

Build a SaaS using Cursor

Build a Complete Software App in 10 Steps

A step-by-step guide using GPT-5 powered Cursor, GitHub, VS Code, Node.js, Git Bash, Docker, Supabase, Neon, Render & Netlify

Quick Links

-  [Cursor](#) — AI-powered coding environment
 -  [GitHub](#) — Code hosting & version control
 -  [VS Code](#) — Local code editor
 -  [Node.js](#) — JavaScript runtime & npm package manager
 -  [Git for Windows \(Git Bash\)](#) /  [Git SCM](#) — Git CLI tools
 -  [Docker](#) — Containerized backend testing & deployment
 -  [Supabase](#) — Cloud backend service for auth, APIs & database
 -  [Neon](#) — Cloud-hosted PostgreSQL with Netlify integration
 -  [Render](#) — Cloud hosting for backend apps
 -  [Netlify](#) — Hosting & deployment for frontend apps
-

Prerequisites

- Cursor account
 - GitHub account
 - Node.js installed
 - (Optional) Docker installed for backend testing
 - (Optional) Custom domain for deployment
-

Step-by-Step Guide

- Go to [Cursor](#)
- Create a new project & describe your app in plain English.
- Example:

"Build a full-stack task manager app — React frontend, Node.js backend, PostgreSQL database."

- Connect **Supabase** or **Neon** if you need a database.
 - Cursor will generate your **frontend, backend, and database schema** automatically.
-

- Link your [GitHub](#) account inside Cursor.
 - Choose/create a repository.
 - Cursor will automatically push code changes to GitHub.
-

- Install [VS Code](#)
- Clone your GitHub repo:

```
git clone <https://github.com/yourusername/your-repo.git>
```

- Open the folder in VS Code to edit code locally.
-

- Download & install [Node.js](#)
- In your project folder:

```
npm install  
npm run dev # or npm start
```

- Install [Git Bash](#)
- Use it to pull, commit, and push updates:

```
git pull  
git add .  
git commit -m "Updated project"  
git push
```

-
- Install [Docker](#)
 - Create a [Dockerfile](#) and `docker-compose.yml` for your backend.
 - Run:

`docker compose up`

- Test APIs locally before deploying.

Option A — Supabase (supabase.com)

- Auth, APIs, database all in one.

Option B — Neon (neon.tech)

- Dedicated PostgreSQL, perfect for Netlify integration.

Add your database credentials to your `.env` file.

-
- Go to [Render](#)
 - Create a new **Web Service** → connect GitHub repo.
 - Add environment variables (DB URL, API keys).
 - Deploy.

-
- Go to [Netlify](#)
 - Import GitHub repo.
 - Build command:

`npm run build`

- Publish directory:

`build`

(or `dist` / `.next` depending on framework)

- Connect a custom domain.

-
- Your **frontend** is live on Netlify.
 - Your **backend** is live on Render.
 - Your **database** is hosted on Supabase or Neon.

Follow [Yash Sanghavi](#) for more such content