

# Học Git - Quản Lí Dự Án

---

## 01. Học Git

### 1.1. Giới thiệu chung

- [Tài Git tại đây](#)

### 1.2. Cài đặt Git

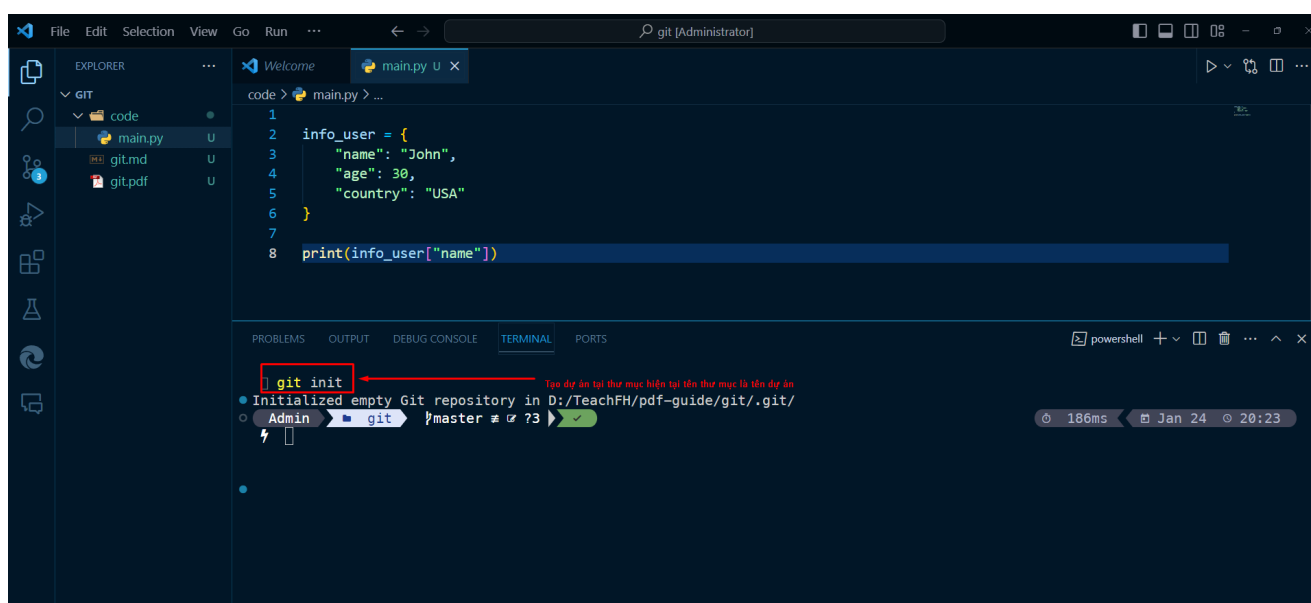
- **Working directory:** Khu vực chứa dự án đang làm việc.
- **Staging area:** Khu vực chứa thông tin thay đổi của các file.
- **Repository (Git directory):** Kho lưu trữ để lưu trữ dữ liệu và lịch sử các phiên bản.

### 1.3. Các thuật ngữ: Working directory, Staging area, Repository

- `git --version`: Kiểm tra phiên bản Git.
- `git --help`: Danh sách các câu lệnh Git.
- `git config`: Cấu hình username và email.
- `git config --global user.name`: Xem thông tin cấu hình username.
- `git config --global user.email`: Xem thông tin cấu hình email.
- `git config --global user.name "Nguyễn Phương Anh Tú"`: Đặt username mới.
- `git config --global user.email "techdev.td1111@gmail.com"`: Đặt email mới.

### 1.4. Các câu lệnh Git sử dụng phổ biến

- `git init`: Khởi tạo kho lưu trữ (repository).

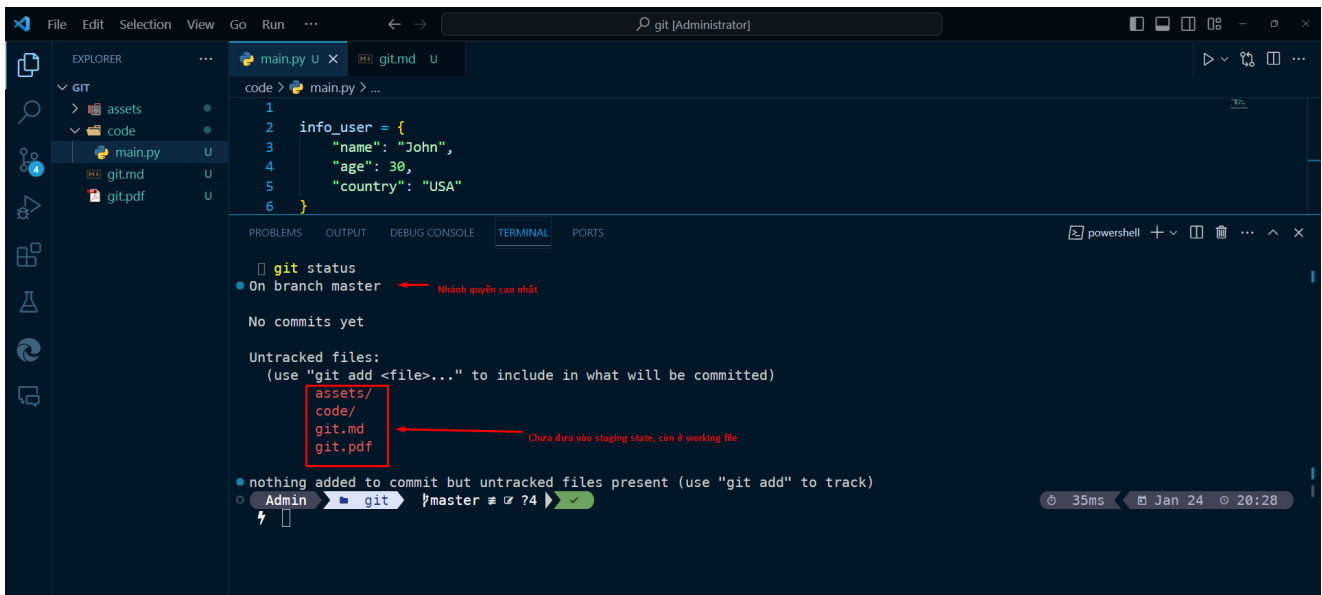


The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project named 'code' with a subdirectory 'main.py'. The main editor displays a Python file 'main.py' with the following code:

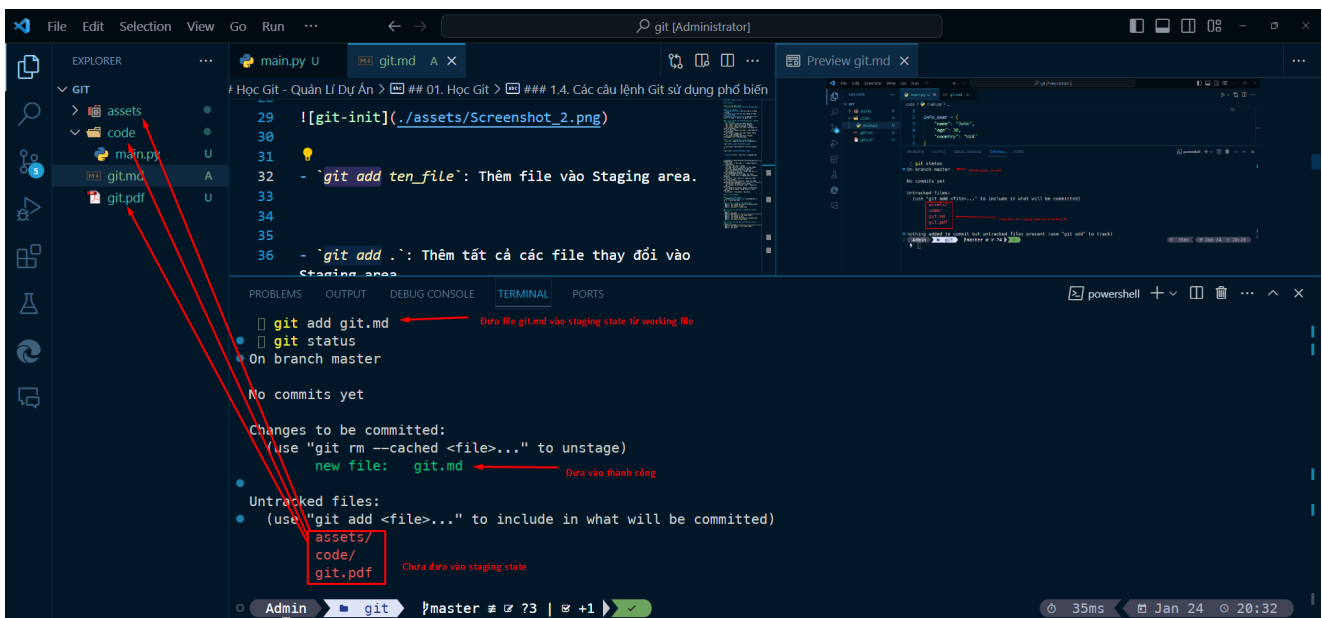
```
1
2 info_user = {
3     "name": "John",
4     "age": 30,
5     "country": "USA"
6 }
7
8 print(info_user["name"])
```

The Terminal panel at the bottom shows the command `git init` being executed. The output indicates that an empty Git repository has been initialized in the directory `D:/TeachFH/pdf-guide/git/.git/`. The terminal prompt is `Admin git /master # 73`.

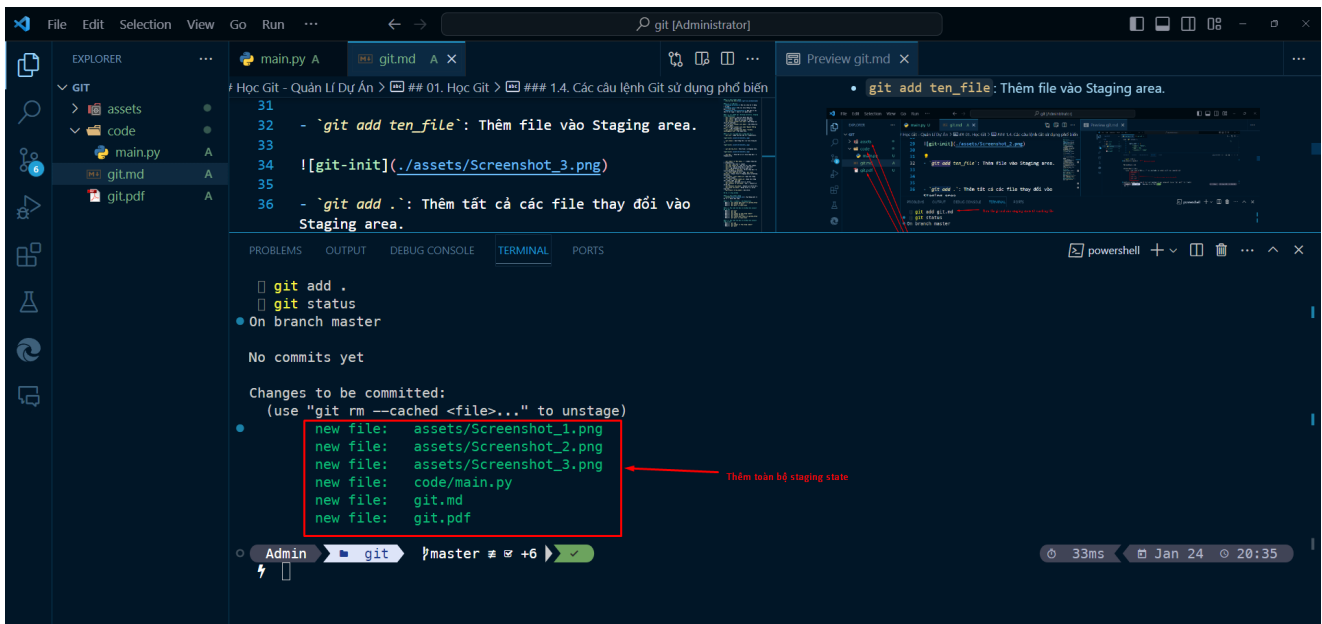
- `git status`: Xem trạng thái của file trong dự án.



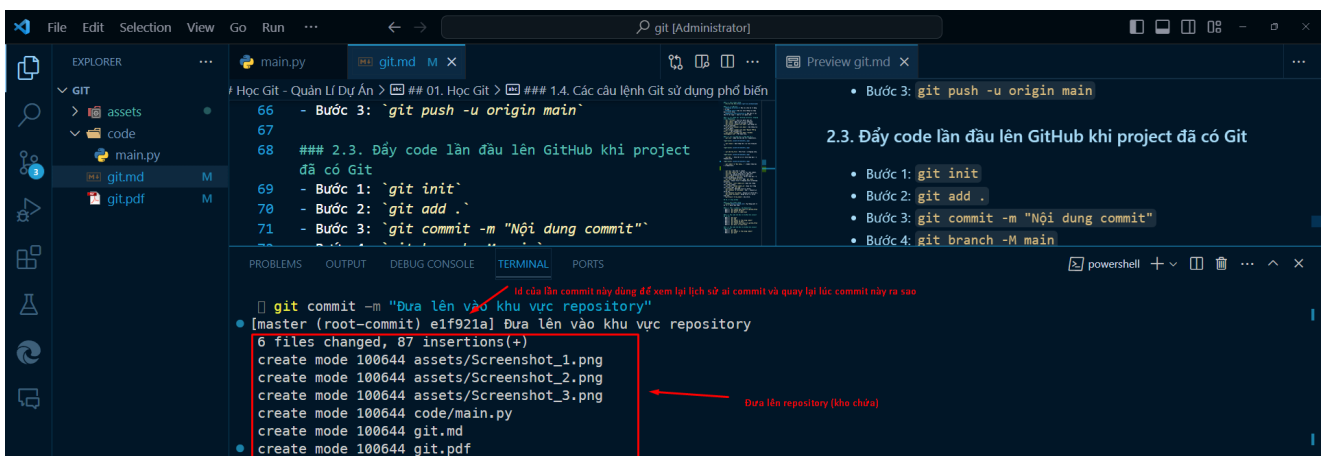
- `git add ten_file`: Thêm file vào Staging area.



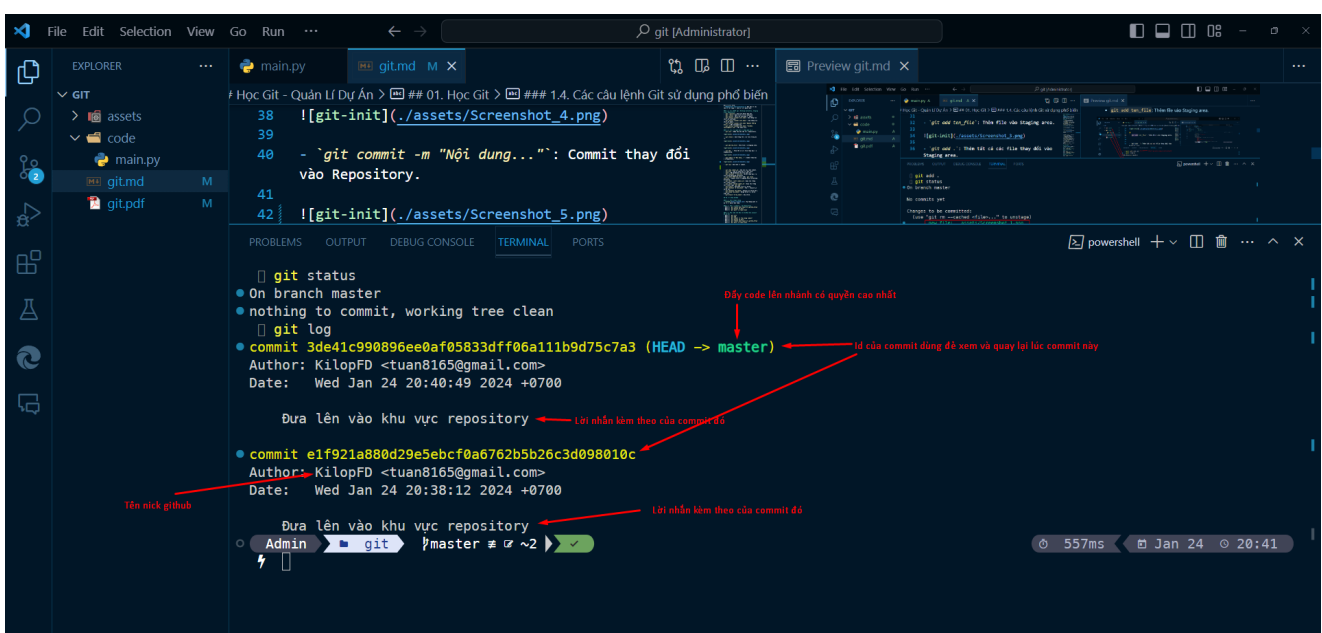
- `git add .`: Thêm tất cả các file thay đổi vào Staging area.



- `git commit -m "Nội dung..."`: Commit thay đổi vào Repository.



- `git log`: Xem lịch sử commit.



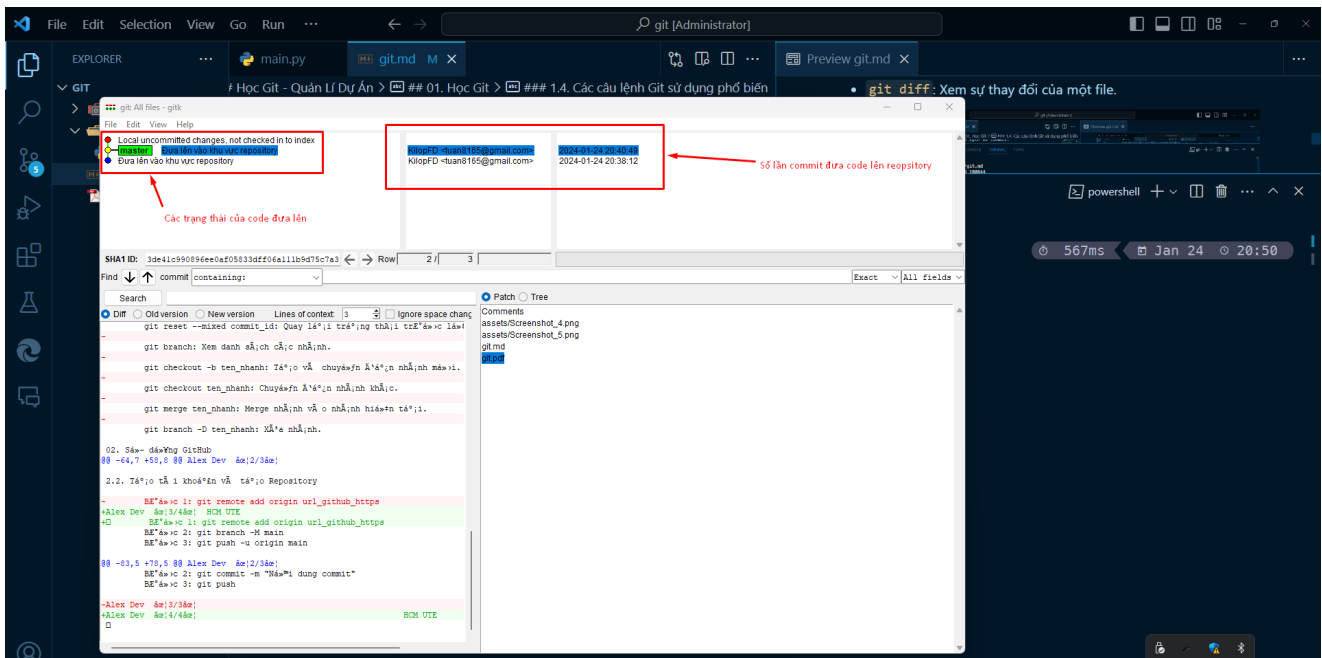
- `git show commit_id`: Xem chi tiết một commit.

```
# Học Git - Quản lý Dự Án > ## 01. Học Git > ### 1.4. Các câu lệnh Git sử dụng phổ biến
45
git diff
diff --git a/assets/Screenshot_1.png b/assets/Screenshot_1.png
new file mode 100644
index 0000000..65fd1d3
Binary files /dev/null and b/assets/Screenshot_1.png differ
diff --git a/assets/Screenshot_2.png b/assets/Screenshot_2.png
new file mode 100644
index 0000000..073ca86
Binary files /dev/null and b/assets/Screenshot_2.png differ
diff --git a/assets/Screenshot_3.png b/assets/Screenshot_3.png
new file mode 100644
index 0000000..1e56f49
Binary files /dev/null and b/assets/Screenshot_3.png differ
diff --git a/code/main.py b/code/main.py
new file mode 100644
index 0000000..85cc894
```

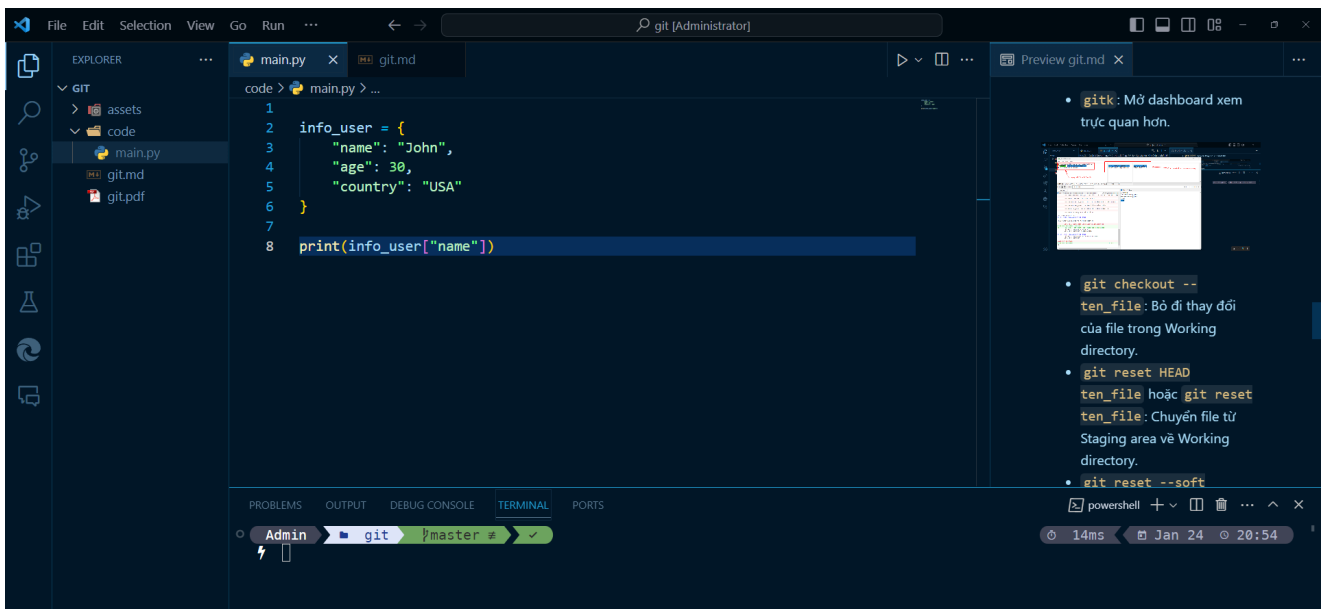
- **git diff**: Xem sự thay đổi của một file.

```
# Học Git - Quản lý Dự Án > ## 01. Học Git > ### 1.4. Các câu lệnh Git sử dụng phổ biến
git diff
diff --git a/git.md b/git.md
index 756cf5e..3ec8629 100644
--- a/git.md
+++ b/git.md
@@ -39,10 +39,16 @@
- `git commit -m "Nội dung..."`: Commit thay đổi vào Repository.
+
+![git-init](./assets/Screenshot_5.png)
- `git log`: Xem lịch sử commit.
+
+![git-init](./assets/Screenshot_6.png)
+
- `git show commit_id`: Xem chi tiết một commit.
+
+![git-init](./assets/Screenshot_7.png)
+
- `git diff`: Xem sự thay đổi của một file.
- `gitk`: Mở dashboard xem trực quan hơn.
- `git checkout -- ten_file`: Bỏ đi thay đổi của file trong Working directory.
diff --git a/git.pdf b/git.pdf
index 2916c4f..00e351c 100644
--- a/git.pdf
+++ b/git.pdf
```

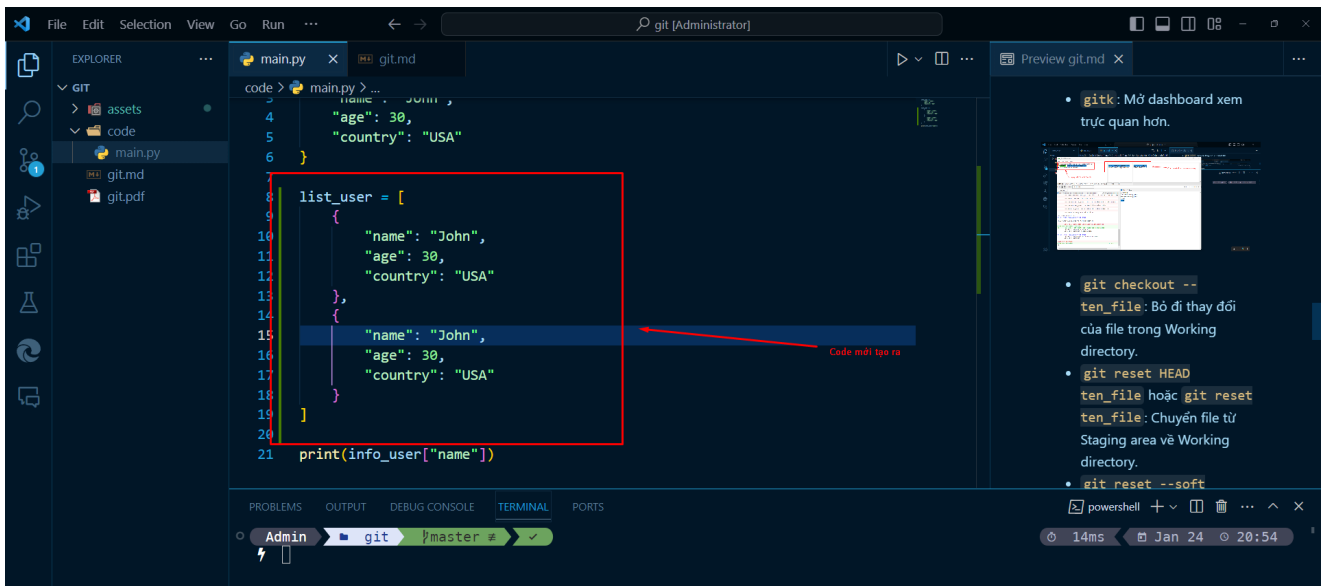
- **gitk**: Mở dashboard xem trực quan hơn.



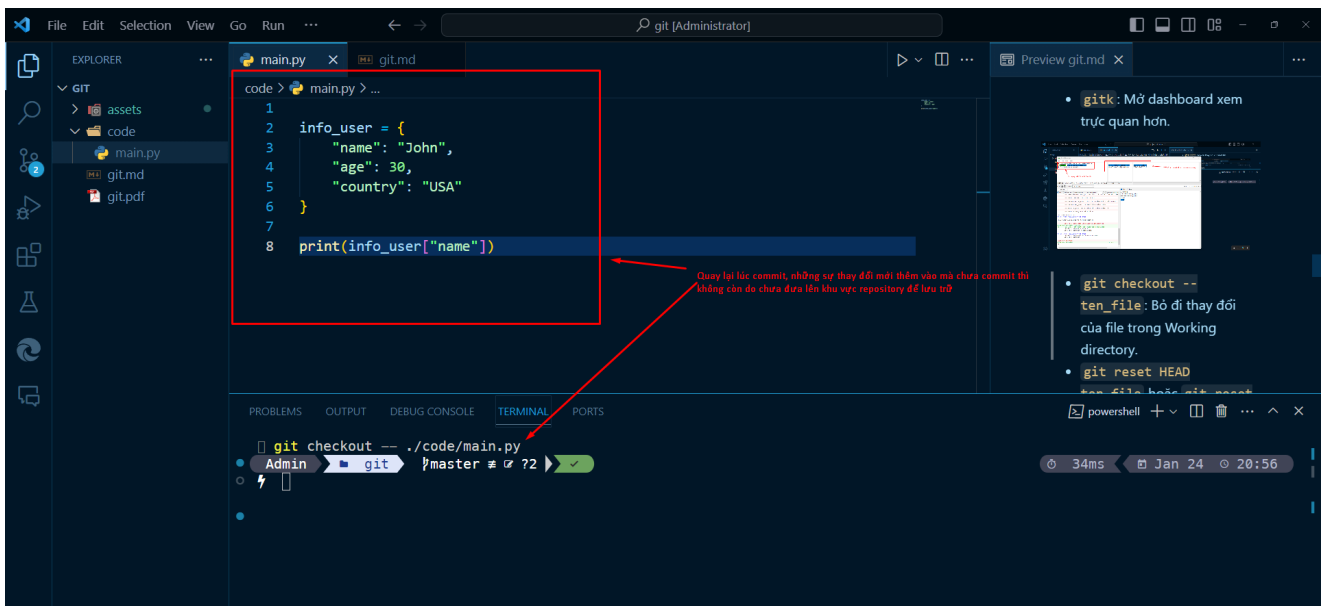
- `git checkout -- ten_file`: Bỏ đi thay đổi của file trong Working directory.



Lúc commit ban đầu

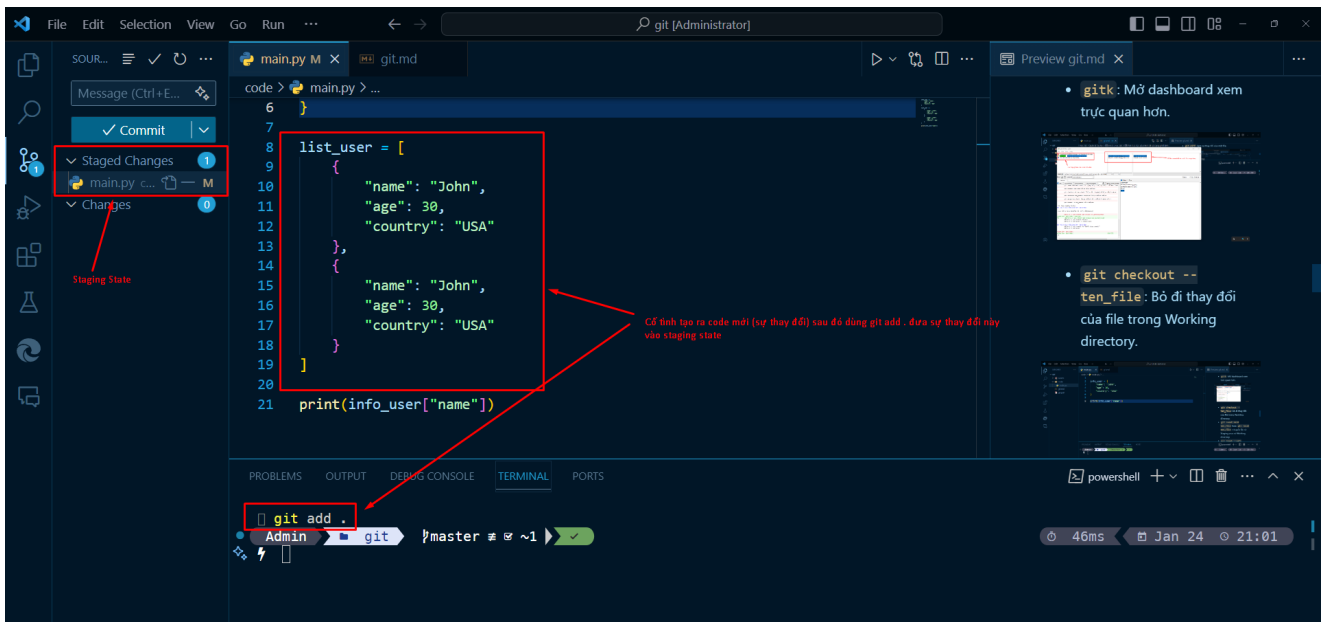


Thêm code nhưng chưa commit(ghi nhận) sự thay đổi.

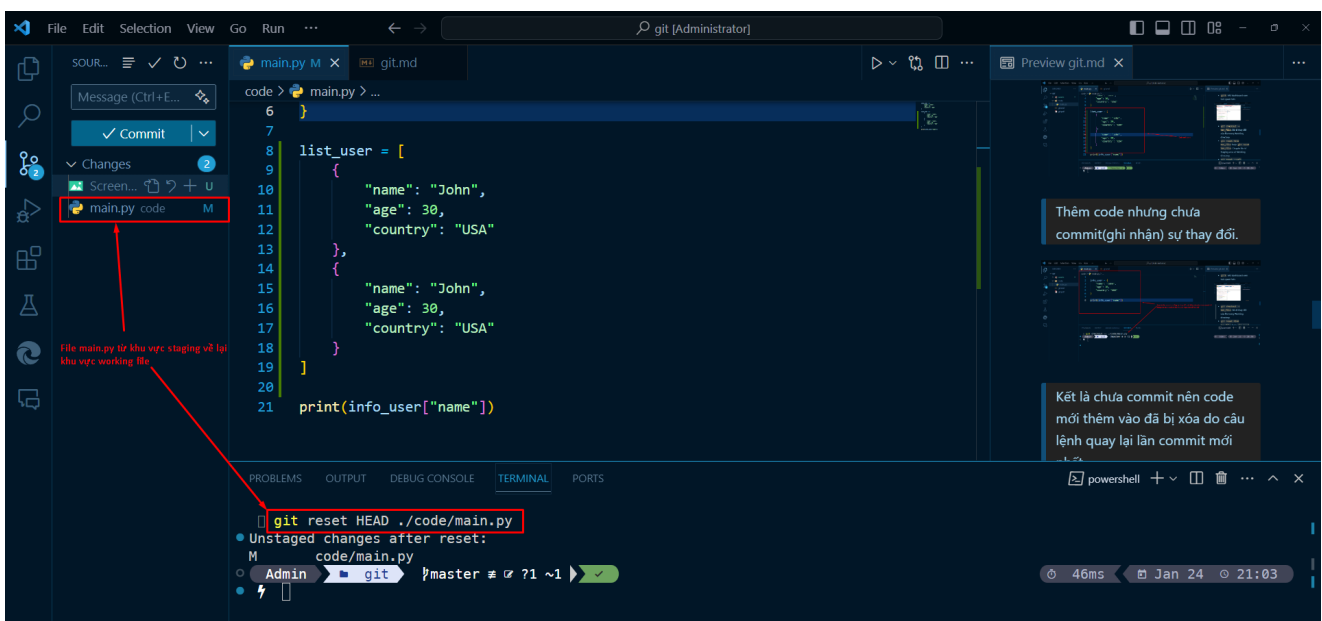


Kết là chưa commit nên code mới thêm vào đã bị xóa do câu lệnh quay lại lần commit mới nhất.

- `git reset HEAD ten_file` hoặc `git reset ten_file`: Chuyển file từ Staging area về Working directory.

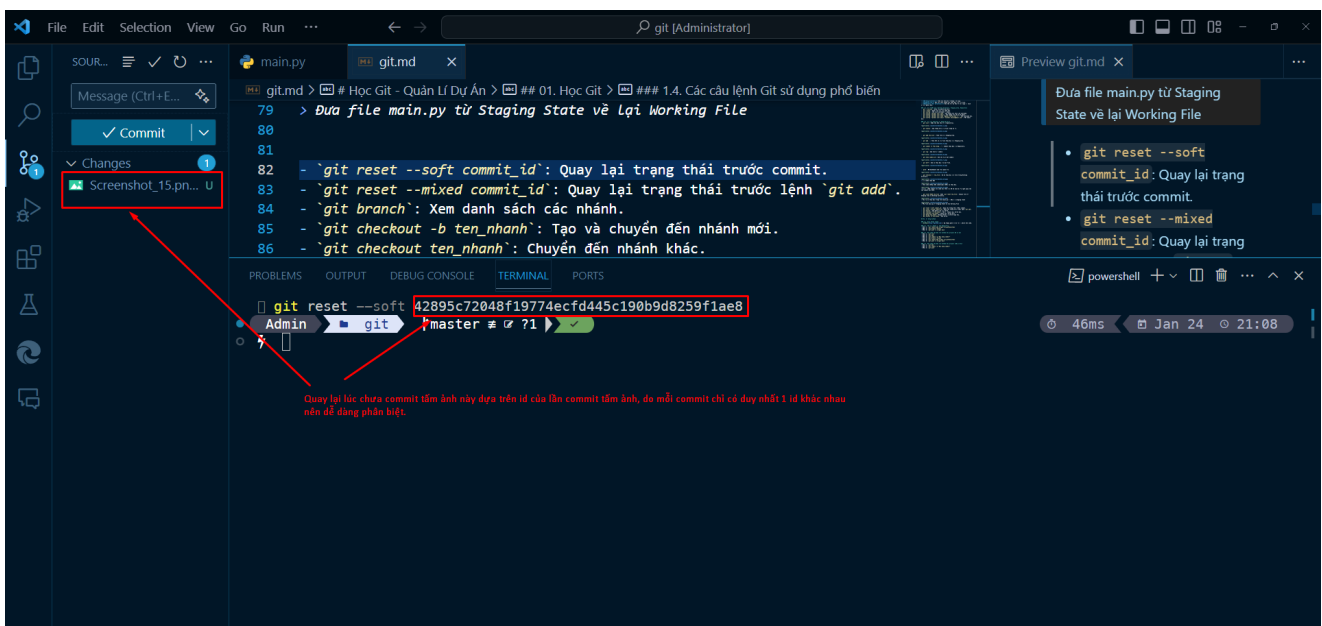
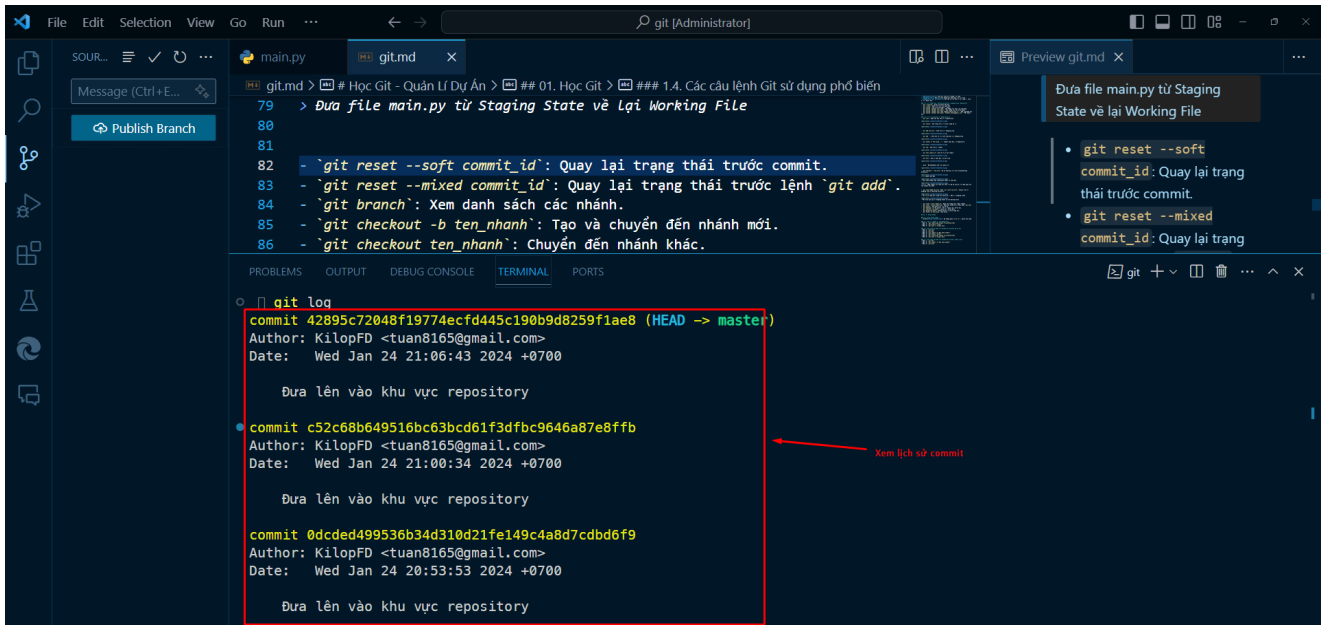


Tạo ra sự thay đổi trong file main.py và đưa vào staging state



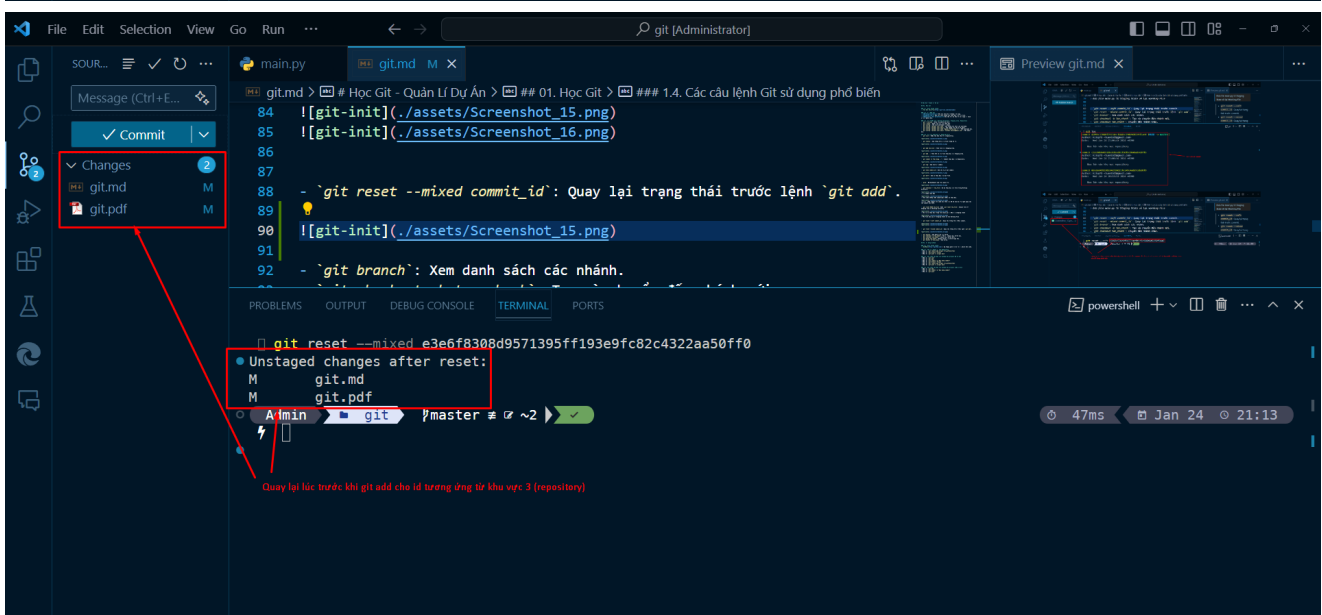
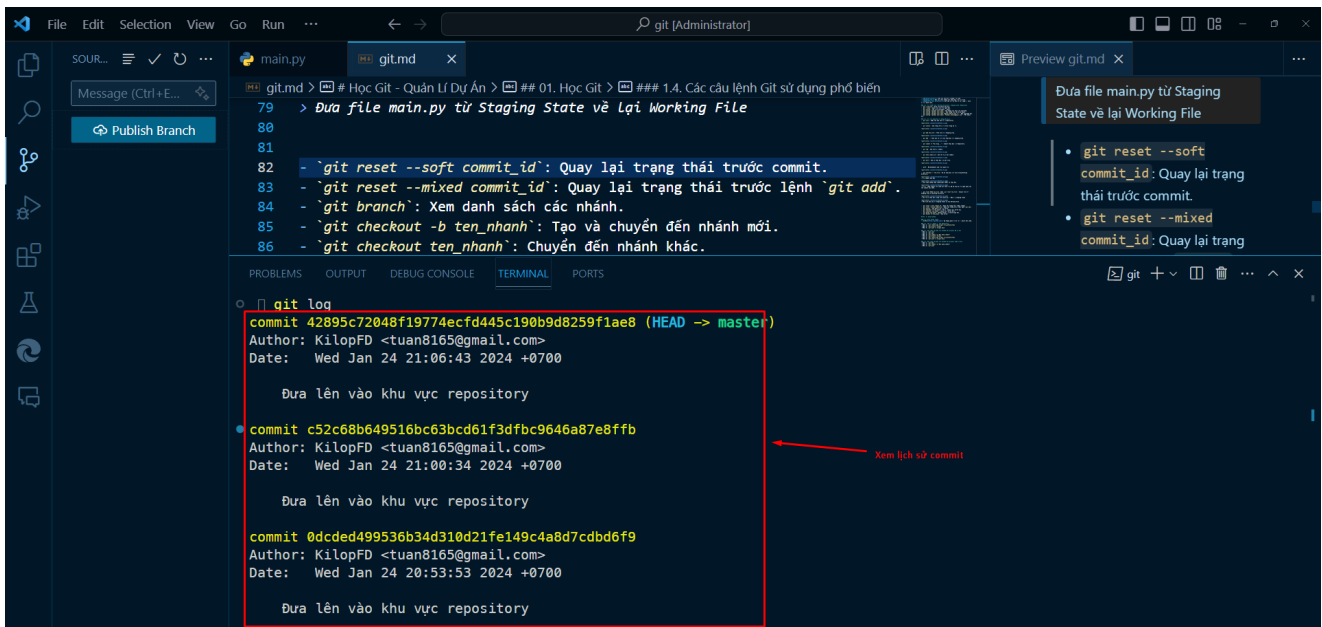
Đưa file main.py từ Staging State về lại Working File

- `git reset --soft commit_id`: Quay lại trạng thái trước commit.

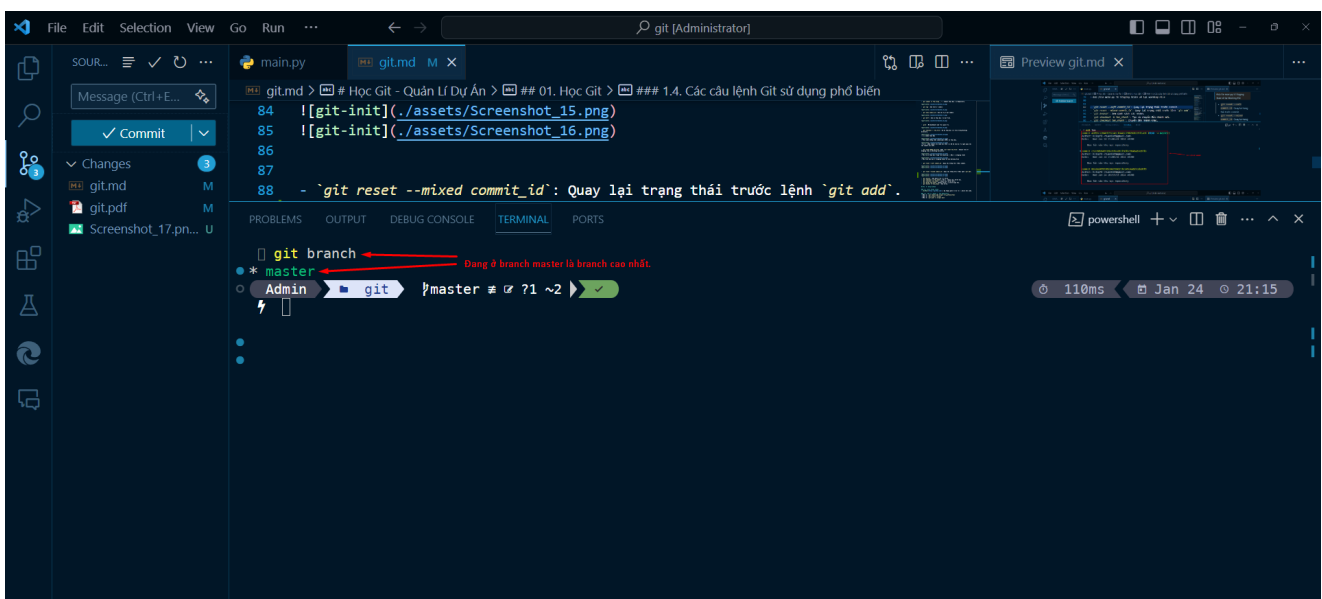


- `git reset --mixed commit_id`: Quay lại trạng thái trước lệnh `git add`.

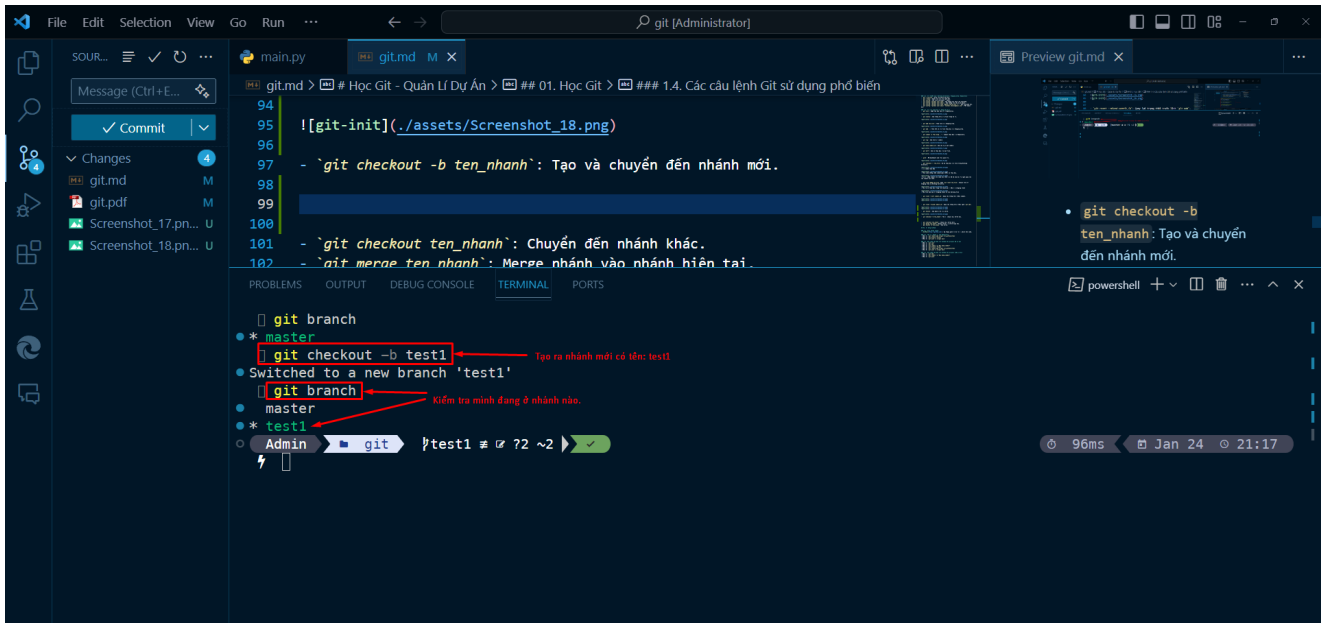




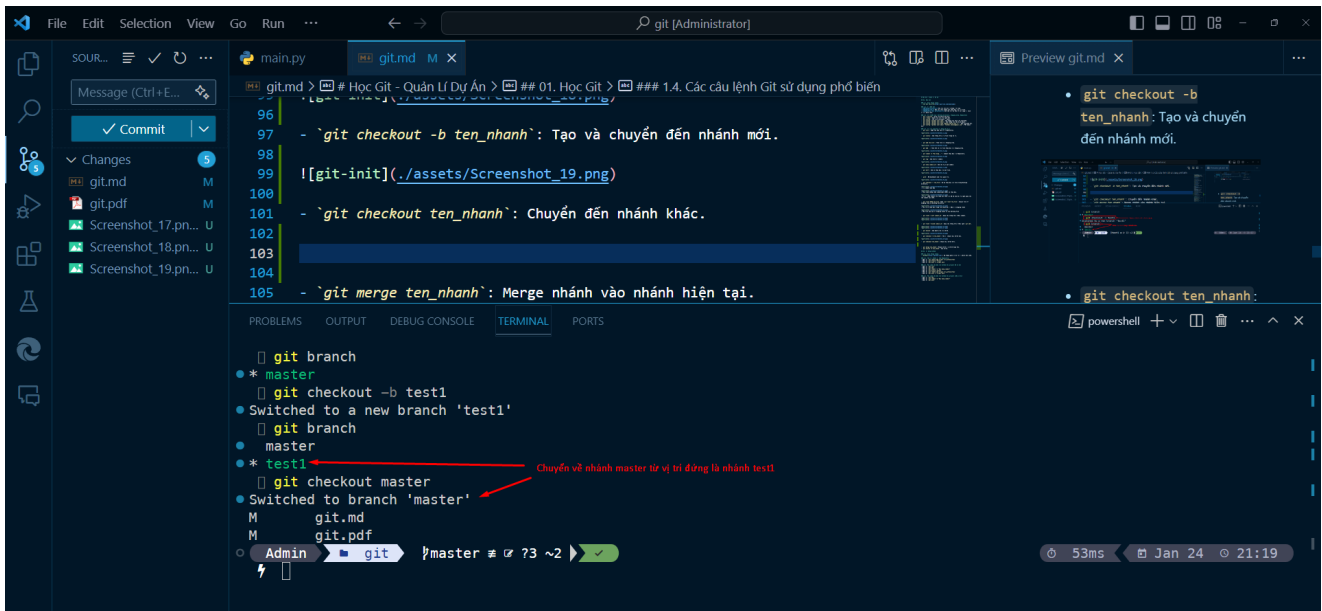
- **git branch**: Xem danh sách các nhánh.



- `git checkout -b ten_nhanh`: Tạo và chuyển đến nhánh mới.



- `git checkout ten_nhanh`: Chuyển đến nhánh khác.



- `git merge ten_nhanh`: Merge nhánh vào nhánh hiện tại.
- `git branch -D ten_nhanh`: Xóa nhánh.

## 02. Sử dụng GitHub

### 2.1. Giới thiệu chung

- [GitHub](#): Hệ thống quản lý dự án và phiên bản code.

### 2.2. Tạo tài khoản và tạo Repository

- Bước 1: `git remote add origin url_github_https`
- Bước 2: `git branch -M main`

- Bước 3: `git push -u origin main`

### 2.3. Đẩy code lần đầu lên GitHub khi project đã có Git

- Bước 1: `git init`
- Bước 2: `git add .`
- Bước 3: `git commit -m "Nội dung commit"`
- Bước 4: `git branch -M main`
- Bước 5: `git remote add origin url_github_https`
- Bước 6: `git push -u origin main`

### 2.4. Đẩy code lần đầu lên GitHub khi project chưa có Git

- Bước 1: `git add .`
- Bước 2: `git commit -m "Nội dung commit"`
- Bước 3: `git push`