WORKSHOP BUILD GUIDE

From GitHub repo to live Next.js app

A step-by-step learner guide for creating a repository, connecting the folder in VS Code,using Claude to build the app, adding Supabase, pushing to GitHub, and deploying through Vercel.

GitHub repo and history	**VS Code** project folder	**Claude** AI build partner
Next.js app framework	**Supabase** database	**Vercel** deployment

The full build path

1 **GitHub repo** Create the online project home.	2 **VS Code folder** Connect the folder to the repo remote.	3 **Claude build** Scaffold the Next.js app.	4 **Supabase schema** Create the database table.
5 **.env.local** Add project URL and anon key.	6 **Push code** Send the working version to GitHub.	7 **Vercel deploy** Import, add keys, and deploy.	8 **Live test** Submit, view, filter, refresh.

STEP 01

Create the GitHub repository

GitHub is the online home for the project code.Before deploying, the project needs a repository so the work can be saved, shared, and connected to Vercel.

1. Go to github.com and sign in.
2. Create a new repository.
3. Use a clear name such as bank-ideas.
4. Keep the repository public or private based on the trainer's instruction.
5. Do not add secret keys or passwords to the repository.

```
Repository name example:
bank-ideas
```

Keep in mind
A repository is not the app itself. It is the saved code history for the app.

STEP 02

Connect your VS Code folder to GitHub

The folder on the laptop needs to know which GitHub repository it belongs to.This connection is called a remote.

1. Open the project folder in VS Code.
2. Open the built-in terminal.
3. Initialize Git if the folder is not already a Git project.
4. Add the GitHub repository URL as the remote.
5. Check that the remote was added correctly.

```
git init
git remote add origin https://github.com/YOUR-USERNAME/bank-ideas.git
git remote -v
```

Keep in mind
If the folder already has a remote, do not add a second one blindly. Ask Claude or the trainer to inspect it first.

STEP 03

Make sure the Claude extension is ready

Claude will help plan and generate the first version of the app.Before asking it to build, confirm the extension can see the project folder.

1. Open VS Code Extensions.
2. Search for the Claude extension approved by the trainer.
3. Install it if it is missing.
4. Sign in if the extension asks you to.
5. Open Claude from inside VS Code while the project folder is open.

[ ] VS Code is open on the correct folder.

[ ] Claude extension is installed.

[ ] Claude can read the project files.

[ ] Terminal is available inside VS Code.

STEP 04

Ask Claude to create the Next.js app

Start with the app structure and user interface.Tell Claude that Supabase and Vercel will be added later, so it should keep the first version ready for integration.

[ ] The app runs locally.

[ ] The page has a clear submit form.

[ ] Ideas appear in a feed or card layout.

[ ] Category filtering is visible.

[ ] The UI is clean enough before database work starts.

```
Act as a beginner-friendly full-stack coach. Create a simple Next.js app in this
folder for an internal Bank Ideas platform.

The app should let employees submit an idea with title, short description, and
category. It should also show submitted ideas in a simple feed and let users filter
by category.

For now, use local sample data or simple state so I can see the app working. Do not
connect Supabase yet, but structure the code so we can integrate Supabase and
deploy to Vercel later.

Before editing, explain the files you will create or change. Then build the
smallest clean version.
```

Keep in mind
Make the first version look good before connecting services. It is easier to debug one layer at a time.

STEP 05

Create the Supabase project

Supabase is where the app stores real data.The project contains the database, API settings, and keys the app uses to talk to Supabase.

1. Go to supabase.com and sign in.
2. Create a new project.
3. Use a simple project name such as bank-ideas.
4. Open the SQL Editor from the project sidebar.
5. Keep the Project URL and anon public key ready for the environment file.

STEP 06

Ask Claude for the Supabase schema

The schema defines what the database table looks like.Ask Claude for a simple first table, then paste it into Supabase SQL Editor.

1. Copy the final SQL from Claude.
2. Paste it into the Supabase SQL Editor.
3. Run the SQL.
4. Open the Table Editor and confirm the ideas table exists.

```
Ask Claude:
Create a simple Supabase SQL schema for my Bank Ideas app. Each idea needs a title,
description, category, and created date. Keep it beginner-friendly and explain each
column.

Starter SQL example:
create table ideas (
id uuid primary key default gen_random_uuid(),
title text not null,
description text not null,
category text not null,
created_at timestamptz not null default now()
);
```

Keep in mind
This table is a starter structure. Later features such as votes, comments, or statuses may need extra columns or extra tables.

STEP 07

Connect Supabase keys in .env.local

Environment variable sare settings the app reads when it runs.They let the code use the correct Supabase project without hardcoding keys directly into app files.

1. Create a file called .env.local in the main project folder.
2. Add the Supabase Project URL.
3. Add the Supabase anon public key.
4. Restart the local development server after changing the file.
5. Do not commit .env.local to GitHub.

```
NEXT_PUBLIC_SUPABASE_URL=your-project-url
NEXT_PUBLIC_SUPABASE_ANON_KEY=your-anon-public-key
```

Keep in mind
NEXT_PUBLIC values can be used by browser code. Do not put private service role keys in browser code.

STEP 08

Ask Claude to integrate Supabase

Now the app can move from temporary sample data to real saved data.Ask Claude to connect only the main idea flow first.

[ ] Submitting an idea saves it to Supabase.

[ ] Refreshing the page still shows saved ideas.

[ ] The category filter works with saved data.

[ ] Errors are visible if Supabase is not configured correctly.

```
Act as a beginner-friendly full-stack coach. Connect this Next.js Bank Ideas app to
Supabase.

Use the ideas table with title, description, category, and created_at. The app
should save a new idea when the form is submitted, load ideas from Supabase, and
keep the category filter working.

First explain which files need to change and why. Then make the smallest clean
change. Include loading, success, and error states.
```

STEP 09

Push the code to GitHub

After the app works locally,save the current checkpoint to GitHub. This gives Vercel something to deploy.

1. Check what changed.
2. Add the files.
3. Commit the checkpoint.
4. Push the code to GitHub.

```
git status
git add .
git commit -m "build bank ideas app"
git push -u origin main
```

Keep in mind
If Git says the branch name is master instead of main, ask Claude or the trainer before changing commands.

STEP 10

Deploy with Vercel

Vercel builds the Next.js app and gives it a live URL. Because Vercel connects to GitHub, future pushes can update the deployed app.

1. Go to vercel.com and sign in with GitHub.
2. Import the Bank Ideas repository. Vercel should detect that it is a Next.js project.
3. Open the Environment Variables section.
4. Add the same Supabase URL and anon public key used in .env.local.
5. Run the deployment.
6. Open the live URL and test the app.

```
NEXT_PUBLIC_SUPABASE_URL=your-project-url
NEXT_PUBLIC_SUPABASE_ANON_KEY=your-anon-public-key
```

STEP 11

Test the live app

A successful deployment means the app built. It does not prove the workflow works. Test the live app the same way a real user would.

[ ] The live URL opens.

[ ] A new idea can be submitted.

[ ] The submitted idea appears in the feed.

[ ] The category filter works.

[ ] The idea stays visible after refresh.

[ ] No secret keys or passwords were pushed to GitHub.

Prompt pack

Use these prompts when you get stuck or when a step needs to be reviewed before moving forward.

```
Explain the project structure in beginner-friendly language. Tell me what each main
folder or file is responsible for, and what I should avoid changing right now.
```

```
Review this change before I continue. Tell me what it changes, what could break,
and how I can test it in the browser.
```

```
The app works locally but not on Vercel. Help me check environment variables,
Supabase connection settings, build errors, and what to test next. Ask me for one
piece of information at a time.
```

Final mental model
VS Code is where you edit. Claude helps you plan and build. Next.js is the app. Supabase is the database. GitHub saves the code. Vercel publishes the app.