

1. **Install and verify Git**

- * Use Homebrew to install Git.
- * Confirm installation by checking the Git version.

```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\adribers>git -v
git version 2.50.1.windows.1
```

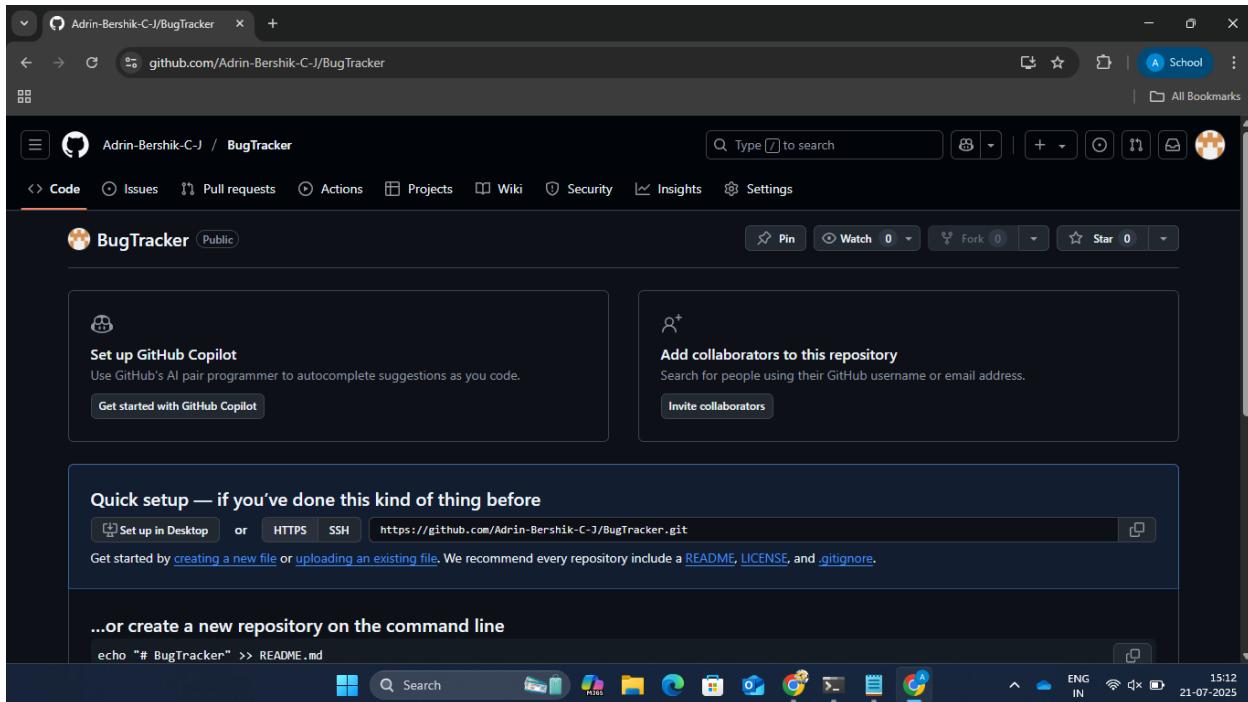
2. **Configure Git**

- * Set your global Git username and email to your real or test identity.

```
C:\Users\adribers>git config --global user.name "Adrin-Bershik"
C:\Users\adribers>git config --global user.email "cjabershik@gmail.com"
C:\Users\adribers>
```

3. **Create GitHub Repository**

- * Go to GitHub and create a new repo named: `BugTracker`
- * Do **not** initialize with README or license.
- * Keep the repo **private** or **public** as needed.



4. **Clone the Repository**

- * Clone the repo to your local system using HTTPS.
- * Use your **GitHub token** to authenticate.

```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop> git clone https://github.com/Adrin-Bershik-C-J/BugTracker.git
Cloning into 'BugTracker'...
warning: You appear to have cloned an empty repository.
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop>
```

5. **Verify `.`git` folder**

- * Inside the cloned folder, use the terminal to check if `.`git` is present.

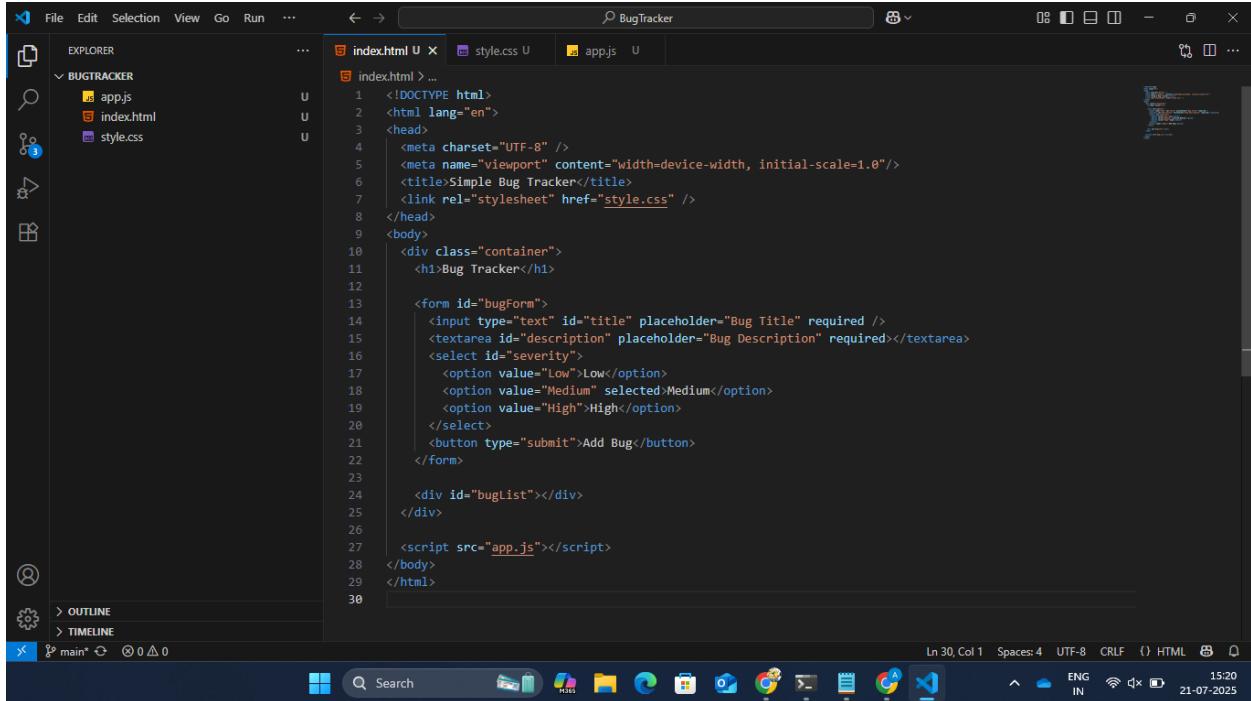
```
adribers@7ZSPHG3 MINGW64 ~/OneDrive - EPS US, LLC/Desktop/BugTracker (main)
$ ls -a
● ./ ../.git/

adribers@7ZSPHG3 MINGW64 ~/OneDrive - EPS US, LLC/Desktop/BugTracker (main)
○ $ |
```

6. **Create Basic Files**

- * Create the following in the root of your repo:

- * `README.md`
- * `index.html` (basic BugTracker layout)
- * `style.css` (simple styles)
- * `app.js` or `main.java` or `Program.cs` (based on backend choice)



```

File Edit Selection View Go Run ... ← → ⌂ BugTracker
EXPLORER BUGTRACKER
  app.js
  index.html
  style.css
index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6    <title>Simple Bug Tracker</title>
7    <link rel="stylesheet" href="style.css" />
8  </head>
9  <body>
10 <div class="container">
11   <h1>Bug Tracker</h1>
12
13   <form id="bugForm">
14     <input type="text" id="title" placeholder="Bug Title" required />
15     <textarea id="description" placeholder="Bug Description" required></textarea>
16     <select id="severity">
17       <option value="Low">Low</option>
18       <option value="Medium" selected>Medium</option>
19       <option value="High">High</option>
20     </select>
21     <button type="submit">Add Bug</button>
22   </form>
23
24   <div id="bugList"></div>
25 </div>
26
27 <script src="app.js"></script>
28 </body>
29 </html>
30

```

Ln 30, Col 1 Spaces: 4 UTF-8 CRLF ⓘ HTML ⚙ 15:20 ENG IN 21-07-2025

7. **Track Your Work**

* Add and commit the above files using **conventional commits**:

- * `feat: initial layout for BugTracker`
- * `style: basic styles for homepage`

```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git commit -m "feat: initial layout for BugTracker"
[main (root-commit) 7e82715] feat: initial layout for BugTracker
 3 files changed, 163 insertions(+)
  create mode 100644 app.js
  create mode 100644 index.html
  create mode 100644 style.css
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git add style.css
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git commit -m "style: basic styles for homepage"
On branch main
Your branch is based on 'origin/main', but the upstream is gone.
(use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.75 KiB | 149.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Adrin-Bershik-C-J/BugTracker.git
 * [new branch]      main -> main
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker>
```

8. **Push Changes to GitHub**

- * Push to the `main` branch using GitHub token if needed.

```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.75 KiB | 149.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Adrin-Bershik-C-J/BugTracker.git
 * [new branch]      main -> main
```

9. **Edit File Remotely (Conflict Setup)**

- * On GitHub, open `index.html` and change an `

` tag.
- * Commit the change with: `fix: updated heading remotely`

BugTracker / index.html

Adrin-Bershik-C-J fix: updated heading remotely

Code Blame 29 lines (26 loc) · 812 Bytes GitHub Copilot Code 55% faster with GitHub Copilot

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Simple Bug Tracker</title>
7   <link rel="stylesheet" href="style.css" />
8 </head>
9 <body>
10 <div class="container">
11   <h1>Bug Tracker Application</h1>
12 
```

10. **Change Same File Locally**

* Locally, change the **same line** in `index.html`.

```
11 |   <h1>Bug Tracker Super Application</h1>
---
```

11. **Trigger Merge Conflict**

* Try to push your local changes without pulling first.

* Observe the **merge conflict error**.

```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git push
To https://github.com/Adrin-Bershik-C-J/BugTracker.git
! [rejected]      main -> main (fetch first)
error: failed to push some refs to 'https://github.com/Adrin-Bershik-C-J/BugTracker.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker>
```

12. **Resolve Conflict**

* Pull from GitHub (`git pull origin main`)

- * Open conflict in VS Code and manually resolve it.
- * Commit the resolved file: `fix: resolved heading conflict`

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Simple Bug Tracker</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
<div class="container">
    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
    <===== HEAD (Current Change)>
    <h1>Bug Tracker Super Application</h1>
    =====
    <h1>Bug Tracker Application</h1>
    <===== 8384fdacd7f78b6157iae341fa866c82005793b6 (Incoming Change)>
    <form id="bugForm">
        <input type="text" id="title" placeholder="Bug Title" required />
        <textarea id="description" placeholder="Bug Description" required></textarea>
        <select id="severity">
            <option value="Low">Low</option>
            <option value="Medium" selected>Medium</option>
            <option value="High">High</option>
        </select>
        <button type="submit">Add Bug</button>
    </form>
    <div id="bugList"></div>
</div>
<script src="app.js"></script>

```

13. **Branch for Frontend**

- * Create a new branch `frontend`
- * Modify/add to `index.html` and `style.css`
- * Commit your changes with a conventional commit.
- * Push the branch and create a **pull request** to `main`.

```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git checkout -b frontend
Switched to a new branch 'frontend'
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker>
```

14. **Branch for Backend**

- * Switch to `main`, then create a new branch `backend`
- * Create/modify your backend logic file (`main.java`, `Program.cs`, etc.)
- * Commit and push.
- * Raise a PR to `main`.

```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git commit -m "feat:updated new backend feature"
[backend 0ccf631] feat:updated new backend feature
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git push origin backend
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 372 bytes | 74.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'backend' on GitHub by visiting:
remote:   https://github.com/Adrin-Bershik-C-J/BugTracker/pull/new/backend
remote:
To https://github.com/Adrin-Bershik-C-J/BugTracker.git
 * [new branch]      backend -> backend
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker>
```

15. **Create `\.gitignore`**

* Add the following:

```
```
node_modules/
*.log
.env
```

```

* Commit it: `chore: add .gitignore file`

16. **Stash Scenario**

- * Edit any file (e.g., `index.html`) but **don't commit**
- * Try pulling new changes from GitHub — Git should block you
- * Use `git stash`, then `git pull`, then `git stash pop`
- * Resolve any conflict if it appears

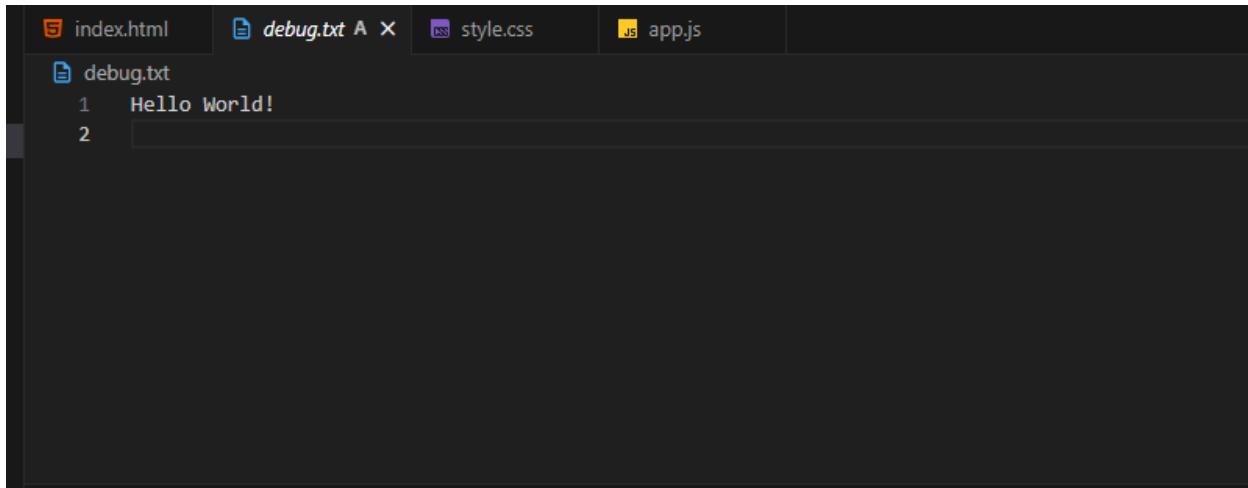
```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1020 bytes | 102.00 KiB/s, done.
From https://github.com/Adrin-Bershik-C-J/BugTracker
 * branch            main      -> FETCH_HEAD
   76bdb05..0af9a07  main      -> origin/main
Updating 76bdb05..0af9a07
error: Your local changes to the following files would be overwritten by merge:
      index.html
Please commit your changes or stash them before you merge.
Aborting
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker>
```

The screenshot shows the Visual Studio Code interface. On the left, the Source Control panel displays a commit message: "Message (Ctrl+Enter to commit on ...)" with a "Commit" button. Below it, the "Changes" section shows a staged file named "index.html". The main area is a code editor displaying the content of "index.html". The code is an HTML document with a form for adding bugs and a script tag for "app.js". The status bar at the bottom shows the author as "Adrin-Bershik (41 minutes ago)", the file path as "main+", and the date and time as "21-07-2025 16:12".

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Simple Bug Tracker</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
<div class="container">
<h1>Bug Tracker Application</h1>
<h1>On frontend branch</h1>
<!-- Change for stash -->
<form id="bugForm">
<input type="text" id="title" placeholder="Bug Title" required />
<textarea id="description" placeholder="Bug Description" required></textarea>
<select id="severity">
<option value="Low">Low</option>
<option value="Medium" selected>Medium</option>
<option value="High">High</option>
</select>
<button type="submit">Add Bug</button>
</form>
<div id="bugList"></div>
</div>
<script src="app.js"></script>
</body>
</html>
```

17. **Reset and Restore**

- * Add and stage a new dummy file (e.g., `debug.txt`)
- * Then **unstage it** using `git reset`
- * Restore any accidental local change using `git restore`



index.html debug.txt style.css app.js

debug.txt

1 Hello World!

2

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git reset debug.txt
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git status
On branch main
Your branch is up to date with 'origin/main'.

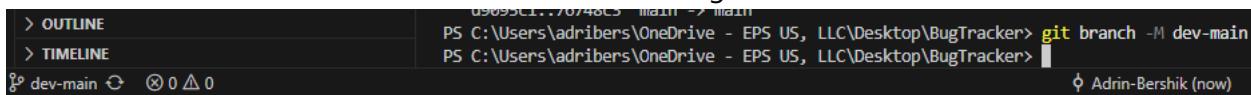
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    debug.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> echo "Hello World!" > debug.txt
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    debug.txt
```

18. **Rename Branch**

* Rename the current `main` branch to `dev-main` using Git



> OUTLINE
> TIMELINE
dev-main ⌂ ⌂ 0 △ 0

```
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker> git branch -M dev-main
PS C:\Users\adribers\OneDrive - EPS US, LLC\Desktop\BugTracker>
```

∅ Adrin-Bershik (now)

19. **View Commit History**

* Use `git log` and take a screenshot showing at least 5 commits with:

- * different commit types ('feat', 'fix', 'chore', etc.)

- * merge conflict resolved commit

```
:  
  
commit 76bdb051f92ef6571da505fb06a4cbd90fb3ee5  
Merge: a6add81 0ccf631  
Author: Adrin-Bershik-C-J <cjabershik@gmail.com>  
Date: Mon Jul 21 15:37:14 2025 +0530  
  
    Merge pull request #2 from Adrin-Bershik-C-J/backend  
  
        feat:updated new backend feature  
  
commit 0ccf6319d7a09aa388319f3c2b6d3394e3e5321c (origin/backend, backend)  
Author: Adrin-Bershik <cjabershik@gmail.com>  
Date: Mon Jul 21 15:35:26 2025 +0530  
  
        feat:updated new backend feature  
  
commit a6add813661bab26c9d032ece4fa62dd6622b242  
Merge: a052ca1 eb8de3a  
Author: Adrin-Bershik-C-J <cjabershik@gmail.com>  
Date: Mon Jul 21 15:32:43 2025 +0530  
  
    Merge pull request #1 from Adrin-Bershik-C-J/frontend  
  
        feat:new line added  
  
commit eb8de3a6857746f6051d4f4ef2d00c0d3644b767 (origin/frontend, frontend)  
Author: Adrin-Bershik <cjabershik@gmail.com>  
Date: Mon Jul 21 15:31:24 2025 +0530  
  
        feat:new line added  
  
commit a052ca1572995f1ca0ab7299d22bc3b2ed700e4f  
Merge: 6932dea 8384fda  
Author: Adrin-Bershik <cjabershik@gmail.com>  
Date: Mon Jul 21 15:29:09 2025 +0530
```

📁 **Final Submission Requirements**

- * A GitHub repo named `BugTracker` with:
 - * At least 2 branches (`frontend`, `backend`)
 - * At least 1 merged pull request with conflict resolution
 - * ` `.gitignore` file committed
 - * A clean commit history using conventional commits
- * Screenshot of:

- * `git status` showing a merge conflict
- * `git log` showing your commit history