**Project Phase 4 (Ege Tan,Kutluhan Berke Kılıçkaya,Efe Sözer) :**

>In this project, we developed a Python script that provides a command-line interface for managing MongoDB collections. The script is designed to perform CRUD (Create, Read, Update, Delete) operations on documents within specified collections. It is built using the PyMongo library to interact with MongoDB, ensuring seamless database operations.

The script starts by establishing a connection to the MongoDB database using the connectDB function. This function is imported from an external file named connect.py, which contains the connection logic. Once the connection is established, the script provides a menu-driven interface for the user to select various operations.

**1)Create Collection:** The create\_collection function checks if the specified collection exists in the database. If it doesn't, the function creates the new collection. This prevents the creation of duplicate collections, maintaining database integrity.

**2)Insert Data:** The insert\_data function allows the user to insert a new document into a specified collection. Before inserting the data, it checks if a document with the same id already exists in the collection. If a duplicate id is found, the insertion is aborted, and a message is displayed, ensuring that each document has a unique id.

**3)Read All Data:** The read\_all\_data function retrieves and prints all documents from the specified collection. This function helps in displaying the entire dataset for review or analysis.

**4)Read Filtered Data:** The read\_filtered\_data function allows the user to filter documents based on the given\_star field. It retrieves and prints documents where the given\_star value is greater than a specified value. This feature is useful for querying specific subsets of data based on certain criteria.

**5)Update Data:** The update\_data function updates a document based on its id. Before performing the update, it checks if a document with the specified id exists. If found, it updates the specified fields; otherwise, it notifies the user that the document does not exist. This ensures that updates are only performed on existing documents.

**6)Delete Data:** The delete\_data function deletes a document based on its id. Similar to the update function, it checks if the document exists before attempting to delete it. This prevents accidental deletions and maintains data integrity.

**7)List Collections:** The list\_collections function lists all the collections in the database. This function is used in various parts of the script to allow the user to select a collection for performing different operations.

The script uses a while loop to display the menu continuously until the user chooses to exit. Each operation prompts the user for necessary inputs, such as the collection name, id, and other relevant fields. This interactive approach makes the script user-friendly and easy to navigate.

Initial Screenshot:

A screenshot of a computer

Description automatically generated

Create Collection(student\_evaluation):

A screenshot of a computer

Description automatically generated

Create Collection(teacher\_evaluation):

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedInsert Data(student\_evaluation):

Insert Data(teacher\_evaluation):

A screenshot of a computer

Description automatically generated

Read all Data(student\_evaluation):

A screenshot of a computer

Description automatically generated

Read all Data(teacher\_evaluation):

A screenshot of a computer

Description automatically generated

Update Data(student\_evaluation)🡪 (Update Ahmet Ekrem Rüzgar):

A screenshot of a computer

Description automatically generated

Update Data(teacher\_evaluation) 🡪 (Update Albert Levi):

A screenshot of a computer

Description automatically generated

Delete Data(student\_evaluation) 🡪 (Delete Ahmet Emre):

A screenshot of a computer

Description automatically generated

Delete Data(teacher\_evaluation) 🡪 (Delete Kamer Kaya):

A screenshot of a computer

Description automatically generated

Read some part of the data(student\_evaluation)🡪 (filter , given\_stars greater than input):

A screenshot of a computer

Description automatically generated

Read some part of the data(teacher\_evaluation)🡪 (filter , given\_stars greater than input):

A screenshot of a computer

Description automatically generated