

GITHUB SETUP

GUIDE

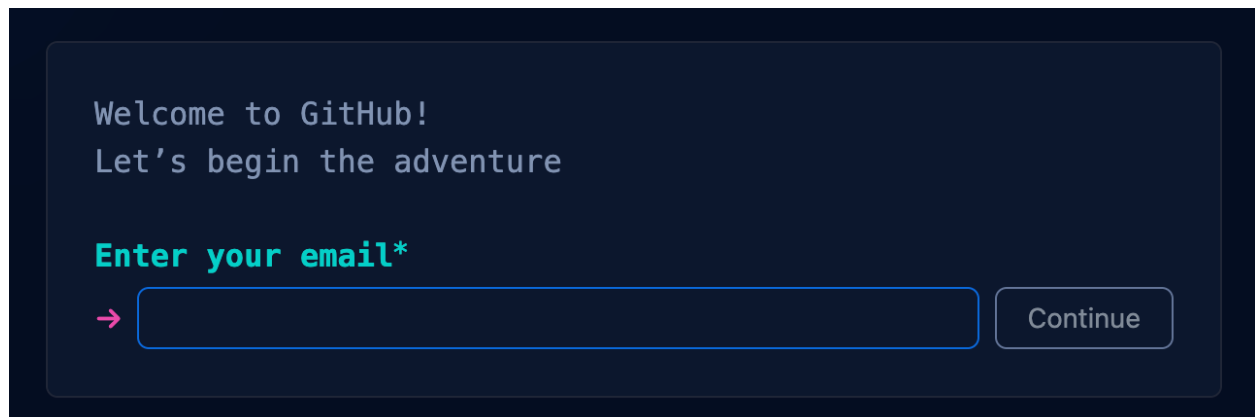
TO DEPLOYING A REACT APP TO GITHUB PAGES

GitHub Pages is a free service from GitHub that allows you to host a website directly from a repository on GitHub. It is a great tool for deploying static websites such as blogs, personal pages, or project documentation.

1. Create a Repository on GitHub.

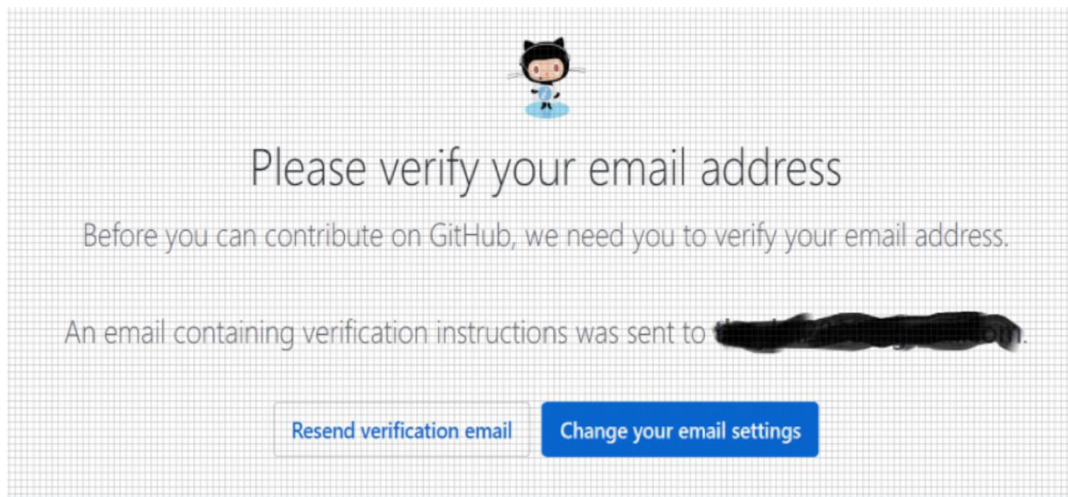
1.1 Create an account and log in to GitHub.

Go to <https://github.com/> to log in or sign up for an account. If you don't have an account, enter your information in the Sign Up form to register.

A screenshot of the GitHub sign-up form. The background is dark blue. The text 'Welcome to GitHub!' and 'Let's begin the adventure' is in a light blue monospace font. Below this, 'Enter your email*' is written in a teal font. There is a red arrow icon to the left of a white rectangular input field. To the right of the input field is a rounded rectangular button with the text 'Continue' in a light blue font.

After providing all the required information and completing the registration process.

GITHUB SETUP



Check your email to see if GitHub has sent a verification link. If so, click on [Verify email address] to proceed with verification. If you haven't received it, go back to the GitHub homepage and click [Resend verification email].

1.2 SIGN IN

Sign in to GitHub

Username or email address

Password

[Forgot password?](#)

Sign in

[Sign in with a passkey](#)

New to GitHub? [Create an account](#)

GITHUB SETUP

2. Create a New Repository.

Click the plus sign icon in the upper-right corner of the screen and select [New Repository].

The screenshot shows the GitHub 'Home' page. At the top, there is a search bar and a navigation bar with icons for repository, codespace, gist, organization, and project. A dropdown menu is open from the repository icon, showing options: 'New repository', 'Import repository', 'New codespace', 'New gist', 'New organization', and 'New project'. The 'New repository' option is selected. Below the navigation bar, the 'Home' section is visible. It includes a 'Start writing code' button and a 'Send feedback' link. The main content area has a heading 'Start a new repository for TQM-Six-Sigma' and a description: 'A repository contains all of your project's files, revision history, and collaborator discussions'. Below this is a form for creating a new repository. It has a 'Repository name' field with the placeholder 'name your new repository...'. There are two radio buttons for visibility: 'Public' (selected) and 'Private'. Below the radio buttons is a green 'Create a new repository' button. At the bottom, there is a section titled 'Introduce yourself with a profile README' with a description: 'Share information about yourself by creating a profile README, which appears at the top of your profile page.' Below this is a text area for the README content, with a 'Create' button. The text area contains a sample README template for 'TQM-Six-Sigma / README.md'.

Home [Send feedback](#)

<> Start writing code

Start a new repository for TQM-Six-Sigma

A repository contains all of your project's files, revision history, and collaborator discussions

Repository name *

name your new repository...

☐ **Public**
Anyone on the internet can see this repository

☒ **Private**
You choose who can see and commit to this repository

[Create a new repository](#)

Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

TQM-Six-Sigma / README.md [Create](#)

```
1 - 🙋 Hi, I'm @TQM-Six-Sigma
2 - 👀 I'm interested in ...
3 - 🌱 I'm currently learning ...
4 - 💖 I'm looking to collaborate on ...
5 - 📫 How to reach me ...
6 - 🗨️ Pronouns: ...
7 - ⚡ Fun fact: ...
8
```


GITHUB SETUP

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 TQM-Six-Sigma ▾

Repository name *

bronxluggage

✓ Your new repository will be created as bronxluggage-.

The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. Need inspiration? How about [bookish-goggles](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)



You are creating a public repository in your personal account.

Create repository

GITHUB SETUP

3. Deploying a React App to GitHub Pages

Open the terminal in Visual Studio Code:

3.1. Install GitHub Pages

```
npm install --save gh-pages
```

3.2. Add to package.json File

```
"homepage": "https://tqm-six-sigma.github.io/bronxluggage/",  
"predeploy": "npm run build",  
"deploy": "gh-pages -d build",
```

```
{  
  "name": "bronxluggage",  
  "version": "0.1.0",  
  "private": true,  
  "homepage": "https://TQM-Six-Sigma.github.io/bronxluggage/",  
  "dependencies": {  
    "@babel/plugin-proposal-private-property-in-object": "^7.16.0",  
    "@fortawesome/free-solid-svg-icons": "^6.6.0",  
    "@fortawesome/react-fontawesome": "^0.2.2",  
    "@testing-library/jest-dom": "^5.17.0",
```

```
    "scripts": {  
      "predeploy": "npm run build",  
      "deploy": "gh-pages -d build",  
      "start": "react-scripts start",  
      "build": "react-scripts build",  
      "test": "react-scripts test",  
      "eject": "react-scripts eject"  
    },
```

GITHUB SETUP

3.3. Push Code to GitHub

- o git init
- o git add .
- o git commit -m "first commit"
- o git branch -M main
- o git remote add origin https://github.com/TQM-Six-Sigma/bronxluggage.git
- o git push -u origin main

3.4. Run the deploy command

- o npm run deploy

After a successful deployment, go back to GitHub: Open your repository on GitHub, click on the "Settings" tab at the top, and scroll down to the "Pages" section. Wait for about 1-5 minutes and you should see the results as shown below.

The screenshot shows the GitHub Pages settings page. On the left is a sidebar with navigation links: General, Access, Collaborators, Moderation options, Code and automation (Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces), Pages (selected), Security (Code security, Deploy keys, Secrets and variables), and Integrations (GitHub Apps, Email notifications). The main content area is titled 'GitHub Pages' and includes the following sections:

- GitHub Pages**: A description stating it's for hosting personal, organization, or project pages. A box shows the site is live at <https://tqm-six-sigma.github.io/bronxluggage/>, last deployed by TQM-Six-Sigma 2 days ago, with a 'Visit site' button.
- Build and deployment**:
 - Source**: A dropdown menu set to 'Deploy from a branch'.
 - Branch**: A text area showing 'gh-pages' and a folder dropdown set to '/ (root)', with a 'Save' button.
 - A link to 'Learn how to add a Jekyll theme to your site.'
 - A note that the site was last deployed to the 'github-pages' environment by the 'pages build and deployment' workflow, with a link to 'Learn more about deploying to GitHub Pages using custom workflows'.
- Custom domain**: A text area for a custom domain, with 'Save' and 'Remove' buttons.
- Enforce HTTPS**: A checked checkbox with a note: 'Required for your site because you are using the default domain (tqm-six-sigma.github.io)'.