debugging	All	NA	Ad Hoc	Complete	

V. **Any documentation on Project specification:** To the best of our ability, we have strictly adhered to the project specifications. Our design document reflects this by avoiding any additional coding beyond what was required.