Class Revision Notes on Git, Conflict Resolution, and Streamlit

Introduction

This class primarily covered concepts related to Git, conflict resolution, and the use of Streamlit for developing applications. The class included both theoretical explanations and practical examples to ensure a comprehensive understanding of these topics.

Git and Version Control

Git Basics

- **Git** is a version control system that allows developers to track changes in their code and collaborate with other developers.
- **Repository**: Think of this as a project folder which can be on your local machine (Local repository) or hosted online (Remote repository, e.g., on GitHub).

Common Git Commands

- **git init**: Initializes a new git repository.
- **git status**: Shows the status of changes as untracked, modified, etc.
- **git add .**: Stages all changes in the current directory for the next commit.
- **git commit -m "message"**: Saves your changes with a descriptive message.
- **git push**: Uploads your changes to a remote repository.
- **git pull**: Fetch and merge changes from the remote repository to your local one【7:16†transcript.txt】.

Branching and Merging

- **Branches** allow you to work on different versions of a project simultaneously. For example, making experimental changes in a different

【7:15†transcript.txt】.

- **Conflict Resolution**: Occurs when Git is unable to automatically resolve differences in code changes between branches. This requires manual intervention to resolve and merge the changes properly 【7:4†transcript.txt】.

GitHub Operations

- **Forking**: Creating your own copy of someone else's project repository.
- **Pull Requests**: Proposing changes from your forked repository back into the original repository【7:10†transcript.txt】.

Conflict Resolution with Real-World Analogy

The class used a cooking analogy to explain conflicts in Git:

1. Imagine a banana cake recipe is your master branch.
2. If one person adds raisins (a new feature) and another adds cashews (another feature) to their copies of the recipe, both changes need to be combined in the master recipe without overwriting each other.
3. This situation represents a conflict that must be resolved before both sets of changes can be merged into the master recipe【7:0†transcript.txt】.

Streamlit for Rapid Application Development

What is Streamlit?

Streamlit is an open-source app framework for Machine Learning and Data Analysis apps. It allows developers, especially those with limited front-end experience, to create web applications with minimal code【7:5†transcript.txt】.

Key Features

- **Ease of Use**: Allows the quick creation of data apps with only a few lines of Python code.

- **Deployment**: Easily deploy applications and share with end-users 【7:5†transcript.txt】.

Using Streamlit

- Create a virtual environment to manage dependencies.
- Install Streamlit and other required Python libraries using pip.
- Write Python scripts using Streamlit's API for building data-driven applications.
- The application is launched on a local server and can be deployed to cloud platforms【7:9†transcript.txt】.

Conclusion

Understanding Git and Streamlit greatly enhances a developer's ability to efficiently manage version control and rapidly create engaging applications. Through practical examples and conceptual analogies, this class provided a thorough understanding of these modern development tools. Practicing these skills within a version control system like GitHub and using frameworks like Streamlit will strengthen your software engineering capabilities.