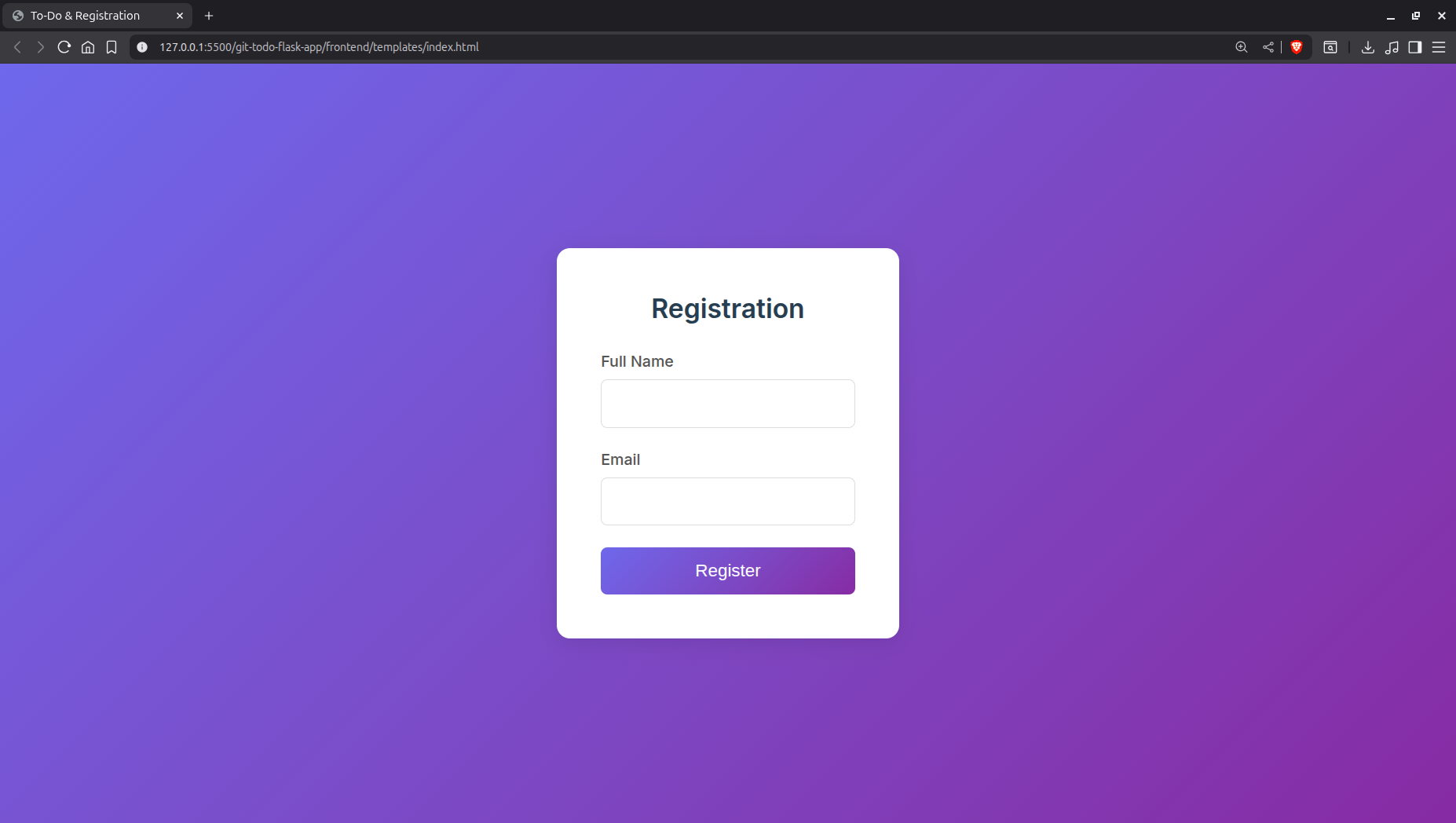
Git & Github Minor Project

# Git & GitHub Workflow for Flask To-Do App Project

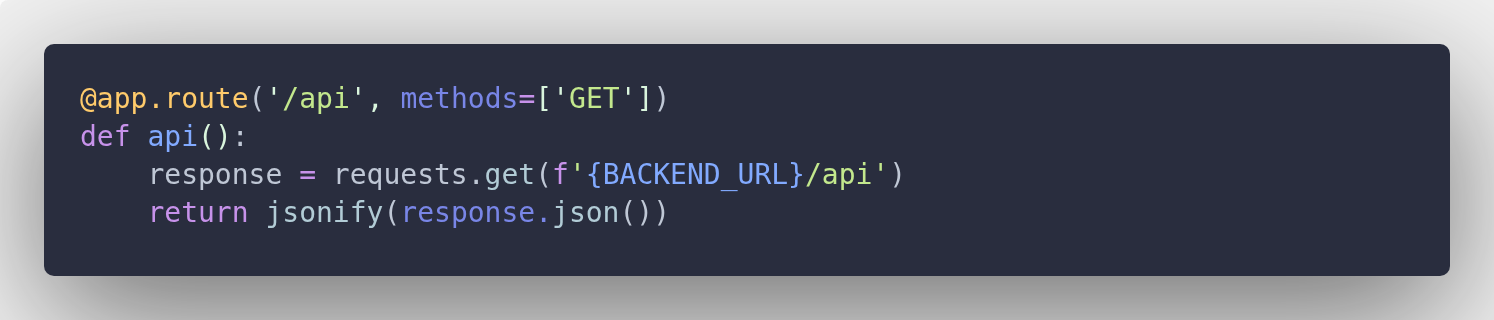
## 1. Repository Setup

- Created a new GitHub repository.  
- Cloned the repository to my local machine using SSH.  
- Created a new branch named “Naman”  
- Added my basic Flask project files to this branch.  
- Commit the changes and merged this branch into the “main” branch.



## 2. New Branch and JSON Update

- Create a new branch named “Naman\_new”  
- Updated the content of the JSON file used for the `/api` route.  
- Merged the “Naman\_new” branch into the “main” branch.  
- Added the resolved changes to staging, commit, and pushed updates to the remote repository.



## 3. Feature Branches

- Created two branches “master\_1” and “master\_2” from the main branch.

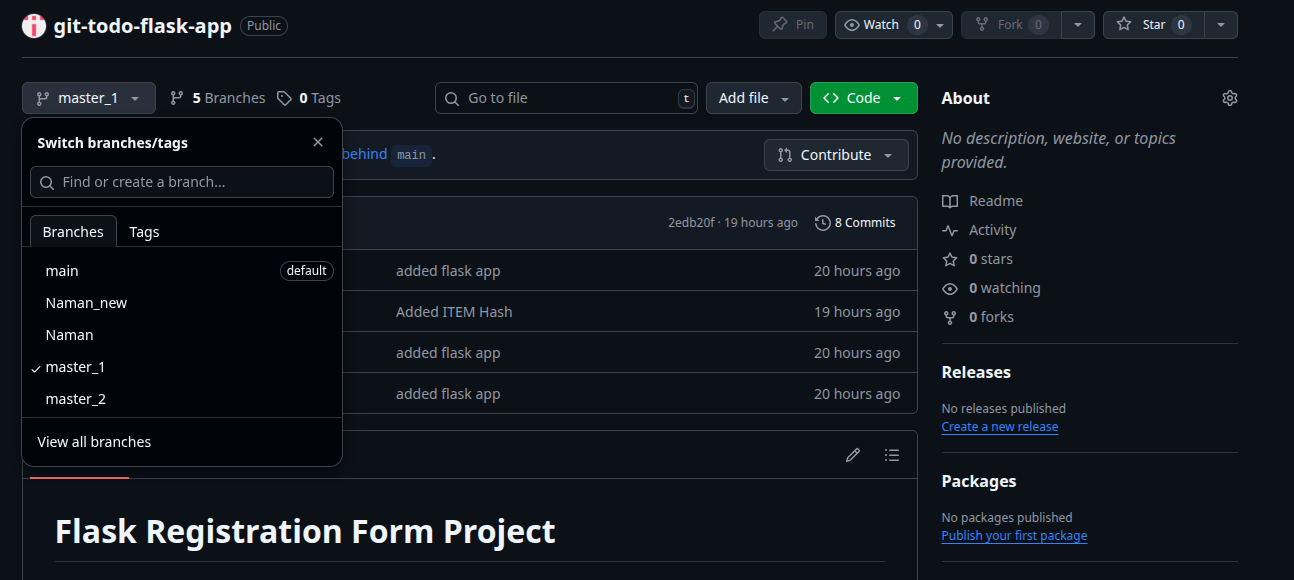
Commands:

- git branch master\_1

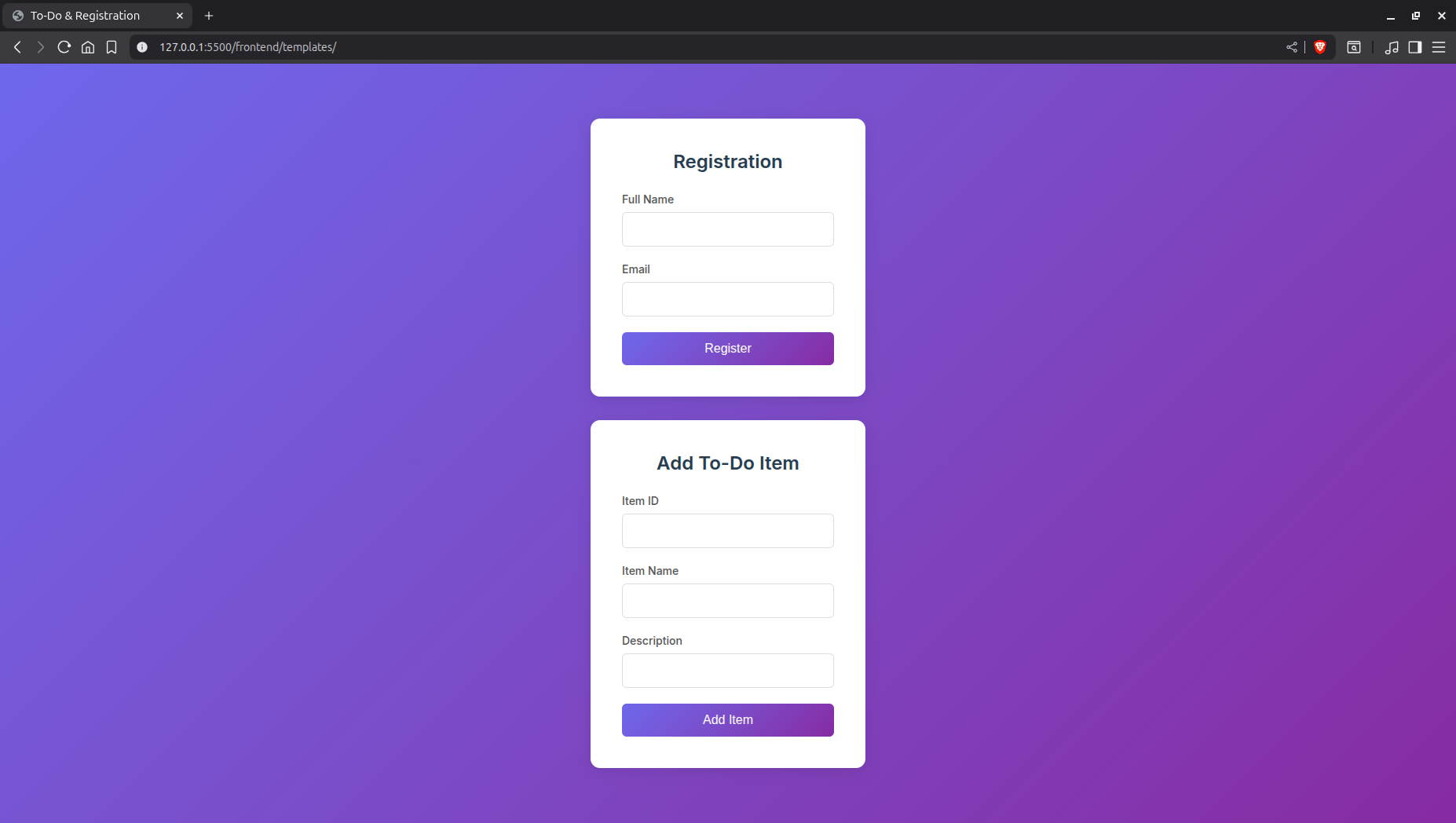
- git checkout main

- git branch master\_2

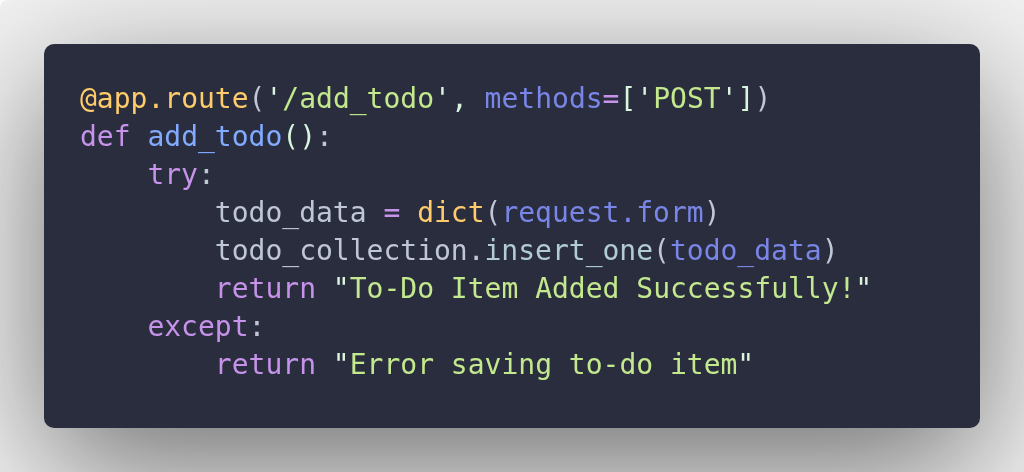
- git checkout main



**Frontend** (master\_1):  
- Added To-Do into the Page in the frontend.  
- The page have a form with fields: Item Name, Item Description.



**Backend** (master\_2):  
- Added a backend route “/add\_todo”.  
- So that the Input objects from form can be directed to the MongoDB i.e,“itemName`” and “itemDescription” via [POST Method].

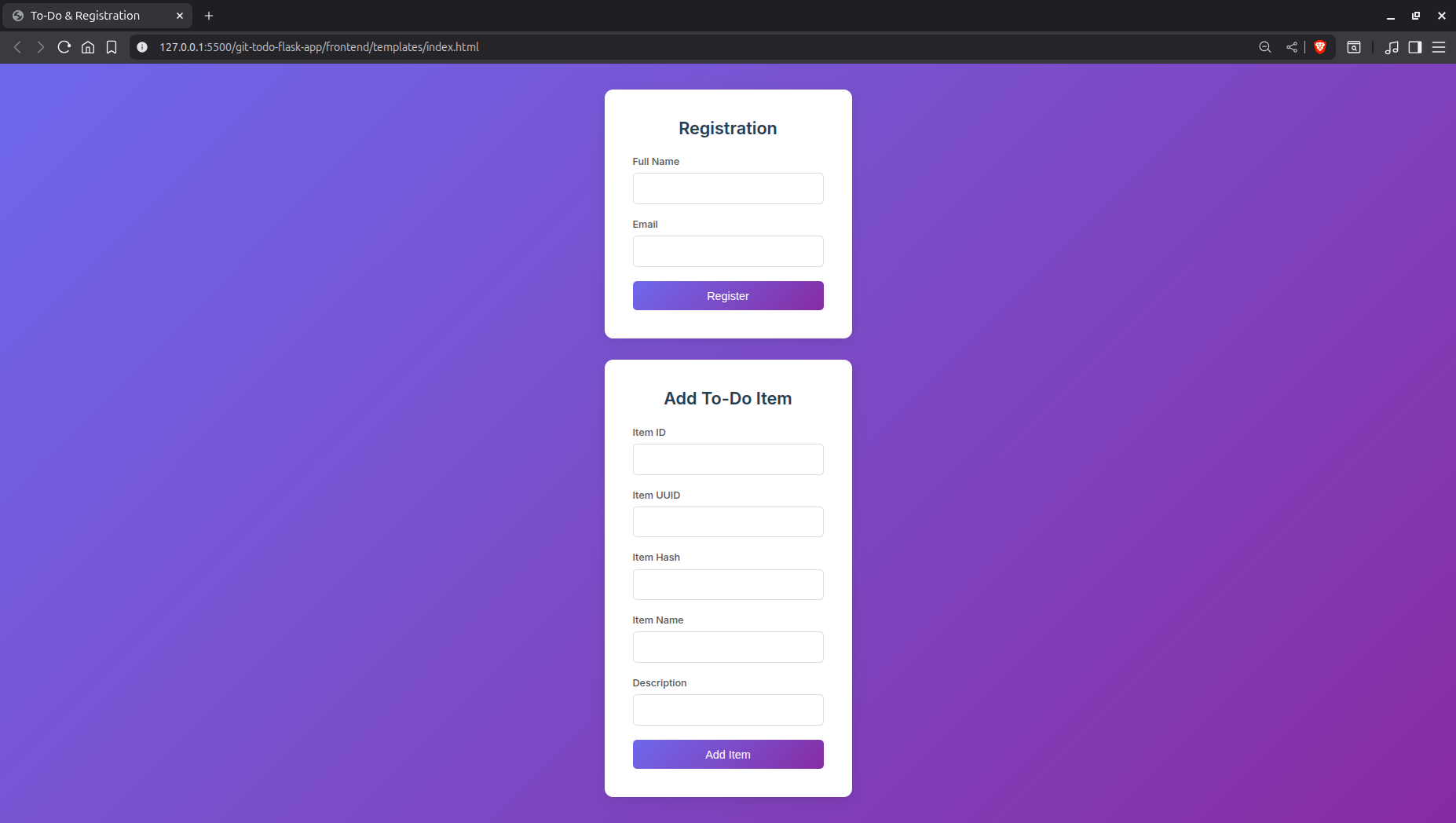


Merging:  
- Merged the changes from both “master\_1” and “master\_2” into the “main” branch.

## 4. Enhancing the To-Do Form (master\_1)

- In `master\_1`, add fields to the form: Item ID, Item UUID, Item Hash.  
- Added commits in sequence:  
 1. Add Item ID field.  
 2. Add Item UUID field.  
 3. Add Item Hash field.

& Merged the “master\_1” branch into the “main” branch.



## 5. Git Reset and Commit Rollback

In the main branch, reset to the commit where only the Item ID field was added.  
- Command:

- git reset –soft fecd0b7711fd9116df360ed93439349sdsdas3430d35

- git add

- git commit -m “re-commit”

- git push

Pushed the updated state to the remote repository.

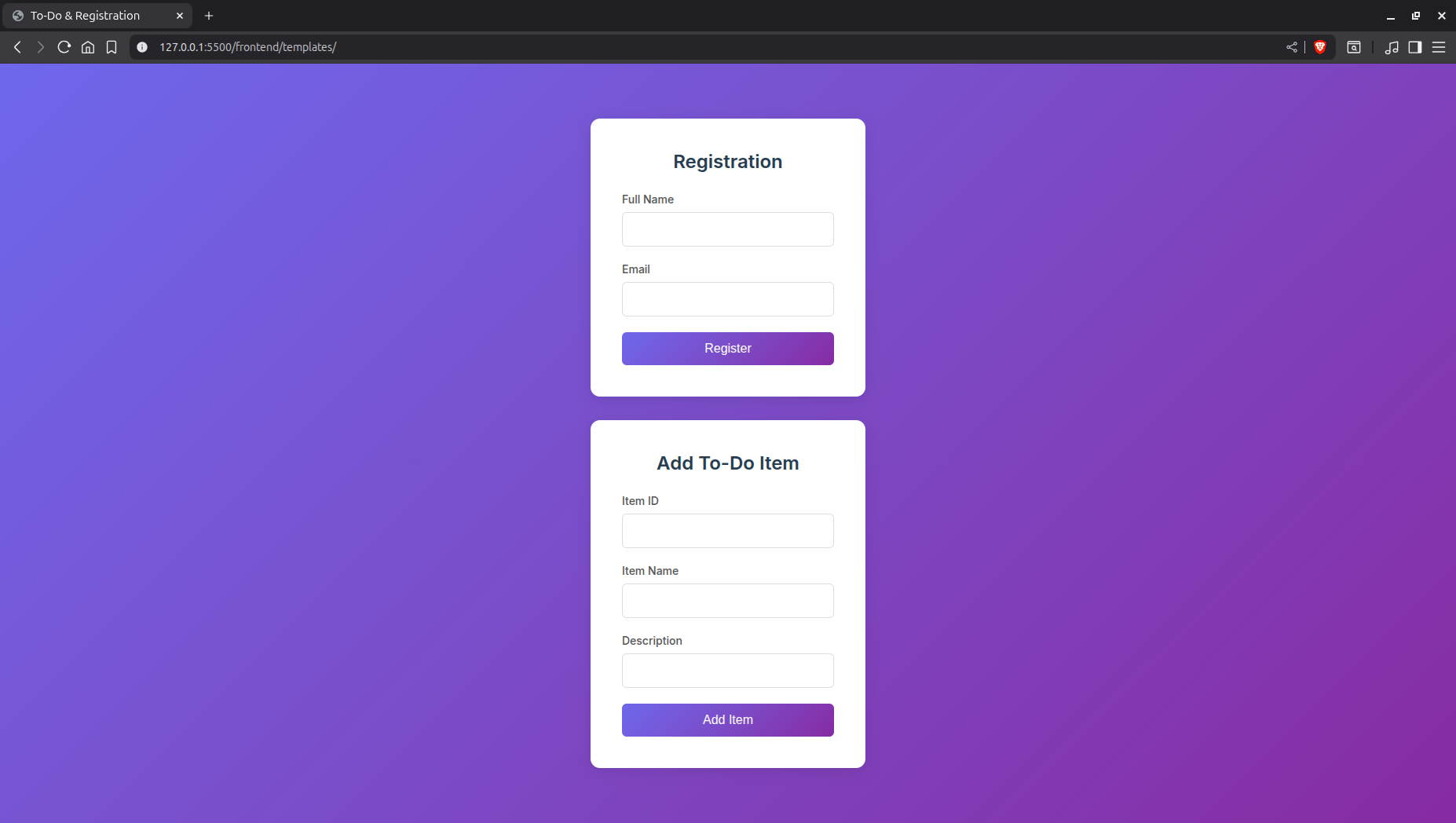
## 6. Rebasing Changes

- Rebase the updated changes in the main branch to the `master\_1` branch.  
- Preserve individual commits during rebasing  
- Command:

`git checkout master\_1`

`git rebase main`

`git add .`

`git rebase –continue’

- This ensures `master\_1` has the latest updates from `main` while keeping its commit history.