

full end-to-end workflow summary, including all previous steps **with command-by-command output**, explanations, and a complete breakdown —

◇ Step 9: Create a New Branch for Feature (**cartpage**)

```
git branch cartpage
```

☑ **Result:** A new branch **cartpage** was created locally, based on current **master**.

◇ Step 10: Switch to **cartpage** Branch

```
git checkout cartpage
```

☑ **Result:** Switched to branch **cartpage**.

◇ Step 11: Add New Page Class File

You created: 📄 **src/main/java/com/qa/appName/pages/CartPage.java**

Check status:

```
git status
```

☑ **Result:** The file is untracked.

◇ Step 12: Stage and Commit the New File

```
git add src/main/java/com/qa/appName/pages/CartPage.java
git commit -m "added cart page code"
```

☑ **Result:** **CartPage.java** created with 7 insertions Commit ID: **4ffc4a2**

◇ Step 13: Push **cartpage** Branch to GitHub

```
git push origin cartpage
```

☑ **Result:** New branch **cartpage** pushed to GitHub 🌀 GitHub shows a message:

"Create a pull request for 'cartpage' on GitHub by visiting:
<https://github.com/green2043/POMSeriesAutomation/pull/new/cartpage>"

☑ **Important: PR is now ready to be created via GitHub UI**

◇ Step 14: Raise a Pull Request (PR) on GitHub

👤 Steps:

1. Go to GitHub → **Pull Requests** tab

2. Click **"New Pull Request"**

3. Select:

- **base:** `master`
- **compare:** `cartpage`

4. Add PR message:

"Naveen, please check CartPage code"

5. Click **Create pull request**

☑ **Result:** PR is created and visible to Naveen.

◇ Step 15: Reviewer Reviews PR

👤 Naveen opens the PR:

- Reviews code
- Adds comment:

"Please add locators, page functions, constructor etc."

- Clicks **Request changes**

☑ **Result:** GitHub sends you a notification.

◇ Step 16: You Address Review Comments Locally

```
git add src/main/java/com/qa/appName/pages/CartPage.java
git commit -m "updated the code as per Naveen's review comment"
git push origin cartpage
```

☑ **Result:** PR is auto-updated on GitHub with new commit (`752ac61`)


◇ Step 17: Reviewer Approves the PR

👤 Naveen:

- Clicks **Review changes**
 - Selects ☒ **Approve**
 - Clicks **Submit review**
- ☒ **Result:** PR is approved and ready to merge.
-

◇ Step 18: Merge PR via GitHub UI

🛠️ Steps:

1. Go to PR on GitHub
 2. Click ☒ **Merge pull request**
 3. Click **Confirm merge**
 4. (Optional) Click  **Delete branch** to clean up
- ☒ **Result:** The **cartpage** branch is merged into **master**.
-

◇ Step 19: Pull Latest **master** Locally

Switch to **master** and pull:

```
git checkout master
git pull origin master
```

- ☒ **Result:** Your local **master** now includes **CartPage.java** (18 insertions)
-

◇ Step 20: Team Members Also Pull the Latest Code

Other team members do:

```
git checkout master
git pull origin master
```

- ☒ **Result:** Everyone’s local **master** is now up-to-date.
-

☒ Final Summary Table:

Action	Who Performs It	Tool Used	Notes
--------	-----------------	-----------	-------

Action	Who Performs It	Tool Used	Notes
Create feature branch	You	Terminal	e.g., <code>git branch cartpage</code>
Write code + push to GitHub	You	Terminal	Push branch to remote
Create PR	You	GitHub Web UI	Compare <code>cartpage</code> → <code>master</code> , add description
Review code + add comments	Reviewer	GitHub Web UI	Comments like “add locators”
Make changes & push again	You	Terminal	GitHub auto-updates the PR
Approve PR	Reviewer	GitHub Web UI	Approves changes
Merge PR	Reviewer/You	GitHub Web UI	Merge button → Confirm
Delete feature branch	Optional	GitHub Web UI	For cleanup
Pull updated <code>master</code>	You & Team	Terminal	Keep everyone in sync