

Case Study 3: Github hands on. Experience!

- Push a local git repository to Github. Demonstrate how to create and manage pull request using Github interface.
- Use either the Github ^{CLI} or the web interface to modify files, commit changes, and track the history of those changes.

Title :-

Pushing a Local Git Repository to GitHub, Managing Pull requests and Tracking file changes

Aim:-

- Pushing an existing local git repository to Github
- Creating and managing pull request via Github web interface.
- Modifying files through either the Github CLI, committing those changes & viewing their history.

Theory:

GitHub is a cloud based hosting service for git repositories.

- Push: Upload local commits to a remote repository.
- Pull Request: A proposed set of commits that you request be merged into another branch.
- Git CLI: A command line tool that allows to perform many Github operations. - Creating PR, provide views issues, editing files directly from your terminal.

- Blame:- Views that show who changed what, when & why enabling audit and rollback.

a]. Push a Local Repository & Manage Pull requests.

1. In your local project folder.

```
git init
```

```
git add.
```

```
git commit -m "Initial code"
```

```
git remote add origin https://github.com/USERNAME  
IREPO.git.
```

```
git push -u origin main
```

2. On GitHub.

- Navigate to your repository's page.
- Click "Compare & pull request"
- Enter a Title and Description.
- Assign reviewers add labels or link issues.
- Click "Create pull request".

3. Managing the PR.

- Reviewers leave comments inline on diffs
- Use "Review conversations" as you address feedback.
- When approved Click "Merge pull request". Then "confirm merge".
- Optionally delete the feature branch to keep the repo tidy.

b. Modify a file, commit and Track history

option 1: GitHub CLI

1. Edit locally.

```
gh repo clone USERNAME IREPO  
cd REPO
```

```
echo "#update" >> README.md.
```

```
git add README.md.
```

```
git commit -m "Add section header to README"
```

```
git push
```

2. View history

```
gh repo view --web.
```

Options 2: GitHub Web Editor.

1. Navigate to README.md in your GitHub repo

2. Click the pencil icon to edit

3. Make your changes, scroll down to "Commit Changes".

- Choose "Create a new branch for this commit"

- Enter a commit message.

- Click "Commit Changes"

4. Tracking history.

- Open the "History" view for that file to see all commits.

- Use "Blame" to see line by line authorship.

Conclusion:- Editing files directly in the GitHub web UI was surprising fast for small files. Examining the file history & blame views gave me confidence to track & troubleshoot issues.