

LEARNING ABOUT GITHUB

GitHub is a web-based platform used for version control and collaboration on software development projects. It provides developers with tools for managing source code, tracking changes made to code over time, and facilitating collaboration among team members.



Features Offered By GitHub

GitHub offers a wide range of features that cater to various aspects of software development, collaboration, and project management. Some of the key features include:

1. **Git Repository Hosting:** GitHub provides hosting for Git repositories, allowing users to store, manage, and collaborate on their codebases. Users can create both public and private repositories.
2. **Collaboration Tools:** GitHub offers features such as pull requests to facilitate collaboration among team members. These tools help streamline communication and ensure that code changes are reviewed and discussed before being merged into the main codebase.
3. **Branching and Merging:** Git's powerful branching and merging capabilities are fully supported by GitHub. Users can create branches to work on new features or bug fixes independently and merge them back into the main branch when ready.
4. **Integrated Development Environment (IDE) Integration:** GitHub integrates with popular IDEs such as Visual Studio Code, IntelliJ IDEA, and Atom, providing seamless workflows for developers.

5. Code Hosting and Version Control: GitHub serves as a platform for hosting code and offers robust version control capabilities through Git. Users can track changes, revert to previous versions, and collaborate on code effectively.

6. Security Features: GitHub offers security scanning tools to identify vulnerabilities in code dependencies and configurations. It also provides features like two-factor authentication and access control to enhance the security of repositories.

7. Community and Social Features: GitHub fosters a vibrant community around open source development. Users can discover projects, contribute to open source initiatives, and engage with other developers through discussions and collaboration.

8. Documentation Hosting: GitHub Pages allows users to host static websites directly from their GitHub repositories. This feature is commonly used for project documentation, blogs, and personal websites.

These are just some of the many features offered by GitHub, making it a versatile platform for software development and collaboration.

SOME IMPORTANT GITHUB TERMINOLOGIES WITH EXAMPLES

LET US BEGIN.....



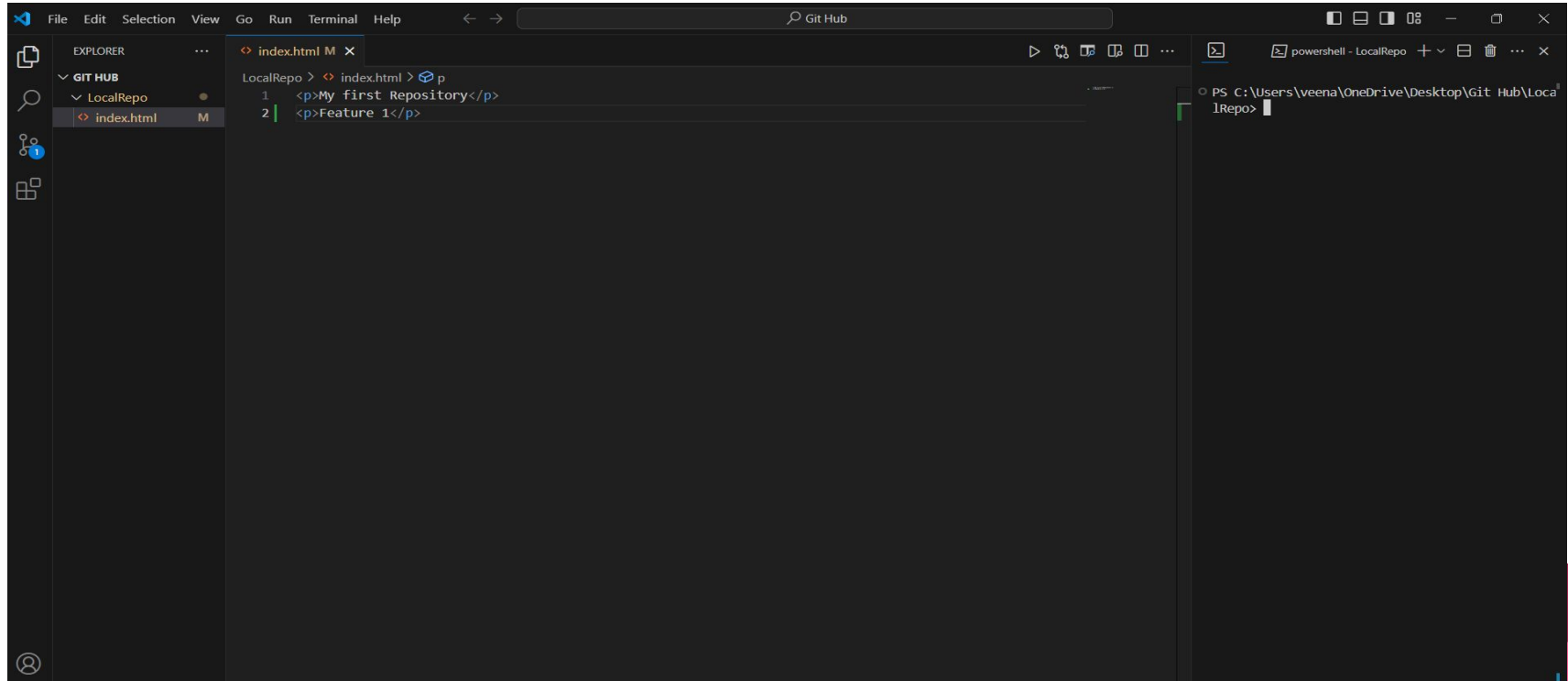
GIT STASH

Git Stash is a command that helps us to move from one branch to the other or one part of the code to the other without committing changes in the current branch/code. This is a very helpful command as it helps people to fix bugs in others' codes, review others' codes or help the other coders without having the pressure of committing the changes before switching to the other branch. This is not possible in case of Git Switch command.

NOTE: git stash list command is used to check for all the current stashes.



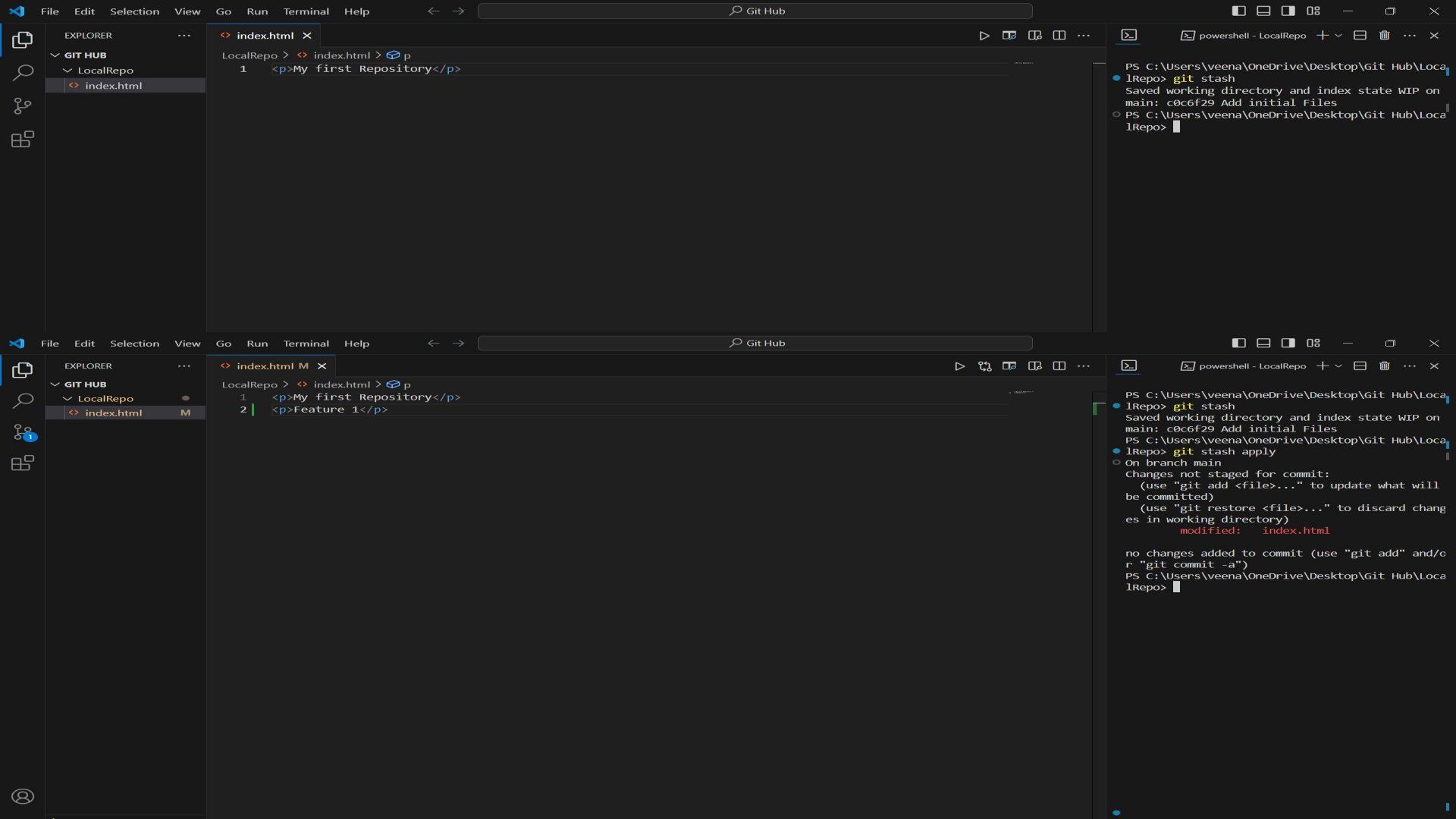
CODE FOR REFERENCE



The screenshot displays the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left shows a project named 'GIT HUB' containing a subfolder 'LocalRepo' and a file 'index.html'. The main editor area shows the content of 'index.html' with two lines of HTML code:

```
1 <p>My first Repository</p>
2 <p>Feature 1</p>
```

The bottom right pane shows a PowerShell terminal window with the current directory set to 'C:\Users\veena\OneDrive\Desktop\Git Hub\LocalRepo'.



GIT BISECT

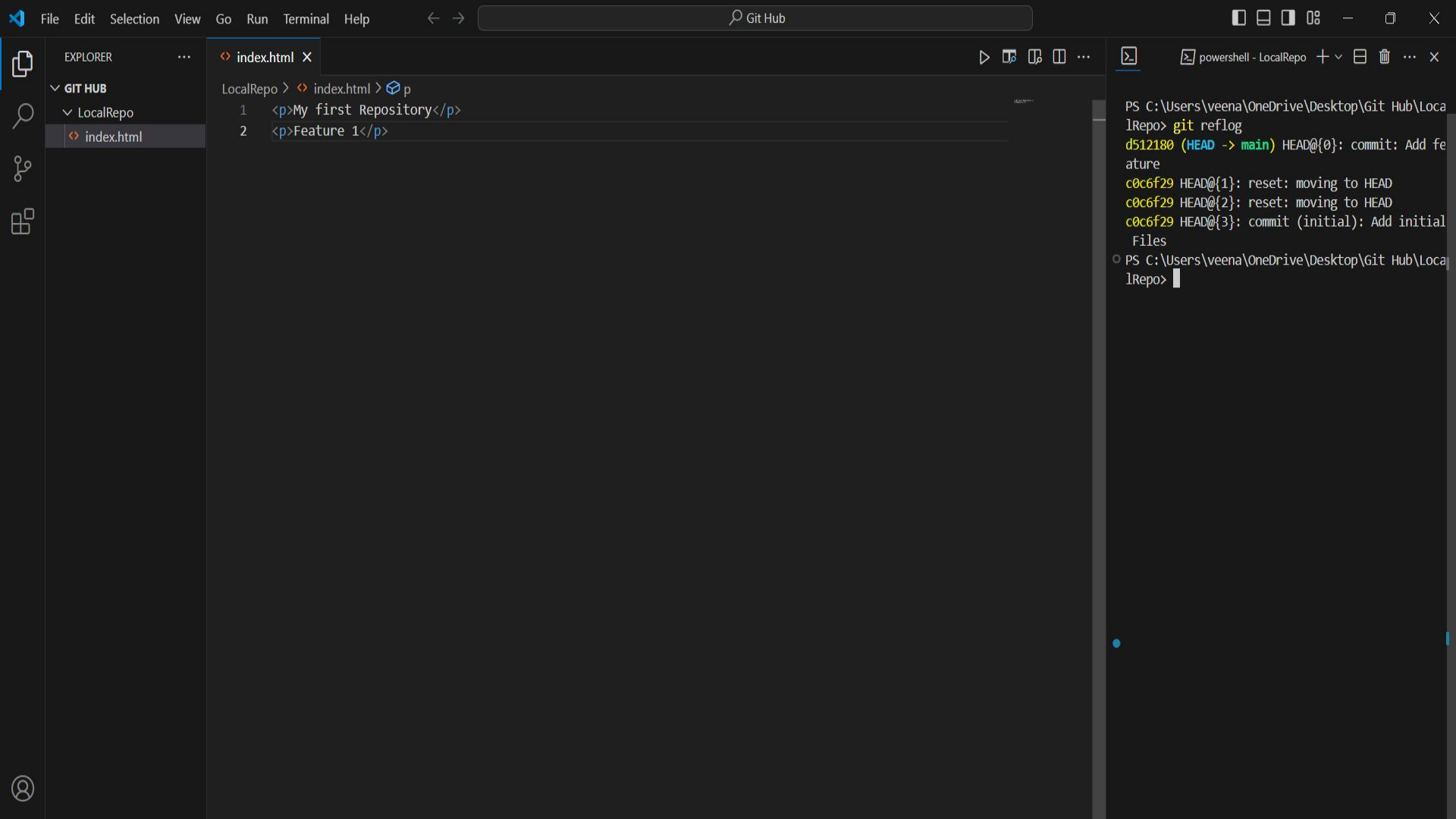
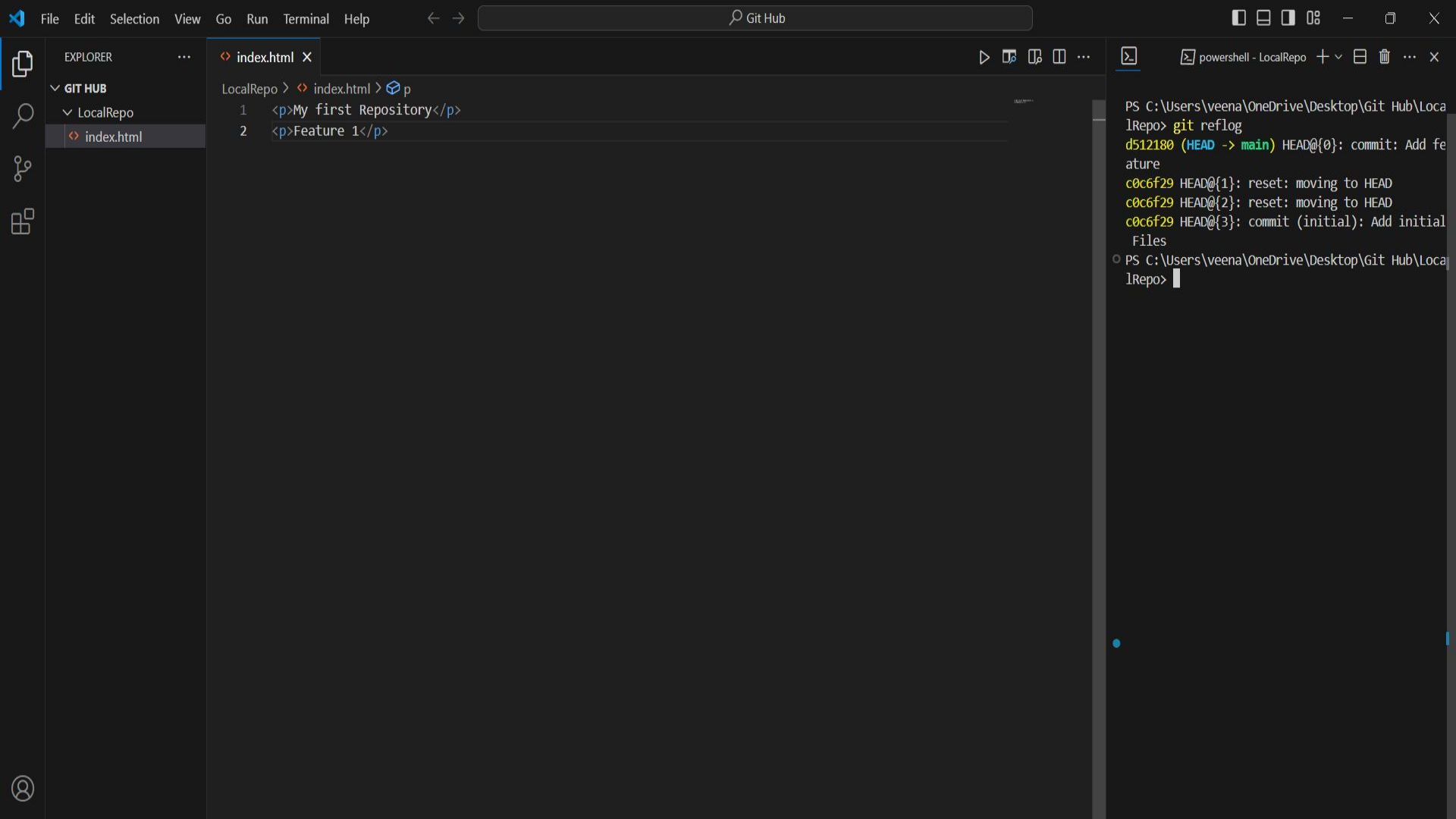
Git Bisect is a command that helps us to identify bugs in our code and hence, is a very useful command. It helps us to find a bad commit in our code. It uses binary search to find out the bad commit. For example, if we are writing a code which we know that was working well when we started but gave a bug till we reached the end then we can start with the git bisect start command. Then mention the current commit as a bad commit by writing the command git bisect bad and write the command git bisect good and the id of the first commit. In this way we will be able to fix the bug in our code.

GIT REFLOG

Git reflog is a command that stands for reference logs. It provides a reference history of actions you have taken in your Git repository, including commits, checkouts, merges, and other operations that affect the state of your repository.

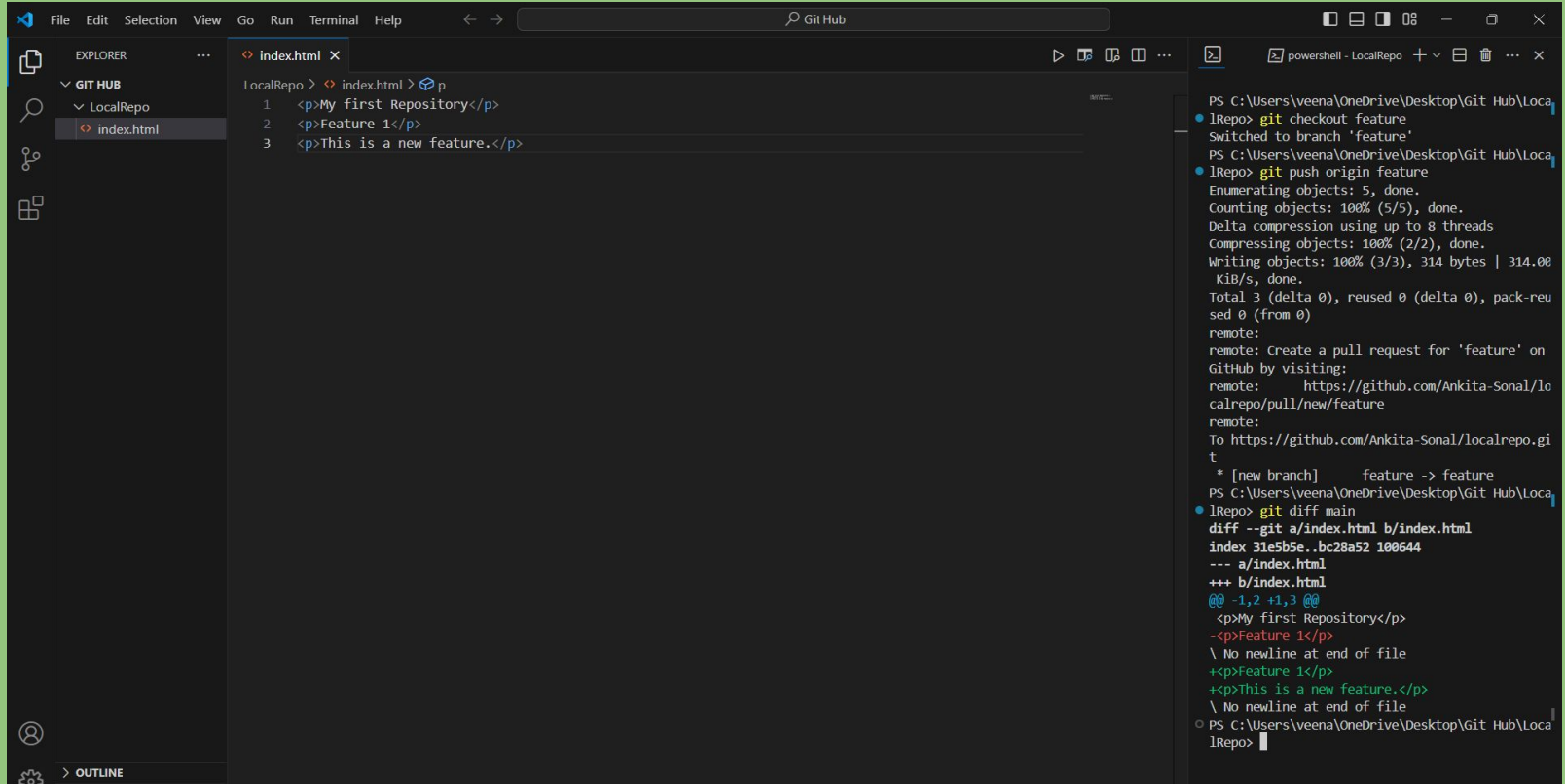
An example of the command comes up in the next page





GIT DIFF

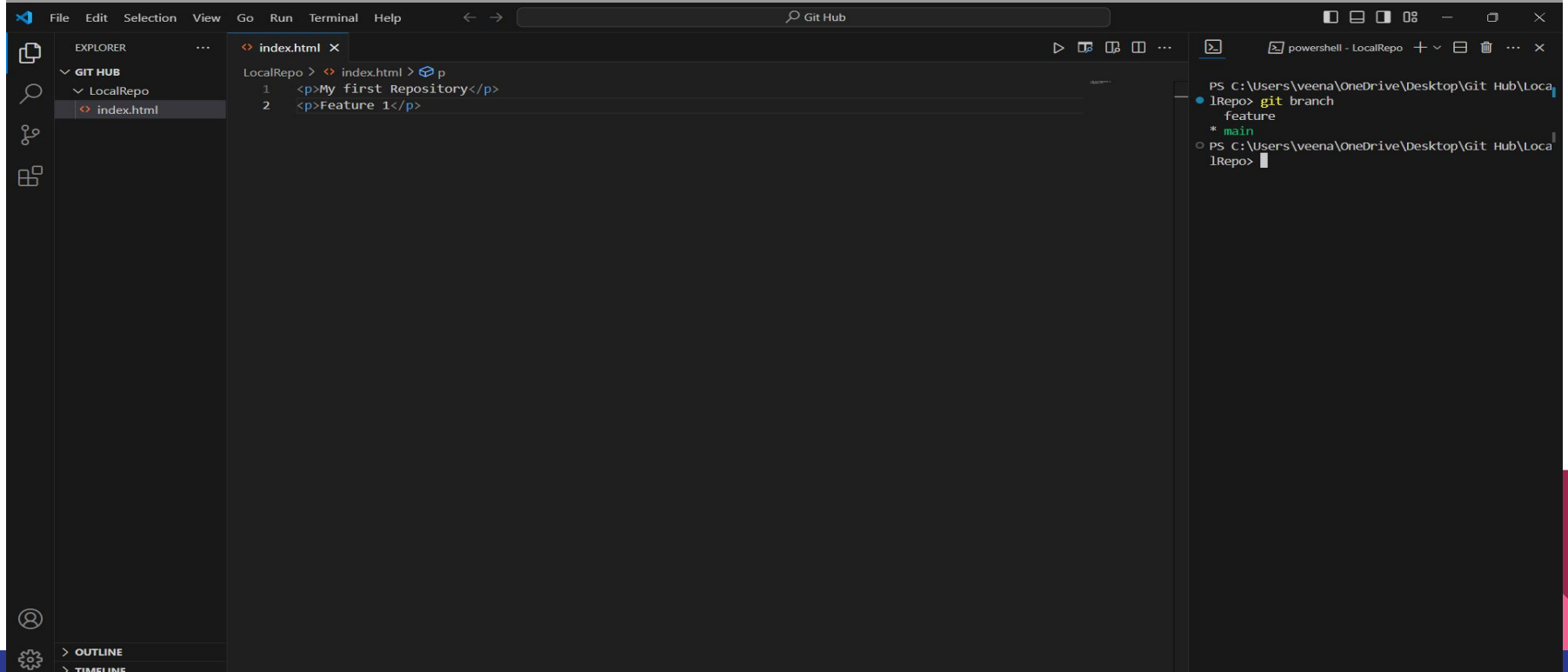
Git diff is a command used in Git to show the difference between changes made to files in the working directory.

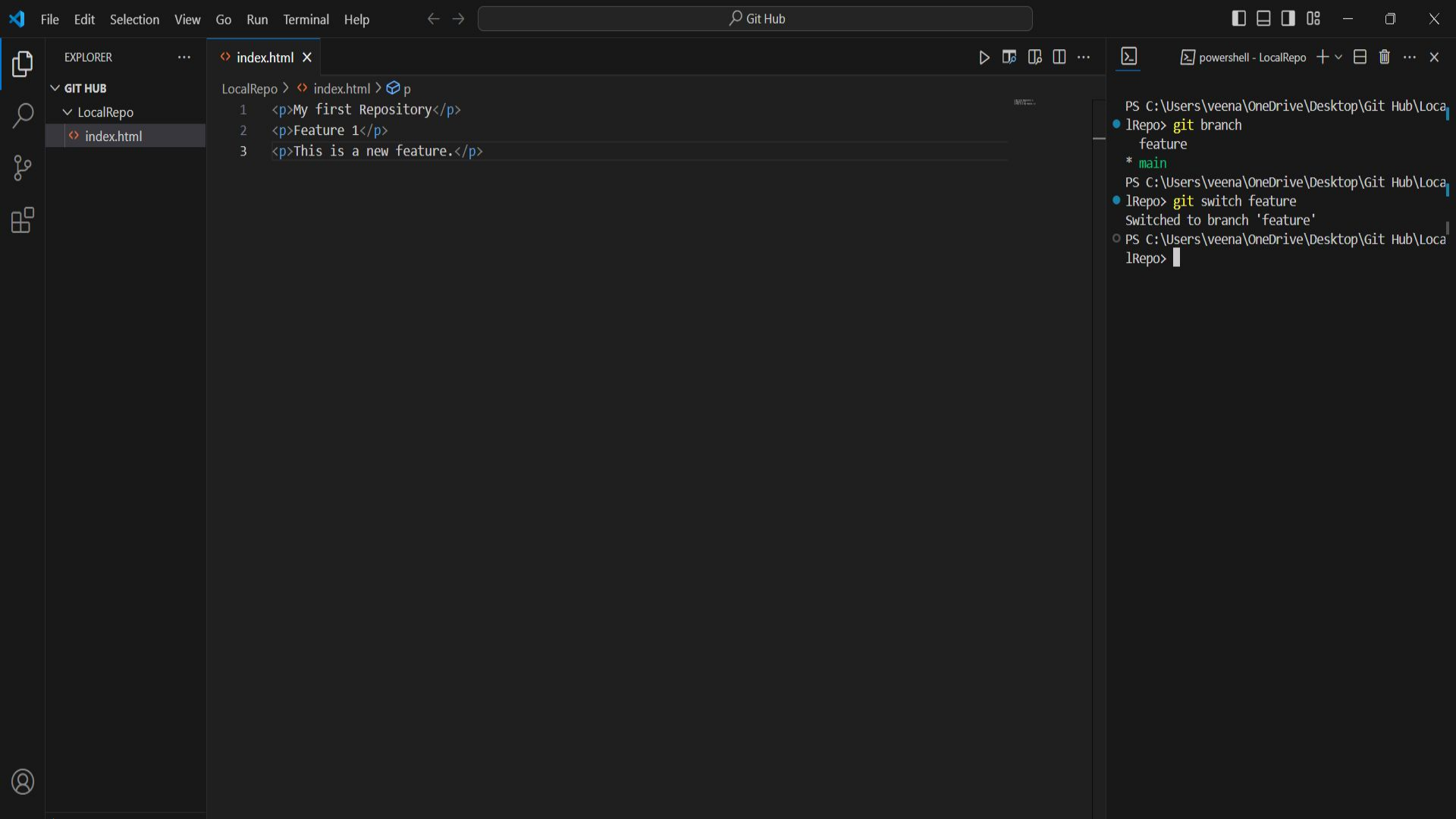
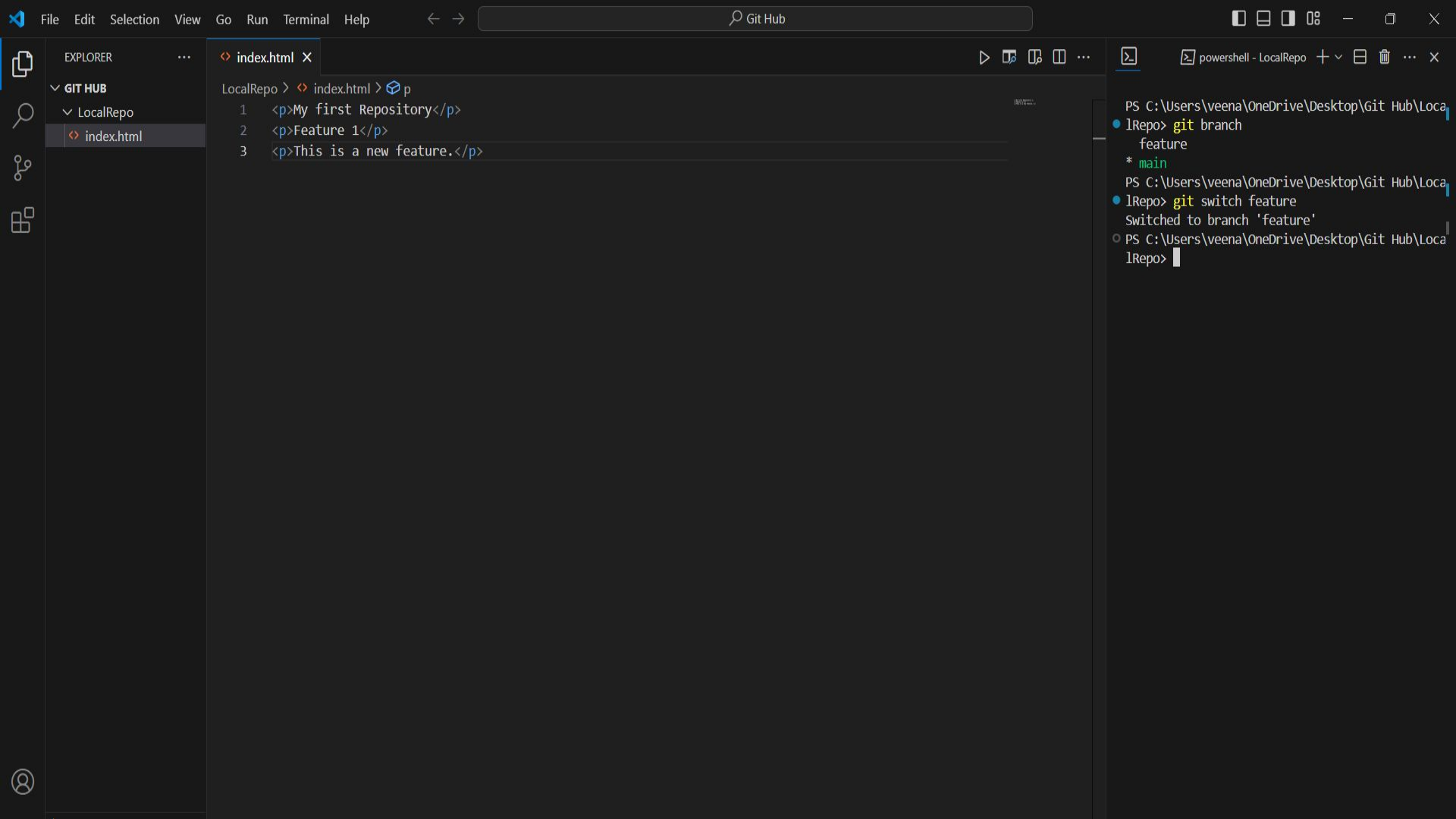


```
File Edit Selection View Go Run Terminal Help
EXPLORER
  GIT HUB
  LocalRepo
    index.html
index.html x
LocalRepo > index.html > p
1 <p>My first Repository</p>
2 <p>Feature 1</p>
3 <p>This is a new feature.</p>
PS C:\Users\veena\OneDrive\Desktop\Git Hub\LocalRepo> git checkout feature
Switched to branch 'feature'
PS C:\Users\veena\OneDrive\Desktop\Git Hub\LocalRepo> git push origin feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 314 bytes | 314.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/Ankita-Sonal/localrepo/pull/new/feature
remote: To https://github.com/Ankita-Sonal/localrepo.git
* [new branch] feature -> feature
PS C:\Users\veena\OneDrive\Desktop\Git Hub\LocalRepo> git diff main
diff --git a/index.html b/index.html
index 31e5b5e..bc28a52 100644
--- a/index.html
+++ b/index.html
@@ -1,2 +1,3 @@
<p>My first Repository</p>
-<p>Feature 1</p>
\ No newline at end of file
+<p>Feature 1</p>
+<p>This is a new feature.</p>
\ No newline at end of file
PS C:\Users\veena\OneDrive\Desktop\Git Hub\LocalRepo>
```

GIT SWITCH

It helps in switching branches.

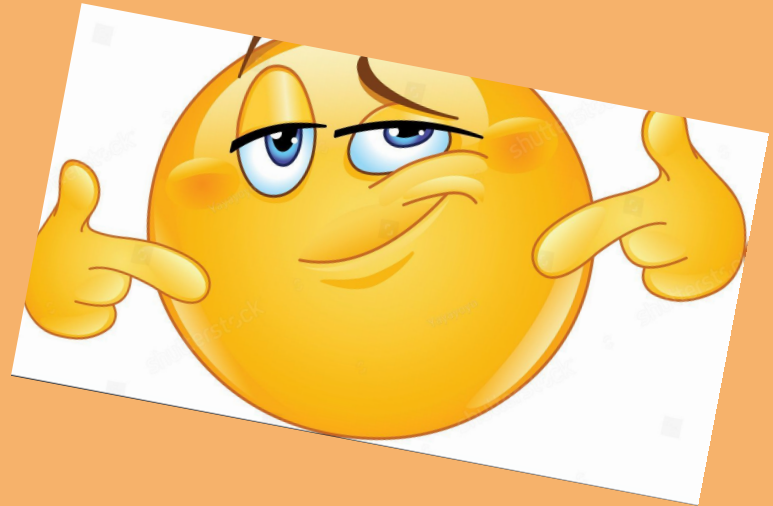


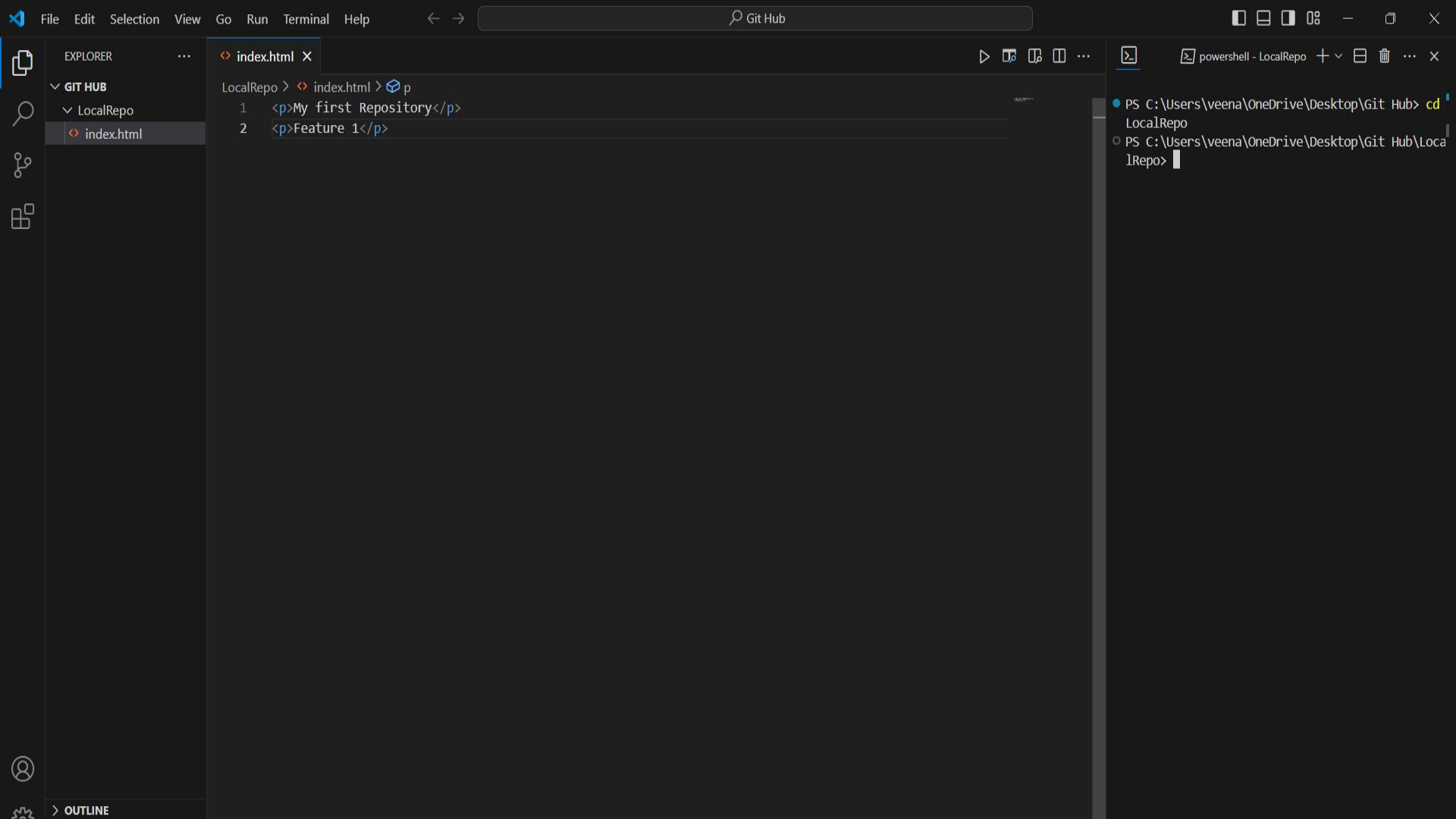


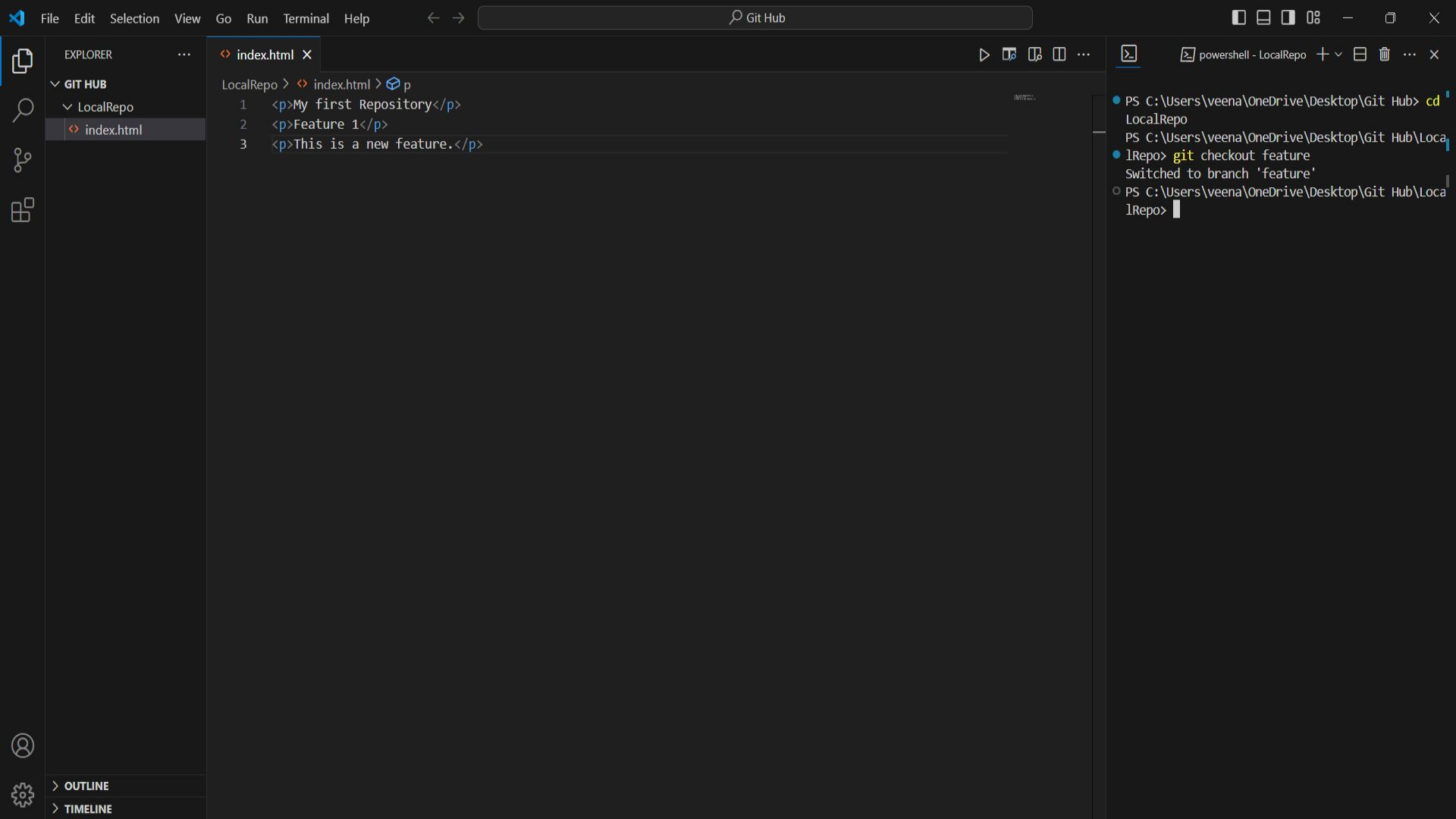
GIT REBASE

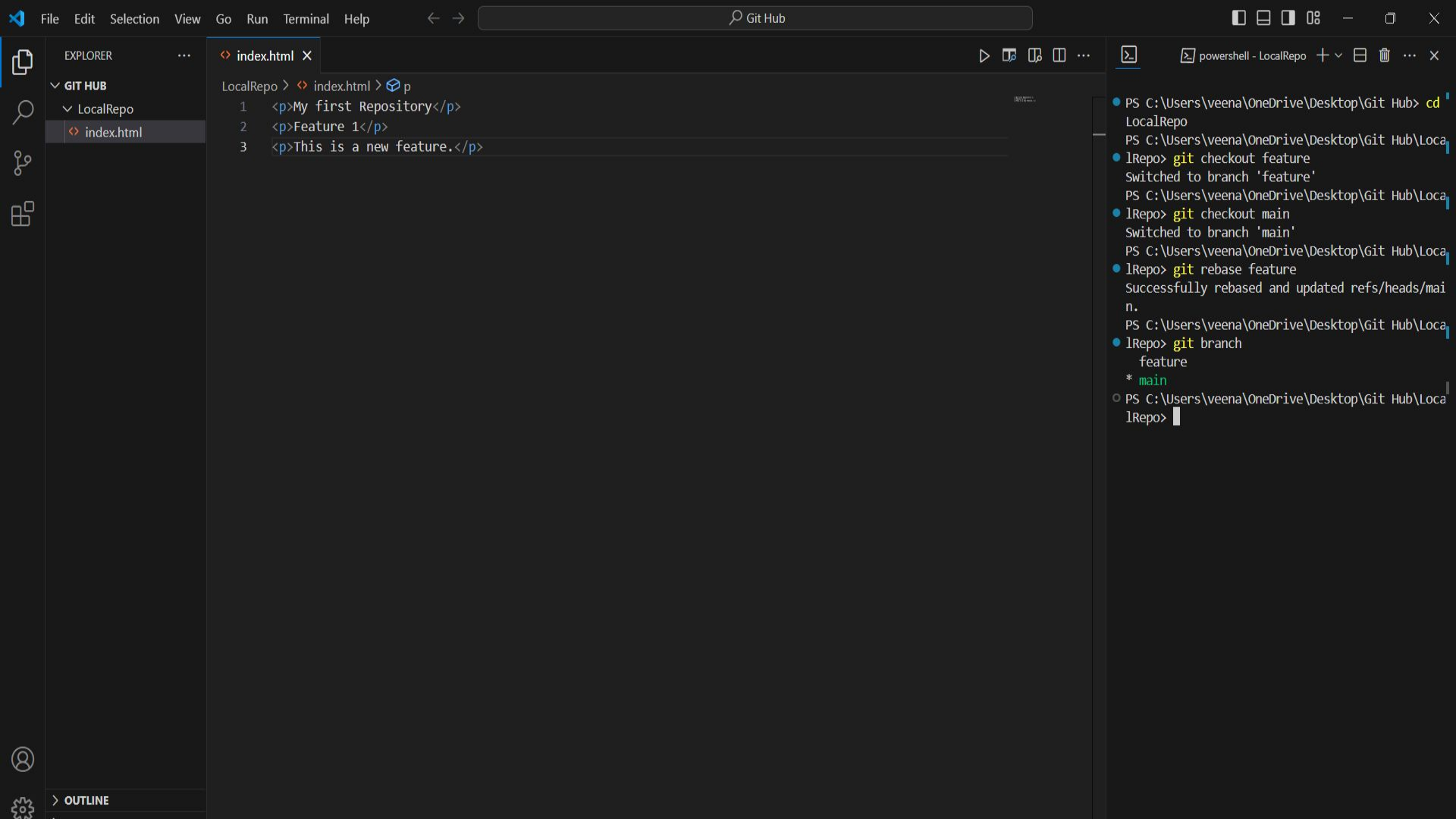
Git rebase is a Git command used to incorporate changes from one branch onto another branch. It allows us to reapply commits on top of another base commit, typically moving or integrating a feature branch onto a main branch like 'master', or 'main'.

LET'S REBASE









GIT CHERRY-PICK

Git cherry-pick is a Git command used to apply a specific commit from one branch to another. It is particularly useful when we want to pick only certain commits from one branch and apply them to another branch without merging entire branch.

AN EXAMPLE FOLLOWS IN THE NEXT PAGE

File

Edit

Selection

View

Go

Run

Terminal

Help

←

→

Git Hub

EXPLORER

...

▼ GIT HUB

▼ LocalRepo

index.html

index.html X

LocalRepo > index.html > p

1 <p>My first Repository</p>

2 <p>Feature 1</p>

3 <p>This is a new feature.</p>

4 <p>Hardwork is the key to success.</p>

5 <p>Honesty is the best policy.</p>

▶

🔍

📄

📁

...

🔍

📄

📁

...

powershell - LocalRepo + - 🗑 ... ✕

PS C:\Users\veena\OneDrive\Desktop\Git Hub\Loca

• lRepo> git add .

PS C:\Users\veena\OneDrive\Desktop\Git Hub\Loca

• lRepo> git commit -m "new changes"

[feature 4f29fa1] new changes

1 file changed, 3 insertions(+), 1 deletion(-)

PS C:\Users\veena\OneDrive\Desktop\Git Hub\Loca

• lRepo> git log

commit 4f29fa18bfc5ffffd26465f02ed4ff393a03bf74f

(HEAD -> feature)

Author: Ankita-Sonal <workharducandoit123@gmail

.com>

Date: Sun Mar 31 00:49:56 2024 +0530

new changes

commit 1f30c62b6cfa4dc32badb19b22bb19ad02b17e0b

(origin/feature, main)

Author: Ankita-Sonal <workharducandoit123@gmail

.com>

Date: Sat Mar 30 22:18:26 2024 +0530

Add this change

commit d5121807d0609bf96005d971c930216ac07eeb2a

(origin/main)

Author: Ankita-Sonal <workharducandoit123@gmail

.com>

Date: Sat Mar 30 20:22:33 2024 +0530

Add feature

commit c0c6f291d8390bfe24672656868d1c7dc5b5dc9d

Author: Ankita-Sonal <workharducandoit123@gmail

.com>

Date: Sat Mar 30 18:15:37 2024 +0530

Add initial Files

PS C:\Users\veena\OneDrive\Desktop\Git Hub\Loca

👤

🔍

📄

📁

...

OUTLINE

File

Edit

Selection

View

Go

Run

Terminal

Help

←

→

Git Hub

EXPLORER

...

▼ GIT HUB

▼ LocalRepo

index.html

index.html X

LocalRepo > index.html > p

1

<p>My first Repository</p>

2

<p>Feature 1</p>

3

<p>This is a new feature.</p>

4

<p>Hardwork is the key to success.</p>

5

<p>Honesty is the best policy.</p>

(HEAD -> feature)

Author: Ankita-Sonal <workharducandoit123@gmail.com>

Date: Sun Mar 31 00:49:56 2024 +0530

new changes

commit 1f30c62b6cfa4dc32badb19b22bb19ad02b17e0b

(origin/feature, main)

Author: Ankita-Sonal <workharducandoit123@gmail.com>

Date: Sat Mar 30 22:18:26 2024 +0530

Add this change

commit d5121807d0609bf96005d971c930216ac07eeb2a

(origin/main)

Author: Ankita-Sonal <workharducandoit123@gmail.com>

Date: Sat Mar 30 20:22:33 2024 +0530

Add feature

commit c0c6f291d8390bfe24672656868d1c7dc5b5dc9d

Author: Ankita-Sonal <workharducandoit123@gmail.com>

Date: Sat Mar 30 18:15:37 2024 +0530

Add initial Files

PS C:\Users\veena\OneDrive\Desktop\Git Hub\LocalRepo> git checkout main

Switched to branch 'main'

PS C:\Users\veena\OneDrive\Desktop\Git Hub\LocalRepo> git cherry-pick 4f29fa18bfc5fffd26465f02ed4ff393a03bf74f

[main 1346163] new changes

Date: Sun Mar 31 00:49:56 2024 +0530

1 file changed, 3 insertions(+), 1 deletion(-)

So finally after so many codes we have come to an end of the presentation.



*But Github has a
lot more concepts which are yet
to be explored.*

THANK YOU!