**Project Folder Structure and Collaboration Guidelines**

**Overview**

To ensure consistency for everyone working on this project, all team members must use the same folder structure in their VS Code workspaces and when pushing/pulling from GitHub. This guide describes how to set up your repository so that the project folder appears inside an outer folder exactly as it does on your local VS Code setup.

**Folder Structure Example**

Your project in VS Code should be organized as follows:

MyWorkspace/ <-- Outer folder (your workspace folder)

└── COSC471FinalProject/ <-- GitHub repository folder

├── BinarySearchTree.java

├── CommandParser.java

├── Database.java

├── DBMS.java

├── DBMSApp.java

├── FileManager.java

├── README.md

├── Table.java

└── dbms\_state.ser

In this example, **COSC471FinalProject** is the repository folder that you push to/pull from GitHub. It sits inside an outer folder (here called **MyWorkspace**), which you may use as your VS Code workspace. All team members should ensure that their local workspace mimics this structure.

**Step-by-Step Instructions**

**1. Create a Workspace Folder on Your Local Machine**

* On your computer, create a folder called MyWorkspace (or another agreed-upon name). This will be your outer folder and VS Code workspace.

**2. Clone the GitHub Repository into the Workspace Folder**

* Open Visual Studio Code.
* Use **File > Open Folder...** and select your workspace folder (MyWorkspace).
* Open the Command Palette (press **Ctrl+Shift+P** on Windows or **Cmd+Shift+P** on macOS).
* Type **Git: Clone** and press Enter.
* When prompted, paste the GitHub repository URL (for example, https://github.com/YourUsername/COSC471FinalProject.git).
* Choose your workspace folder (MyWorkspace) as the destination.
* VS Code will create a folder named **COSC471FinalProject** inside your workspace folder. This folder is your project’s repository.

**3. Open the Repository in VS Code**

* Once the repository is cloned, open the folder **COSC471FinalProject** as your project.
* Your VS Code Explorer should display the entire structure exactly as it appears on GitHub:

COSC471FinalProject/

├── BinarySearchTree.java

├── CommandParser.java

├── Database.java

├── DBMS.java

├── DBMSApp.java

├── FileManager.java

├── README.md

├── Table.java

└── dbms\_state.ser

**4. Keeping the Structure Consistent**

* **Always work within the COSC471FinalProject folder:**  
  Do not move or rename files and folders within this directory unless everyone agrees.
* **Every team member should use VS Code and clone the repository into the same parent (workspace) folder structure.**
* **When pushing and pulling changes:**
  1. Open VS Code in the workspace folder.
  2. Confirm that the repository folder (COSC471FinalProject) is present within your workspace.
  3. Use the Source Control panel in VS Code to pull changes from GitHub before you start working.
  4. After changes, commit and push from within the same folder structure so that the GitHub repository always reflects the same internal hierarchy.

**5. Synchronizing Changes**

* **Before making any changes:**  
  Always perform a **Pull** from GitHub so your local copy is up to date.
* **After making changes:**  
  Stage, commit, and push your changes from the **COSC471FinalProject** folder.
* **Verify on GitHub:**  
  After pushing, check your GitHub repository online. The folder structure should be:

COSC471FinalProject/

├── [all project files…]

This confirms that everyone is using the same structure.

**6. Resolving Folder or Merge Conflicts**

* If conflicts arise because files or folders have been moved or renamed, coordinate with your team to manually resolve these issues.
* **Do not change the overall structure** (i.e., keep the repository inside the workspace folder with the same directory names).

**Summary Checklist**

1. **Workspace Setup:**  
   Create or use a common workspace folder (e.g., MyWorkspace).
2. **Clone Repository:**  
   Clone the GitHub repository into this workspace so the repository folder (e.g., COSC471FinalProject) is inside your workspace folder.
3. **Work Consistently:**
   * Open the project from within the workspace folder in VS Code.
   * Always work from the repository folder as it appears in the workspace.
   * Do not change the folder structure.
4. **Pull, Commit, and Push:**
   * Pull before you start working each day.
   * Commit and push changes from the repository folder to keep the structure synchronized.
5. **Verify:**  
   Check the GitHub repository online to ensure the project directory structure remains as expected.

By following these instructions, all team members will ensure the project directory in GitHub matches the local structure in VS Code, avoiding inconsistencies and merge conflicts due to differences in folder hierarchy.