

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.

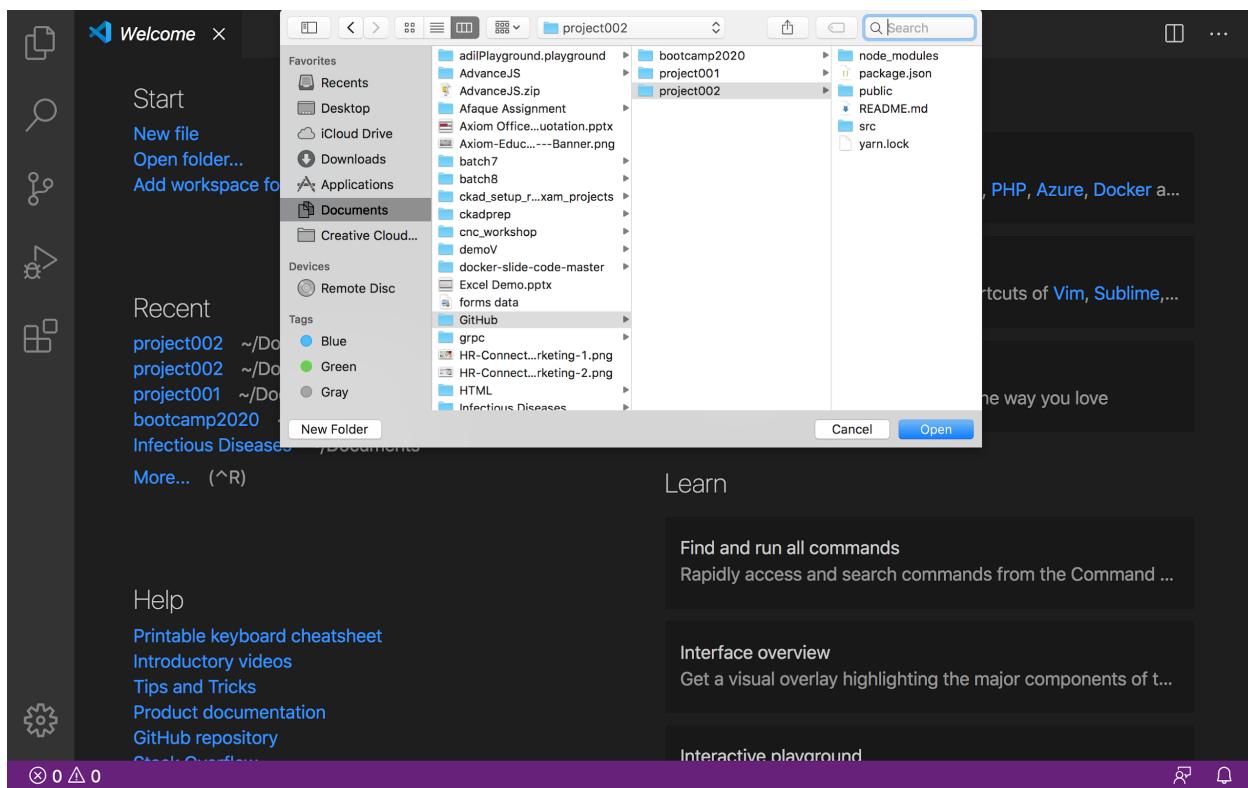
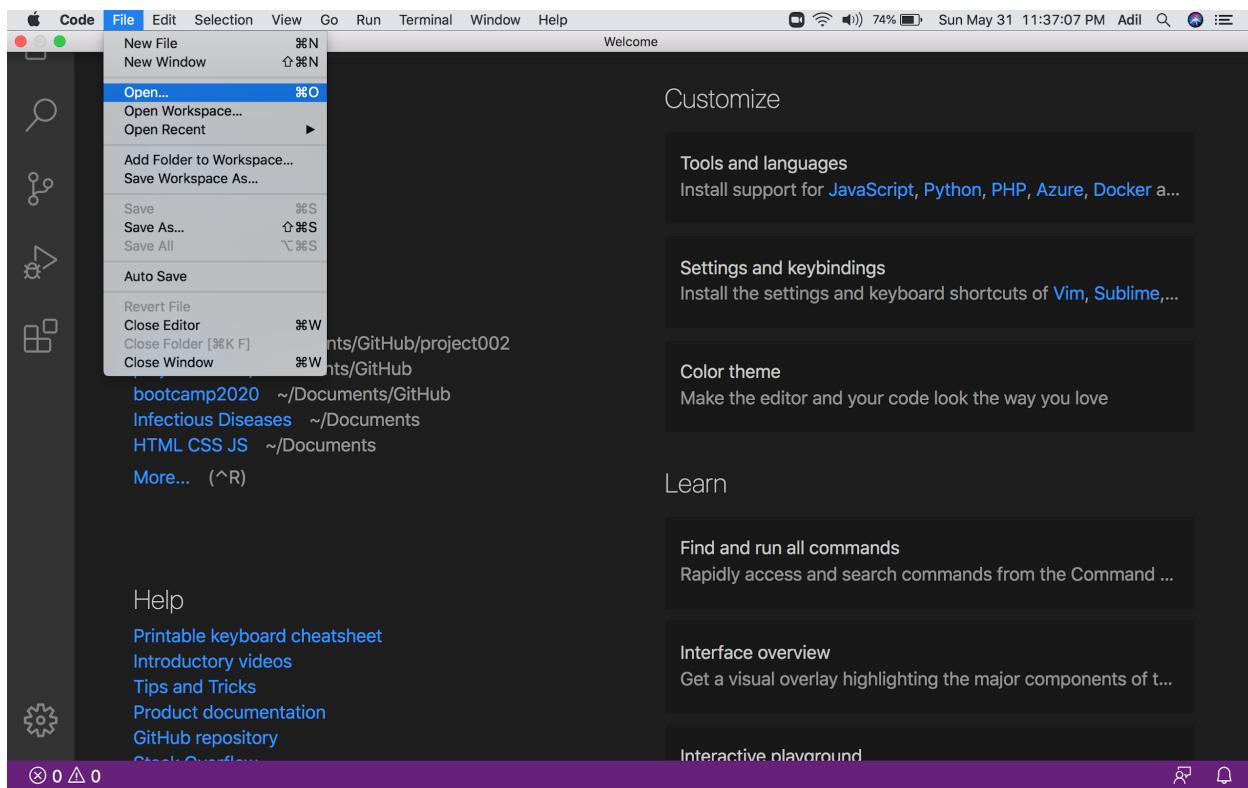
yarn eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!

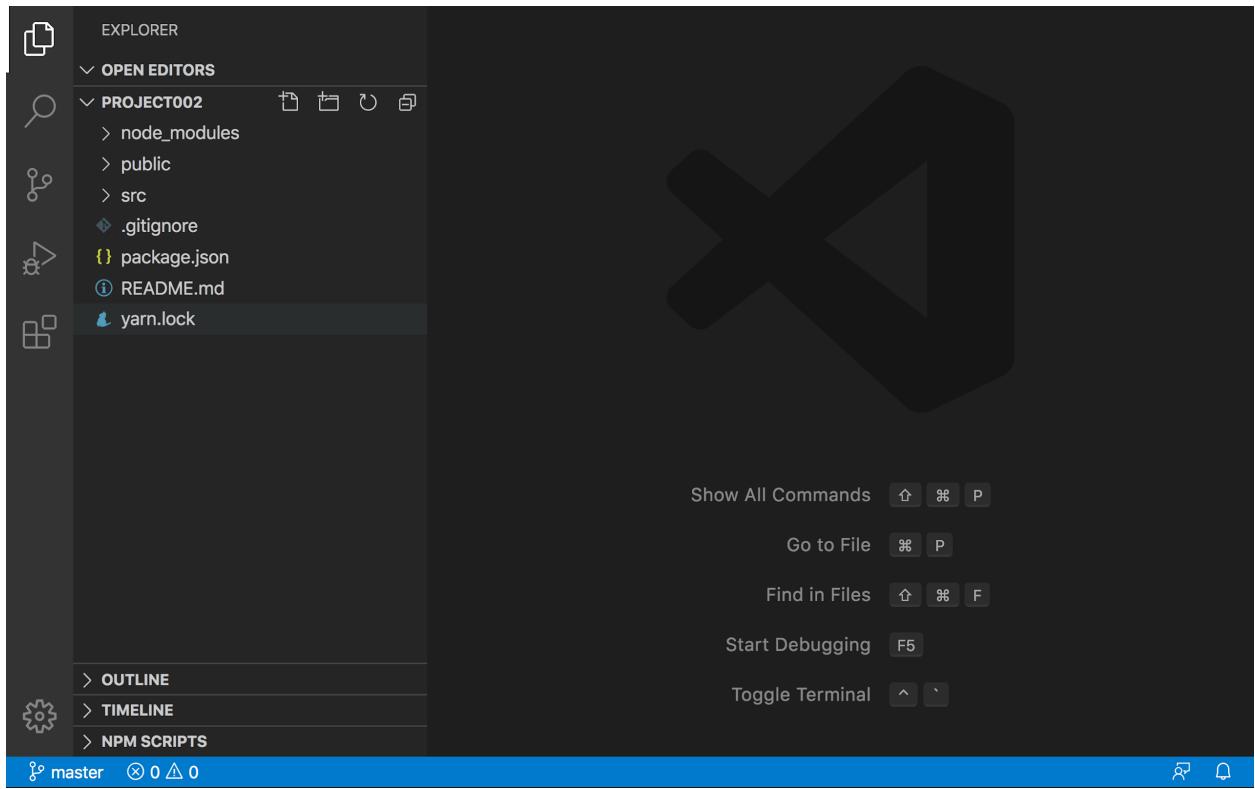
We suggest that you begin by typing:

cd project002
yarn start

Happy hacking!
Adil:GitHub adil$
```

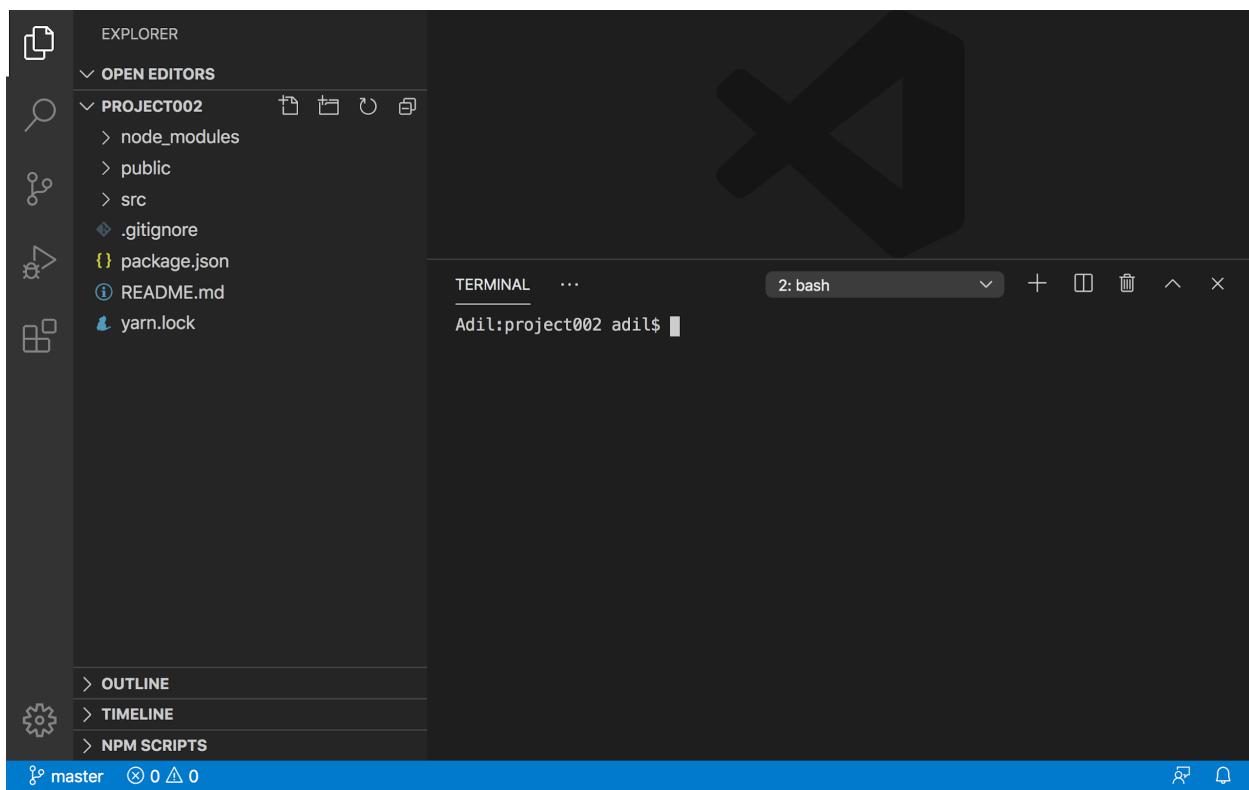
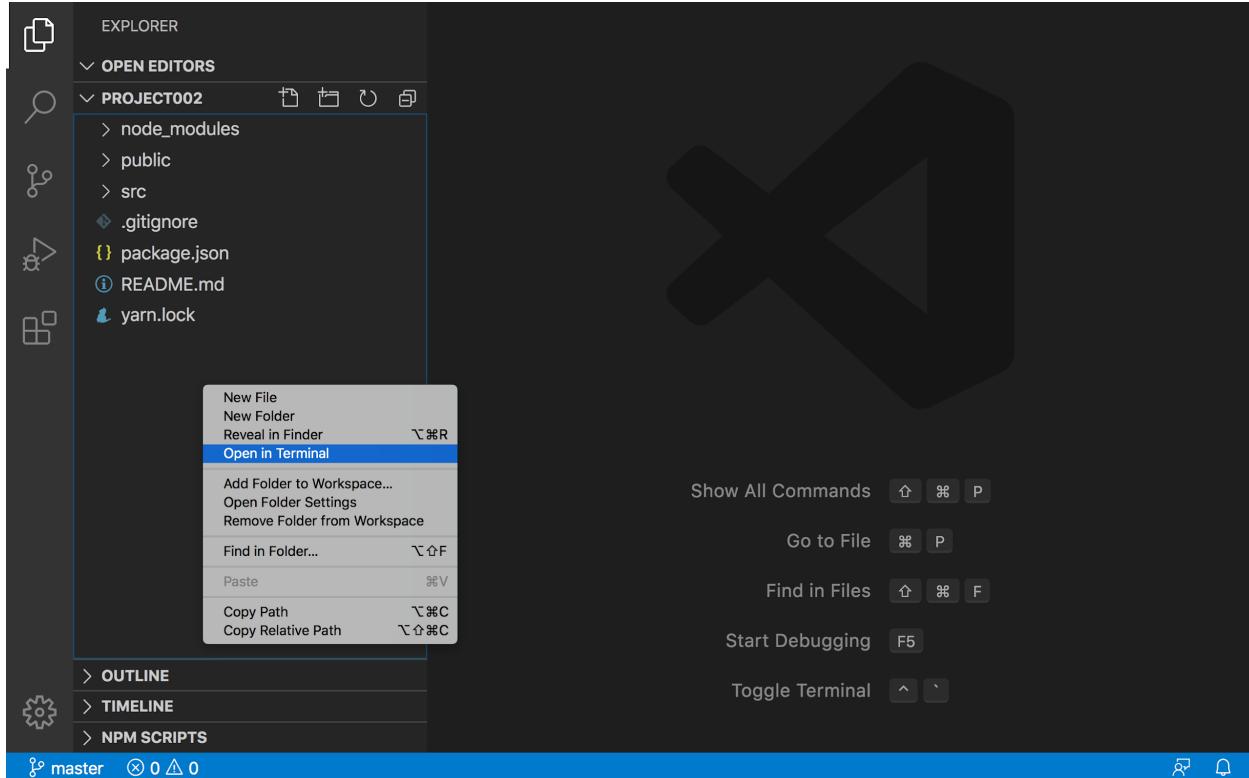
Part II - Open the Project with VS Code





Part III - View the Project in the Browser

First, let's open the project in the terminal



The screenshot shows the Visual Studio Code interface. The left sidebar has icons for Explorer, Open Editors, Project, Search, Git, and Terminal. The 'OPEN EDITORS' section shows 'PROJECT002' expanded, containing 'node_modules', 'public', 'src', '.gitignore', 'package.json', 'README.md', and 'yarn.lock'. The bottom left also lists 'OUTLINE', 'TIMELINE', and 'NPM SCRIPTS'. The main area is a dark-themed terminal window titled 'TERMINAL ...' with tab '2: bash'. The command 'Adil:project002 adil\$ npm start' is typed into the terminal. The status bar at the bottom shows 'master' and other git-related information.

The screenshot shows the Visual Studio Code (VS Code) interface. On the left, the Explorer sidebar displays a project structure for 'PROJECT002' containing 'node_modules', 'public', 'src', '.gitignore', 'package.json', 'README.md', and 'yarn.lock'. The terminal window on the right shows the output of a build process:

```
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.
```

The status bar at the bottom indicates the current branch is 'master' with 0 changes, and there are 0 untracked files.



Part IV - Add Repository to GitHub with the CLI

Step 1: Create a new repository on GitHub

The screenshot shows the GitHub homepage. On the left, there's a sidebar with 'Repositories' and a search bar. Below it, under 'Working with a team?', there's a section about GitHub being built for collaboration and a 'Create an organization' button. The main area shows 'Recent activity' with notifications for completed assignments and followings from users like mnomanfarooq, owsy123, and RooshanAhmed. To the right, there are sections for 'Explore repositories' featuring projects like jina-ai/jina, Python, facebook/react, and Jupyter Notebook. A blue banner at the top right announces 'GitHub is now free for teams'.

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

The screenshot shows the 'Create a new repository' page. At the top, it says 'Create a new repository'. Below that, it asks if you already have a project repository elsewhere and provides a link to 'Import a repository'. The form fields include 'Owner' (set to 'adil-innovation-lab') and 'Repository name *' (set to 'project002'). There's a note about repository names being short and memorable, with an example of 'upgraded-pancake'. The 'Description (optional)' field contains 'Basic React App'. Below that, there are two radio buttons for visibility: 'Public' (selected) and 'Private'. The 'Public' option is described as allowing anyone to see the repository. The 'Private' option is described as allowing the owner to choose who can see and commit. At the bottom, there are several buttons: 'Search', 'Copy', 'Send to My Flow', 'Share...', 'Skip this step if you're importing an existing repository.' (with a note), 'Initialize this repository with a README' (with a note), 'Add .gitignore: None', 'Add a license: None', and a large green 'Create repository' button.

The screenshot shows a GitHub repository page for 'adil-innovation-lab / project002'. At the top, there's a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation is a search bar and a header with the repository name and statistics: 1 unwatched, 0 stars, and 0 forks. A tab bar at the bottom includes Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security (0), Insights, and Settings.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/adil-innovation-lab/project002.git>

Get started by creating a new file or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

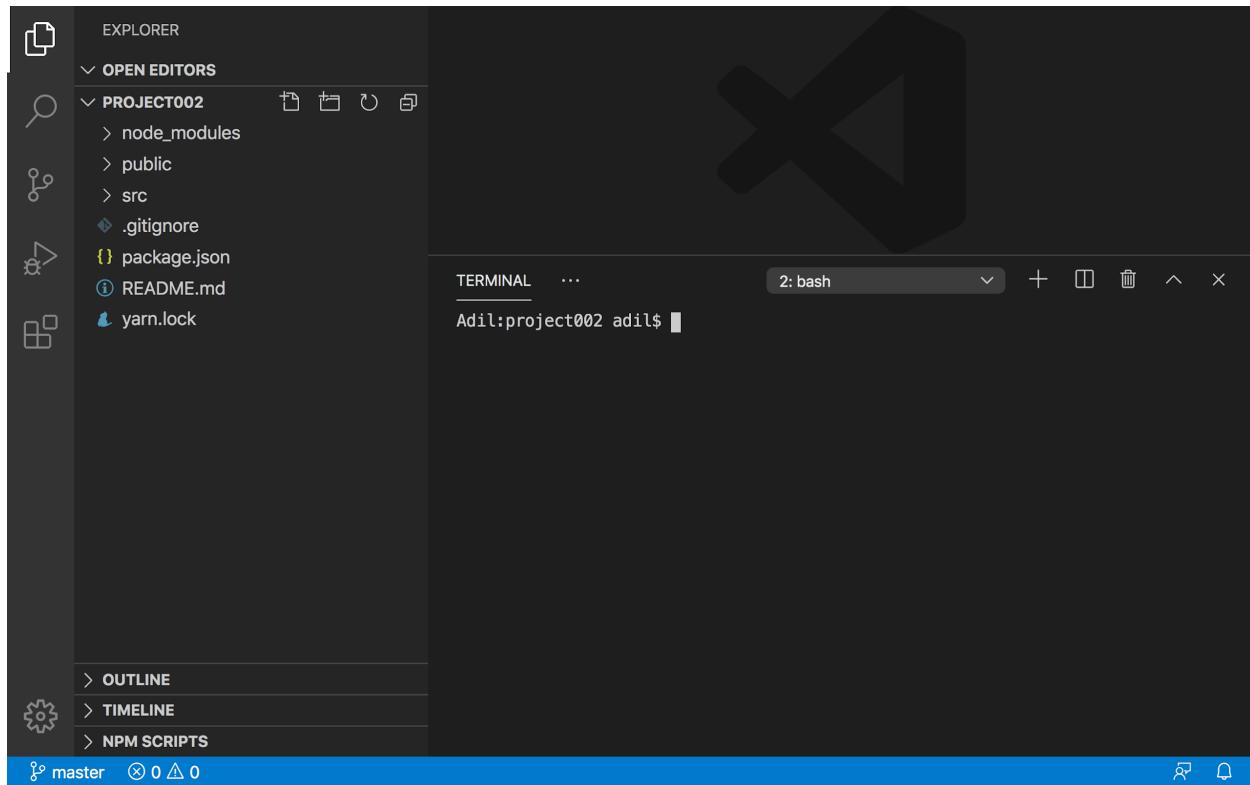
```
echo "# project002" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/adil-innovation-lab/project002.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/adil-innovation-lab/project002.git
git push -u origin master
```

Step 2: Initialize the local directory as a git repository:

Open terminal in our local folder:



A screenshot of the Visual Studio Code interface. On the left is the Explorer sidebar with a tree view of a project named 'PROJECT002' containing files like node_modules, public, src, .gitignore, package.json, README.md, and yarn.lock. Below the tree are sections for Outline, Timeline, and NPM Scripts. The main area shows the VS Code logo. At the bottom is a blue status bar with the text 'master' and some icons.

TERMINAL ...

2: bash

```
Adil:project002 adil$ git init
```

A screenshot of the Visual Studio Code interface, identical to the one above but with output in the terminal. The terminal window shows the command 'git init' being run and its output: 'Reinitialized existing Git repository in /Users/adil/Documents/GitHub/project002/.git/'. The status bar at the bottom remains the same.

TERMINAL ...

2: bash

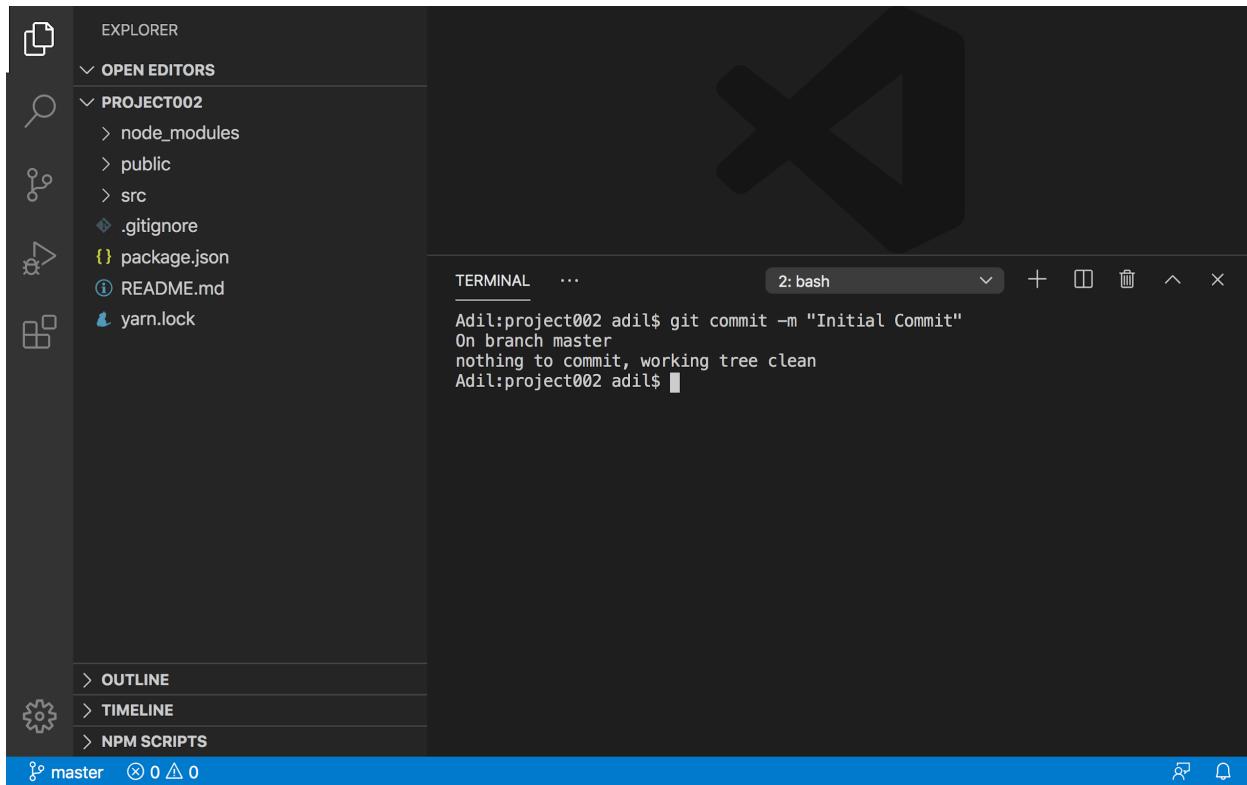
```
Adil:project002 adil$ git init
Reinitialized existing Git repository in /Users/adil/Documents/GitHub/project002/.git/
Adil:project002 adil$
```

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

A screenshot of the Visual Studio Code interface. On the left is the Explorer sidebar, which shows a project named 'PROJECT002' containing files like 'node_modules', 'public', 'src', '.gitignore', 'package.json', 'README.md', and 'yarn.lock'. Below the project list are sections for 'OUTLINE', 'TIMELINE', and 'NPM SCRIPTS'. At the bottom of the sidebar, it says 'master' with a green checkmark icon, '0' changes, and '0' staged changes. The main area is the Terminal, which has a dark background with a large 'X' logo. It shows the command 'Adil:project002 adil\$ git add .' entered. The status bar at the bottom indicates '2: bash'.

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

A screenshot of the Visual Studio Code interface, identical to the previous one but with a different terminal command. The Explorer sidebar shows the same project structure. The Terminal shows the command 'Adil:project002 adil\$ git commit -m "Initial Commit"' entered. The status bar at the bottom indicates '2: bash'.



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

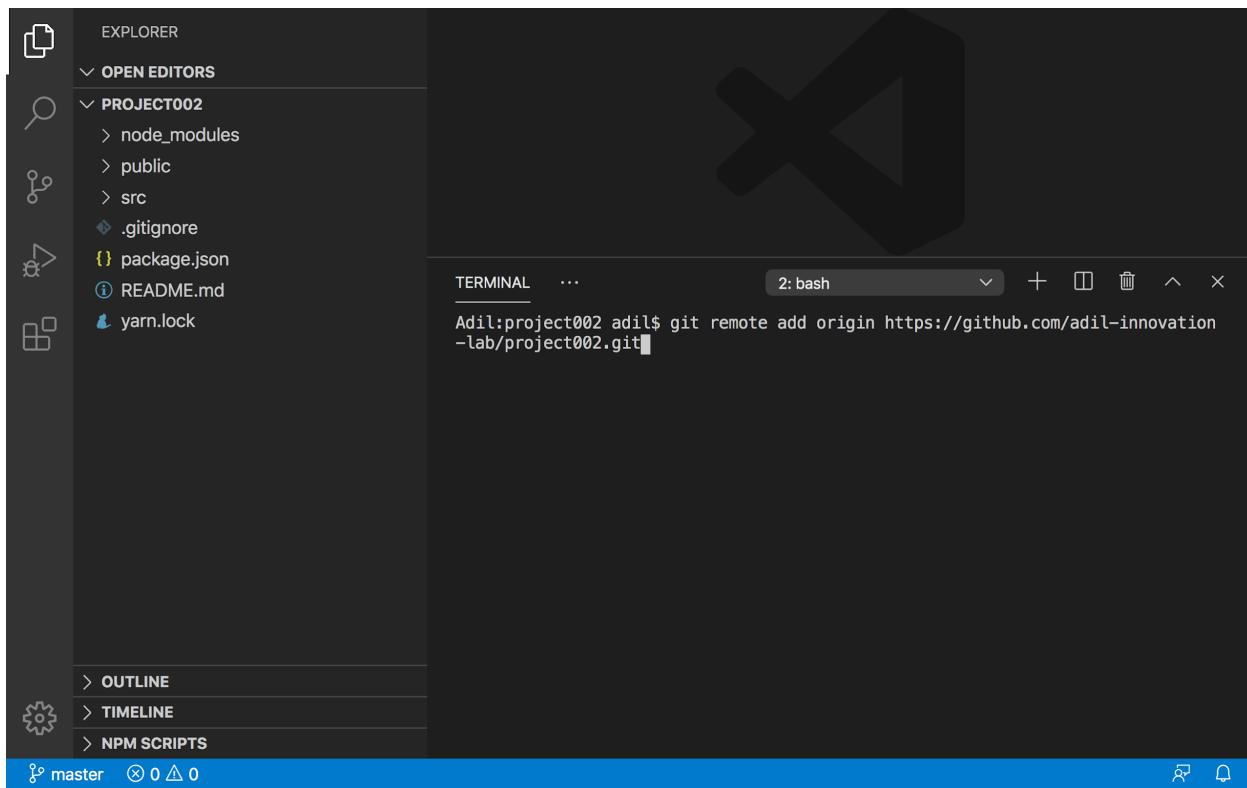
A screenshot of a GitHub repository page for 'adil-innovation-lab / project002'. The page header shows the repository name and a link to 'github.com/adil-innovation-lab/project002'. Below the header, there are buttons for 'Unwatch', 'Star', 'Fork', and a copy icon. A navigation bar below the header includes links for 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area features a 'Quick setup — if you've done t...' section with a 'Copy' button and a highlighted 'HTTPS' link, showing the URL 'https://github.com/adil-innovation-lab/project002.git'. Below this, instructions say 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.' A '...or create a new repository on the command line' section contains the following git commands:

```
echo "# project002" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/adil-innovation-lab/project002.git
git push -u origin master
```

A '...or push an existing repository from the command line' section contains the following git commands:

```
git remote add origin https://github.com/adil-innovation-lab/project002.git
git push -u origin master
```

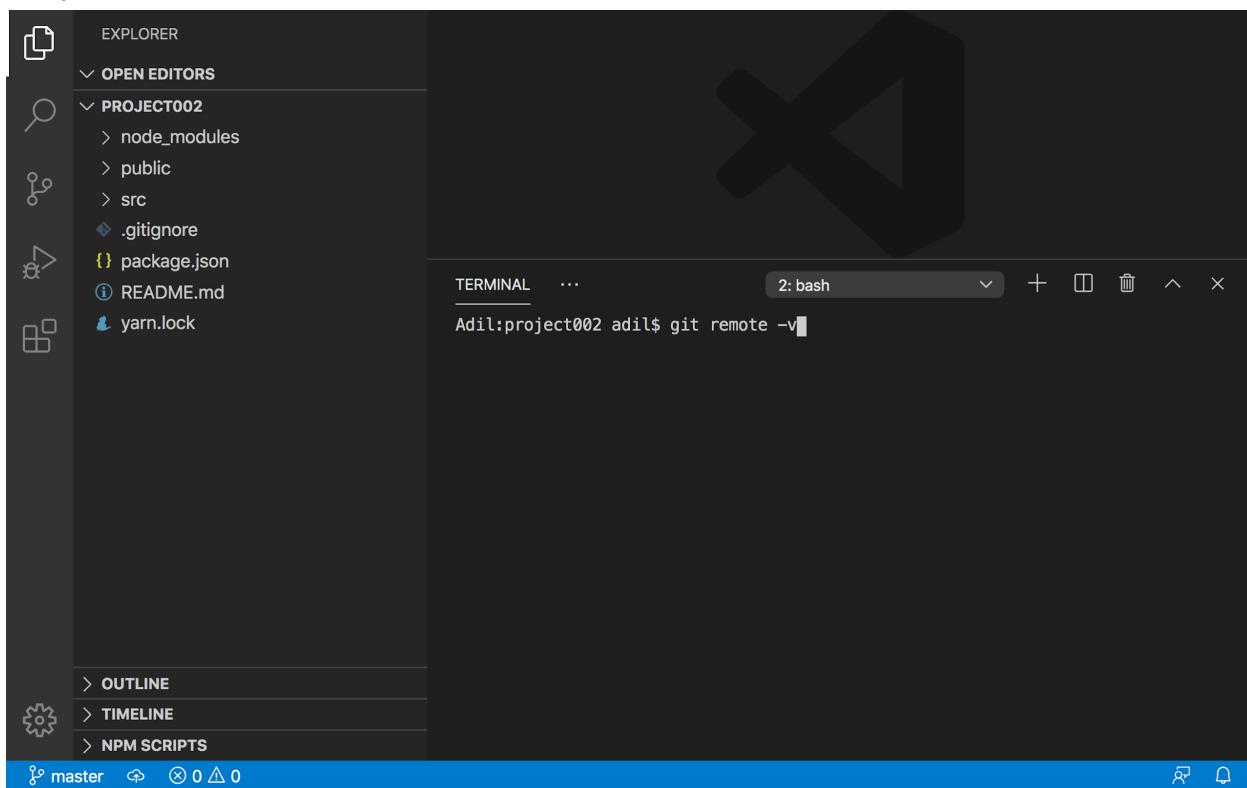
Step 6: Add remote URL to git via the terminal so local repository can be pushed to the right place



The screenshot shows the VS Code interface with a dark theme. On the left is the Explorer sidebar, which displays a project structure for 'PROJECT002' containing files like node_modules, public, src, .gitignore, package.json, README.md, and yarn.lock. Below the project tree are sections for Outline, Timeline, and NPM Scripts. The main area is the Terminal, which shows the command 'git remote add origin https://github.com/adil-innovation-lab/project002.git' being typed. The status bar at the bottom indicates the current branch is 'master'.

```
Adil:project002 adil$ git remote add origin https://github.com/adil-innovation-lab/project002.git
```

Verify the remote URL



This screenshot is identical to the previous one, showing the same VS Code interface and terminal command. The terminal now displays the output of the command, which lists the configured remotes for the repository.

```
Adil:project002 adil$ git remote -v
```

A screenshot of the Visual Studio Code interface. On the left is the Explorer sidebar with a tree view of a project named 'PROJECT002'. The tree includes 'node_modules', 'public', 'src', '.gitignore', 'package.json', 'README.md', and 'yarn.lock'. Below the tree are sections for 'OUTLINE', 'TIMELINE', and 'NPM SCRIPTS'. The main workspace is dark-themed with a large 'VS' logo. At the bottom is a blue status bar showing 'master' and other icons.

TERMINAL ... 2: bash + ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

```
Adil:project002 adil$ git remote -v
```

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

A screenshot of the Visual Studio Code interface, identical to the previous one but with a different terminal command. The Explorer sidebar shows the same project structure. The terminal at the bottom now displays the command 'git push -u origin master'.

TERMINAL ... 2: bash + ⌂ ⌂ ⌂ ⌂ ⌂ ⌂

```
Adil:project002 adil$ git push -u origin master
```

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with 'OPEN EDITORS' expanded, showing a project structure with files like node_modules, public, src, .gitignore, package.json, README.md, and yarn.lock. Below that are sections for OUTLINE, TIMELINE, and NPM SCRIPTS. At the bottom of the sidebar is a status bar showing 'master' and other icons. The main area is a dark-themed terminal window titled 'TERMINAL ...' with the command 'git push -u origin master' run. The output shows the process of pushing changes to a GitHub repository, including object enumeration, counting, compressing, and writing objects, along with a note about a new branch being created.

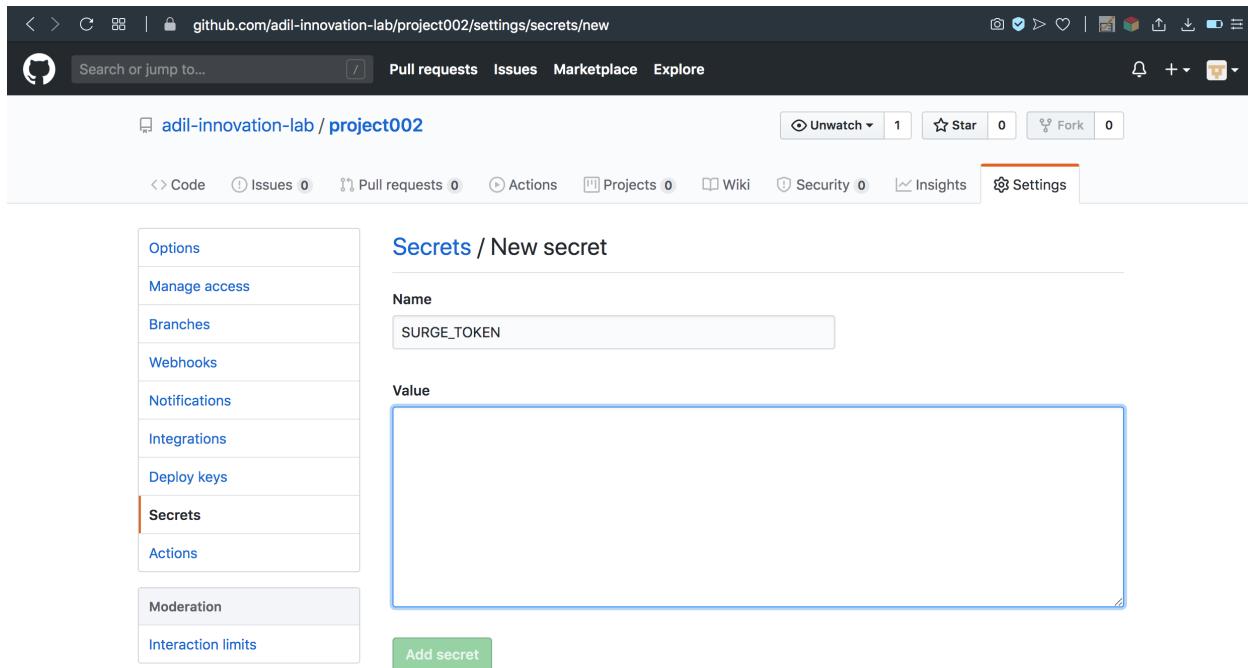
```
Adil:project002 adil$ git push -u origin master
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 4 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (22/22), 200.64 KiB | 6.08 MiB/s, done.
Total 22 (delta 0), reused 0 (delta 0)
To https://github.com/adil-innovation-lab/project002.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
Adil:project002 adil$
```

Step 8: Verify changes pushed to remote git repository

The screenshot shows a GitHub repository page for 'adil-innovation-lab / project002'. The page header includes links for Pull requests, Issues, Marketplace, and Explore. Below the header, there's a summary bar with metrics: 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. A 'Clone or download' button is visible. The main content area displays a commit history for the 'master' branch. The first commit, by 'adil-innovation-lab', initializes the project using Create React App. Subsequent commits show the creation of the public, src, .gitignore, README.md, package.json, and yarn.lock files. The commits were made 42 minutes ago. At the bottom of the page, a note states: 'This project was bootstrapped with [Create React App](#)'.

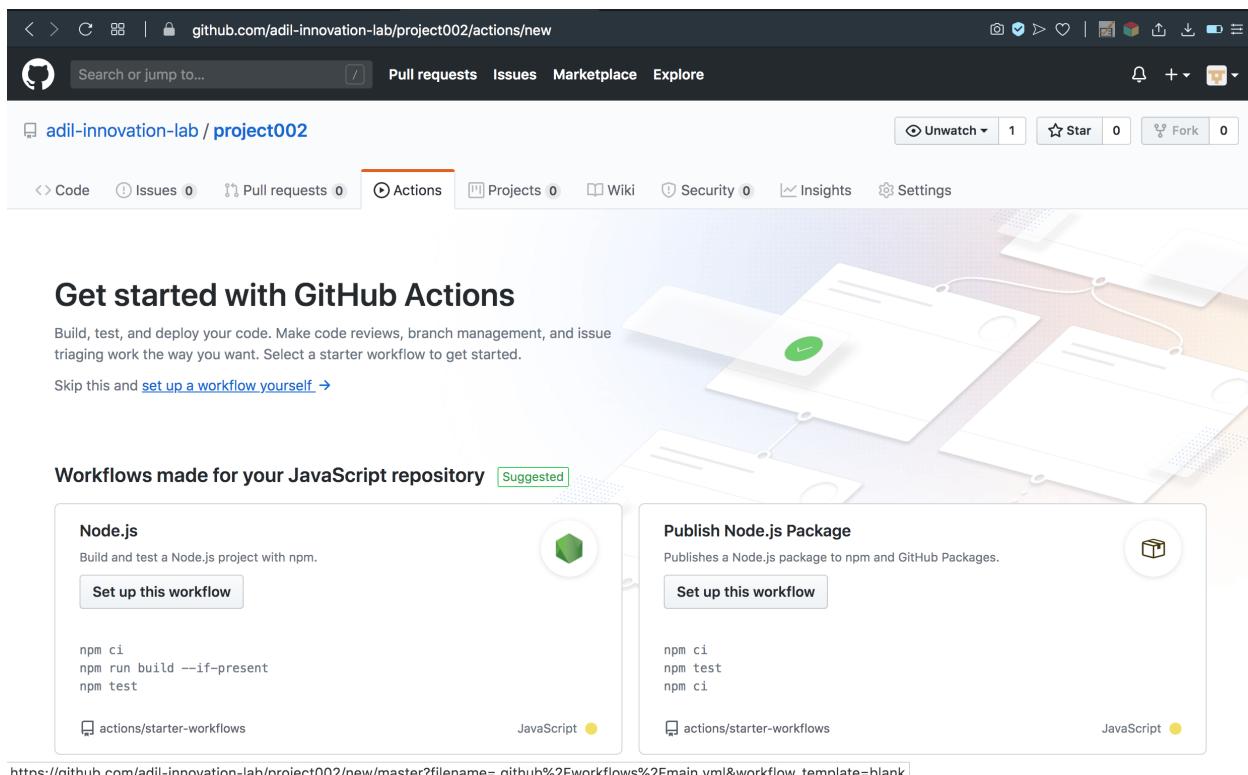
Part V - Deploy React App to Surge with a GitHub Workflow

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



The screenshot shows the GitHub repository settings page for 'adil-innovation-lab / project002'. On the left, a sidebar menu is open with options like Options, Manage access, Branches, Webhooks, Notifications, Integrations, Deploy keys, Secrets (which is selected), Actions, Moderation, and Interaction limits. The main area is titled 'Secrets / New secret'. It has two fields: 'Name' containing 'SURGE_TOKEN' and 'Value' which is currently empty. At the bottom is a green 'Add secret' button.

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



The screenshot shows the GitHub Actions setup page for 'adil-innovation-lab / project002'. The 'Actions' tab is selected in the top navigation bar. The main section is titled 'Get started with GitHub Actions' with the sub-instruction 'Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a starter workflow to get started.' Below this, there's a link 'Skip this and [set up a workflow yourself](#) →'. A large illustration of three smartphones is visible in the background. At the bottom, there are two workflow templates: 'Node.js' (Suggested) and 'Publish Node.js Package'. Both have a 'Set up this workflow' button. The 'Node.js' template includes the command 'npm ci\nnpm run build --if-present\nnpm test'. The 'Publish Node.js Package' template includes the command 'npm ci\nnpm test\nnpm ci'. The URL at the bottom of the page is 'https://github.com/adil-innovation-lab/project002/new/master?filename=.github%2Fworkflows%2Fmain.yml&workflow_template=blank'.

```

1 # This is a basic workflow to help you get started with Actions
2
3 name: CI
4
5 # Controls when the action will run. Triggers the workflow on push or pull request
6 # events but only for the master branch
7 on:
8   push:
9     branches: [ master ]
10    pull_request:
11      branches: [ master ]
12
13 # A workflow run is made up of one or more jobs that can run sequentially or in parallel
14 jobs:
15   # This workflow contains a single job called "build"
16   build:
17     # The type of runner that the job will run on
18     runs-on: ubuntu-latest
19
20   # Steps represent a sequence of tasks that will be executed as part of the job
21   steps:
22     # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it

```

Use **Control + Space** to trigger autocomplete in most situations.

Step 3: Write the YAML for the workflow

```

25   # Install NodeJS
26   - name: Install NodeJS
27     uses: actions/setup-node@v2-beta
28     with:
29       node-version: 12
30
31   # Install Yarn
32   - uses: borales/actions-yarn@v2.0.0
33     with:
34       cmd: install # will run `yarn install` command
35
36   # Build React App with Yarn
37   - name: Build React App
38     run: yarn build
39
40   # Install Surge
41   - name: Install Surge
42     run: npm install --global surge
43
44   # Deploy Surge
45   - name: Deploy React App to Surge
46     run: surge ./build project002-adil.surge.sh --token ${ secrets.SURGE_TOKEN }

```

Use **Control + Space** to trigger autocomplete in most situations.

```

25  # Install NodeJS
26  - name: Install NodeJS
27  uses: actions/setup-node@v2-beta
28  with:
29    node-version: 12
30
31  # Install Yarn
32  - uses: borales/actions-yarn@v2.0.0
33  with:
34    cmd: install # will run `yarn install` command
35
36  # Build React App with Yarn
37  - name: Build React App
38  run: yarn build
39
40  # Install Surge
41  - name: Install Surge
42  run: npm install --global surge
43
44  # Deploy Surge
45  - name: Deploy React App to Surge
46  run: surge ./build project002-adil.surge.sh --token ${secrets.SURGE_TOKEN}

```

Step 4: Commit the workflow to the master branch

Commit new file

Created Workflow Deploy_Surge.yml

Created a workflow to deploy the react app to Surge

⚡ Commit directly to the `master` branch.

🌟 Create a new branch for this commit and start a pull request. [Learn more about pull requests](#).

Commit new file

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

The screenshot shows a GitHub repository page for 'adil-innovation-lab / project002'. The 'Actions' tab is selected, displaying a workflow named 'Created Workflow Deploy_Surge.yml'. A green checkmark icon indicates the workflow has succeeded. The workflow step 'build' is expanded, showing a list of tasks: Set up job, Build borales/actions-yarn@v2.0.0, Run actions/checkout@v2, Install NodeJS, Run borales/actions-yarn@v2.0.0, Build React App, Install Surge, Deploy React App to Surge, Post Run actions/checkout@v2, and Complete job. All tasks are marked with green checkmarks and have execution times listed next to them. The total duration of the workflow is 1m 36s.

| Task | Time |
|-----------------------------------|------|
| Set up job | 3s |
| Build borales/actions-yarn@v2.0.0 | 13s |
| Run actions/checkout@v2 | 6s |
| Install NodeJS | 0s |
| Run borales/actions-yarn@v2.0.0 | 41s |
| Build React App | 13s |
| Install Surge | 17s |
| Deploy React App to Surge | 3s |
| Post Run actions/checkout@v2 | 0s |
| Complete job | 0s |

Part VI - Learn React

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

Part VII - Report Project Completed as an Issue

The screenshot shows a GitHub repository page for 'panacloud/Base-React-App'. The repository has 1 star, 4 forks, and 0 issues. The 'Issues' tab is selected, showing one open issue. The issue is titled 'Adil Altaf submitting assignment #1'. It was opened by 'adil-innovation-lab' 2 minutes ago. The comment section contains a message from 'adil-innovation-lab' providing GitHub and Surge URLs for the project. Below the comment is a text input field for leaving a comment, rich text editing tools, and a file attachment area. A note at the bottom reminds users to follow GitHub Community Guidelines. To the right of the issue are sections for assignees, labels, projects, milestones, and linked pull requests.

panacloud / Base-React-App

Watch 1 Star 4 Fork 0

Code Issues 1 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights

Adil Altaf submitting assignment #1

adil-innovation-lab opened this issue 2 minutes ago · 0 comments

adil-innovation-lab commented 2 minutes ago

GitHub Repo URL: <https://github.com/adil-innovation-lab/project002>
Surge URL: <http://project002-adil.surge.sh>

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Linked pull requests
Successfully merging a pull request may close this issue.
None yet