GitHub & Vercel Deployment Notes (Step-by-Step Guide)

For Frontend Projects (like MERN Blog)

## Step 1: Check Git Version

Open CMD / PowerShell / VS Code terminal and type:  
  
git --version  
  
Then type:  
  
git  
  
This will display all available Git commands.

## Step 2: Create a GitHub account

#### 1. ****Go to GitHub Website****

Open your browser and go to:  
[**https://github.com/**](https://github.com/)

#### 2. ****Click "Sign up"****

You'll see a **"Sign up"** button on the top right corner — click on it.

#### 3. ****Enter Your Details****

**Email address:** Enter your valid email.

**Password:** Create a strong password.

**Username:** Choose a unique username (it will be your GitHub URL too).

**Verify you're human** (CAPTCHA).

#### 4. ****Verify Your Email****

Check your email inbox for a verification code from GitHub.

Enter the code on the GitHub page.

#### 5. ****Choose Plan****

Select the **Free plan** (sufficient for most users).

Click **Continue**.

#### 6. ****Finish Setup****

You may be asked a few optional questions (like your experience, how you

plan to use GitHub).

Click **Submit** or **Skip** to finish.

## Step 3: Create a GitHub Repository

1. Go to: https://github.com  
   2. Click on `+` > New repository  
   3. Give a name like: blog\_front\_end\_eveing or mern-blog  
   4. Do NOT check “Initialize with README”  
   5. Click Create repository

Note : Click the repository name

1. GitHub will give you a link like:  
   <https://github.com/yourusername/your-repo-name.git>

## Step 4: Initialize Git in Project Folder

Open Vs Code terminal and go to your project folder:  
cd D:\BBD\bbd\_blog\_eveing

Then create . gitignore file add those file not want add / push on git Repository

## Create `.gitignore` File

.env  
node\_modules  
  
# Logs  
logs  
\*.log  
npm-debug.log\*  
yarn-debug.log\*  
yarn-error.log\*  
pnpm-debug.log\*  
lerna-debug.log\*  
  
dist  
dist-ssr  
\*.local  
  
# Editor directories and files  
.vscode/\*  
!.vscode/extensions.json  
.idea  
.DS\_Store  
\*.suo  
\*.ntvs\*  
\*.njsproj  
\*.sln  
\*.sw?

## Step 5: Initialize Git in Project Folder

Now run:  
git init  
  
You will see:  
Initialized empty Git repository in D:/BBD/bbd\_blog\_eveing/.git/

Now it will create a local repo displaying this

Initialized empty Git repository in D:/BBD/bbd\_blog\_eveing/front-end/.git/

## Step 6: Add Files to Git

git add .

(Use . to add all files. You can also add files individually.)

We can track status of file . using this command  
git status

## Step 7: Commit Your Code

git commit -m "Initial commit"  
  
You can write any meaningful commit message here.

"Initial commit" => It is not mandatory to write "Initial commit" only.

You can write any meaningful message.

## Step 8: Set Branch to `main`

git branch -M main (my branch is default master)

Note: If master is present then we can create branch

git branch -M master

## Step 9: Connect to GitHub (Add Remote Origin)

Click Own repository name and see down

Replace URL with your own repo URL:  
git remote add origin https://github.com/yourusername/your-repo-name.git  
  
Example:  
git remote add origin https://github.com/rajeevkumar/blog\_front\_end\_eveing.git  
  
Note: You can check whether the origin has been successfully set or not using the command

git remote -v

## Step 10: Push Code to GitHub

git push -u origin main (here main replace with master)

You can check whether the origin has been successfully set or not using the command

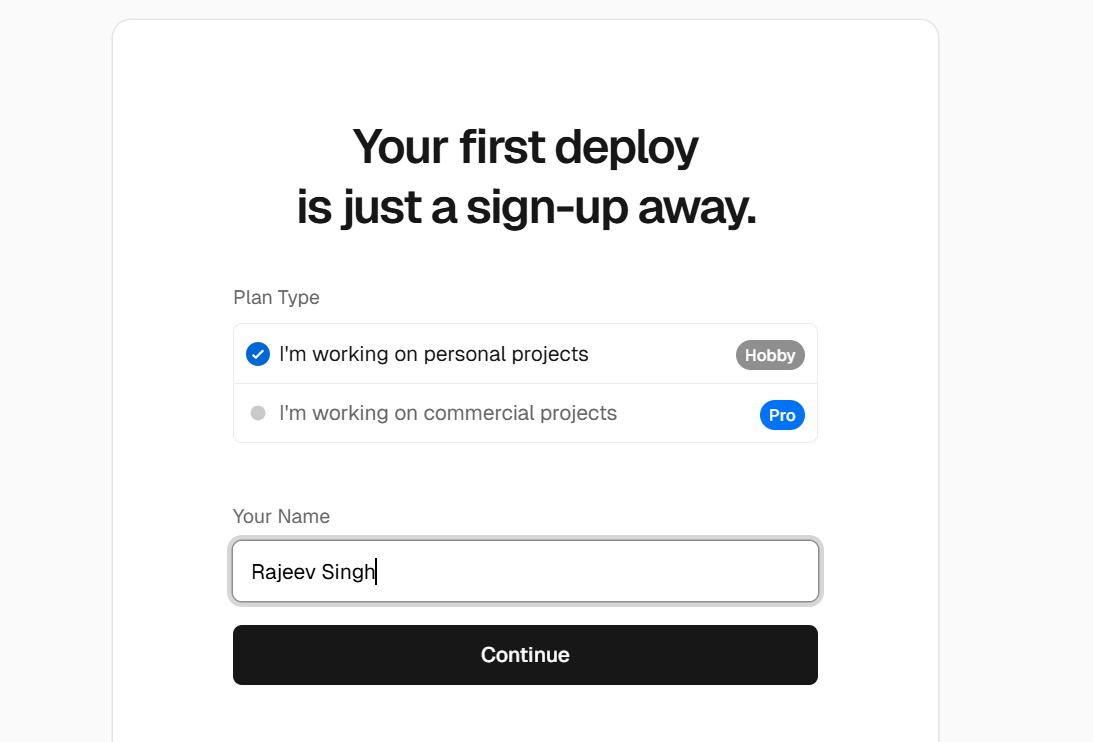
git remote -v

If GitHub asks for username & password:  
- Enter your GitHub username  
- For password, use Personal Access Token (PAT)

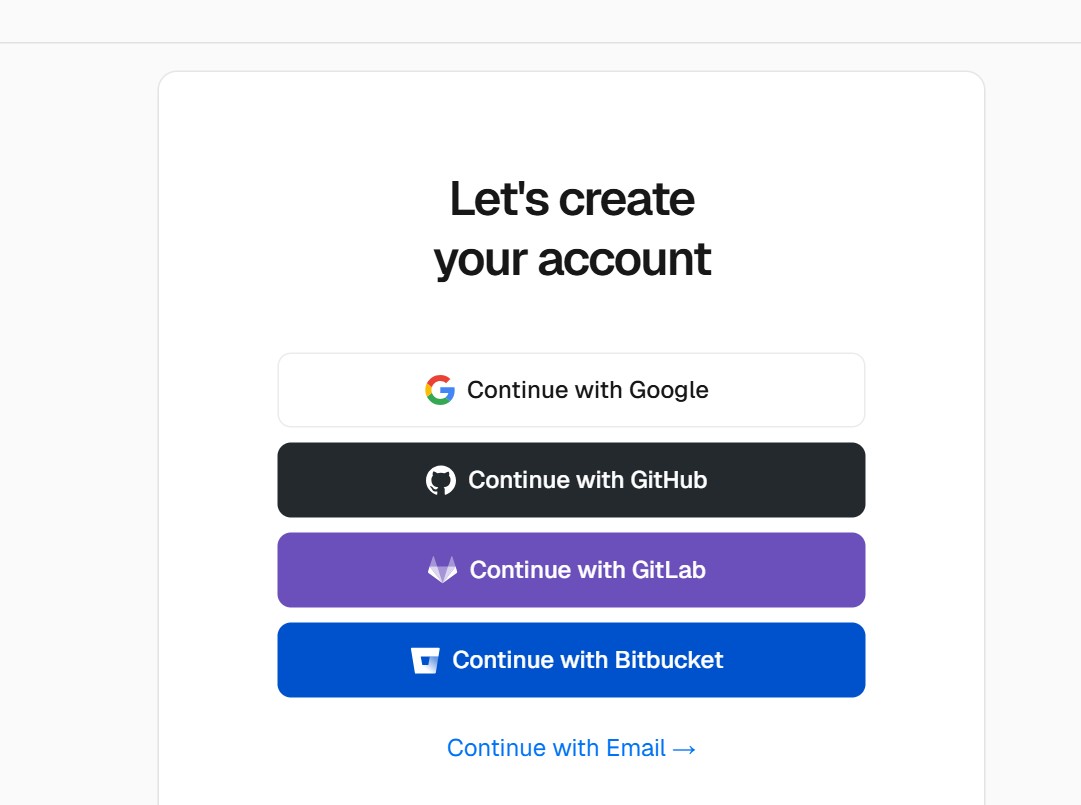
* Done! Your project is now live on GitHub.

## Step 2: Sign Up on Vercel

1. Go to: <https://vercel.com>



1. Click on "Continue with GitHub"



3. Authorize your GitHub account  
4. You're now ready to deploy projects from GitHub to Vercel

## Step 11: Deploy to Vercel

1. Go to https://vercel.com/dashboard  
2. Click on "Add New Project"  
3. Click "Import Git Repository"  
4. Click Install  
5. Select only the repository you want to deploy  
6. Click Import

## Step 12: Set `.env` Variables (if needed)

1. Go to Settings > Environment Variables  
2. Add key-value pairs like:  
 - VITE\_API\_URL = http://localhost:5000/api  
 - VITE\_MODE = production  
3. Click Save and Redeploy

## Done!

Your frontend project is now live on GitHub and Vercel.

## Bonus Tips:

You can add a README.md for project info.  
To push updates:  
git add .  
git commit -m "your message"  
git push