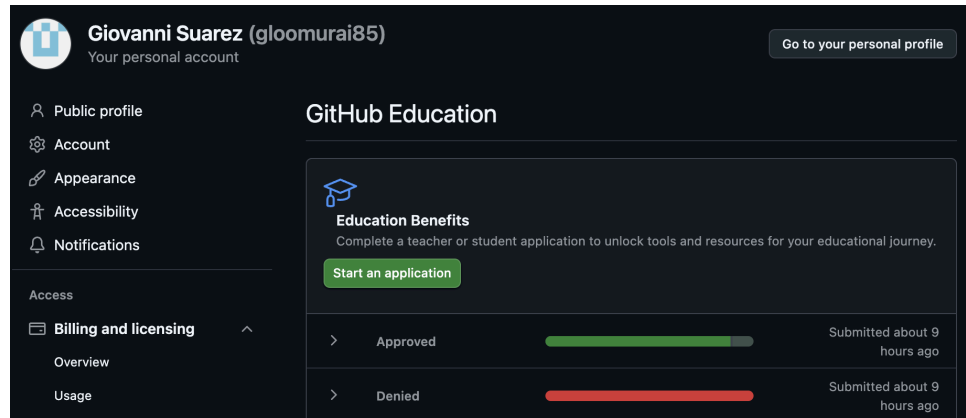


## Part a — GitHub Education



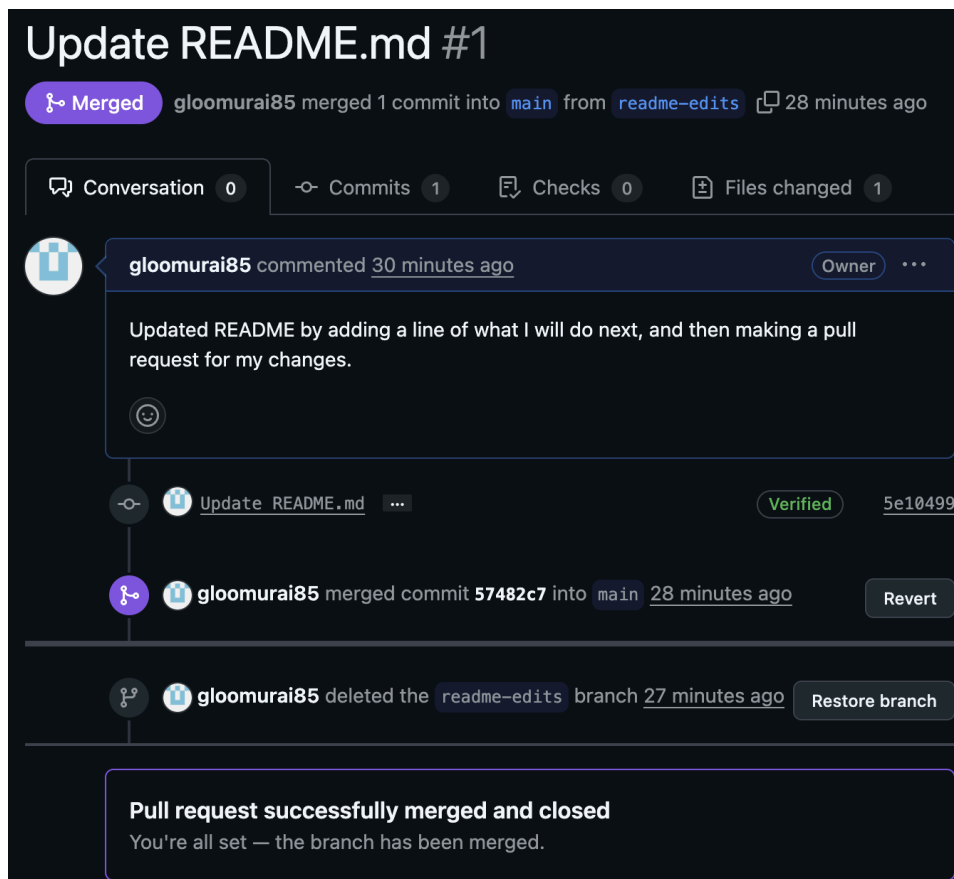
## Part b — GitHub Tutorial from Web UI

### What was done:

- Created a repository named **hello-world**.
- Added a **README.md** on **main**.
- Created a feature branch **readme-edits**, edited the README, and committed.
- Opened a Pull Request from **readme-edits** to **main**, merged, and deleted the branch.

### What I learned:

- Safe change-making via branches.
- Pull requests = discussion, review, and merge point.
- Merge records the history and preserves main's stability.



## Part c — Atlassian Notes

### 1. What is Version Control?

- **Main idea:** Version control tracks file changes over time, enabling rollback, comparison, branching/merging, and collaborative editing without overwriting work.
- **Why care:** It creates a reliable history ("who changed what, when, and why") and supports experiments via branches without risking the main code.
- **Useful Concepts:** Branches for features/experiments; commits with meaningful messages; diffs to review changes; tags for milestones/releases.
- **Actions:** I will commit small, logical chunks with descriptive messages so teammates (and future me) can understand intent quickly.

### 2. What is Git?

- **Main idea:** Git is a distributed VCS. Every clone has the full history; commits are snapshots identified by hashes; branching is fast and cheap.

- **Core mechanics:** Staging area (index), commits (atomic change sets), remotes (ex. **origin**), and workflows (feature branches, PRs).

**Merging & conflicts:** Most merges are automatic; conflicts require manual resolution but remain localized and traceable.

- **Actions/Takeaway:** I'll keep **main** always releasable, develop on short-lived feature branches, and use PRs to merge back.
- Understanding that Git snapshots (not diffs only) and cheap branching encourages frequent, low-risk experimentation.

### 3. Source Code Management

- **Main idea:** SCM builds on VCS with collaboration practices such as reviews, PRs, issues, permissions, and integration with CI/CD.
- **Team workflows:** Feature-branch, Gitflow, or trunk-based; each aligns team structure with release cadence and risk tolerance.
- **Traceability:** Issues → commits → PRs → releases create an auditable chain from requirements to code.
- **Actions/Takeway:** I'll associate commits/PRs with issues, request reviews for meaningful changes, and use status checks before merge.
- Even solo, SCM discipline (issues, PRs, checklists) reduces defects and clarifies decisions for later coursework reuse.

## Part d — Git from Command Line

Verified configuration with **git config --global --list**

## Part e — Cloning Tutorial Repo Locally

**Outcome:** Local working copy with **.git** present and **README.md** visible.

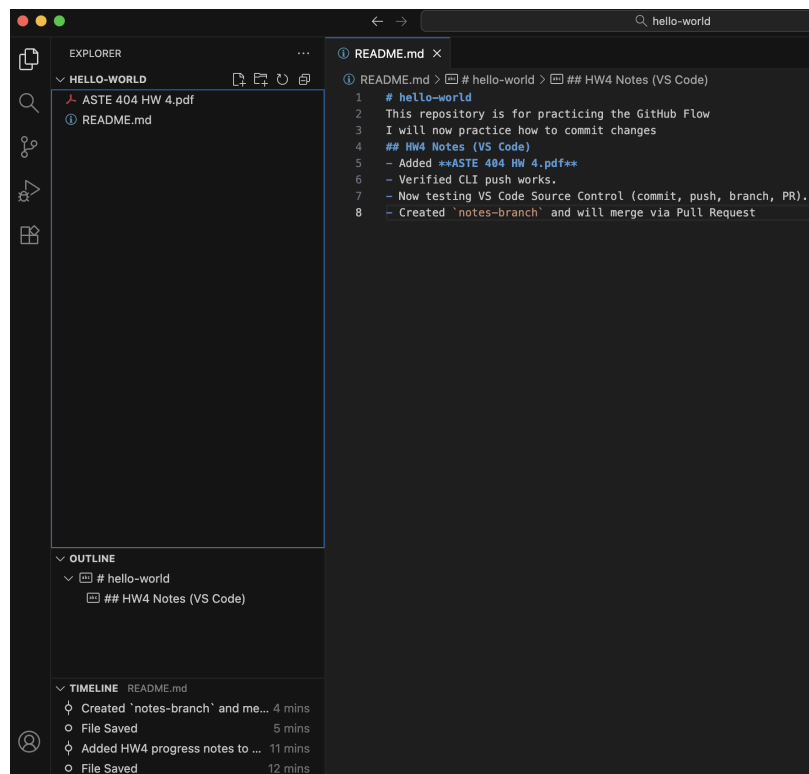
```
(base) giovannisuares@KA-R3N-30 HW4 % git clone https://github.com/gloomurai85/hello-world.git
Cloning into 'hello-world'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
(base) giovannisuares@KA-R3N-30 HW4 % cd hello-world
(base) giovannisuares@KA-R3N-30 hello-world % ls -la
total 8
drwxr-xr-x  4 giovannisuares  staff  128 Oct 20 09:32 .
drwxr-xr-x  3 giovannisuares  staff   96 Oct 20 09:32 ..
drwxr-xr-x 12 giovannisuares  staff  384 Oct 20 09:32 .git
-rw-r--r--  1 giovannisuares  staff  106 Oct 20 09:32 README.md
(base) giovannisuares@KA-R3N-30 hello-world %
```

## Part f — Committing Changes from Command

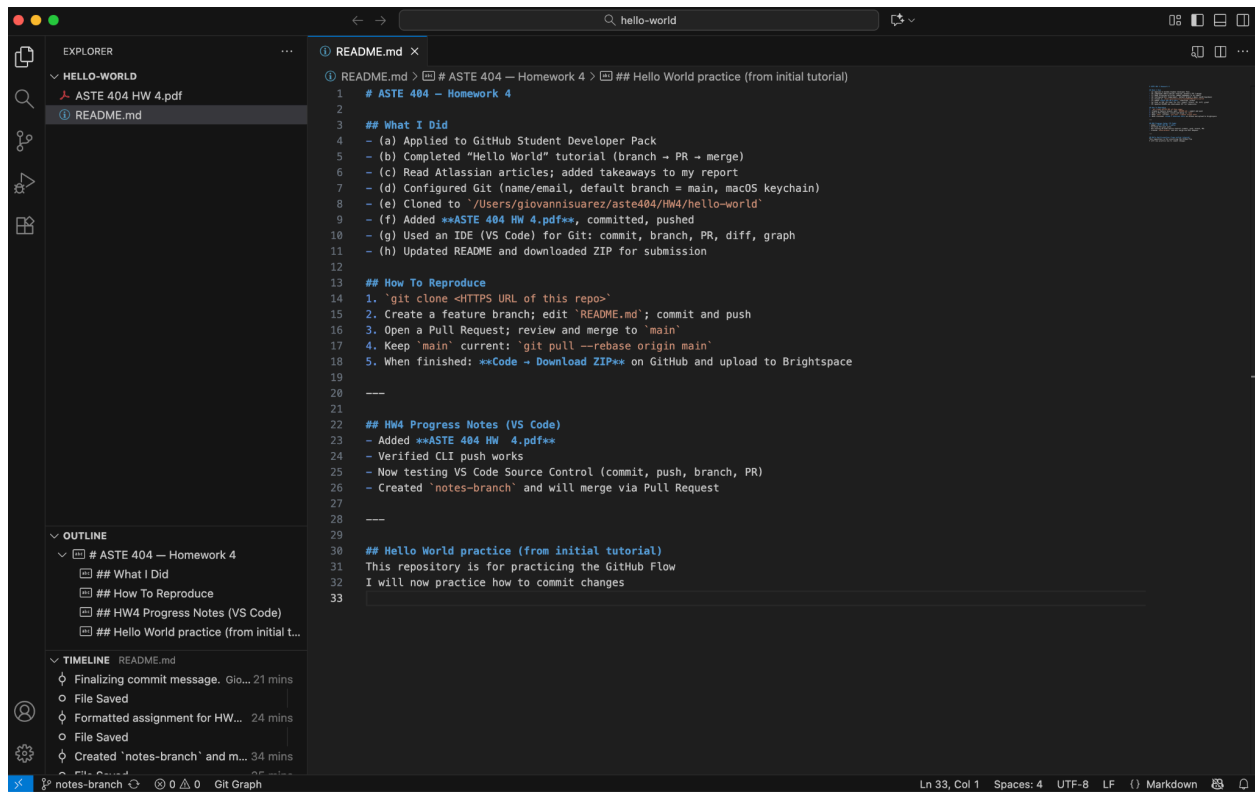
```
(base) giovannisuares@KA-R3N-30 hello-world % git add "ASTE 404 HW 4.pdf"
[(base) giovannisuares@KA-R3N-30 hello-world % git commit -m "Add HW4 report PDF"
[[main c549267] Add HW4 report PDF
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 ASTE 404 HW 4.pdf
(base) giovannisuares@KA-R3N-30 hello-world % git branch --show-current
main
(base) giovannisuares@KA-R3N-30 hello-world % git push origin main
[Username for 'https://github.com': git push -u origin $(git branch --show-current)
Password for 'https://git push -u origin $(git branch --show-current)@github.com':

Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 428.08 KiB | 38.92 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/gloomurai85/hello-world.git
 57482c7..c549267  main -> main
(base) giovannisuares@KA-R3N-30 hello-world % git log --oneline -n 3
c549267 (HEAD -> main, origin/main, origin/HEAD) Add HW4 report PDF
57482c7 Merge pull request #1 from gcsuarez85/readme-edits
5e10499 Update README.md
```

## Part g — Git Integration in IDE



## Part h — Updating README + Report



The screenshot shows a VS Code editor window with a README.md file open. The Explorer sidebar on the left shows a project structure with 'HELLO-WORLD' and 'ASTE 404 HW 4.pdf'. The README.md file contains the following content:

```
1 # ASTE 404 - Homework 4
2
3 ## What I Did
4 - (a) Applied to GitHub Student Developer Pack
5 - (b) Completed "Hello World" tutorial (branch -> PR -> merge)
6 - (c) Read Atlassian articles; added takeaways to my report
7 - (d) Configured Git (name/email, default branch = main, macOS keychain)
8 - (e) Cloned to '/Users/giovannisuares/aste404/HW4/hello-world'
9 - (f) Added **ASTE 404 HW 4.pdf**, committed, pushed
10 - (g) Used an IDE (VS Code) for Git: commit, branch, PR, diff, graph
11 - (h) Updated README and downloaded ZIP for submission
12
13 ## How To Reproduce
14 1. 'git clone <HTTPS URL of this repo>'
15 2. Create a feature branch; edit 'README.md'; commit and push
16 3. Open a Pull Request; review and merge to 'main'
17 4. Keep 'main' current: 'git pull --rebase origin main'
18 5. When finished: **Code -> Download ZIP** on GitHub and upload to Brightspace
19
20 ---
21
22 ## HW4 Progress Notes (VS Code)
23 - Added **ASTE 404 HW 4.pdf**
24 - Verified CLI push works
25 - Now testing VS Code Source Control (commit, push, branch, PR)
26 - Created 'notes-branch' and will merge via Pull Request
27
28 ---
29
30 ## Hello World practice (from initial tutorial)
31 This repository is for practicing the GitHub Flow
32 I will now practice how to commit changes
33
```

The Outline sidebar on the left shows the following structure:

- ASTE 404 — Homework 4
  - ## What I Did
  - ## How To Reproduce
  - ## HW4 Progress Notes (VS Code)
  - ## Hello World practice (from initial t...
- TIMELINE README.md
  - Finalizing commit message. Glo... 21 mins
  - File Saved
  - Formatted assignment for HW... 24 mins
  - File Saved
  - Created 'notes-branch' and m... 34 mins

The status bar at the bottom shows 'Ln 33, Col 1', 'Spaces: 4', 'UTF-8', 'LF', and 'Markdown'.