Bootcamp 2020 Class 2: React App

Bootcamp 2020 Class 2: React App **1**
- Part I - Create React Project using Create React App 2
- Part II - Open the Project with VS Code 4
- Part III - View the Project in the Browser 6
- Part IV - Add Repository to GitHub with the CLI 9
- Part V - Deploy React App to Surge with a GitHub Workflow 17
- Part VI - Learn React 21
- Part VII - Report Project Completed as an Issue 22

Part I - Create React Project using Create React App

Step 1: Create a new local repository
First, I navigate to my local GitHub folder where all my local repositories are kept.

```
Adil:~ adil$ cd /Users/adil/Documents/GitHub_
```

Now I create my local repository by creating my basic react application with:

npx create-react-app <app-name>

```
Adil:GitHub adil$ npx create-react-app project002
npx: installed 98 in 8.958s

Creating a new React app in /Users/adil/Documents/GitHub/project002.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

yarn add v1.19.1
[1/4] 🔍  Resolving packages...
[2/4] 🚚  Fetching packages...
[3/4] 🔗  Linking dependencies...
warning "react-scripts > @typescript-eslint/eslint-plugin > tsutils@3.
17.1" has unmet peer dependency "typescript@>=2.8.0 || >= 3.2.0-dev ||
 >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.
6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta".
[4/4] 🔨  Building fresh packages...
success Saved lockfile.
success Saved 14 new dependencies.
info Direct dependencies
├─ cra-template@1.0.3
├─ react-dom@16.13.1
├─ react-scripts@3.4.1
```

```
Success! Created project002 at /Users/adil/Documents/GitHub/project002
Inside that directory, you can run several commands:

  yarn start
    Starts the development server.

  yarn build
    Bundles the app into static files for production.

  yarn test
    Starts the test runner.

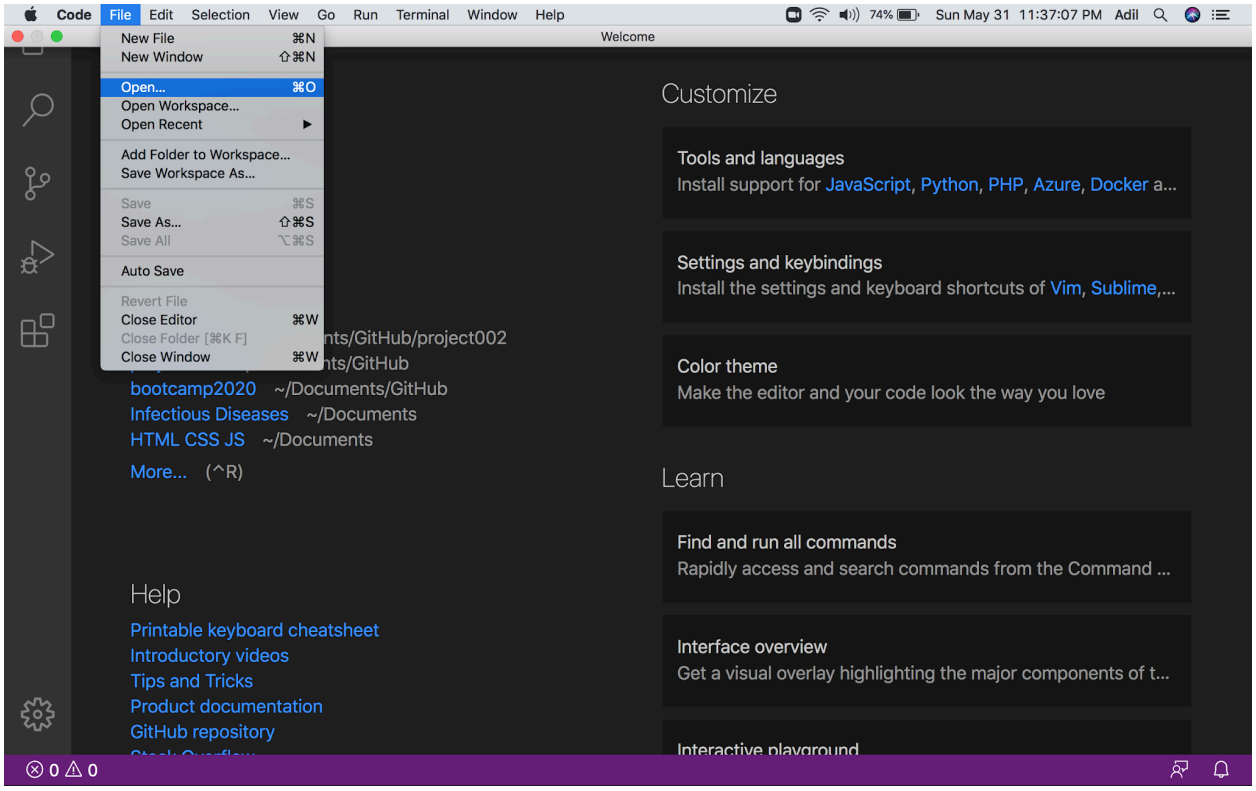
  yarn eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can’t go back!

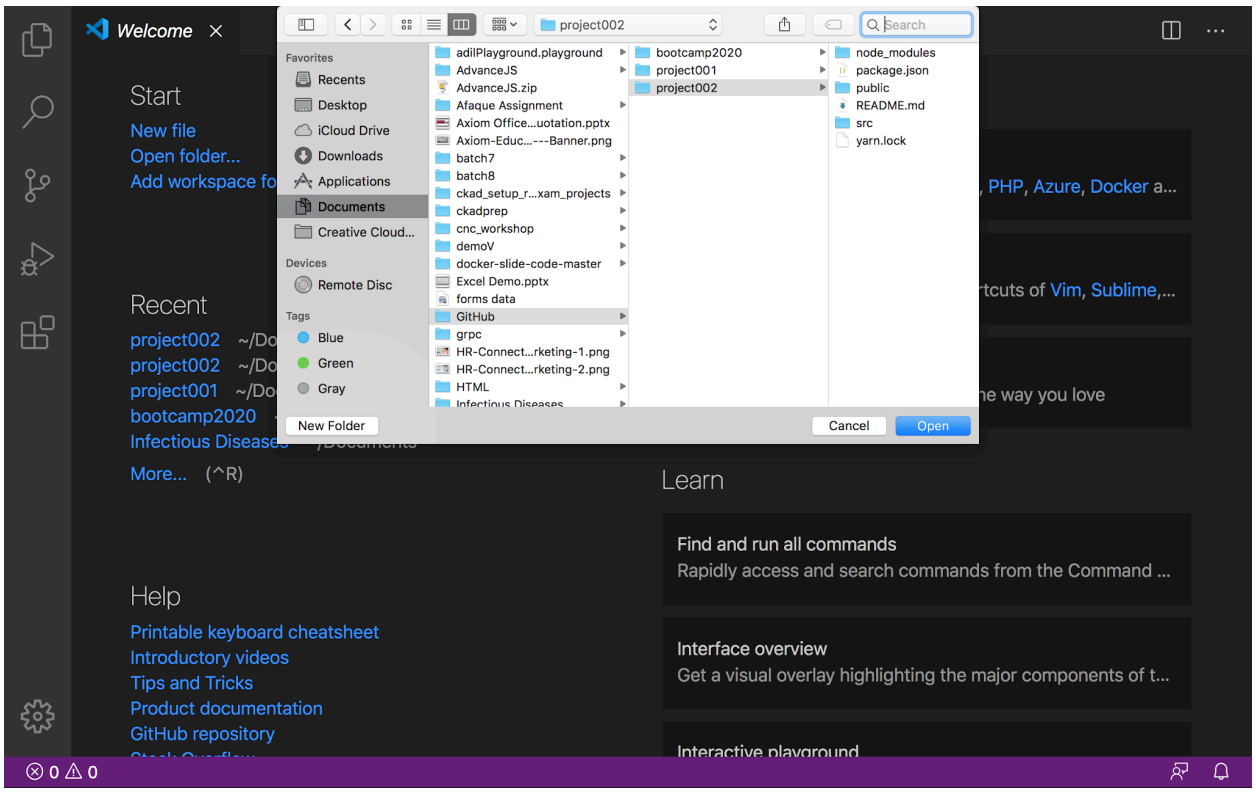
We suggest that you begin by typing:

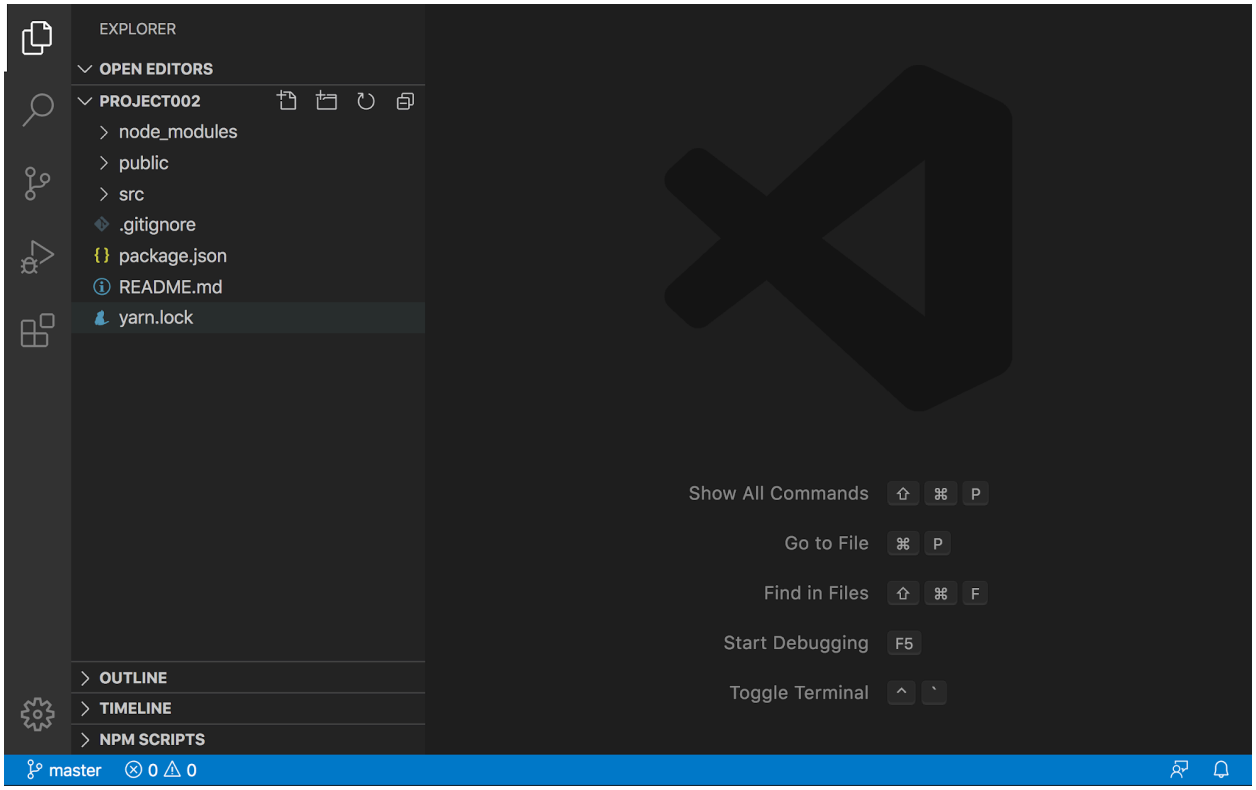
  cd project002
  yarn start

Happy hacking!
Adil:GitHub adil$
```

Part II - Open the Project with VS Code

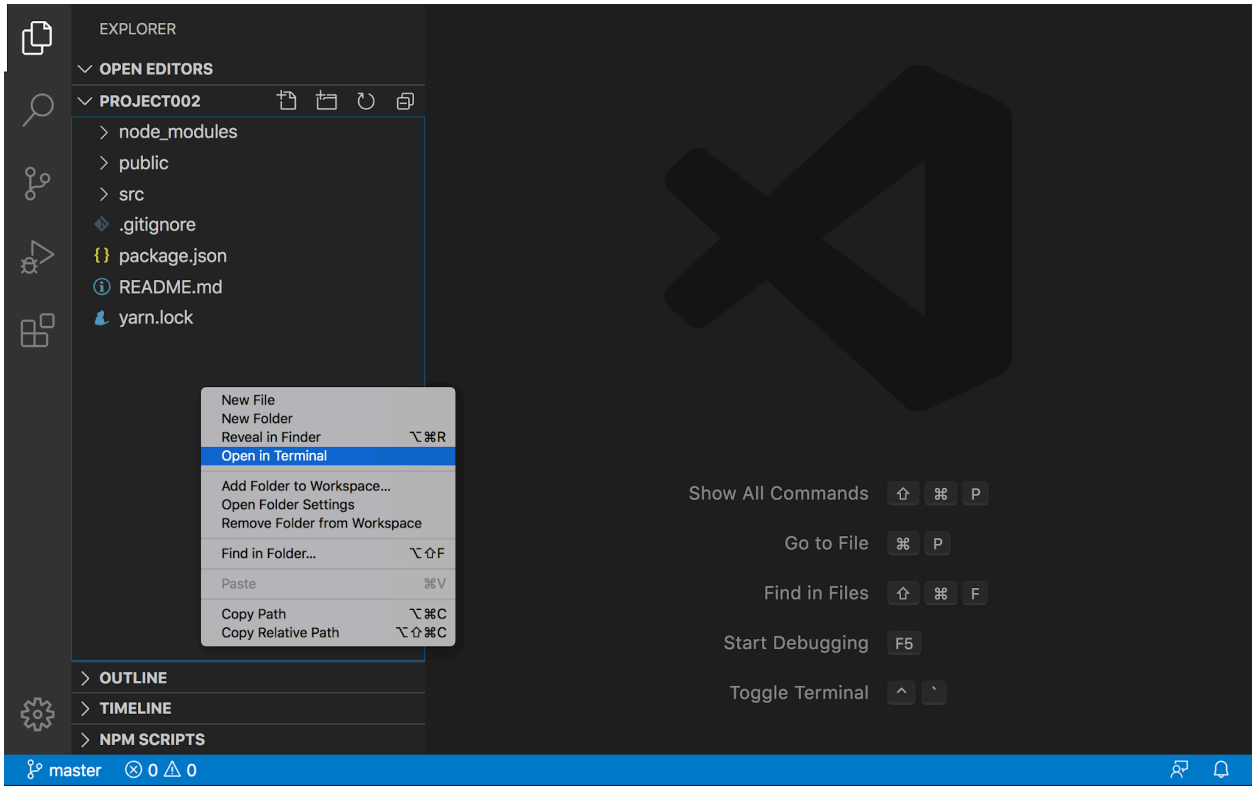


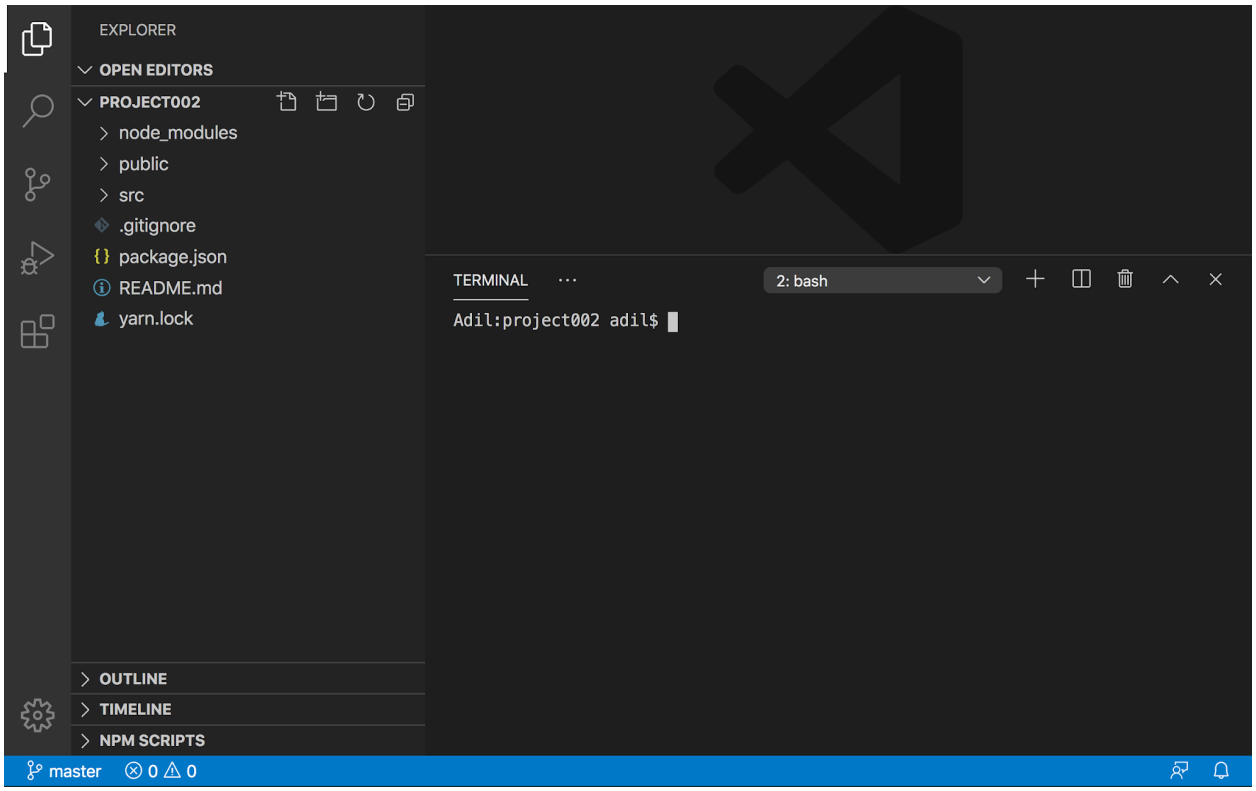


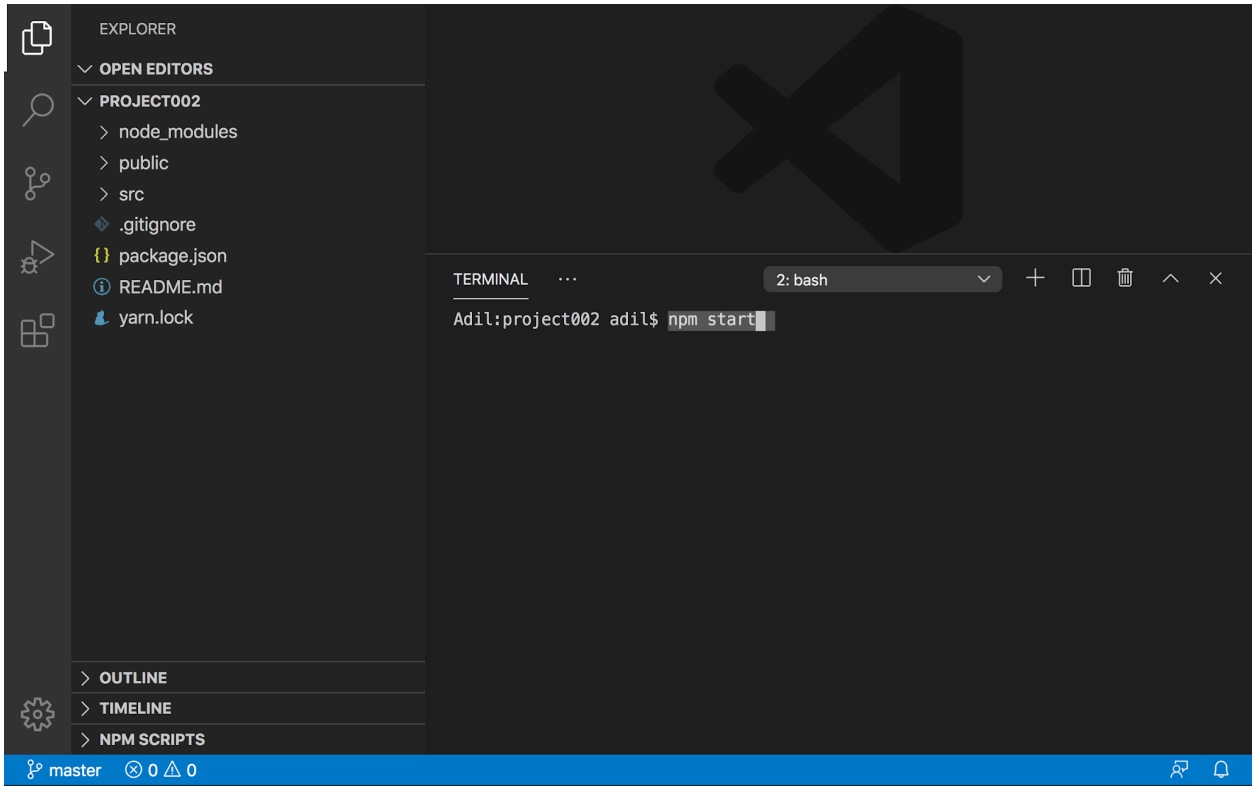

EXPLORER
OPEN EDITORS
PROJECT002
node_modules
public
src
.gitignore
package.json
README.md
yarn.lock
OUTLINE
TIMELINE
NPM SCRIPTS
Show All Commands ⇧ ⌘ P
Go to File ⌘ P
Find in Files ⇧ ⌘ F
Start Debugging F5
Toggle Terminal ^ `
master ⊗ 0 ⚠ 0

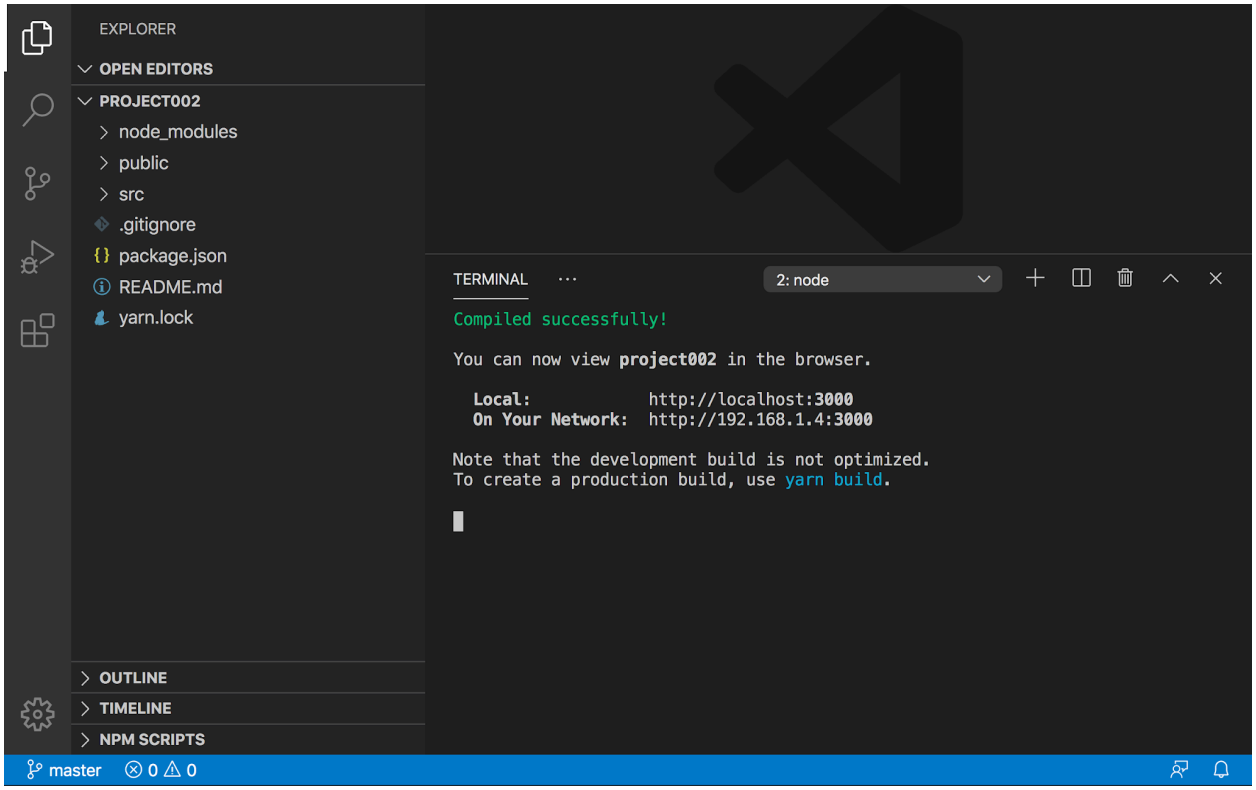
Part III - View the Project in the Browser

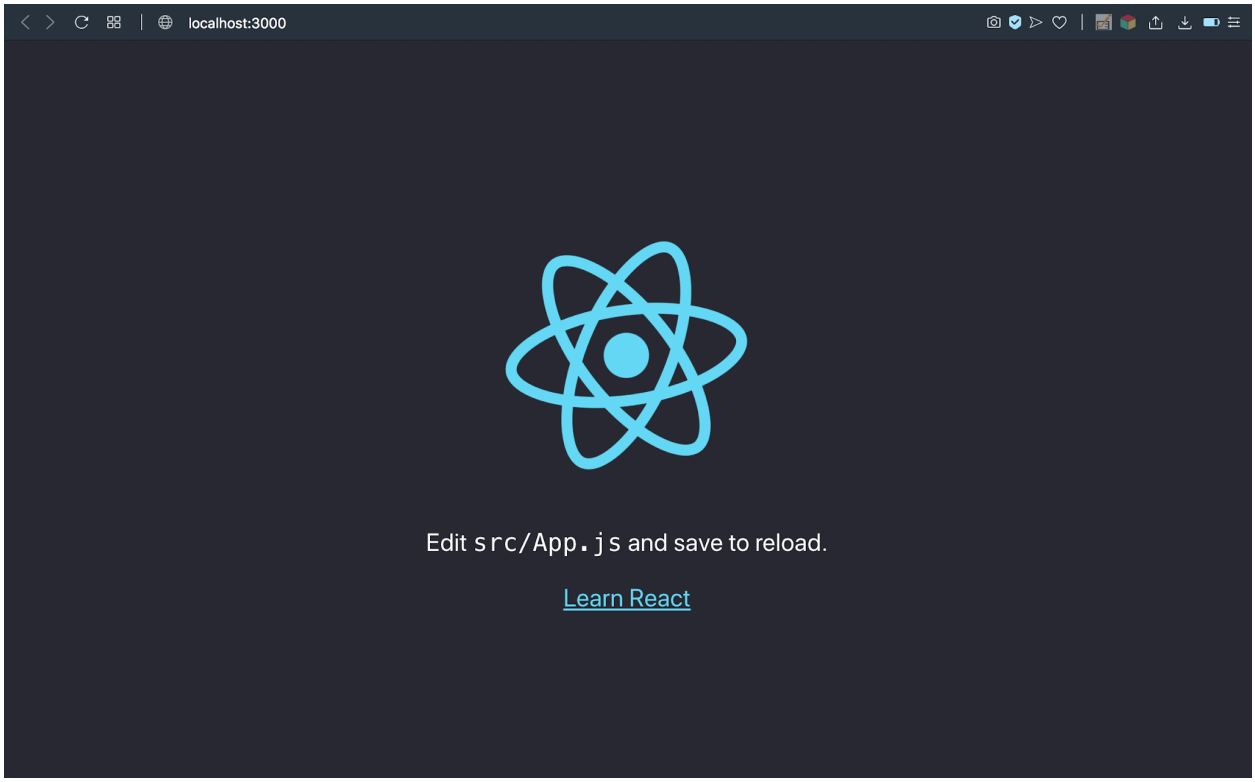
First, let's open the project in the terminal





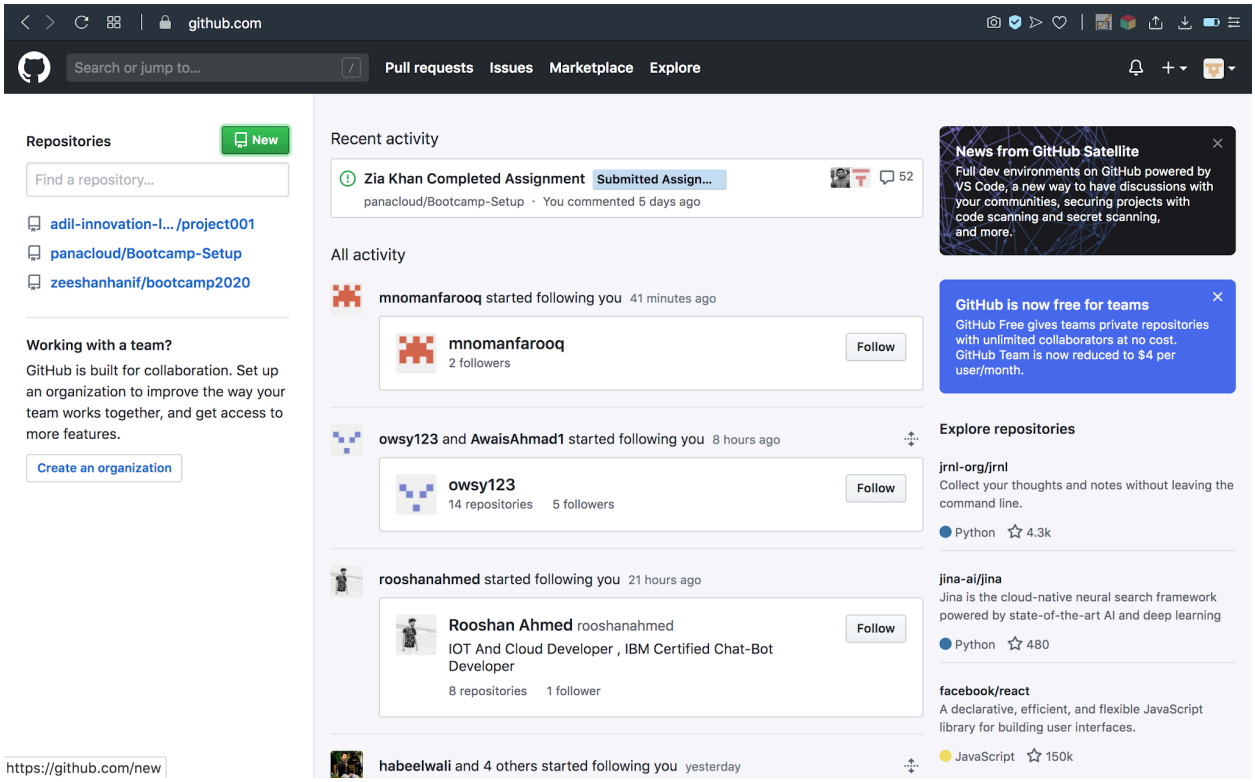

EXPLORER
OPEN EDITORS
PROJECT002
node_modules
public
src
.gitignore
package.json
README.md
yarn.lock
OUTLINE
TIMELINE
NPM SCRIPTS
TERMINAL
2: bash
Adil:project002 adil$ npm start
master
0 0


EXPLORER
OPEN EDITORS
PROJECT002
node_modules
public
src
.gitignore
package.json
README.md
yarn.lock
OUTLINE
TIMELINE
NPM SCRIPTS
TERMINAL
2: node
Compiled successfully!
You can now view project002 in the browser.
Local: http://localhost:3000
On Your Network: http://192.168.1.4:3000
Note that the development build is not optimized.
To create a production build, use yarn build.
master
0 0

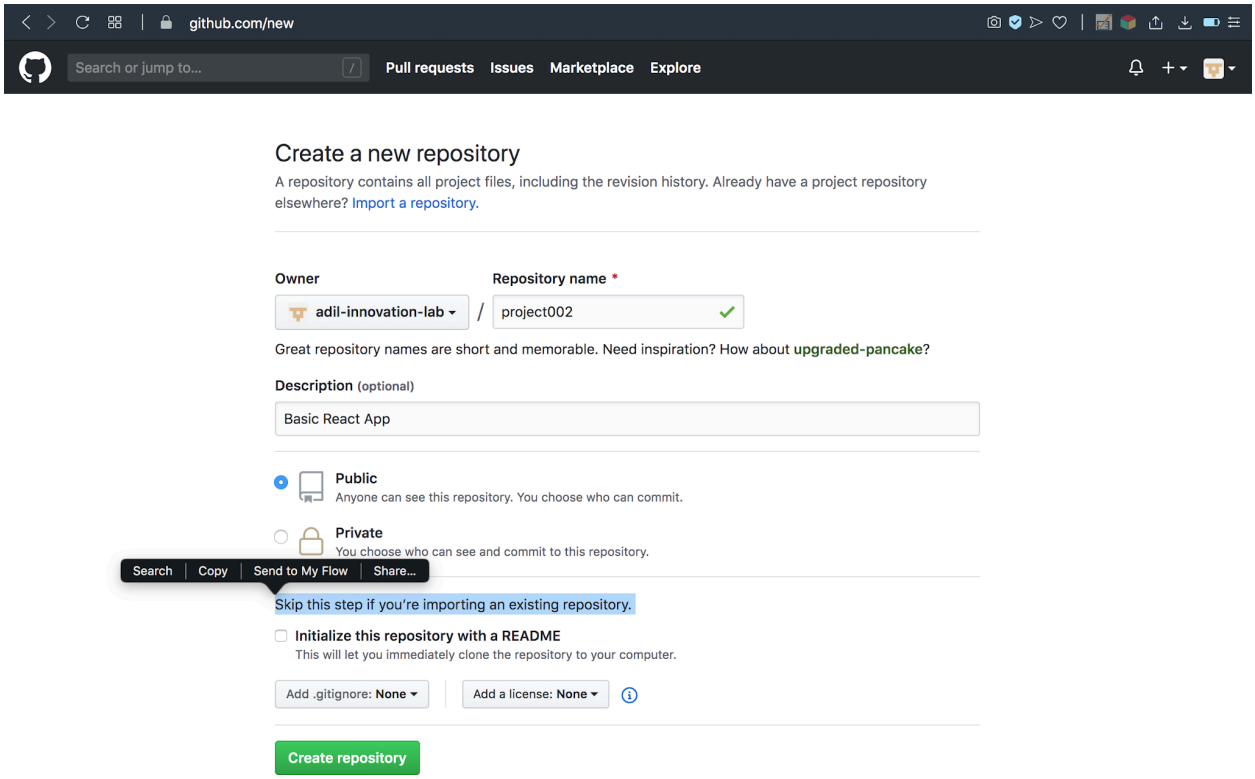

localhost:3000
Edit src/App.js and save to reload.
Learn React

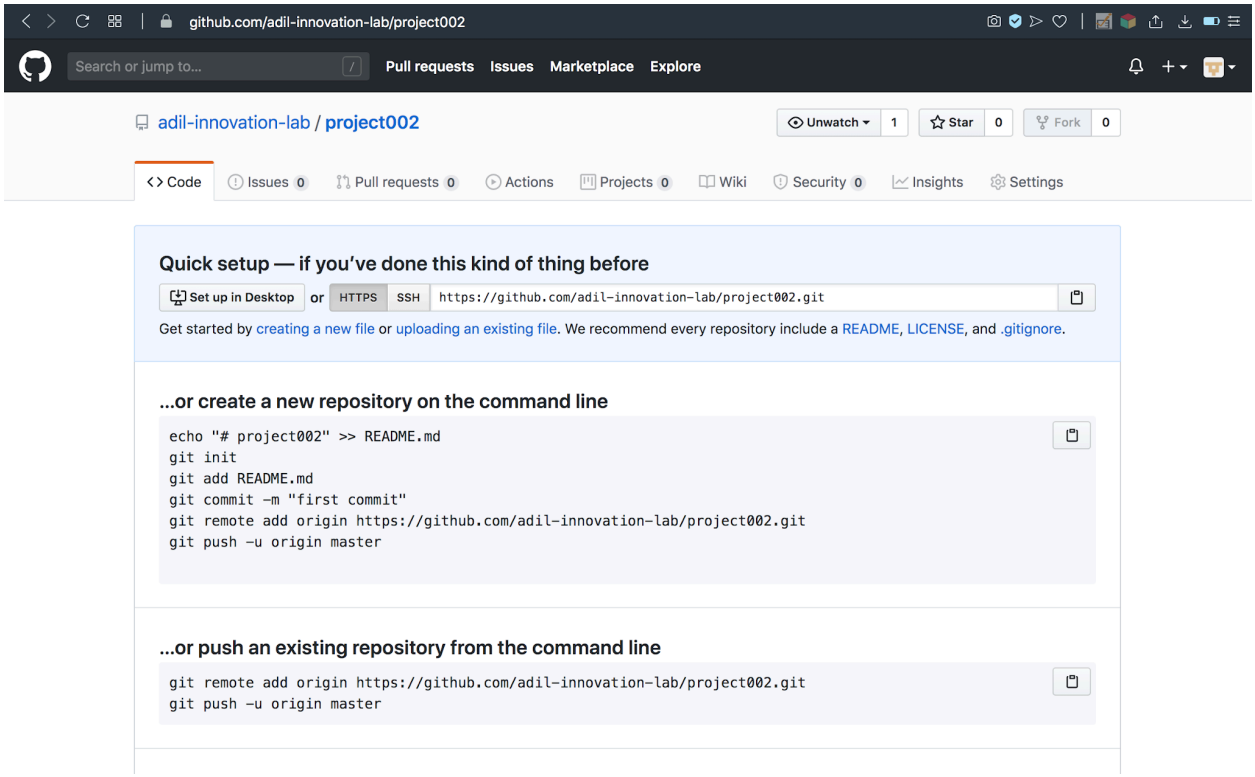
Part IV - Add Repository to GitHub with the CLI

Step 1: Create a new repository on GitHub

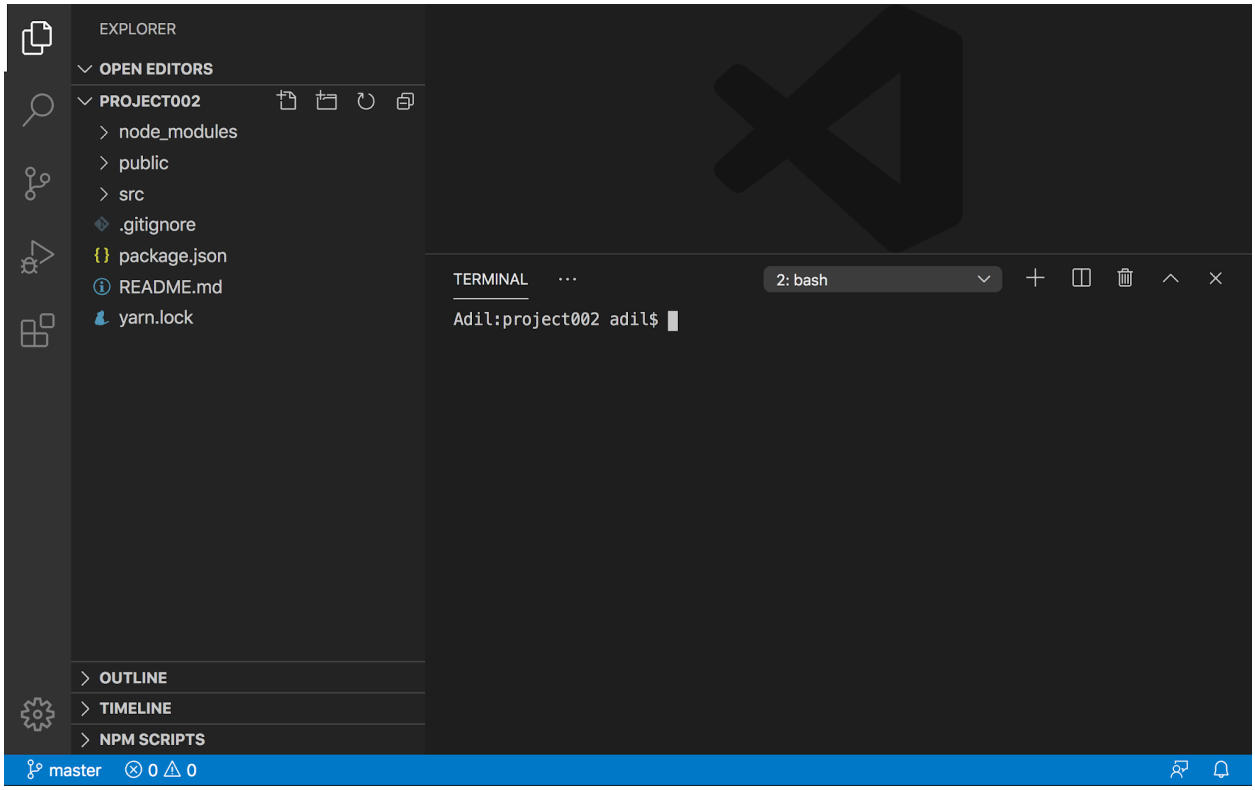


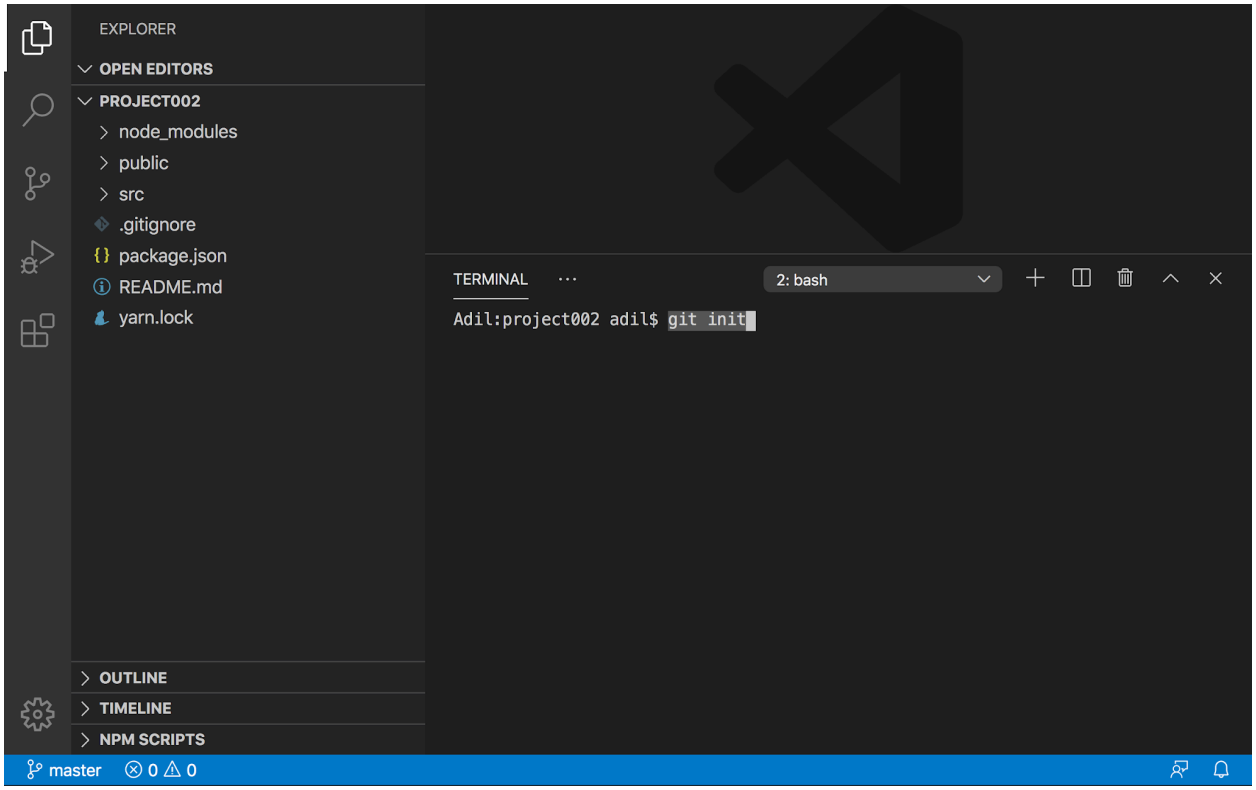
Do not initialize with a readme file, gitignore, or license:

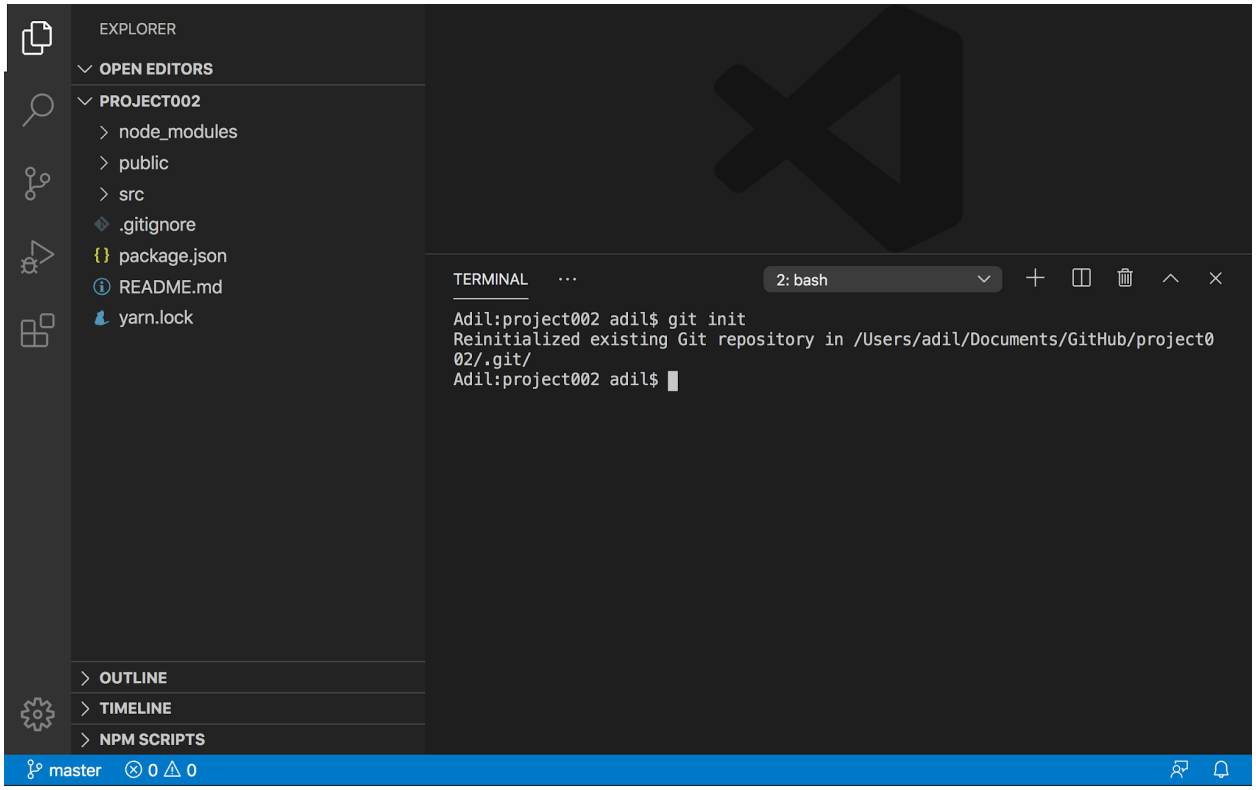




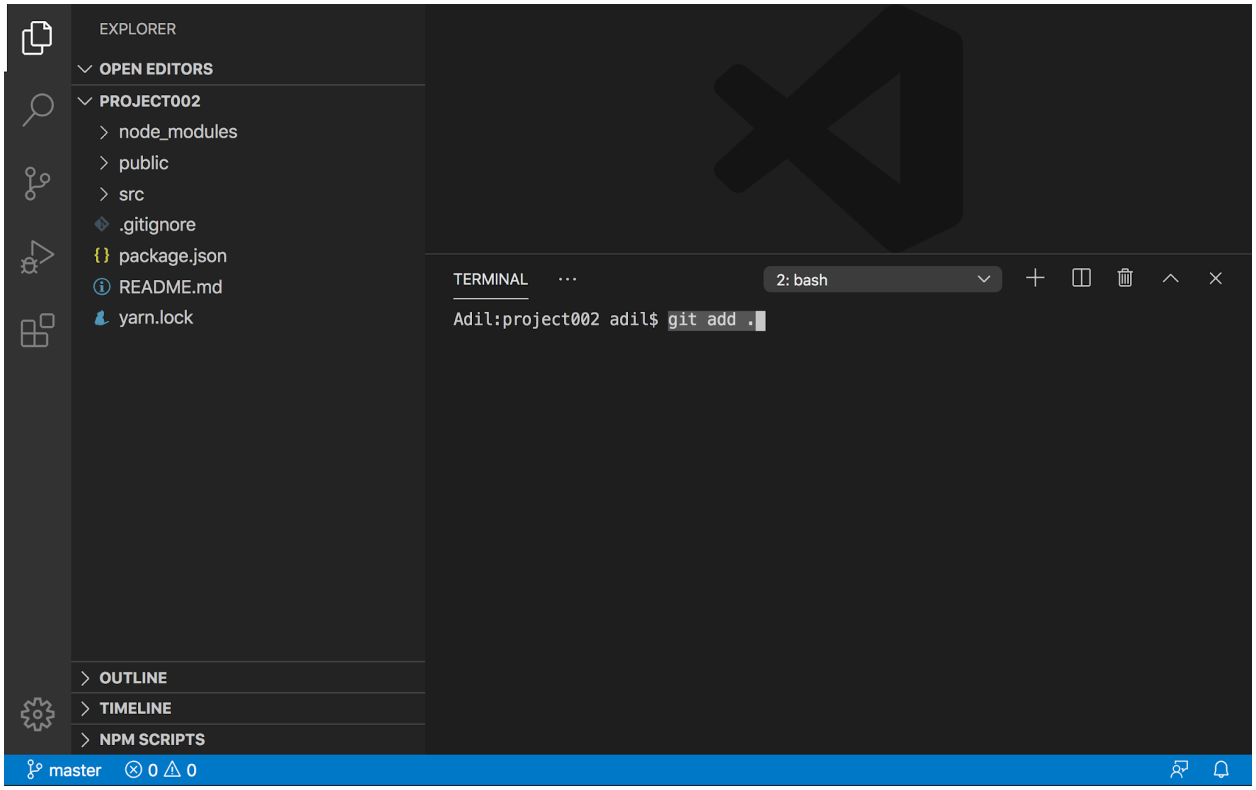
Step 2: Initialize the local directory as a git repository:
Open terminal in our local folder:



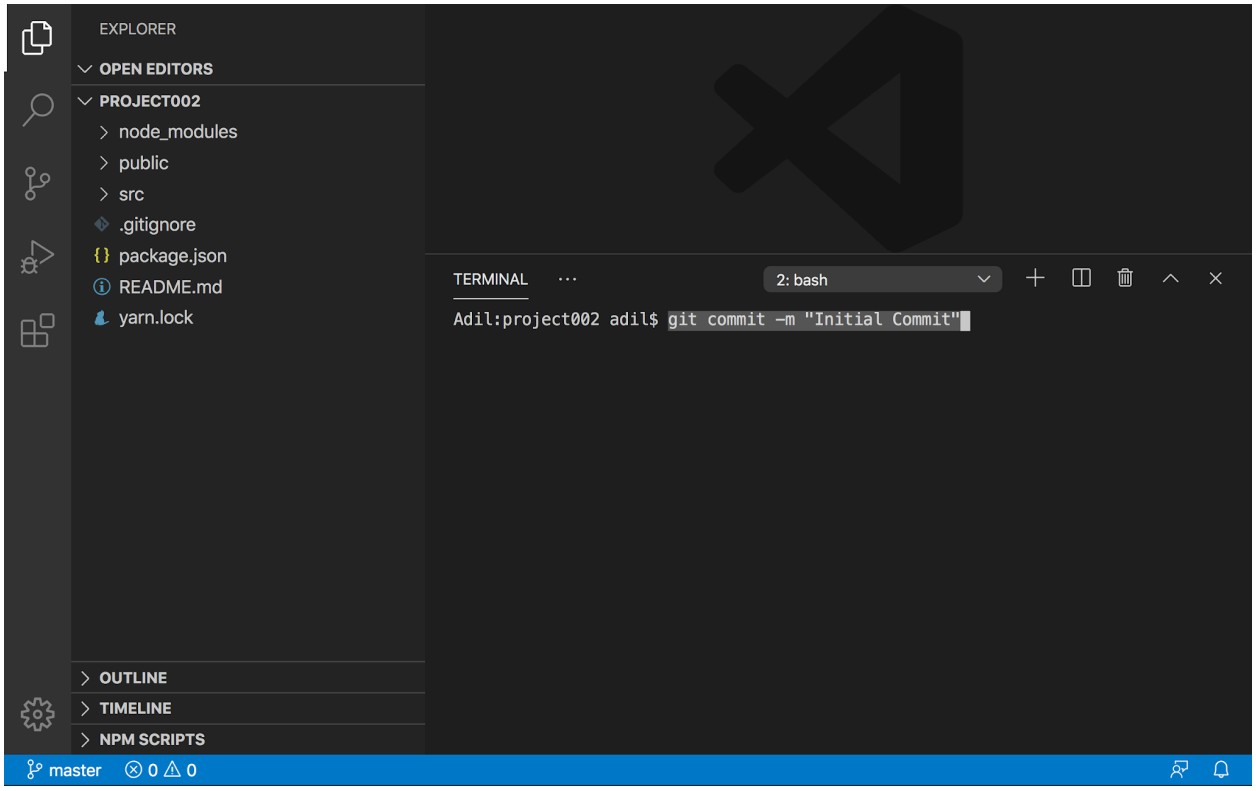

EXPLORER
OPEN EDITORS
PROJECT002
node_modules
public
src
.gitignore
package.json
README.md
yarn.lock
OUTLINE
TIMELINE
NPM SCRIPTS
TERMINAL
2: bash
Adil:project002 adil$ git init
master
0 0

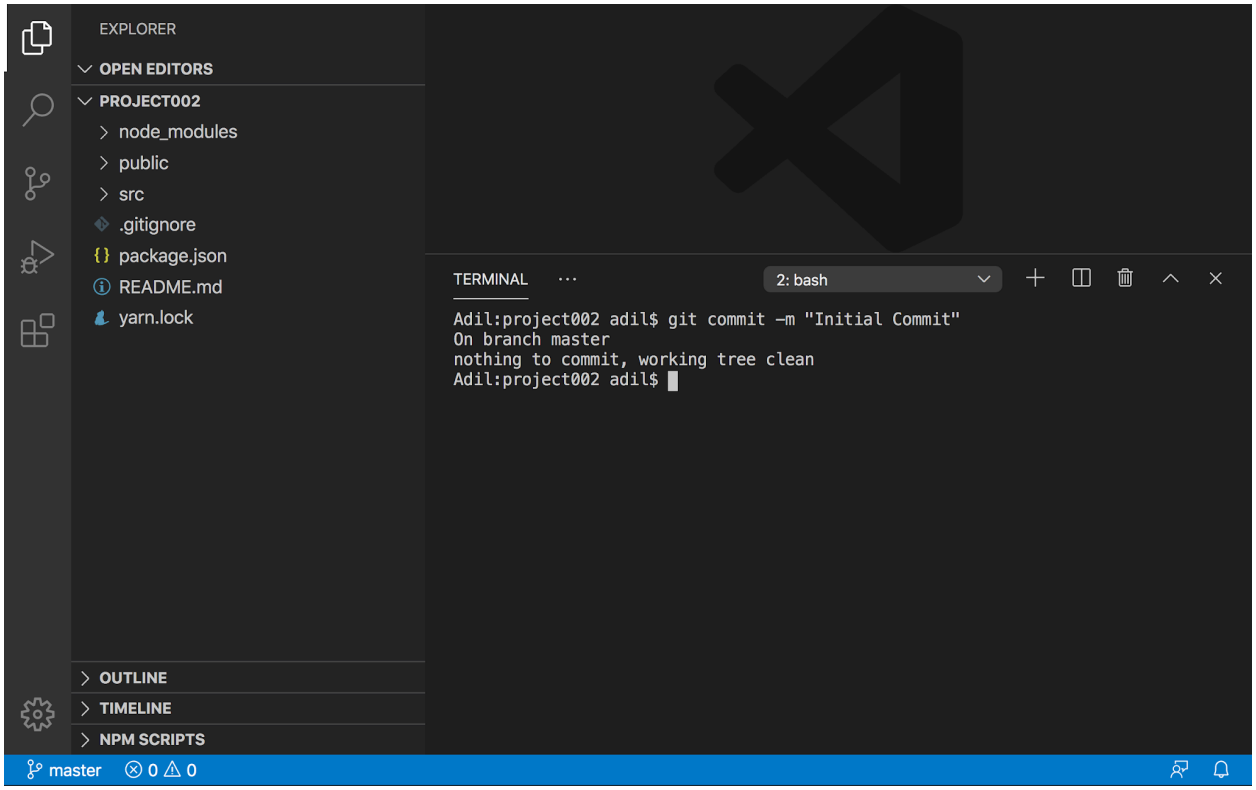

EXPLORER
OPEN EDITORS
PROJECT002
node_modules
public
src
.gitignore
package.json
README.md
yarn.lock
OUTLINE
TIMELINE
NPM SCRIPTS
TERMINAL
2: bash
Adil:project002 adil$ git init
Reinitialized existing Git repository in /Users/adil/Documents/GitHub/project002/.git/
Adil:project002 adil$
master
0 0

Step 3: Add files from local directory to git staging area to prepare for commit

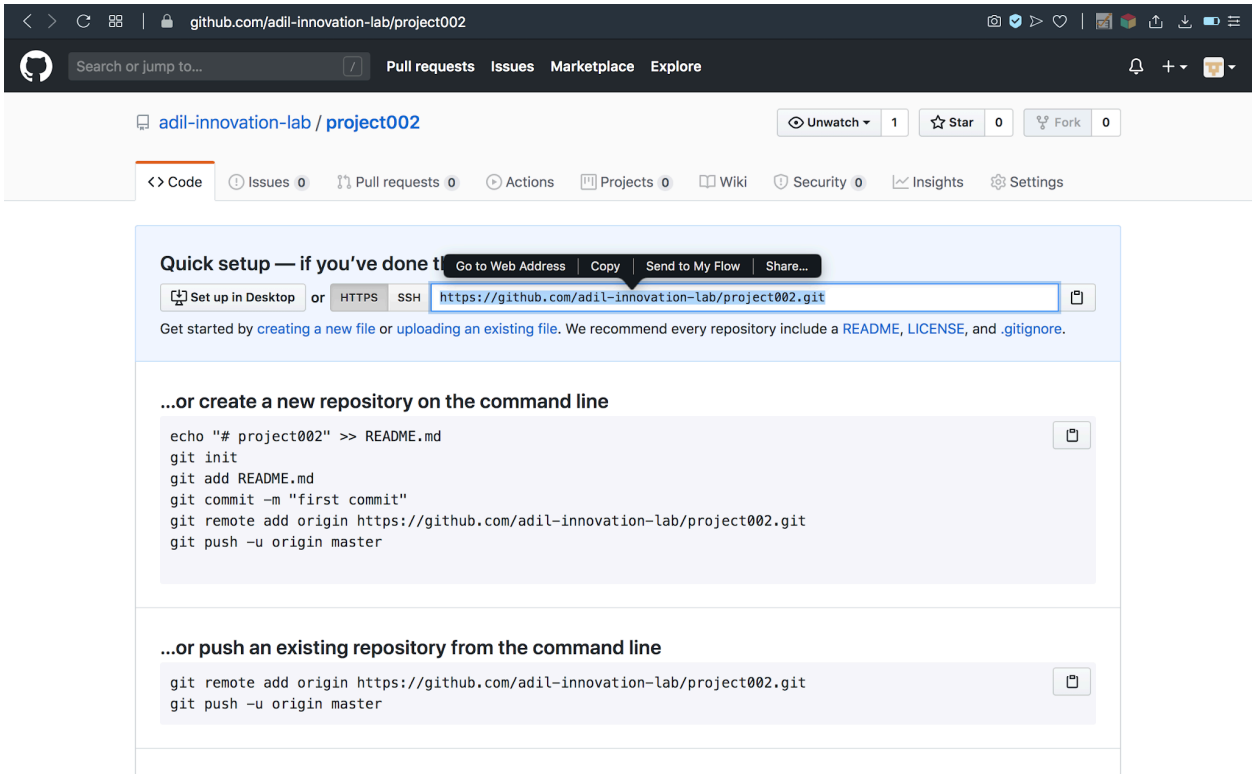


Step 4: Commit files from staging area to local git repository

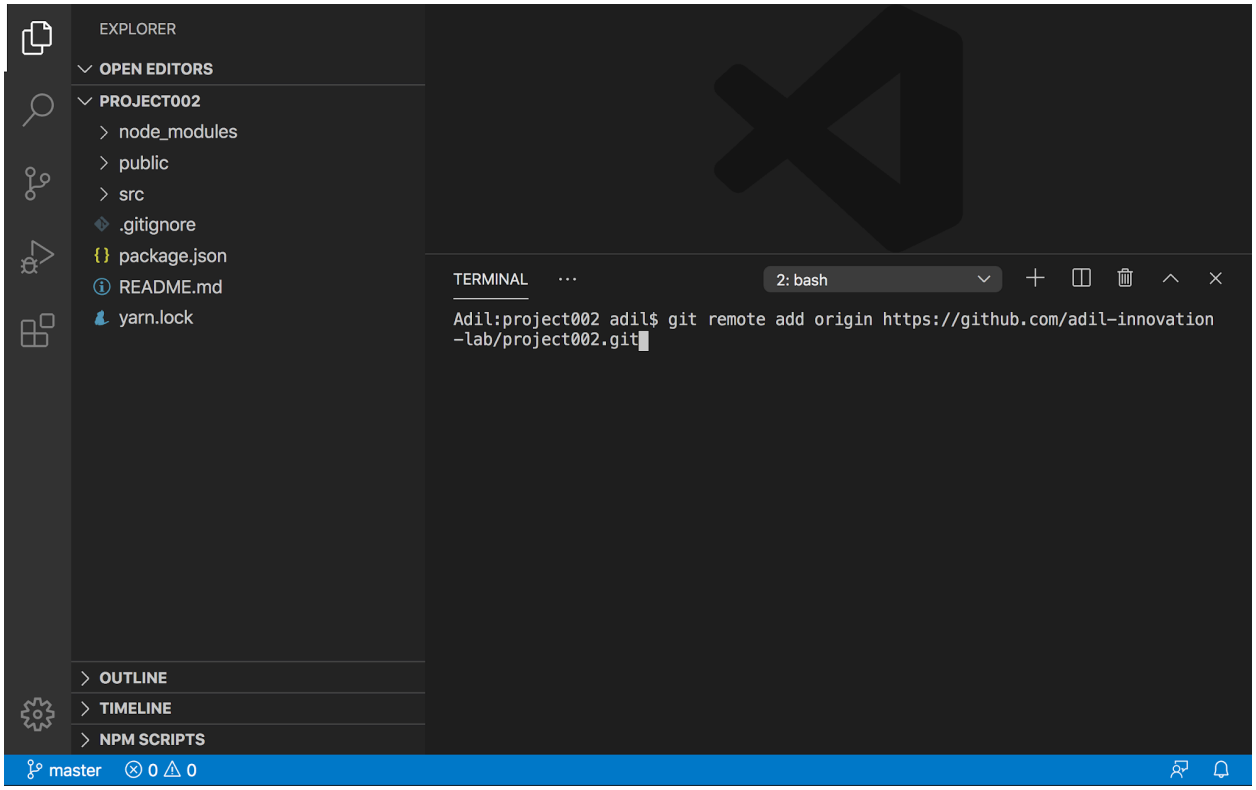




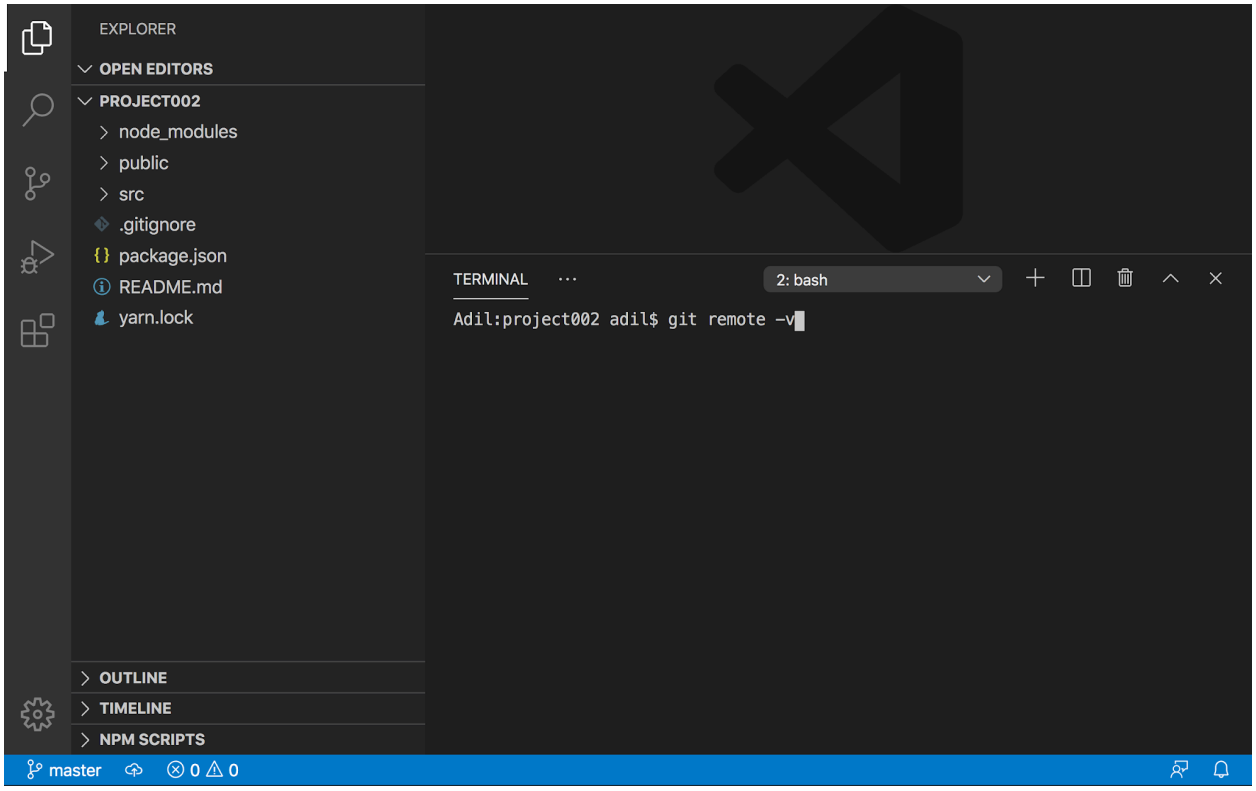
Step 5: Get the URL of the remote repository from GitHub:

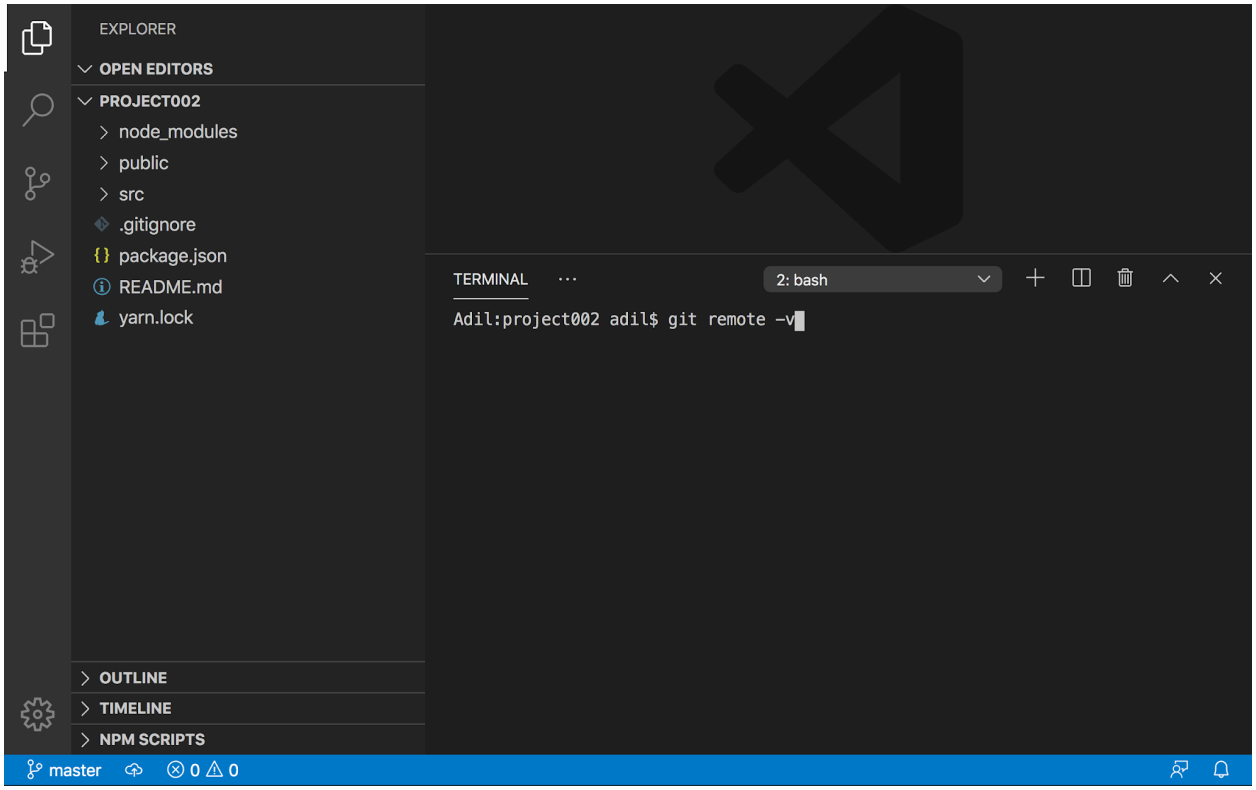


Step 6: Add remote URL to git via the terminal so local repository can be pushed to the right place

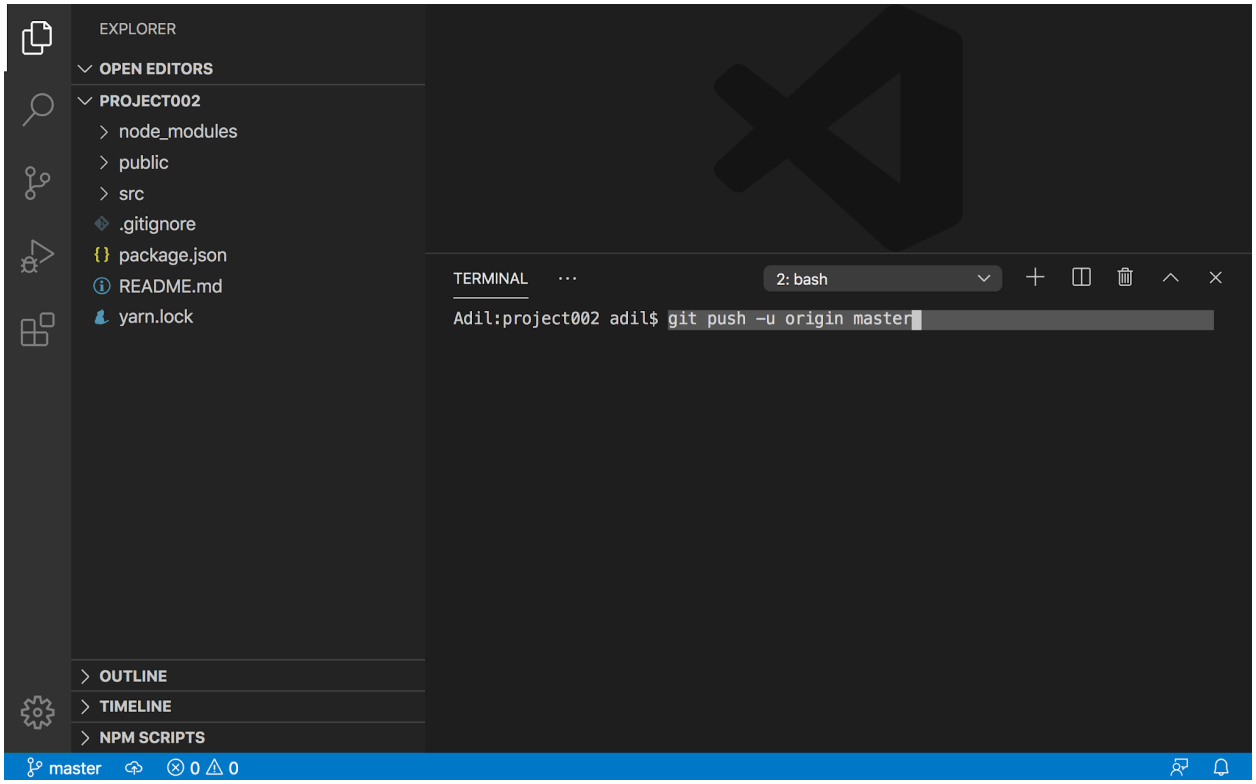


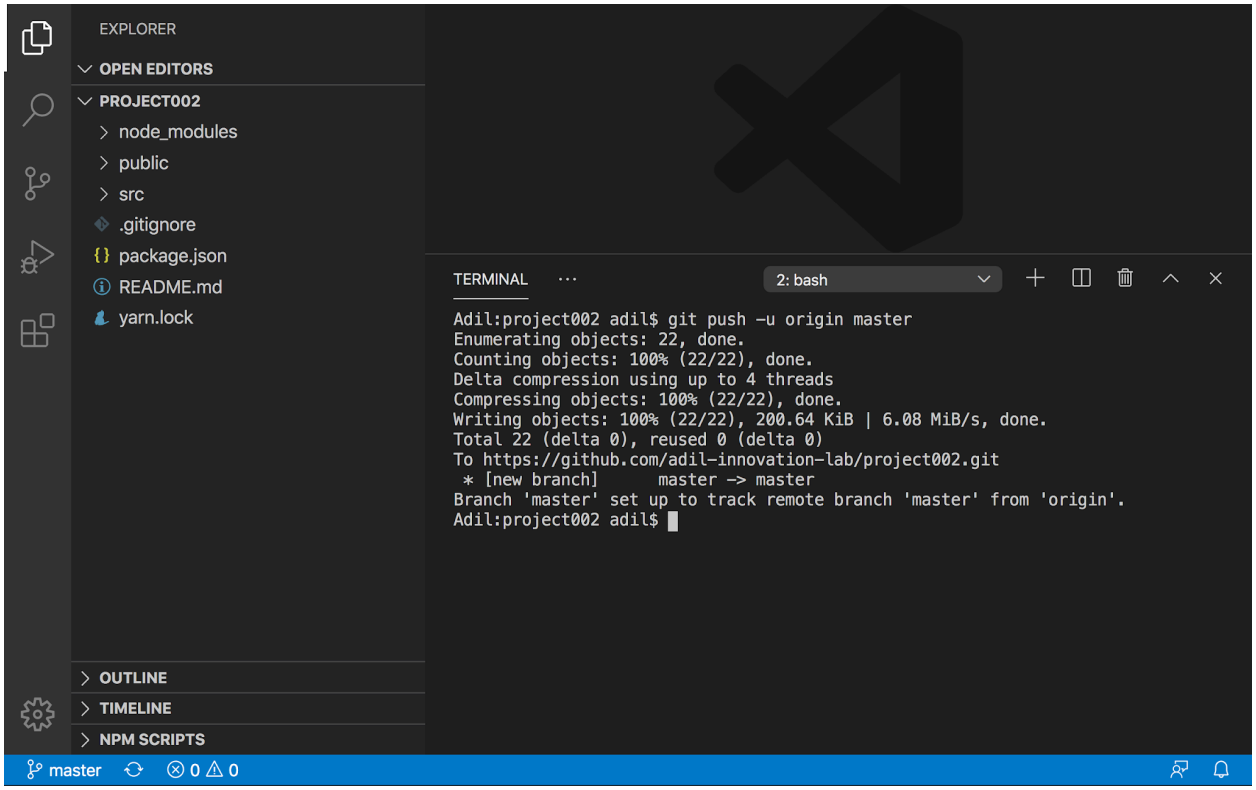
Verify the remote URL



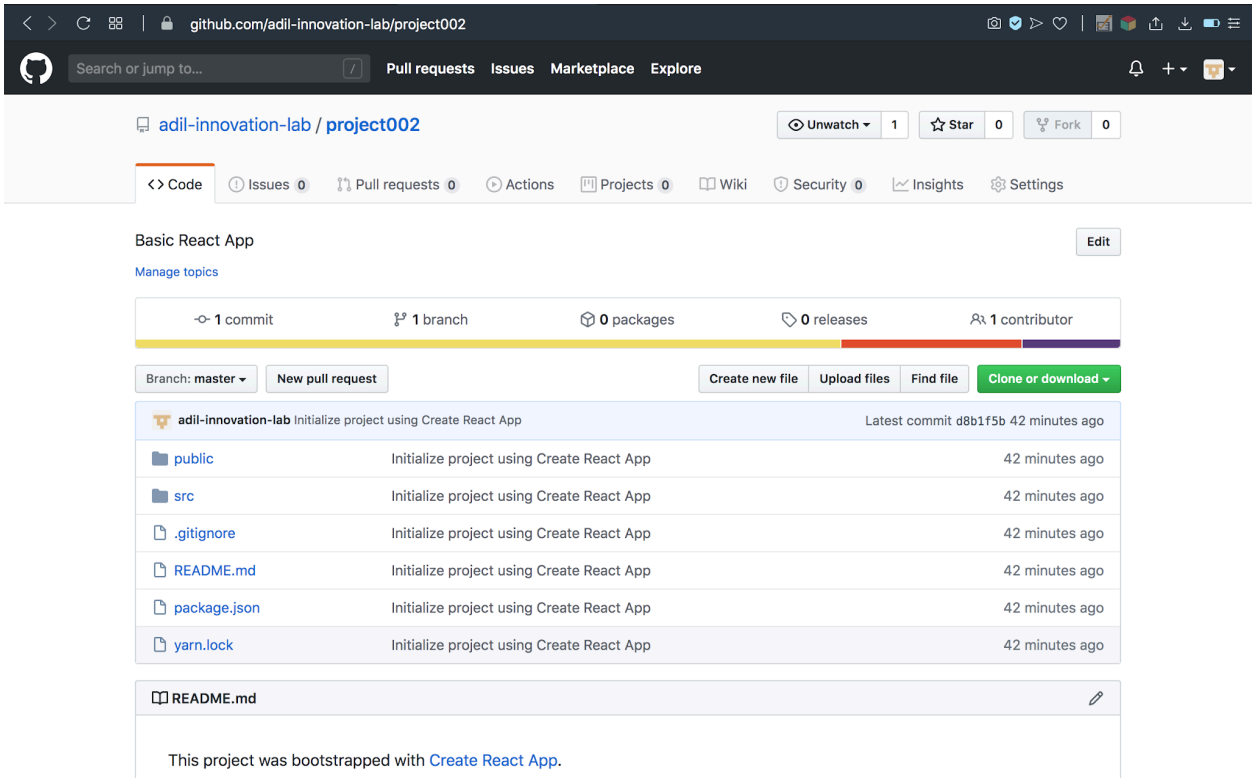


Step 7: Push the code from local git repository to the remote git repository



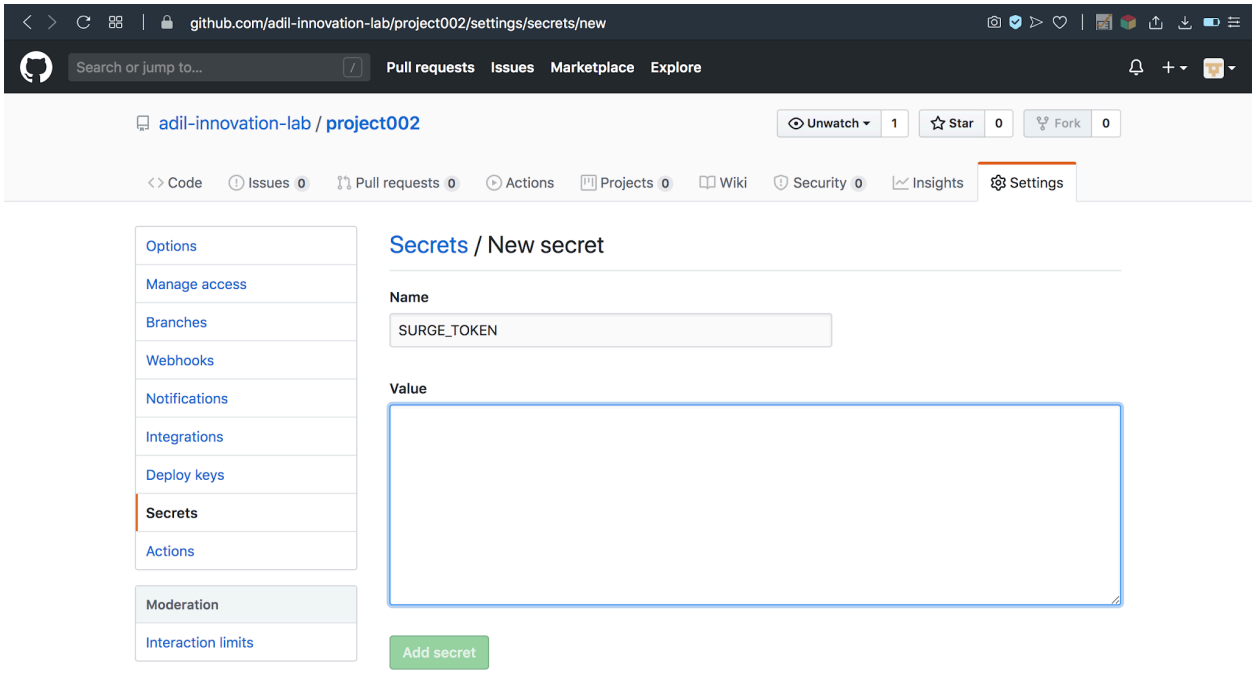


Step 8: Verify changes pushed to remote git repository

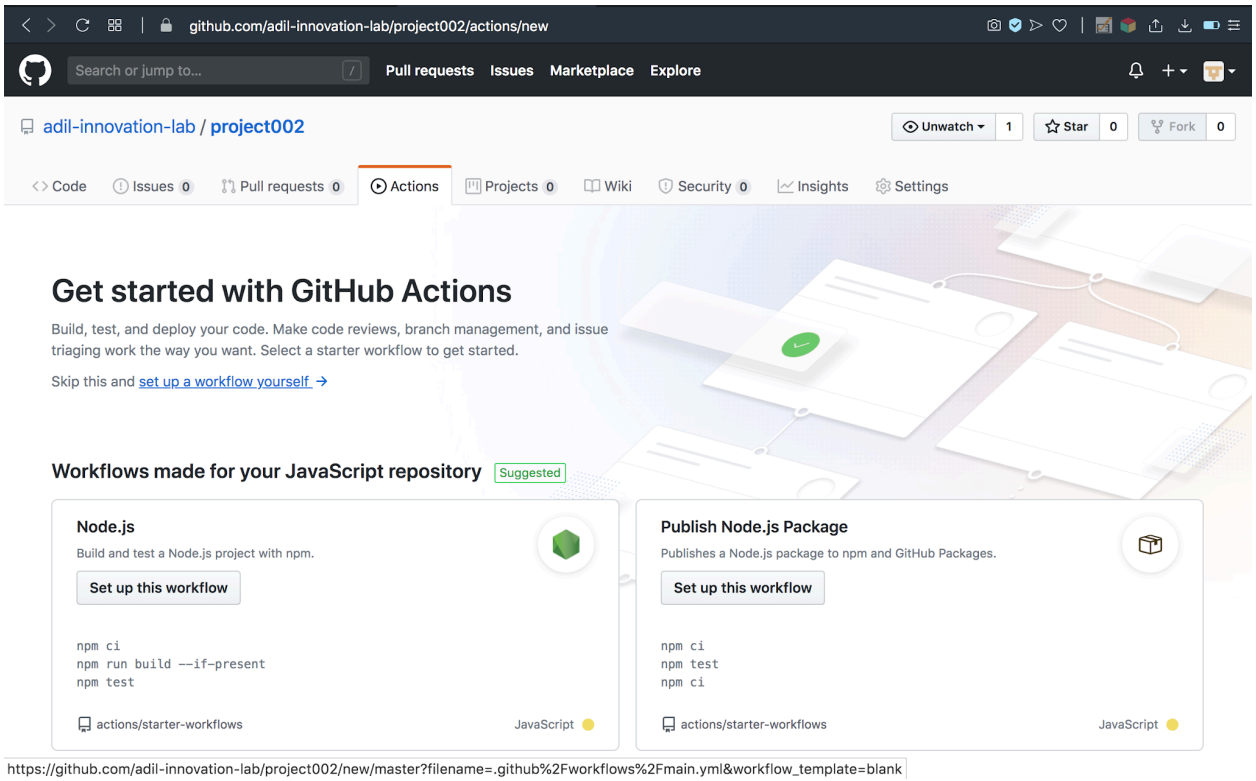


Part V - Deploy React App to Surge with a GitHub Workflow

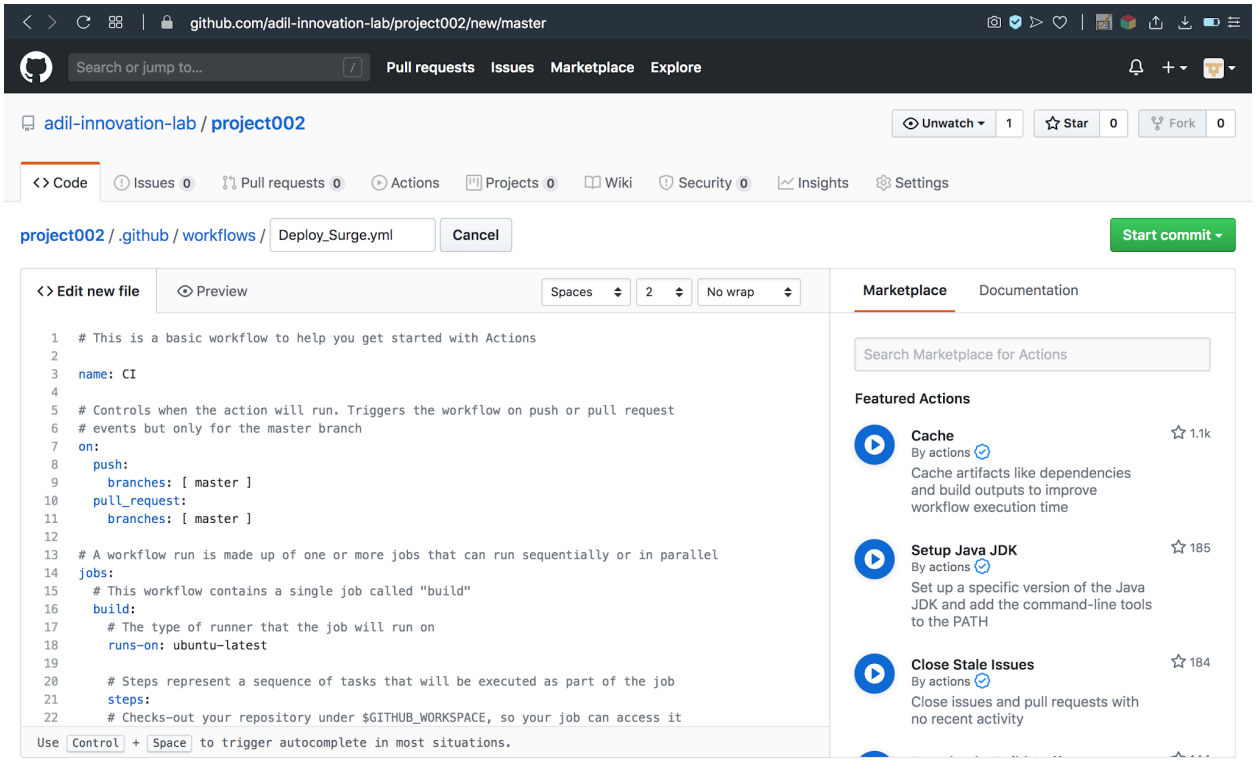
Step 1: Create surge token and add a secret to the GitHub repository



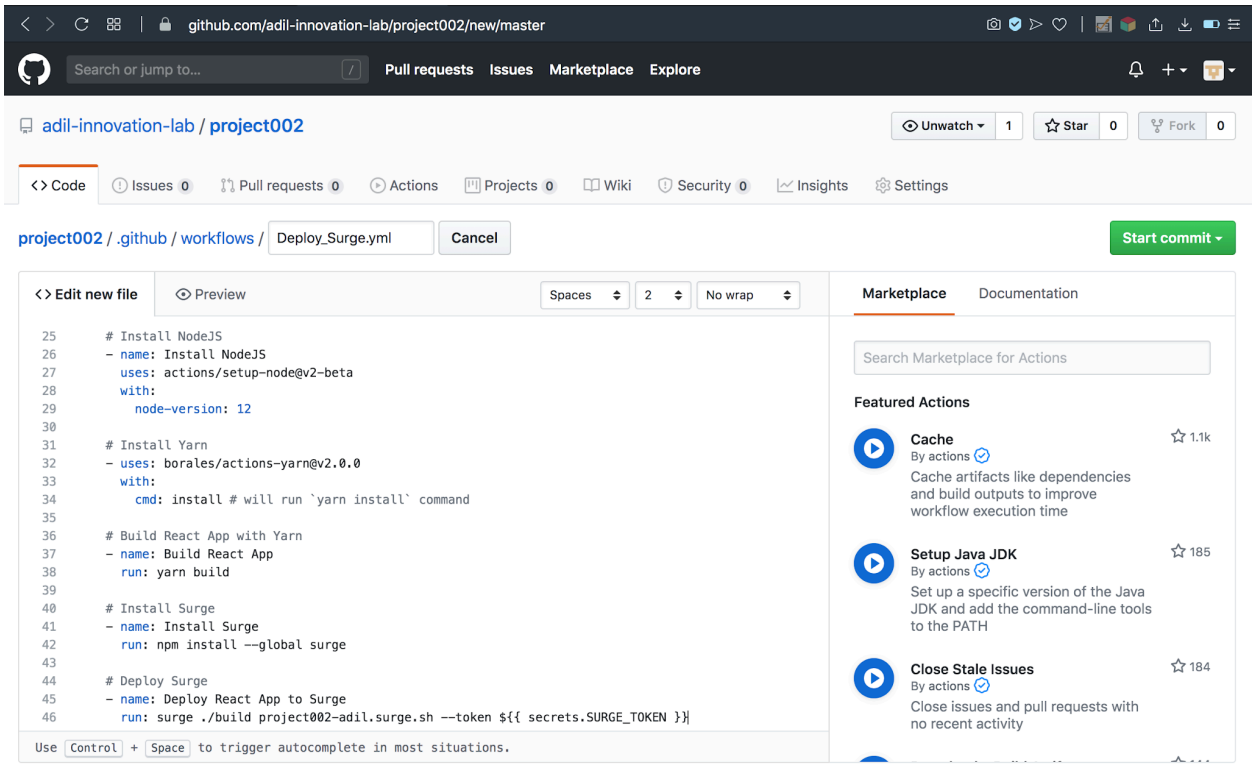
Step 2: Go to the Actions tab in the GitHub repository on github.com and click the set up a workflow yourself link

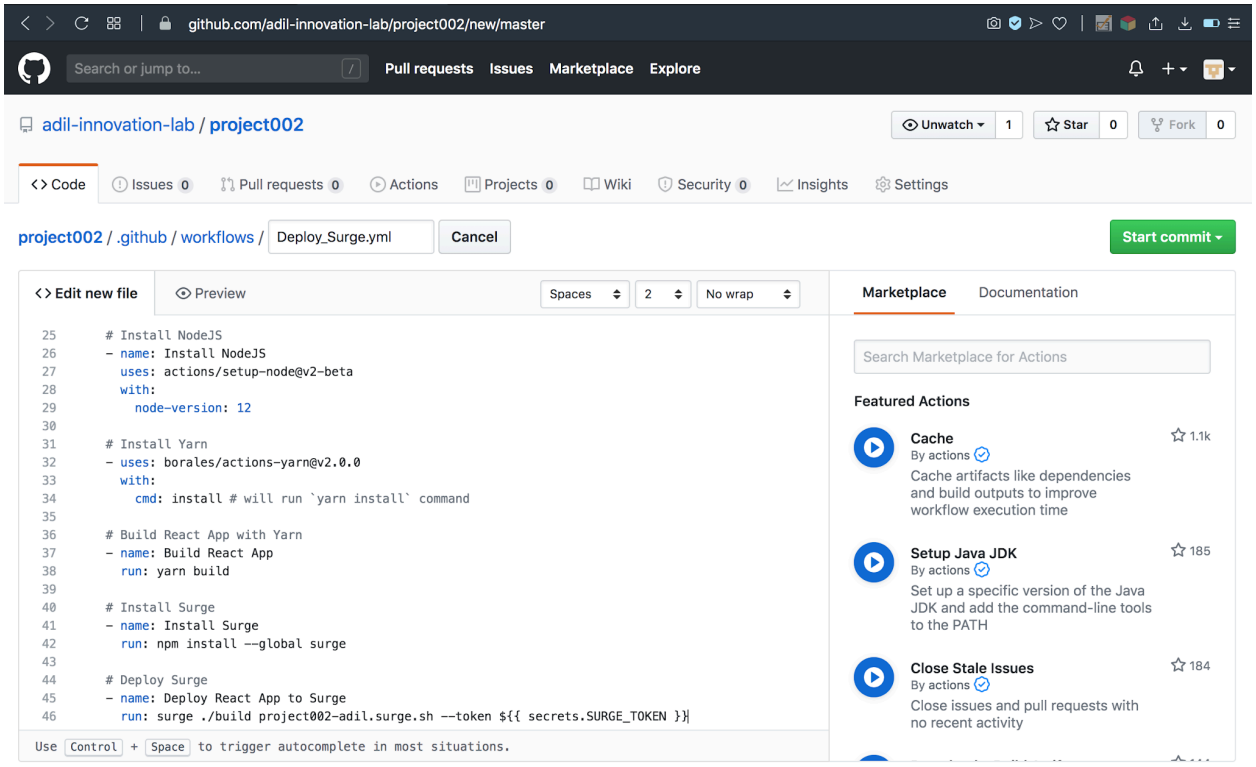


https://github.com/adil-innovation-lab/project002/new/master?filename=.github%2Fworkflows%2Fmain.yml&workflow_template=blank

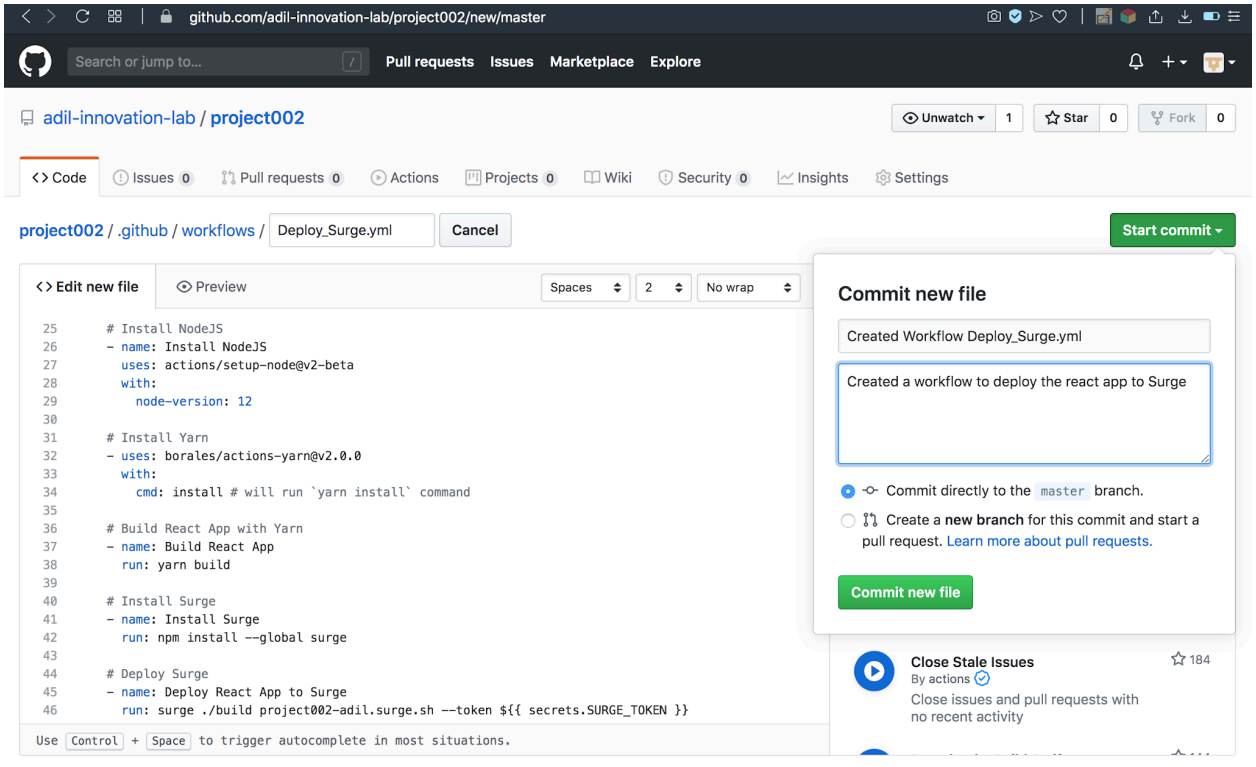


Step 3: Write the YAML for the workflow

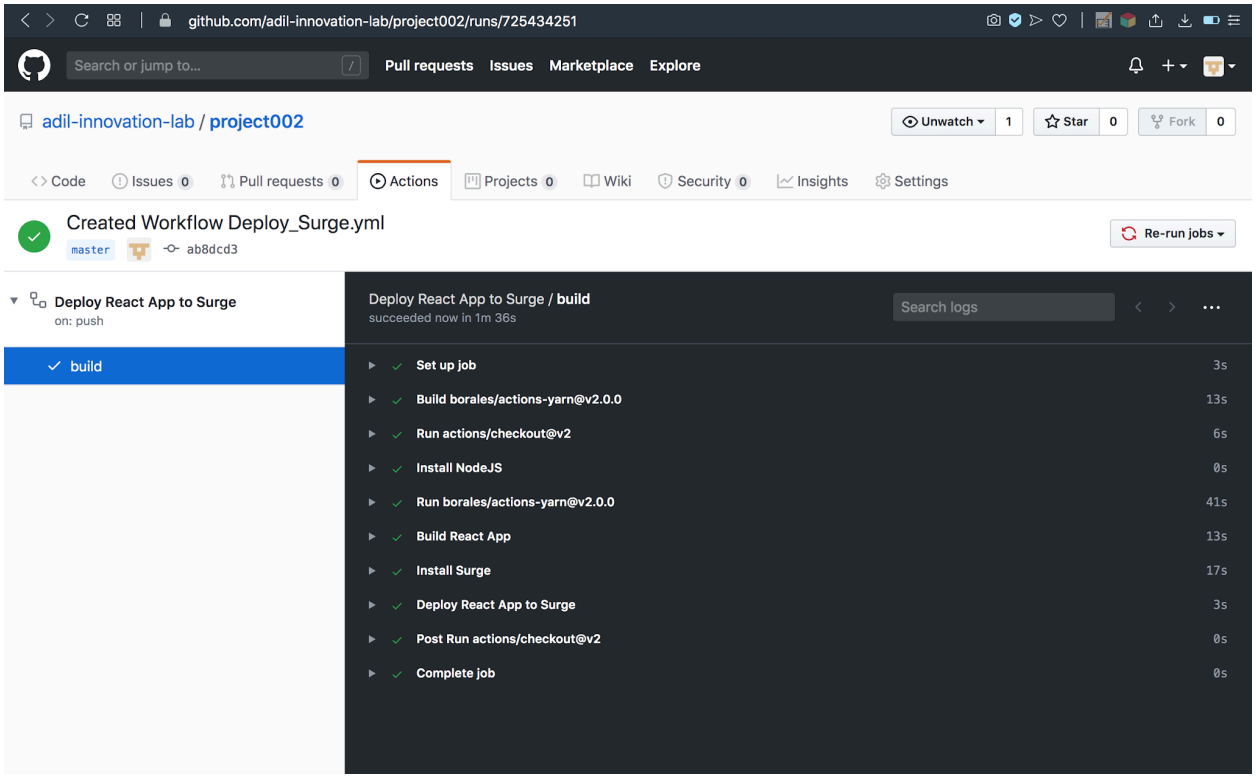




Step 4: Commit the workflow to the master branch



Step 5: Check the Actions tab for the workflow to deploy successfully



Part VI - Learn React

https://daveceddia.com/react-tutorial/

Part VII - Report Project Completed as an Issue

