**SCRIPT MODULE 5**

By the end of this module, you will understand how GitHub enables **team collaboration**, **task management**, **code reviews**, and **project tracking** using **Pull Requests, Issues, Labels, and GitHub Projects**.

**🔹 1. GitHub Issues, Labels, Assigning & Comments**

**➤ Step-by-Step:**

**✅ Step 1: Create an Issue**

1. Go to your repo on GitHub.
2. Click on **"Issues" > "New Issue"**.
3. Add a **title** like: Improve project description in README.
4. Add a **description** for clarity.
5. Click **Submit new issue**.

“Think of Issues like digital sticky notes that help your team know *what needs to be done*. It could be a bug, improvement, or a new feature request.”

**✅ Step 2: Add Labels to Issues**

1. Open your issue.
2. On the right-hand side, click **"Labels"**.
3. Select labels like: enhancement, documentation, or create custom ones.

“Labels help categorize work. For example, bug, enhancement, urgent — just like putting colored tags on physical files.”

**✅ Step 3: Assign the Issue**

1. Click **"Assignees"** on the right.
2. Choose a contributor (yourself if working solo).

“Assigning issues makes it clear *who is responsible* for the task — avoiding confusion in teams.”

**✅ Step 4: Add Comments**

1. Go to the issue and add a comment like:  
   "I’ll work on this and submit a PR shortly."

“Comments keep the team updated — like leaving notes on a task board.”

**🔹 2. Create a Branch, Make a Commit, Link to the Issue**

**➤ Step-by-Step:**

**✅ Step 5: Create a Branch for the Issue**

In terminal or VS Code:

Bash

git clone <https://github.com/SoniaRajput/myrepo.git>

cd my repo

code .

git checkout -b issue-1-update-readme

**✅ Step 6: Edit the README and Commit**

Update something in readme.md, then:

bash

git add .

git commit -m "Fixes #1: Updated project description"

git push origin issue-1-update-readme

“By writing Fixes #1 in the commit message, GitHub automatically links this commit to the issue and closes it when merged.”

**🔹 3. Pull Requests: Review, Approve, Merge**

**➤ Step-by-Step:**

**✅ Step 7: Open a Pull Request (PR)**

1. Go to GitHub repo → **Pull Requests > New Pull Request**.
2. Choose base as main and compare with issue-1-update-readme.
3. Add a title & description. Click **Create Pull Request**.

“Pull Requests are how we propose and review changes before merging into the main project. This is where teamwork and feedback happen.”

**✅ Step 8: Review the PR (with Comments)**

1. As a reviewer, click **"Files Changed"**.
2. Click the + icon to add comments.
3. Click **Review changes > Approve** or **Request changes**.

“Reviewers can approve, comment, or reject PRs — just like peer reviews in writing or design.”

**✅ Step 9: Merge the PR**

1. After approval, click **Merge Pull Request**.
2. Confirm merge and delete the branch.

You’ll notice **Issue #1 is now closed automatically.**

**🔹 4. GitHub Projects (Boards + Automation)**

**➤ Step-by-Step:**

**✅ Step 10: Create a GitHub Project**

1. Go to your repo > **Projects > New Project**.
2. Choose **Board** layout (Kanban style).
3. Add columns like: To do, In Progress, Done.

“Think of this like a digital whiteboard to organize tasks visually — just like Trello or Jira.”

A project is an adaptable spreadsheet, task board and roadmap that integrates with your issues and pull requests on Github to help you plan and track your work effectively. We can create and customize multiple views by filtering, sorting, grouping your issues and pull requests, visualize your work

**✅ Step 11: Add Issues to the Project Board**

1. Click + under the “To do” column.
2. Add the existing issues to the board.

**✅ Step 12: Enable Automation**

1. Go to “Settings” > “Automation”.
2. Set it to move issues to **"Done"** once PR is merged.

“Automation reduces manual tracking. Issues move on the board based on pull request activity — making workflows smoother.”

**🔹 5. Profile-Level README & Showcasing Work**

**✅ Step 13: Create Profile README**

1. Go to GitHub and create a new repo with **your GitHub username exactly**.
   * Example: SoniaRajput/SoniaRajput
2. Add a file: README.md
3. Add your bio, badges, projects, and GitHub stats.

Example:

markdown

- Hi, I'm Sonia Rajput!

- Learning Git, Python & AI

- Projects: [Resume Builder](#), [AI Toy Recommender](#)

“Your profile README is your personal brand page on GitHub — a great way to showcase your skills and projects to recruiters and collaborators.”

**Summary of Hands-On Activities:**

| **Task** | **Tools Involved** |
| --- | --- |
| Create issue, label, assign | GitHub UI |
| Link issue in commit | Git CLI |
| Pull request & code review | GitHub UI |
| Create & manage Project Board | GitHub Projects |
| Profile README setup | GitHub Repo |