User Manual for StudentVLE MongoDB Application

1. Introduction

Welcome to the **StudentVLE MongoDB Application**! This application allows you to upload student-related data (such as courses, assessments, and student info) into MongoDB, clean up duplicate records, and create indexes for faster queries. The application also provides a graphical user interface (GUI) built with **Streamlit**, which you can use to interact with the MongoDB data.

This manual will guide you through the steps required to set up, run, and use the application on **macOS** and **Windows** systems, including how to map the project folder to the **Documents** directory.

---

2. Requirements

Before you can run the application, ensure that the following requirements are met:

1. **Python**: Version 3.7 or higher. Ensure that its added to environment variables.
2. **MongoDB**: The MongoDB server should be running locally or remotely.
3. **Libraries**: All required Python libraries should be installed, which are specified in the `requirements.txt` file.

---

3. Installing MongoDB

3.1. Install MongoDB Locally (macOS)

Install MongoDB Using Homebrew
Install Homebrew: If Homebrew is not installed, open Terminal and run:
bash:
```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Add MongoDB Tap: After installing Homebrew, tap the MongoDB repository:
bash:
```
brew tap mongodb/brew
```

Install MongoDB: Install the MongoDB Community Edition:
bash:
```
brew install mongodb-community@6.0
```

Start MongoDB: Start the MongoDB service using Homebrew:
bash:
```
brew services start mongodb/brew/mongodb-community@6.0
```

Verify MongoDB is Running: Check the status of MongoDB:
bash:
```
brew services list
```

3.2. Install MongoDB Locally (Windows)

Install MongoDB on Windows

1. **Download MongoDB**:
    - Go to the MongoDB Downloads page.
    - Select **Windows** as your OS and download the installer.

2. **Install MongoDB**:
    - Run the installer and follow the installation prompts.
    - Make sure to check the box for **"Install MongoDB as a Service"** so MongoDB starts automatically after installation.

3. **Start MongoDB**:

MongoDB will be installed as a service, so it should automatically start. If it doesn't, open Command Prompt (with Administrator privileges) and run:
bash:
```
net start MongoDB
```

4. **Verify MongoDB**:

Open Command Prompt and run:
bash:
```
mongosh
```

If you enter the MongoDB shell (`test>`), MongoDB is successfully installed.

3.3. MongoDB URI (Skip if unnecessary)

MongoDB URI: The MongoDB connection URI is specified in the `.env` file. If you're running MongoDB locally, the URI is:
arduino:
```
mongodb://localhost:27017/
```

Create a Database: The application uses a database named `StudentVLE`. If it's not created automatically, you can create it manually by connecting to MongoDB through the MongoDB shell (`mongosh`) and running:
bash:
```
use StudentVLE
```

3.4. MongoDB Installation Verification

Check MongoDB Version: After installation, verify the MongoDB version by running:
bash:
```
mongod --version
```

Run MongoDB Shell: To connect to the MongoDB instance and interact with the database, use:
bash:
```
mongosh
```

If you enter the shell (`test>`), MongoDB is successfully installed and running.

---

4. Setting Up the Python Project

4.1. Installing Dependencies

1. **Clone or Download the Project**:
    - Extract the ZIP file you received, or clone the repository containing the project. For our demonstration, save the “MongoDB_Project” in the documents directory.

2. **Install Python and MongoDB**:
    - Ensure **Python** (version 3.7 or higher) is installed on your system.
    - **MongoDB** should be running locally, or the `.env` file should be correctly configured for a remote instance.

3. **Install Project Dependencies**: Navigate to the project folder in the terminal/command prompt and run the following command to install the required

Python libraries:
bash:
```
pip install -r requirements.txt
```

This will install all the necessary dependencies like `pandas`, `pymongo`, `streamlit`, etc.

4.2. Set Up the `.env` File

1. **Environment Variables**: Ensure that the `.env` file is included with the project. This file contains sensitive information, such as MongoDB credentials.
2. **Modify `.env`**: The `.env` file should contain the MongoDB connection URI template, as shown below:
   Env:
   ```
   MONGO_URI_TEMPLATE=mongodb+srv://{username}:{password}@inf2003.isemd.mongodb.net/?retryWrites=true&w=majority&appName=INF2003
   ```

---

5. Mapping the Project to the Documents Directory

5.1. macOS Users:

1. **Move the Project to the Documents Folder**:
   - If you want to store the entire project in the **Documents** folder, move the entire project folder (containing all your scripts) to the **Documents** directory:
     bash:
     ```
     mv /path/to/your/project ~/Documents/
     ```
2. **Update File Paths in the `upload_data.py` Script**: If your CSV files are stored within the `Documents` directory, make sure to reference the correct file paths in your script. In the project file, the CSV files are already uploaded to the “StudentVle” Folder.
   For example, update the `csv_files` dictionary like so:
   Python:
   ```
   csv_files = {"courses":"/Users/your-username/Documents/StudentVLE/studentVle/courses.csv","studentInfo":"/Users/your-username/Documents/StudentVLE/studentVle/studentInfo.csv","studentRegistration":"/Users/your-username/Documents/StudentVLE/studentVle/studentRegistration.csv","assessments":"/Users/your-username/Documents/StudentVLE/studentVle/assessments.csv","studentAssessment":
   ```

`"/Users/your-username/Documents/StudentVLE/studentVle/studentAssessment.csv",}`

Replace `your-username` with your actual macOS username.

5.2. Windows Users:

1. **Move the Project to the Documents Folder**:
    - If you want to store the entire project in the **Documents** folder, move the entire project folder (containing all your scripts) to the **Documents** directory:
        - Open **File Explorer** and move the folder to `C:/Users/your-username/Documents`

2. **Update File Paths in the `upload_data.py` Script**: For **Windows**, update the `csv_files` dictionary in your `upload_data.py` to point to the correct path in the **Documents** directory. In the project file, the CSV files are already uploaded to the "StudentVle" Folder.:
   python:
   ```
   csv_files = {
   ```

```
    "courses":
"C:/Users/your-username/Documents/StudentVLE/studentVle/courses.csv"
,
    "studentInfo":
"C:/Users/your-username/Documents/StudentVLE/studentVle/studentInfo.
csv",
    "studentRegistration":
"C:/Users/your-username/Documents/StudentVLE/studentVle/studentRegis
tration.csv",
    "assessments":
"C:/Users/your-username/Documents/StudentVLE/studentVle/assessments.
csv",
    "studentAssessment":
"C:/Users/your-username/Documents/StudentVLE/studentVle/studentAsses
sment.csv",
}
```

Replace `your-username` with your actual Windows username.

---

6. Running the Application

6.1. Upload Data to MongoDB (Optional)

The MongoDB database is already uploaded with the necessary csv files, so it is unnecessary to run this code unless needed.

6.1.1. Modify the File Paths for CSV Files in the Documents Directory

If you haven't done so already, update the paths for the CSV files in the `upload_data.py` script to reflect their location in the **Documents** directory.

6.1.2. Run the Data Upload Script

1. **Run the Data Upload Script**:
    - Open a terminal (macOS) or Command Prompt (Windows) and navigate to the project directory.
    - Run the following command to upload the CSV data into MongoDB:
      bash:
      `python upload_data.py`

      The script will prompt if you want to upload the data. Type `yes` to begin the upload process. In our case, type `no`.

   Note: This will load the data into the `StudentVLE` database and automatically remove any duplicate records.

6.2. Launch the GUI (Streamlit Application)

1. **Launch the Streamlit GUI**:
    - Open a terminal (macOS) or Command Prompt (Windows) and navigate to the directory where `app.py` is located.
    - Run the following command to start the Streamlit GUI:
      bash:
      `streamlit run app.py`
    - The application will start and open automatically in your default web browser at `http://localhost:8501`.

---

7. Using the Streamlit GUI

7.1. Login to MongoDB

- Upon launching the GUI, you'll be prompted to enter your **MongoDB Username** and **Password**. For demonstration of the project,
  For **Read-Write** Access:
  Username: userRW1

  Password: readwritedb

  For **Read Only** Access:
  Username: userREAD2

  Password: readonlydb

- After entering the credentials, the application will authenticate and allow you to access the data. (NOTICE: Click twice on login!)

7.2. Menu Options

After logging in, you can select from the following menu options:

1. **Insert Student**: Insert new student records into the database.
2. **Find Student**: Search for an existing student by **Student ID**.
3. **Update Student**: Update an existing student's information.
4. **Delete Student**: Delete a student by entering their **Student ID**.
5. **Identify At-Risk Students**: View students who may be at risk based on certain criteria.
6. **Module Difficulty Comparison**: Compare the difficulty of modules based on student assessments.
7. **Student Performance Clustering**: Perform clustering on student performance data.
8. **Module Success Prediction**: Predict the likelihood of student success in a module.

7.3. Logout

- To log out of the application, click the **Logout** button in the sidebar.(NOTICE: Click twice on logout!)

---

8. Troubleshooting

If you encounter any issues, here are some common troubleshooting steps:

1. **MongoDB Connection Issues**:
   - Ensure MongoDB is running and accessible. If using a local MongoDB instance, verify it is running on `localhost:27017`.
2. **File Not Found**:
   - Ensure that the **Documents** directory path in the `upload_data.py` script is correctly set.
   - Double-check that the CSV files are located in the correct folder.

---

9. Conclusion

You have successfully set up the **StudentVLE MongoDB Application**! You can now upload, view, and manage student data easily through the GUI. If you encounter any issues or need further assistance, feel free to reach out.