

☐ Creating Your First Pull Request: The VS Code Way ☐

Welcome to the team! We're excited to have you contributing. This guide is for anyone who is new to Git or prefers to use the visual tools inside VS Code instead of the command line. No scary commands, just friendly buttons!

We use **Pull Requests** (PRs) to review code before it gets added to our main `master` branch. Think of it as asking a teammate, "Hey, can you double-check my work before we make it official?" It's a crucial part of writing great software together.

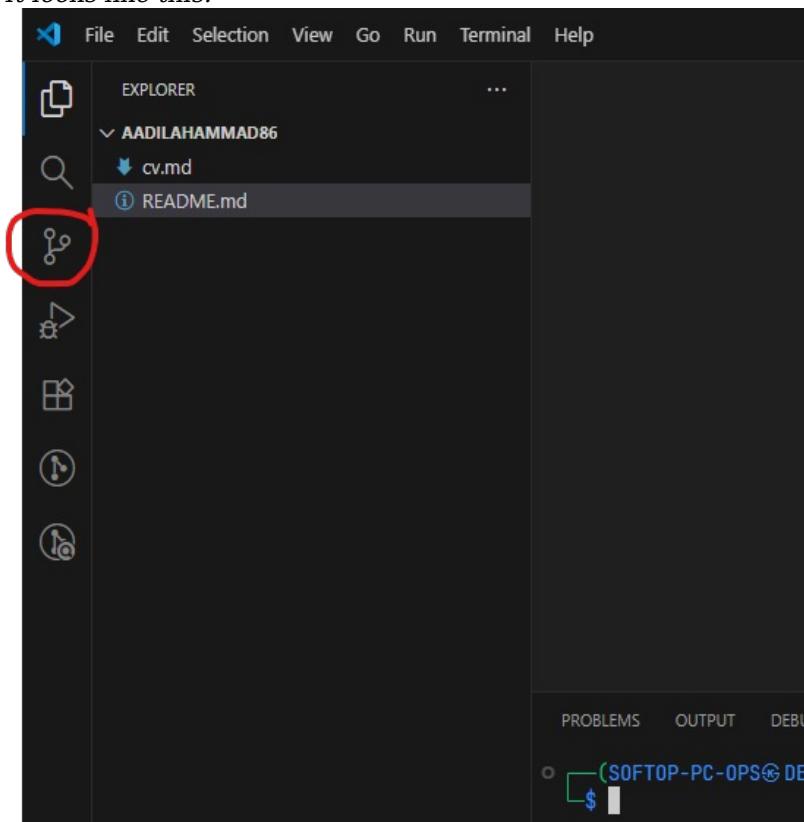
Let's walk through the entire process, step-by-step.

The Workflow: From Idea to Pull Request

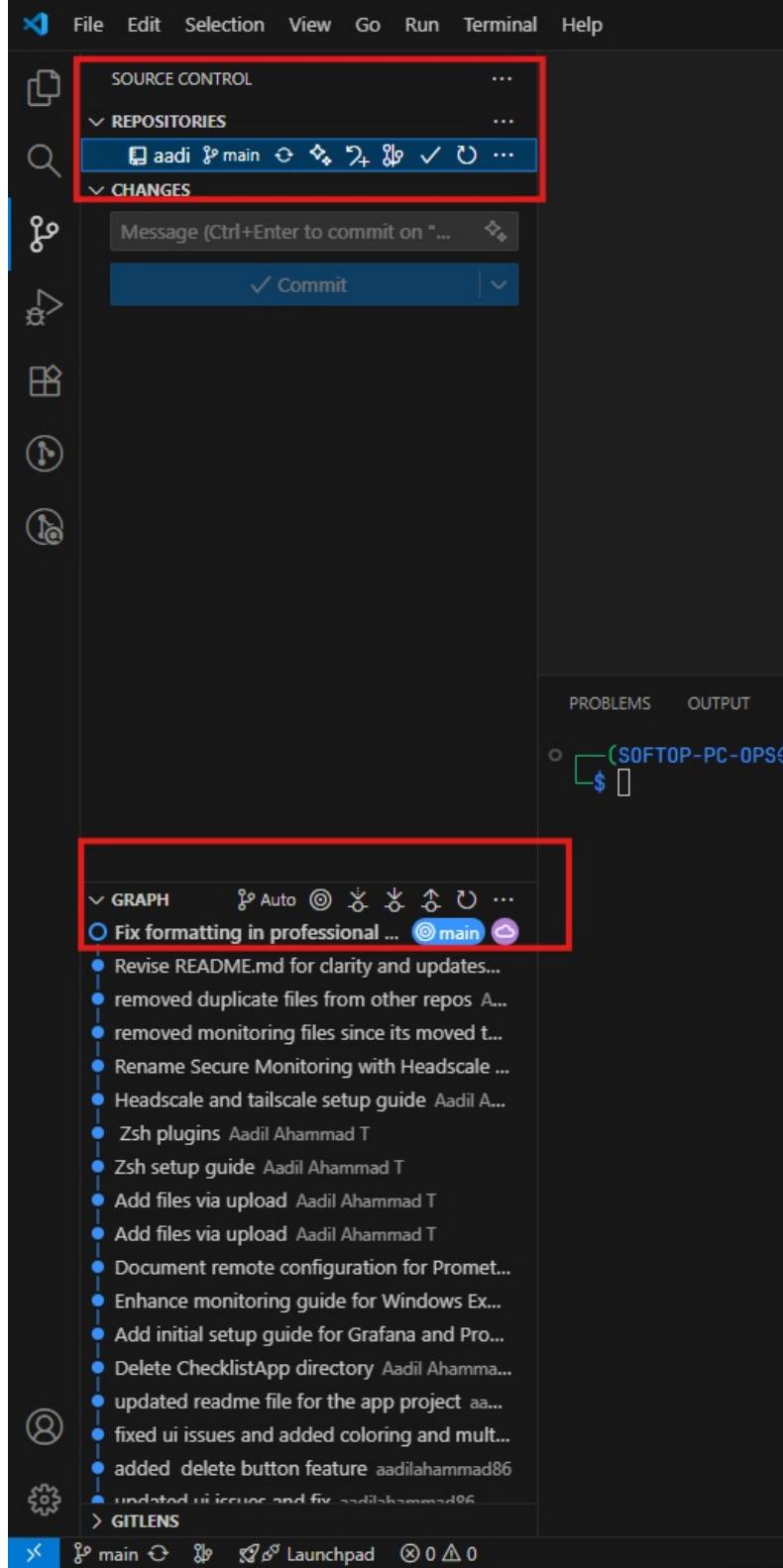
☐ Step 1: Get the Latest Code (Sync Up!)

Before you start writing any code, you need to make sure you have the most recent version of the project.

1. Open the **Source Control** view by clicking its icon on the left-side activity bar.
 - *It looks like this:*



2. Click the ... menu at the top of the panel, go to **Pull**, **Push**, and select **Sync**. This downloads any updates your teammates have made and ensures you're starting from the right place.

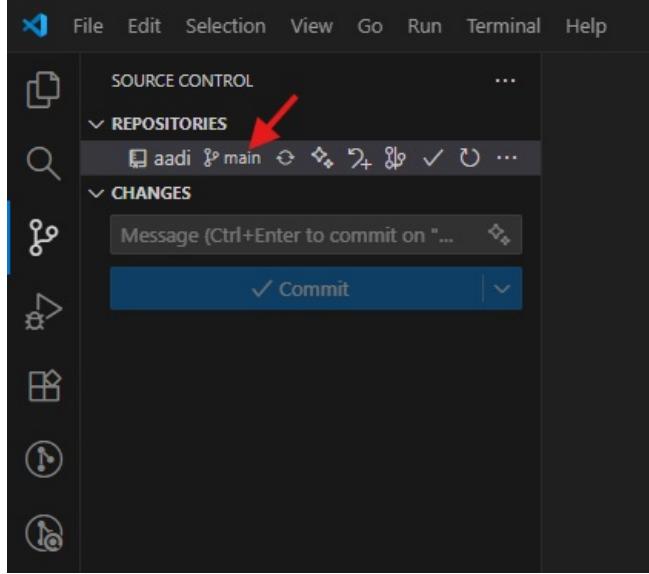


Why we do this: Starting with the latest code prevents a lot of headaches and messy conflicts later on!

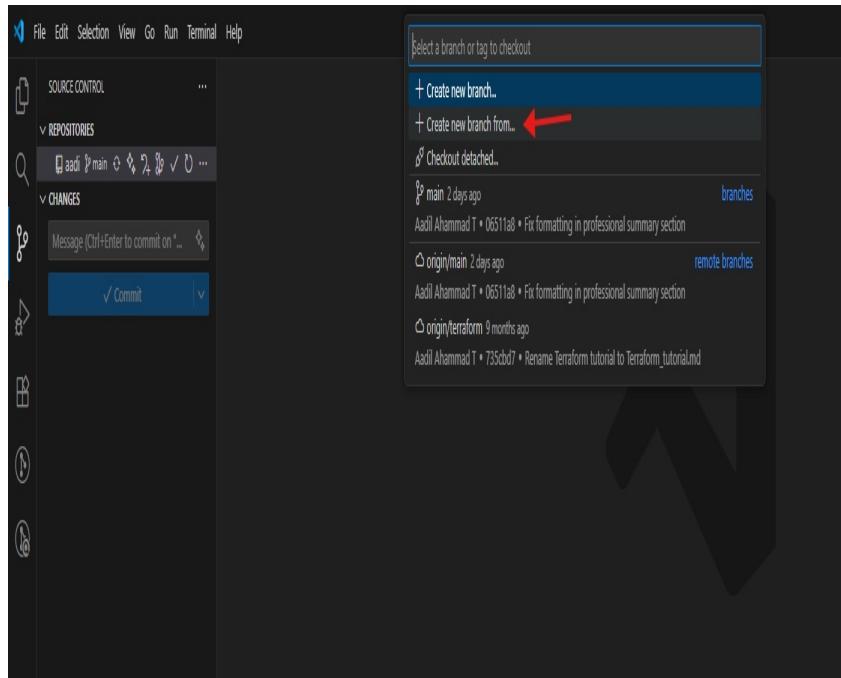
□ Step 2: Create Your Own Branch

You should **never** work directly on the `main` or `master` branch. Instead, you'll create your own copy (a "branch") to work in safely.

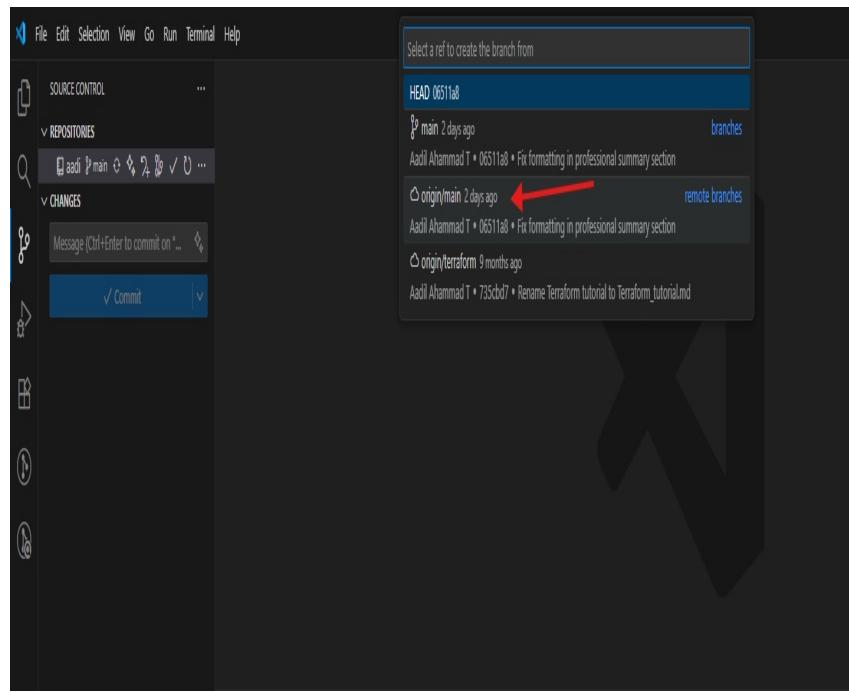
1. In the **Source Control** view, click on the current branch name



2. A menu will appear at the top. Click on **+ Create new branch from....**

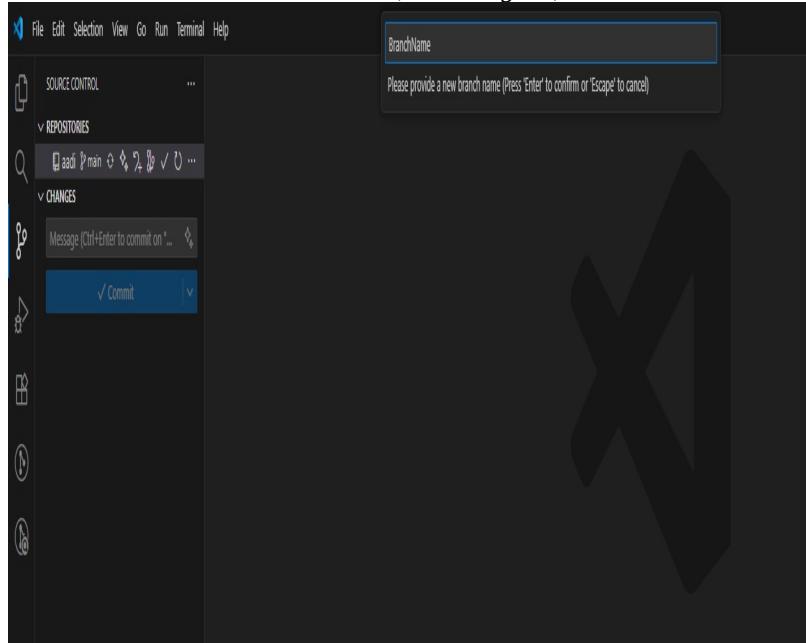


3. It will ask you to "Select a ref to create the branch from". **This is important!** Always choose the latest version of the main project branch, which is usually `origin/main` or `origin/master`.



- Finally, give your branch a descriptive name and press Enter. Good names are like:

- feat/add-user-login-page (for a new feature)
- fix/incorrect-tax-calculation (for a bug fix)



You are now on your new, safe branch!

Why we do this: Your branch is your personal playground. If you make a mistake, it won't affect the main project. It keeps our main codebase clean and working at all times.

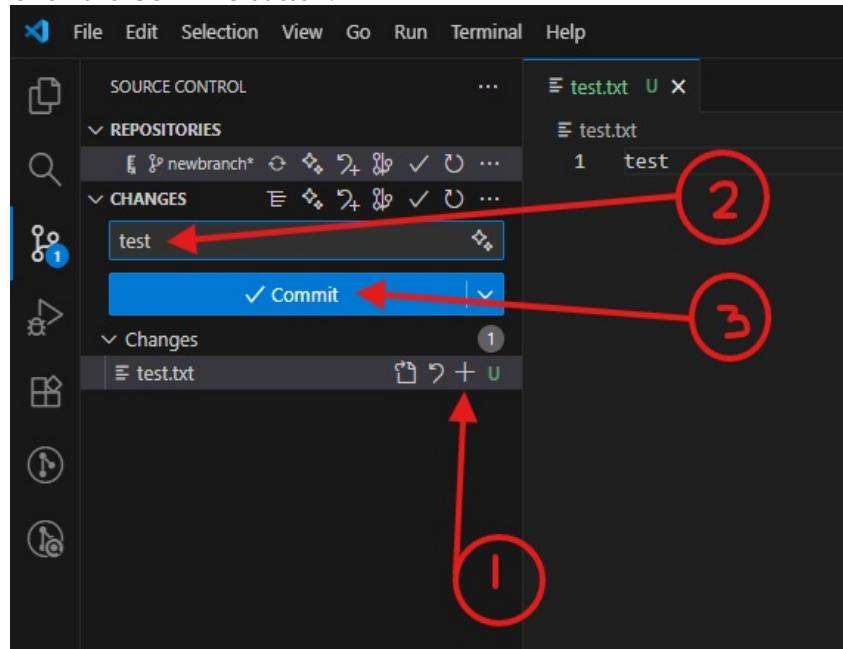
Step 3: Make Your Code Changes

This is the fun part! Go ahead and create new files, edit existing ones, and build your feature or fix your bug. VS Code will automatically track everything you change and list them in the Source Control panel.

staging & committing Step 4: Stage and Commit Your Changes

A “commit” is a snapshot of your changes. It’s like sealing a box and putting a clear label on it describing what’s inside.

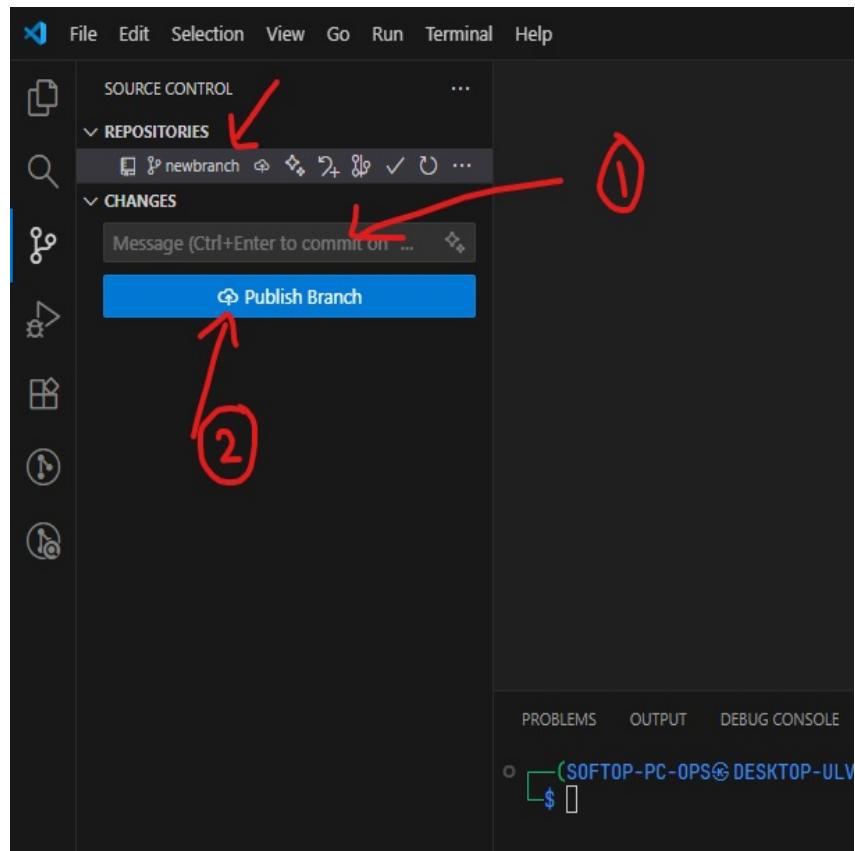
1. In the Source Control view, look at the files under “Changes”. Click the + icon next to each file you want to include in this snapshot. This moves them to “Staged Changes”.
2. In the “Message” box, write a clear, concise **commit message**.
 - **Good Example:** feat: Implement user password reset form
 - **Bad Example:** stuff or more changes
3. Click the **Commit** button.



□ Step 5: Push Your Branch (Share Your Work)

Your new branch and its commits are still only on your computer. You need to “push” it to the remote repository (Azure DevOps in our case) so others can see it.

1. After you commit, a “**Publish Branch**” button will appear. Click it!

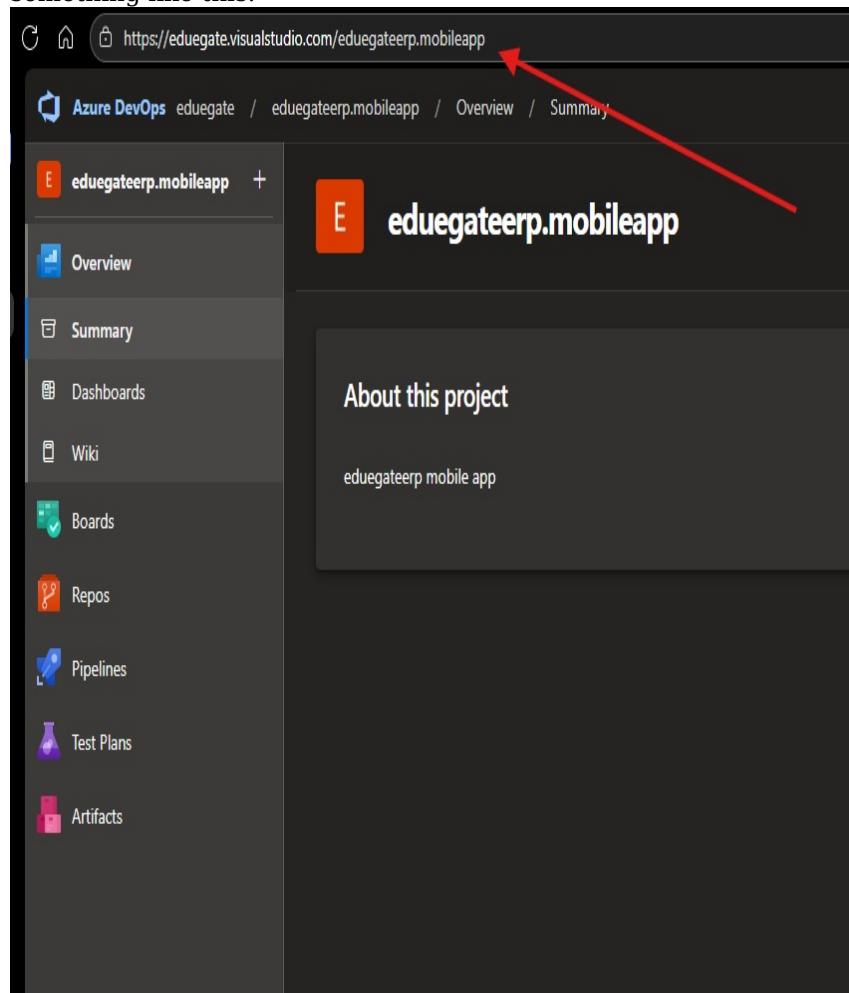


This uploads your branch to the server. You can confirm this by looking at the GRAPH view, where you'll see your new branch has split off from `main` and now exists on the `origin` (the server).

□ Step 6: Create the Pull Request in Azure DevOps

The final step! This is where you formally ask for your changes to be reviewed and merged.

1. Go to your project's repository in Azure DevOps. The URL will look something like this:

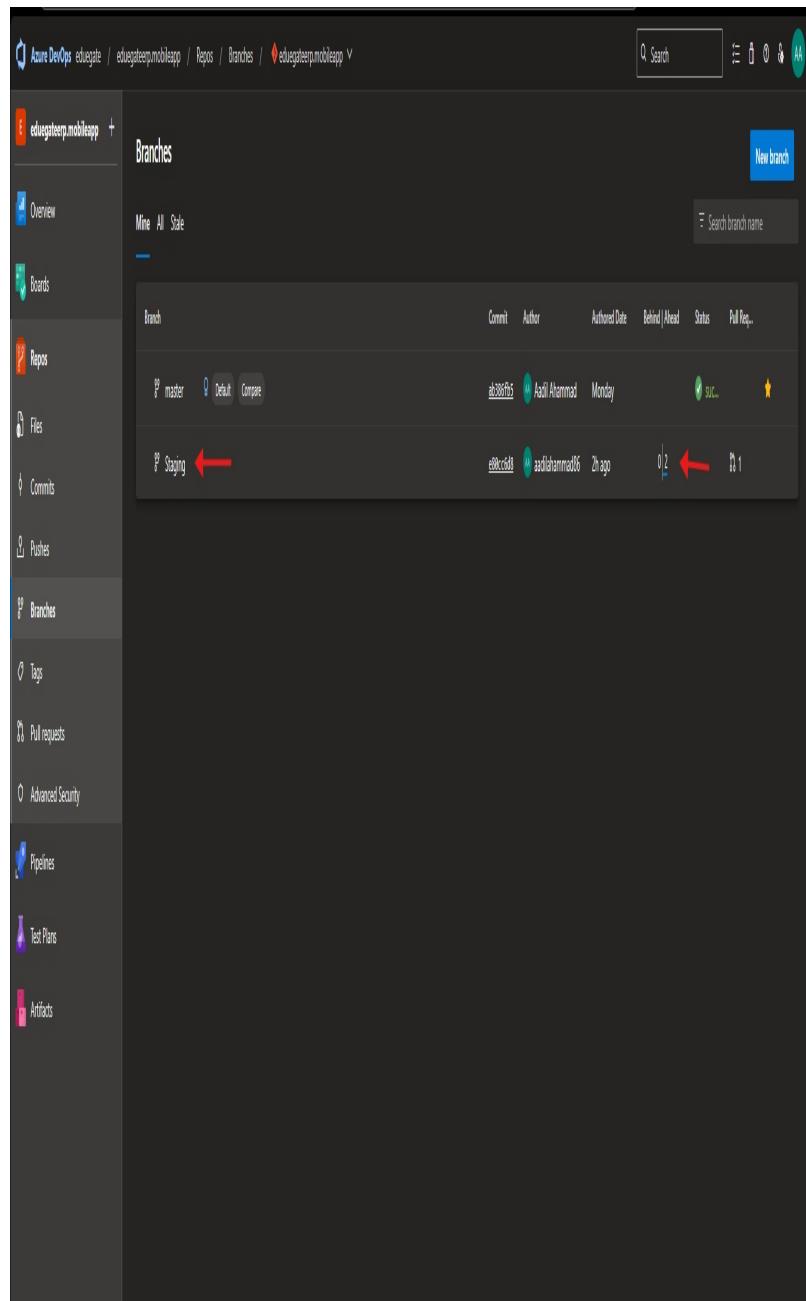


2. In the left-hand navigation pane, click on **Repos**.

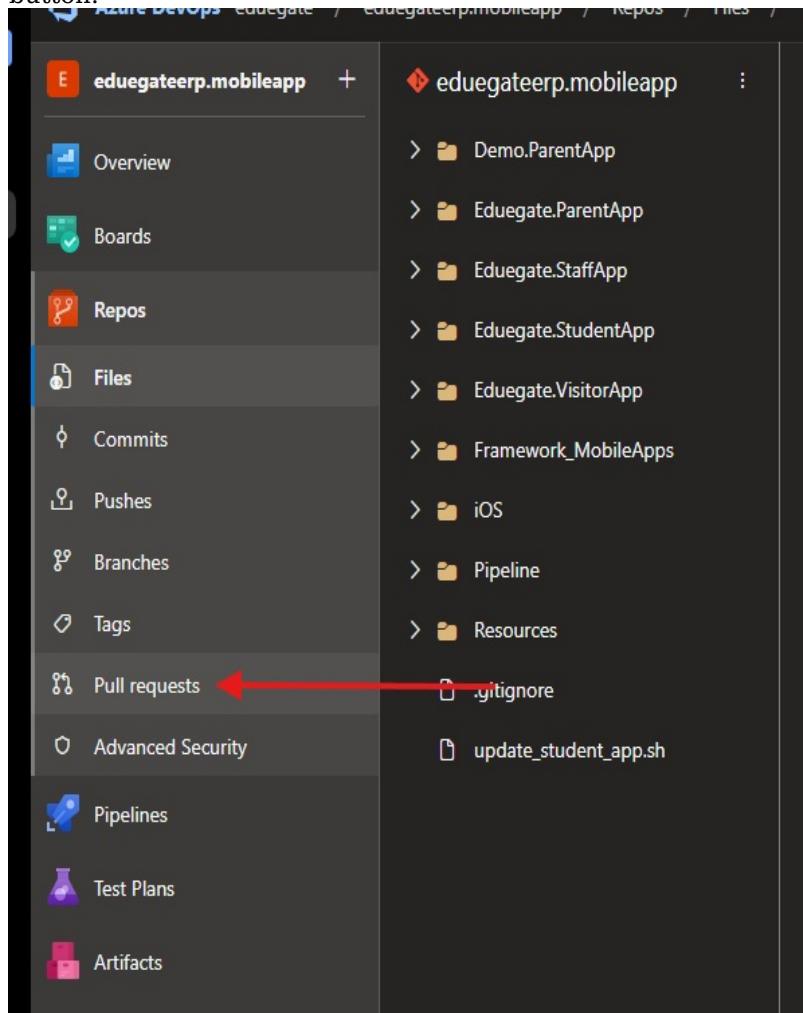
The screenshot shows the Azure DevOps interface for a project named 'eduegateerp.mobileapp'. The left navigation pane is visible, listing various repository-related options: Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Advanced Security, Pipelines, Test Plans, and Artifacts. Two items in the list are highlighted with red arrows pointing to them: 'Repos' and 'Branches'. The main content area displays a list of repository branches under the heading 'eduegateerp.mobileapp'. The branches listed are: Demo.ParentApp, Eduegate.ParentApp, Eduegate.StaffApp, Eduegate.StudentApp, Eduegate.VisitorApp, Framework_MobileApps, iOS, Pipeline, Resources, .gitignore, and update_student_app.sh. The 'Branches' section also includes an 'Ahead' column which is currently empty.

3. You have two easy options:

- Go to **Branches**. You will see your newly pushed branch listed, often with a count in the “Ahead” column. Azure DevOps usually shows a special icon or button here to create a PR.



- Go directly to **Pull Requests** and click the “New pull request” button.



□ Guide in Progress! □

This guide is currently being updated. The steps above cover creating and pushing your branch. The next section on **filling out the Pull Request form in Azure DevOps** is coming soon