Step 1: Set up the Visual Studio Windows Console Project

Open Visual Studio and create a new Windows Console Project.

Name the project and select a location for it.

Once the project is created, you can start working on the implementation.

Step 2: Create the student data text file

Create a simple text file externally using Notepad or any text editor.

The file should follow the specified data format: "Name, Class" on each line.

Add some sample student data to the file, and save it with a meaningful name (e.g., "students.txt").

Step 3: Read data from the text file

Implement a function to read data from the "students.txt" file and load it into memory.

Parse each line of the file and extract the name and class of each student.

Store the student data in a data structure like a list or dictionary.

Step 4: Sort the student data by name

Implement a sorting algorithm (e.g., bubble sort, merge sort, quicksort) to sort the student data by name.

Apply the sorting algorithm to the loaded student data.

Step 5: Allow searching students by name

Implement a function that prompts the user to enter a name to search for.

Iterate through the student data to find a match for the entered name.

Display the details of the student if found, or show a message if the name is not present.

Step 6: Display the sorted student data

Implement a function to display the sorted student data on the console.

Display the student names and their corresponding classes in a formatted manner.

Step 7: Version Control and GitHub

Initialize a new Git repository for your project.

Track and commit your source code regularly as you progress through the steps.

Push your code to a GitHub repository and make sure it is accessible.

Step 8: Documentation

Document each step involved in completing the task, including explanations of the functions and algorithms used.

Include instructions on how to set up and run the program.

Mention the files that are ignored in the Git repository and the reasons for excluding them (e.g., sensitive data, build artifacts).

Provide any additional information or considerations that might be helpful for users or future developers.

Step 9: Testing

Test your program with different scenarios, including edge cases, to ensure it works as expected.

Check if the sorting and searching functionalities are functioning correctly.

Remember to keep your code clean, well-commented, and organized. Also, consider error handling and user-friendly messages when dealing with user input or file operations.

Step 10: GitHub Repository Link

https://github.com/rastogi102/Project2-RetrieveStudentData.git