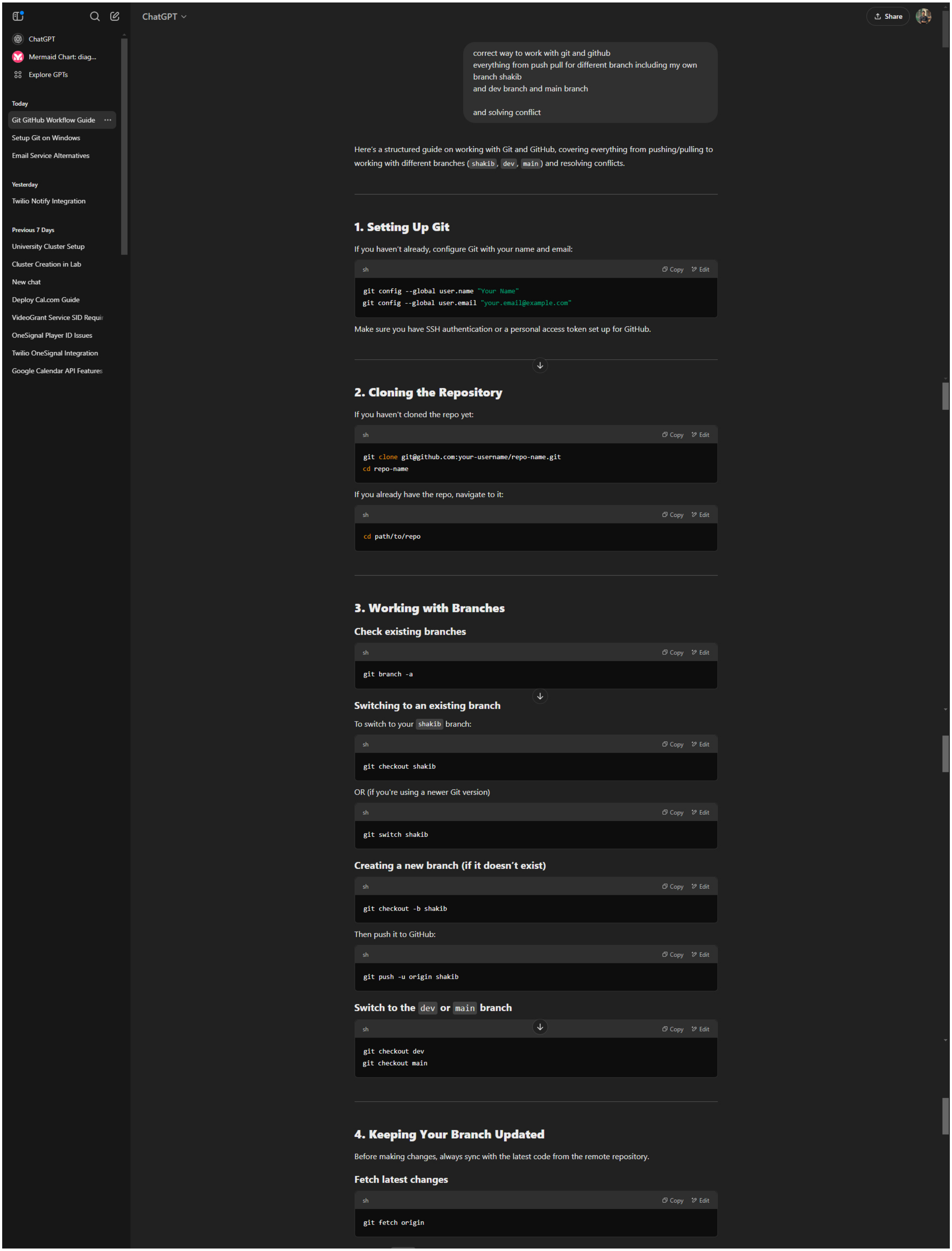

ChatGPT
Mermaid Chart: diag...
Explore GPTs
Today
Git GitHub Workflow Guide
Setup Git on Windows
Email Service Alternatives
Yesterday
Twilio Notify Integration
Previous 7 Days
University Cluster Setup
Cluster Creation in Lab
New chat
Deploy Cal.com Guide
VideoGrant Service SID Requir
OneSignal Player ID Issues
Twilio OneSignal Integration
Google Calendar API Features
ChatGPT
Share
sh
Copy
Edit

correct way to work with git and github
everything from push pull for different branch including my own branch shakib
and dev branch and main branch

and solving conflict

Here's a structured guide on working with Git and GitHub, covering everything from pushing/pulling to working with different branches (`shakib`, `dev`, `main`) and resolving conflicts.

---

1. Setting Up Git

If you haven't already, configure Git with your name and email:

```sh
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

Make sure you have SSH authentication or a personal access token set up for GitHub.

---

2. Cloning the Repository

If you haven't cloned the repo yet:

```sh
git clone git@github.com:your-username/repo-name.git
cd repo-name
```

If you already have the repo, navigate to it:

```sh
cd path/to/repo
```

---

3. Working with Branches

Check existing branches

```sh
git branch -a
```

Switching to an existing branch

To switch to your `shakib` branch:

```sh
git checkout shakib
```

OR (if you're using a newer Git version)

```sh
git switch shakib
```

Creating a new branch (if it doesn't exist)

```sh
git checkout -b shakib
```

Then push it to GitHub:

```sh
git push -u origin shakib
```

Switch to the `dev` or `main` branch

```sh
git checkout dev
git checkout main
```

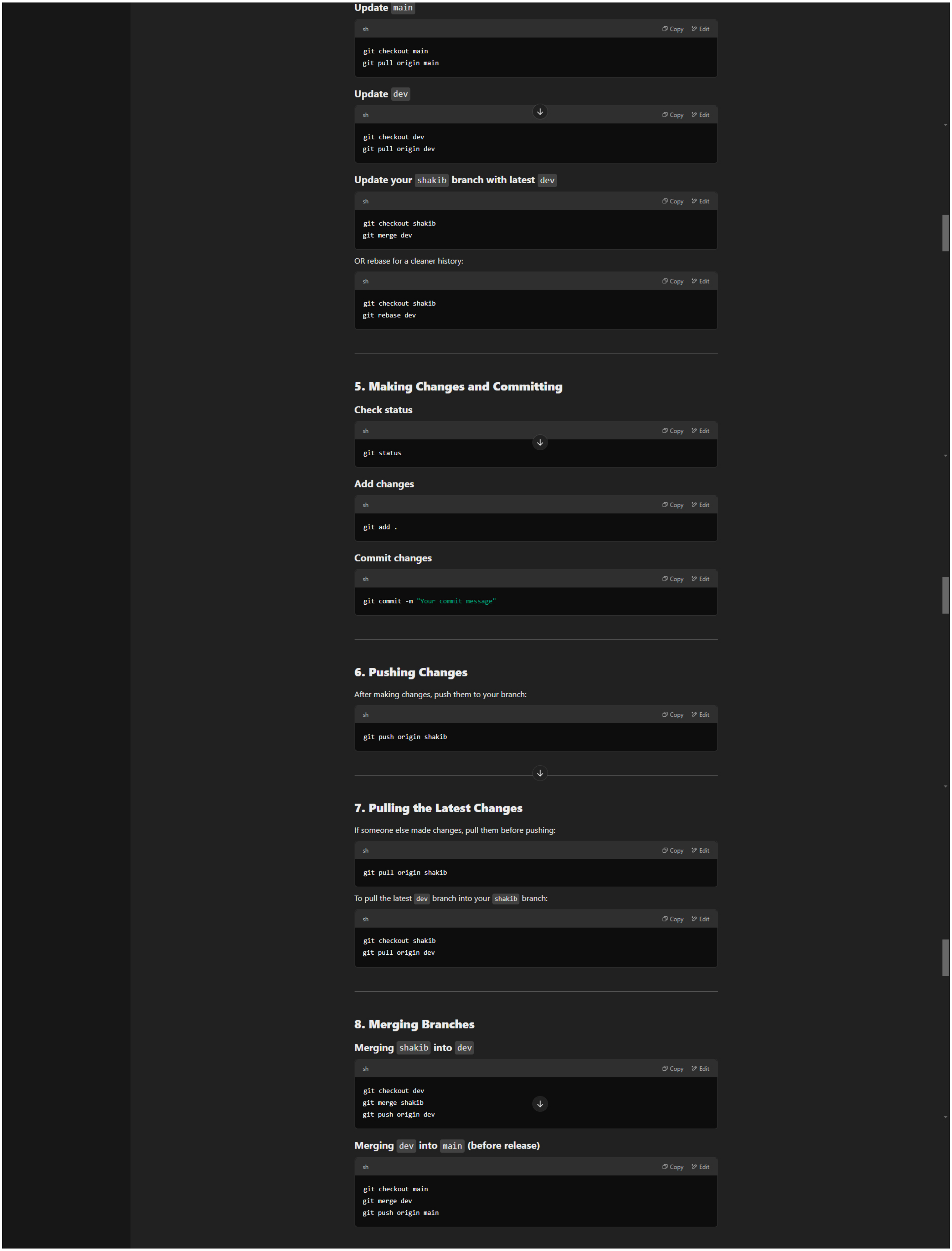
---

4. Keeping Your Branch Updated

Before making changes, always sync with the latest code from the remote repository.

Fetch latest changes

```sh
git fetch origin
```



Update `main`

```sh
git checkout main
git pull origin main
```

Update `dev`

```sh
git checkout dev
git pull origin dev
```

Update your `shakib` branch with latest `dev`

```sh
git checkout shakib
git merge dev
```

OR rebase for a cleaner history:

```sh
git checkout shakib
git rebase dev
```

---

5. Making Changes and Committing

Check status

```sh
git status
```

Add changes

```sh
git add .
```

Commit changes

```sh
git commit -m "Your commit message"
```

---

6. Pushing Changes

After making changes, push them to your branch:

```sh
git push origin shakib
```

---

7. Pulling the Latest Changes

If someone else made changes, pull them before pushing:

```sh
git pull origin shakib
```

To pull the latest `dev` branch into your `shakib` branch:

```sh
git checkout shakib
git pull origin dev
```

---

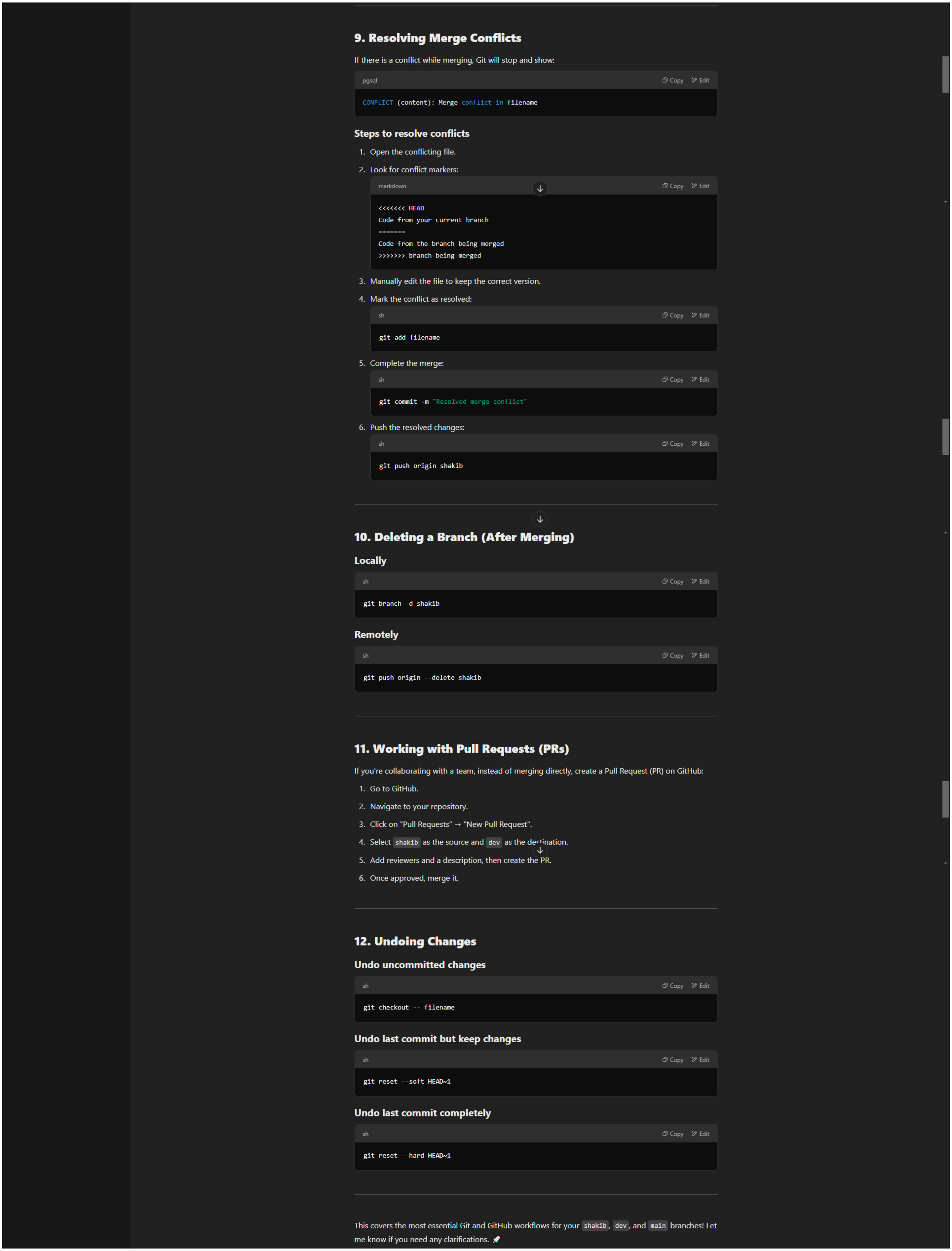
8. Merging Branches

Merging `shakib` into `dev`

```sh
git checkout dev
git merge shakib
git push origin dev
```

Merging `dev` into `main` (before release)

```sh
git checkout main
git merge dev
git push origin main
```



9. Resolving Merge Conflicts

If there is a conflict while merging, Git will stop and show:

```pgsql
CONFLICT (content): Merge conflict in filename
```

Steps to resolve conflicts

1. Open the conflicting file.
2. Look for conflict markers:
   ```markdown
   <<<<<<< HEAD
   Code from your current branch
   =======
   Code from the branch being merged
   >>>>>>> branch-being-merged
   ```
3. Manually edit the file to keep the correct version.
4. Mark the conflict as resolved:
   ```sh
   git add filename
   ```
5. Complete the merge:
   ```sh
   git commit -m "Resolved merge conflict"
   ```
6. Push the resolved changes:
   ```sh
   git push origin shakib
   ```

---

10. Deleting a Branch (After Merging)

Locally

```sh
git branch -d shakib
```

Remotely

```sh
git push origin --delete shakib
```

---

11. Working with Pull Requests (PRs)

If you're collaborating with a team, instead of merging directly, create a Pull Request (PR) on GitHub:

1. Go to GitHub.
2. Navigate to your repository.
3. Click on "Pull Requests" → "New Pull Request".
4. Select `shakib` as the source and `dev` as the destination.
5. Add reviewers and a description, then create the PR.
6. Once approved, merge it.

---

12. Undoing Changes

Undo uncommitted changes

```sh
git checkout -- filename
```

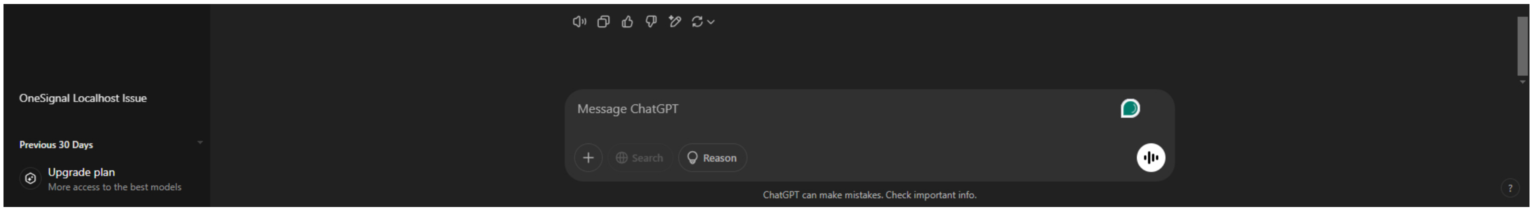
Undo last commit but keep changes

```sh
git reset --soft HEAD~1
```

Undo last commit completely

```sh
git reset --hard HEAD~1
```

This covers the most essential Git and GitHub workflows for your `shakib`, `dev`, and `main` branches! Let me know if you need any clarifications. 🚀


OneSignal Localhost Issue
Previous 30 Days
Upgrade plan
More access to the best models
Message ChatGPT
Search
Reason
ChatGPT can make mistakes. Check important info.
?