

Publishing a Website from React in VSCode to Vercel

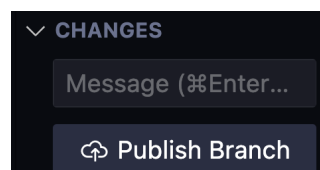
This guide explains how to deploy a modern website built in React, using Visual Studio Code (VSCode), to Vercel – a free web hosting service used by many developers. It is written for students and beginner developers who want to publish a website online with minimal effort and no cost. This guide assumes the user has a functional, publishable website in VSCode, is signed into a GitHub account they created in VSCode, and is ready to publish their site live.

This guide is intended for college students and beginner web developers with basic knowledge of Git and React-based frameworks. The user likely already has their code in VSCode with their GitHub account connected and is ready to publish their site online with either their own domain or a default vercel domain.

They may be working on many assignments or need to quickly finish this part of development, so instructions should be visually clear and to the point. Short paragraphs, bold keywords, and spacing are used to help them follow along and keep their place, even when taking breaks or getting distracted. To follow along a GitHub and Vercel account are needed as well as the React based code being used in VSCode logged into GitHub. Their code must run locally before deployed through this guide and any commit after this should be held to the same standard.

Ensure your code not only runs locally, but also builds locally. Do not commit any secrets or private keys into your repository if it is public. Ensure you are signed into GitHub in VSCode and have the credentials to that GitHub account. Note that GitHub verifies users through the mobile app many times, so this may also be required down the line.

1. In **VSCode** select the '**SOURCE CONTROL**' icon from the left-most menu
 - a. Enter "**First Publish**" as text in the "**Message...**" box. You could also add any other text describing your commit.
 - b. Select "**Commit**" to add your code to your local git.
 - c. Select "**Publish Branch**" to push the latest local commit to GitHub.

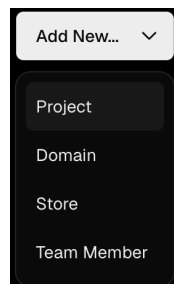


- d. At the top, select either a new **private** or **public** repository, depending on the privacy of the project. When selecting a public repository, ensure no secrets or private keys are in any committed code.

- i. If you have a `.env` file, ensure there is also a `.gitignore` file at the root of your project that contains `".env"`.
2. If you have previously connected Vercel to GitHub and limited Vercel's Access, go to **github.com**
 - a. **Log in** to your account.
 - b. Select the **profile icon** on the **top right** that matches your profile picture.
 - c. Select **"Settings"** from the right menu pane that just opened.
 - d. In the left menu pane, select **"Applications"** under **"Integrations"**. You may have to scroll toward the bottom to see this.
 - e. Select **"Configure"** to the right of **"Vercel"**. If you have only connected Vercel, this will be the only option. If Vercel is not an option, continue to step 3.



- f. Scroll down and ensure the repository you just created is accessible by Vercel, especially if you have limited Vercel's access.
3. Go to **vercel.com**
 - a. **Log in**
 - i. Select **GitHub** when logging in
 - ii. **Authorize** Vercel to access your **GitHub** account
 - b. Create a new project
 - i. Select **"Add New" -> "Project"**.



- c. **Import** your **repository** from GitHub
 - i. Select the name of the repository you just added. This will have the name of the directory your VSCode project is in by default.
 - d. Vercel will detect the framework and automatically prepare for that deployment. If the framework shown does not match your framework, select the icon for the

framework you used. Keep in mind that Next.js is built on React but has its own, separate logo.

- e. Configure Settings
 - i. Leave **most setting** at their **default**
 - ii. Set the build command if it does not match “npm run [found in package.json under scripts]”
 - 1. This is found in VSCode if you navigate to the file at the root directory named “package.json”. This file has a section of scripts and one will be what you used to run your project locally. Select this option.
 - iii. If you have **environment variables** (eg. a .env file) add them to the project.
 - 1. Copy your entire **.env**.
 - 2. Paste it into the environment variables section.
 - f. Select “**Deploy**”
 - i. Vercel will install the dependencies your project uses and build the site efficiently, including any environment variables added in the last step
 - g. View your site by following the url shown under the site preview. For a quick view, simply click the preview.
4. Set up a **custom domain**, if you would like.
- a. Navigate to “**Settings**” in your project view.
 - b. Under “Domains” select “**Add**”.
 - c. You can either register a domain, connect one you own, or edit the vercel default url.
 - i. To add a custom domain you own, navigate to the site where you bought it and follow Vercel’s instructions, adding cname and other tags.

If the deployment fails, check that the code you pushed is not broken, your project builds correctly locally, and your environment variables are correctly set up.

Deploying to vercel is a simple, mostly straightforward way to get your website to the internet without any cost. From now on, every push to GitHub will result in a rerender of the site. If the site is broken in the latest push, the last successful build will be used as a fallback. Vercel makes this process simple and repeatable for developers, especially those using GitHub to store their codebase. It is a powerful and low effort way to manage a plethora of projects in one centralized place.