Documentation Generator - GitHub Setup Guide

Project Structure

Create the following folder structure for your project:

```
doc-generator/
├── public/
│   └── index.html
├── src/
│   ├── App.jsx
│   └── index.js
├── .gitignore
├── package.json
├── README.md
├── LICENSE
└── tailwind.config.js
```

Step-by-Step Setup

1. Initialize Your Local Project

```
bash

# Create project directory
mkdir doc-generator
cd doc-generator

# Initialize git repository
git init

# Initialize npm project
npm init -y
```

2. Install Dependencies

```
bash
```

```
npm install react react-dom
npm install -D vite @vitejs/plugin-react tailwindcss postcss autoprefixer
npm install lucide-react

# Initialize Tailwind CSS
npx tailwindcss init -p
```

3. Create Configuration Files

package.json - Update with these scripts:

```
json

{
  "name": "doc-generator",
  "version": "1.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "lucide-react": "^0.263.1"
  },
  "devDependencies": {
    "@vitejs/plugin-react": "^4.0.0",
    "autoprefixer": "^10.4.14",
    "postcss": "^8.4.24",
    "tailwindcss": "^3.3.2",
    "vite": "^4.3.9"
  }
}
```

vite.config.js:

```
javascript
```

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

export default defineConfig({
  plugins: [react()],
})
```

tailwind.config.js:

```javascript
/** @type {import('tailwindcss').Config} */
export default {
  content: [
    "./index.html",
    "./src/**/*.{js,ts,jsx,tsx}",
  ],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

.gitignore:

```
# Dependencies
node_modules/

# Build output
dist/
build/

# Environment variables
.env
.env.local
.env.*.local

# IDE
.vscode/
.idea/
*.swp
*.swo
*~

# OS
.DS_Store
Thumbs.db

# Logs
logs/
*.log
npm-debug.log*
```

4. Create Source Files

public/index.html:

```
html
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Documentation Generator</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/index.js"></script>
  </body>
</html>
```

src/index.js:

```javascript
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
)
```

src/index.css:

```css
```

```css
@tailwind base;
@tailwind components;
@tailwind utilities;

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
```

src/App.jsx - Copy the component code from the artifact I created earlier.

5. Create README.md

See the README file I'll create separately for the complete content.

6. Create GitHub Repository

Option A: Using GitHub CLI

```bash
bash

# Install GitHub CLI if you haven't: https://cli.github.com/

# Authenticate
gh auth login

# Create repository
gh repo create doc-generator --public --source=. --remote=origin

# Push code
git add .
git commit -m "Initial commit: Documentation Generator"
git push -u origin main
```

Option B: Using GitHub Website

1. Go to https://github.com/new

2. Create a new repository named "doc-generator"

3. Don't initialize with README (we already have one)

4. Run these commands:

```bash
git add .
git commit -m "Initial commit: Documentation Generator"
git branch -M main
git remote add origin https://github.com/YOUR_USERNAME/doc-generator.git
git push -u origin main
```

7. Add Environment Variables (Optional)

If you want to add API key support, create a `.env.example` file:

```
# .env.example
VITE_ANTHROPIC_API_KEY=your_api_key_here
```

8. Test Locally

```bash
# Run development server
npm run dev

# Build for production
npm run build

# Preview production build
npm run preview
```

9. Deploy (Optional)

Deploy to Vercel:

```bash
npm install -g vercel
vercel
```

Deploy to Netlify:

```bash
npm install -g netlify-cli
netlify deploy --prod
```

Quick Commands Reference

```bash
# Clone your repo (after pushing to GitHub)
git clone https://github.com/YOUR_USERNAME/doc-generator.git
cd doc-generator

# Install dependencies
npm install

# Run locally
npm run dev

# Build for production
npm run build
```

Important Notes

⚠️ **API Key Security**: The current implementation uses the Anthropic API without requiring an API key (handled by Claude.ai). If you deploy this publicly, you should:

- Add proper API key authentication
- Implement rate limiting
- Add backend API proxy to protect your keys

Next Steps

1. ✅ Create local project structure
2. ✅ Copy all files
3. ✅ Test locally with `npm run dev`
4. ✅ Create GitHub repository

5. ✅ Push your code

6. 🎉 Share your project!

Repository URL: https://github.com/YOUR_USERNAME/doc-generator

Replace `YOUR_USERNAME` with your actual GitHub username.