Full-stack Development

GServe 2024

Install Prerequisites

Node JS: https://nodejs.org/en

Git: https://www.git-scm.com/downloads

VS Code: https://code.visualstudio.com/download

Other Prerequisites:

Browser (eg:Chrome)

Gmail Account for Cloud Login

Goals

Goals

- **Introduce** the students to the concepts of web development and deployment
- **Familiarize** the students with various technologies and giving them a quick start tutorial
- **Inculcate curiosity** and enable **self-learning** and exploration
- NOT a comprehensive course by any means
- NOT to cover every concept of each topic discussed here.

How?

- Build and deploy a basic website using Vercel & Supabase
- Use readily available courses on the internet and reference them so students can find the material easily

Topics

Topics

Development:

- Development environment (VSCode)
- React / NextJS
- Tailwind CSS
- Node JS
- PostgreSQL

Deployment:

- Version Control (GIT / GitHub)
- Web Hosting
  - Vercel
  - Supabase
- Continuous Deployment

Dev Environment (VSCode)

Microsoft Visual Studio Code is a free, powerful, lightweight code editor for Windows, macOS and Linux. Based on open source, it is highly customizable with several extensions and supports multiple coding languages.

https://code.visualstudio.com/

NodeJS

Node JS

- Node.js is an open source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server

NPM

NPM is a package manager for Node.js packages. www.npmjs.com hosts thousands of free packages to download and use.

- The NPM program is installed on your computer when you install Node.js
- A **package** in Node.js contains all the files you need for a module.
  - Modules are JavaScript libraries you can include in your project.
- More Info:
  - https://www.w3schools.com/nodejs/

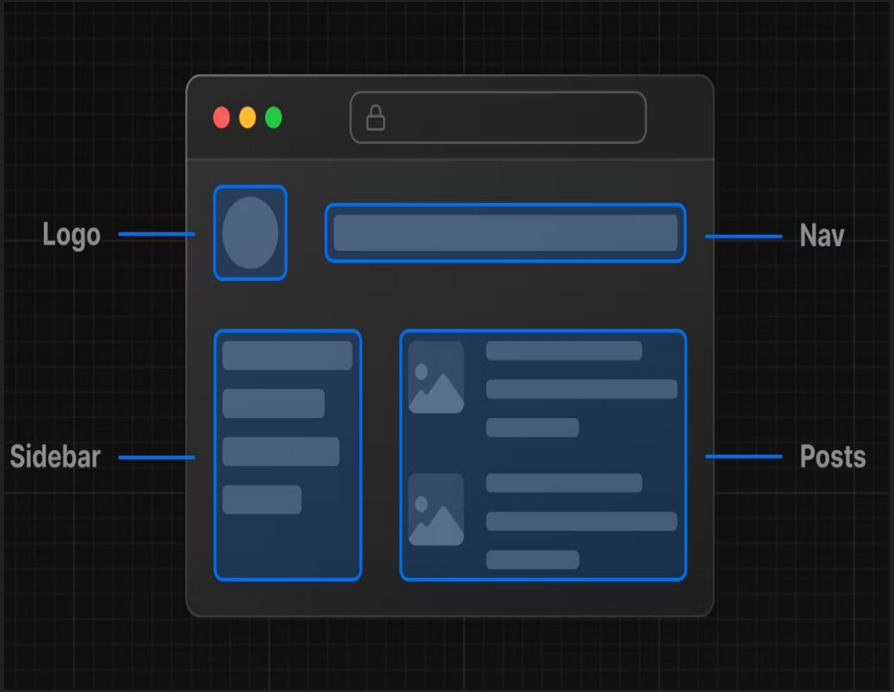
ReactJS

ReactJS

ReactJS is a JavaScript library for building interactive user interfaces.

Official Definition: React is a library for creating modular user interfaces.

ReactJS



- React provides helpful functions (APIs) to build UI
- It is unopinionated about the other aspects of building applications.
    - flourishing ecosystem of third-party solutions & tools
- Building a complete React application from the ground up requires effort for things like routing, caching, authentication etc.

Let’s get started

Get Started

- Create a new React application
  - Open command line and run:

```
npx create-next-app@latest
```

```
npm run dev
```

  - Open your browser and point to http://localhost:3000/
  - Run vscode from the folder that you created your application in.

More info: https://nextjs.org/learn

Quick React Tutorial

What is a component?

<div>, <span>
<ActionButton>, <Counter>

Creating your first component

Create a folder called “sample”

Create 2 files called “action-button.tsx” and “page.tsx”

In page.tsx:

```
export default function SamplePage() {
  return (<main className="p-12 min-h-screen background-white">
      <h4>Action Button Test</h4>
      <ActionButton text="My Action" />
</main>);}
```

http://localhost:3000/sample

```
<ActionButton text=”Book Flight” onAction={someFunc} />
```

src/app/sample/action-button.tsx

```
import { MouseEventHandler } from "react";


export default function ActionButton({text, onAction}:  {text: string, onAction?:
MouseEventHandler<HTMLButtonElement>}) {
    return (




    );
}
```

```
<ActionButton text=”Book Flight” onAction={someFunc} />
```

src/app/sample/action-button.tsx

```
import { MouseEventHandler } from "react";

export default function ActionButton({text, onAction} : {text: string, onAction?:
MouseEventHandler<HTMLButtonElement>}) {
    return (
      <button className="">
        <span>Click</span>
      </button>
    );
}
```

```
<ActionButton text=”Book Flight” onAction={someFunc} />
```

src/app/sample/action-button.tsx

```tsx
import { MouseEventHandler } from "react";

export default function ActionButton({text, onAction}: {text: string, onAction?:
MouseEventHandler<HTMLButtonElement>}) {
    return (
      <button className="">
        <span>{text}</span>
      </button>
    );
}
```

```tsx
<ActionButton text=”Book Flight” onAction={someFunc} />
```

src/app/sample/action-button.tsx

```tsx
import { MouseEventHandler } from "react";

export default function ActionButton({text, onAction}:  {text: string, onAction?:
MouseEventHandler<HTMLButtonElement>}) {
    return (
      <button className="" onClick={onAction}>
        <span>{text}</span>
      </button>
    );
}
```

```tsx
<ActionButton text=”Book Flight” onAction={someFunc} />
```

src/app/sample/action-button.tsx

```
import { MouseEventHandler } from "react";

export default function ActionButton({text, onAction}: {text: string, onAction?:
MouseEventHandler<HTMLButtonElement>}) {
    return (
      <button className="px-4 py-2 bg-slate-200 hover:bg-slate-500 border border-slate-700"
onClick={onAction}>
        <span>{text.toUpperCase()}</span>
      </button>
    );
}
```

Tail Wind CSS styles allow for the element to be styled.

```
<ActionButton text=”Book Flight” onAction={someFunc} />
```

More info: https://tailwindcss.com/

```
<Counter initialCount={4} />
```

src/app/sample/counter.tsx

```
'use client'
import { useState } from "react";
import ActionButton from "./action-button";

export default function Counter({ initialCount }: { initialCount: number }) {
    const [count, setCount] = useState(initialCount);

    function increment() { setCount(count + 1); }
    function decrement() { setCount(count - 1); }

    return (
        <div>
            <h3>Count: {count}</h3>
            <ActionButton text='-1' onAction={increment} />
            <ActionButton text='+1' onAction={decrement} />
        </div>
    );
}
```

Informs this component is client side rendered

Store the value into a local state

Pass a method to the child component

```
<Counter initialCount={4} />
```

src/app/sample/page.tsx

```
import ActionButton from "./action-button";
import Counter from "./counter";
export default function Sample() {
  return (
    <main className="p-12 min-h-screen background-white">
      <h4>Action Button Test</h4>
      <ActionButton text="My Action" />
      <h4>Counter</h4>
      <Counter initialCount={4} />
</main>);
}
```

`<Counter initialCount={4} />`

src/app/sample/page.tsx

```
import ActionButton from "./action-button";
import Counter from "./counter";
export default function Sample() {
  return (  <main className="p-12 min-h-screen background-white">
      <h4>Action Button Test</h4>
      <ActionButton text="My Action" />
     <div className="bg-slate-500 p-4 mt-4">
       <h4>Counter 1</h4>
       <Counter initialCount={4} />
      </div>
      <div className="bg-slate-700 p-4 mt-4">
       <h4>Counter 2</h4>
       <Counter initialCount={104} />
      </div>
</main>);}
```

http://localhost:3000/sample

GIT

Version control system

What is Git

Git is a popular version control system.

It is used for:

- Coding collaboration
- Tracking code changes
- Tracking who made changes

What does Git do?

- Manage projects with Repositories
- **Clone** a project to work on a local copy
- Control and track changes with **Staging** and **Committing**
- **Branch** and **Merge** to allow for work on different parts and versions of a project
- **Pull** the latest version of the project to a local copy
- **Push** local updates to the main project
- See the full history of every **commit**.

Why Git?

- Over 70% of developers use Git!
- Developers can work together from anywhere in the world.
- Developers can see the full history of the project.
- Developers can revert to earlier versions of a project.

What is GitHub?

- Git is not the same as GitHub.
- GitHub makes tools that use Git.
- GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.
- Source: https://www.w3schools.com/git/

Login to GitHub.com

Please login with your credentials to Github.com

Create an account if you do not have one.

Create the GIT repository

Track changes locally with git:

```
git init

git commit -a -m "codelab starting point"

git branch -M main
```

Create a new GitHub repository:

Go to https://github.com/new

Name it anything you'd like.

GitHub will give you a new repository URL that looks like either https://github.com//.git or git@github.com:/.git. Copy this URL.

Commit your changes to the GIT repository

```
git remote add origin <your-repository-url>

git push -u origin main
```

You should now see your code in your GitHub repository.

Branches

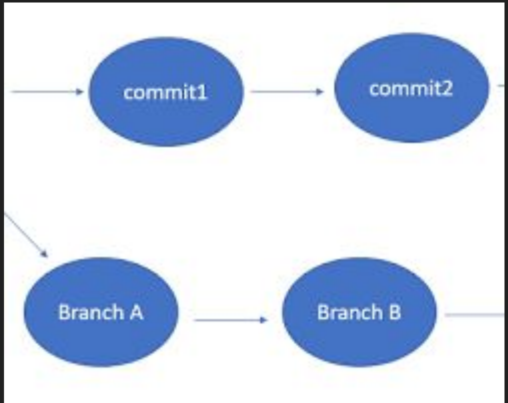
Branches are your Parallel Workspaces

Why Branches?

- Isolation of new features
- Fix bugs without disrupting the main project
- Safely experiment

Types:

- main: The stable, production-ready code
- Feature branches: For developing specific features
- Hotfix branches: For urgent bug fixes



Commits: Snapshots of Your Work

What's a Commit?

- A saved state of your project at a specific point in time.

Includes:

- Changes to files
- A commit message (explaining the changes)

Branching Workflow

Create a Branch: git branch <branch-name>

Switch to the Branch: git checkout <branch-name>

Make Changes & Commit: git add . & git commit -m "Your message"

Repeat (make more changes, more commits)

Pull Requests

What's a PR?

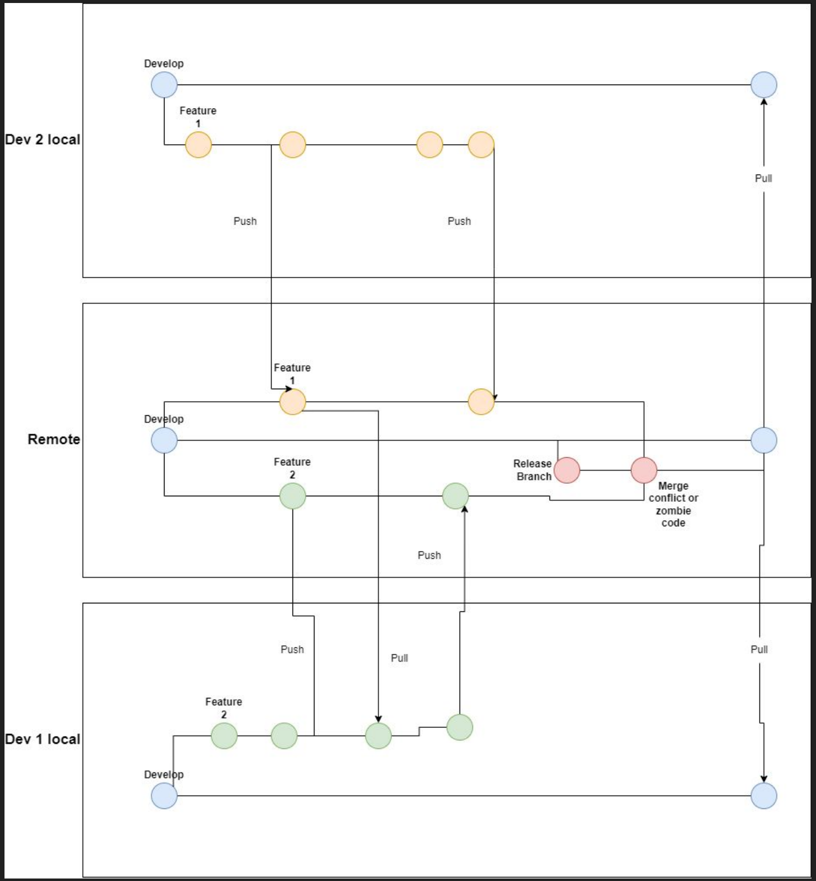
A request to merge your branch into another branch (usually main).

Includes:

- Code changes
- Discussion/review from others

Benefits:

- Code review
- Quality control
- Collaboration

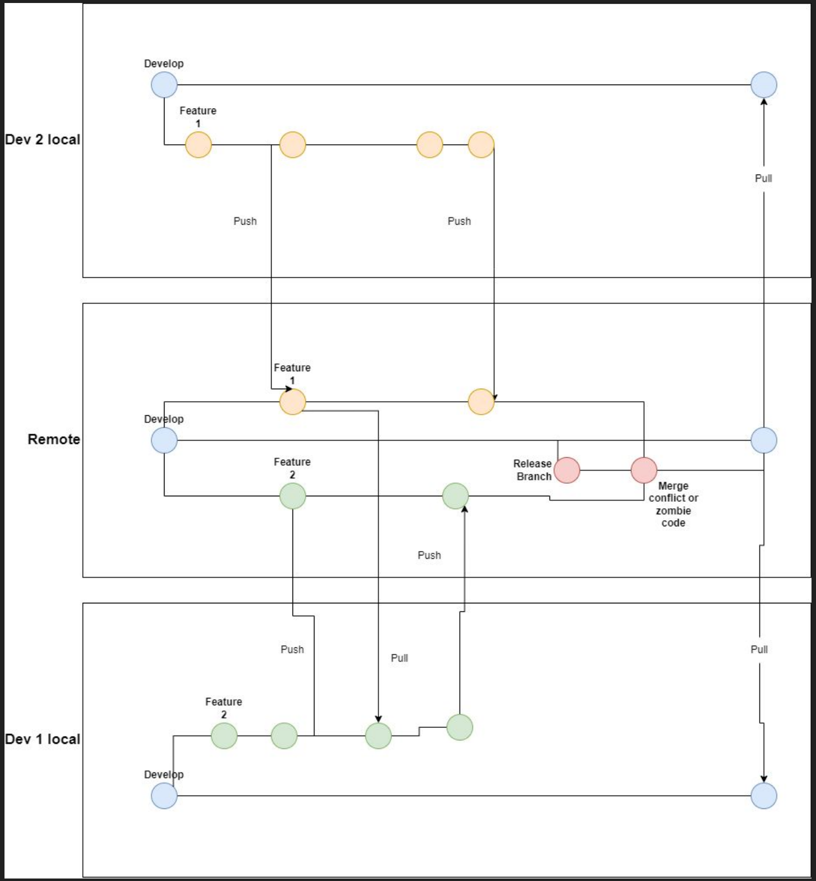


PR Workflow

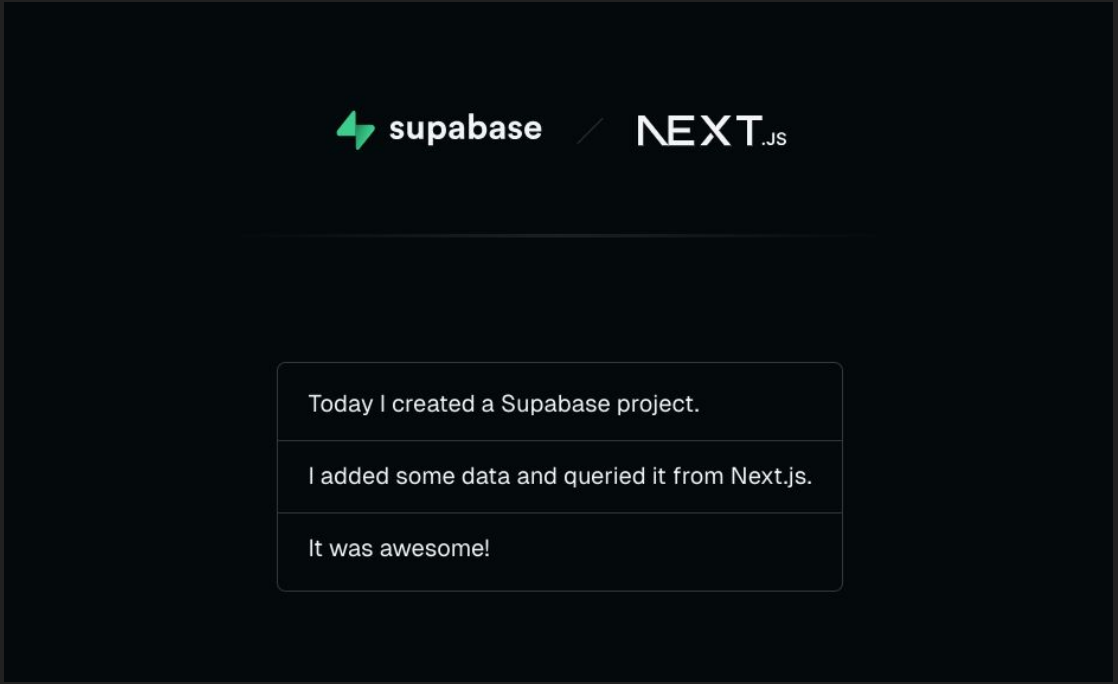
Push your branch to GitHub:

git push origin <branch-name>

- Open a PR on GitHub:
- Describe your changes
- Request reviewers
- Address feedback (if any)
- Merge the PR (when approved)



Let’s build!



Today I created a Supabase project.

I added some data and queried it from Next.js.

It was awesome!

Vercel & Supabase

Vercel

Vercel is the platform for developers who want to build and deploy modern web experiences quickly and easily.

- Effortless Deployment: Vercel simplifies the process of deploying websites and web applications. Push your code, and Vercel handles the rest.
- Integrations: Has great Integrations with other platforms like supabase, mongodb, mysql etc in addition to the popular front-end frameworks like Next.js, React, Vue, and Angular.
- Serverless Functions: Easily add dynamic functionality to your sites with serverless functions.
- Preview Deployments: Share a live preview of your work with your team or clients before it goes live.
- Custom Domains: Connect your own domain name to your Vercel projects.
- Collaboration: Work together with your team on projects, with features like environment variables and team settings.

Supabase

Supabase provides a complete backend solution, including a Postgres database, authentication, real-time subscriptions, and storage.

- Open Source: Built on top of open-source technologies like Postgres and PostgREST, ensuring flexibility and community support.
- Developer-Friendly: Easy to set up and use, with a powerful dashboard, APIs, and SDKs for various languages.
- Real-Time: Enable real-time updates and collaboration in your applications.
- Scalable: Supabase can scale to meet the needs of your growing projects.
- Authentication: Built-in authentication with support for email/password, social logins, and more.
- Storage: Easily store and manage files like images and videos.

Next JS

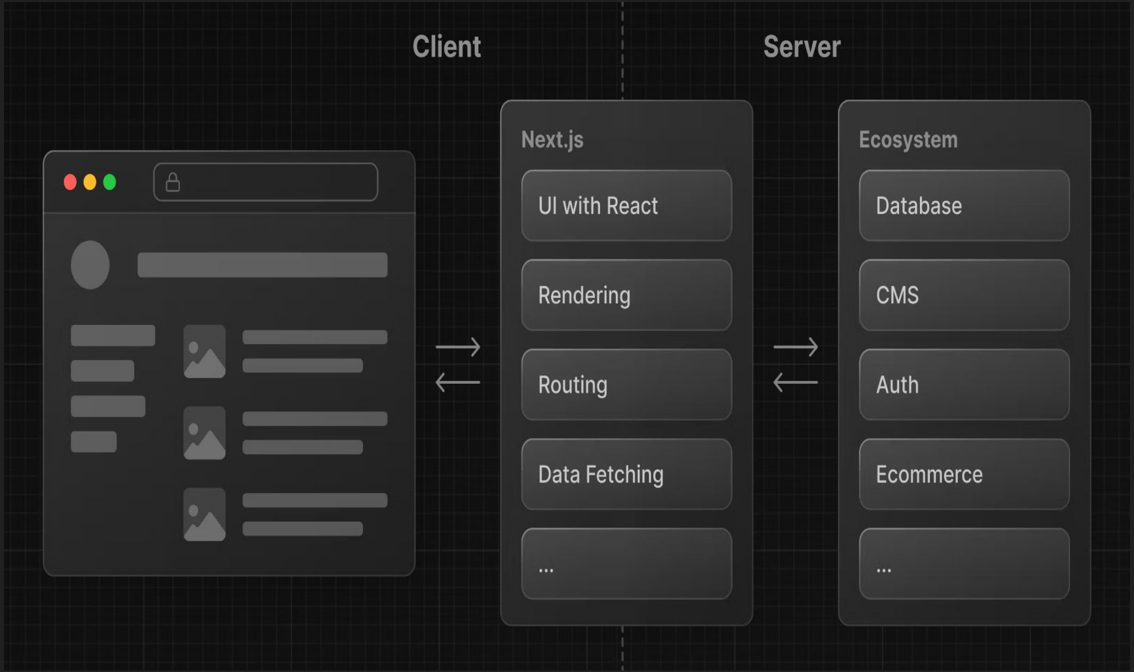
- Next.js is an open-source web development framework based on React and has gained significant popularity due to its amazing features.
- It is developed by Vercel and Next.js
  - Built-in Routing and SSR: Unlike React, which lacks native routing, Next.js provides seamless routing functionality out of the box. Additionally, it supports server-side rendering, improving performance and SEO.
  - Faster Development: Next.js accelerates development by offering built-in features and conventions. Developers can focus on building features rather than configuring complex setups.
  - SEO Optimization: Next.js enhances SEO by addressing slow rendering and loading times associated with client-side rendering. Its SSR capabilities ensure that search engines can efficiently crawl and index your content.

What is Next.js?

Next.js is a React framework that gives you building blocks to create web applications.

Next.js handles the tooling and configuration needed for React, and provides additional structure, features, and optimizations for your application.

As you've seen in the first example (sample), the routing is simply through folder structure and useState is a simple api reference.



Why NextJS?

- React provides helpful functions (APIs) to build UI, but leaves it up to the developer where to use those functions in their application.
- Part of React's success is that it is relatively un-opinionated about the other aspects of building applications. This has resulted in a flourishing ecosystem of third-party tools and solutions, including Next.js.
- Building a complete React application from the ground up requires some effort.
- Developers need to spend time configuring tools and reinventing solutions for common application requirements.

Let’s deploy!!

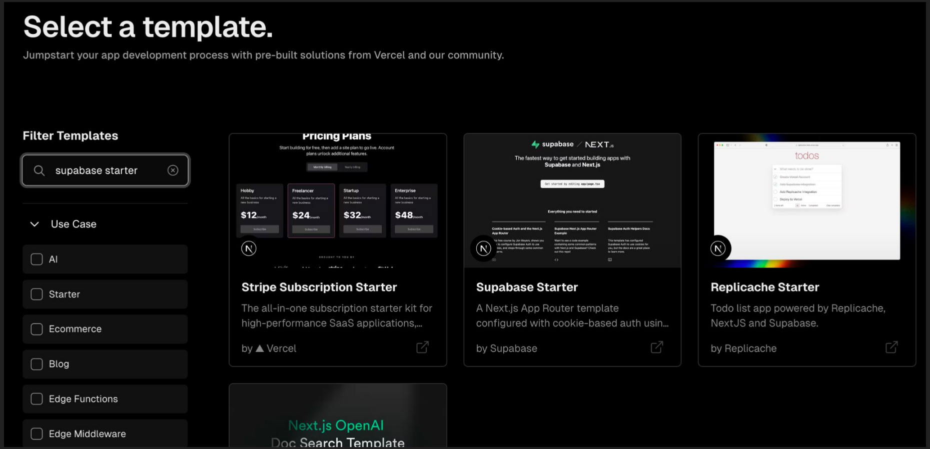
Create a new site - Sign up

Let’s deploy a brand new site on the internet using Vercel :)

1. Go to supabase.com
2. Signup using your GitHub Credentials
3. Headover to vercel.com
4. Do the same - Signup using your GitHub credentials

Create a new site - Find the template

1. On the create project screen: Find the template called "*Supabase Starter*"
2. Click on the "Deploy" button
   a. Vercel will ask to create a new Github Repository.
      i. You can keep the defaults or rename it.
      ii. Click on create
   b. After deployment, visit the site



Select a template.

Jumpstart your app development process with pre-built solutions from Vercel and our community.

Filter Templates

supabase starter

Use Case

☐ AI

☐ Starter

☐ Ecommerce

☐ Blog

☐ Edge Functions

☐ Edge Middleware

Stripe Subscription Starter

The all-in-one subscription starter kit for high-performance SaaS applications,...

by ▲ Vercel

Supabase Starter

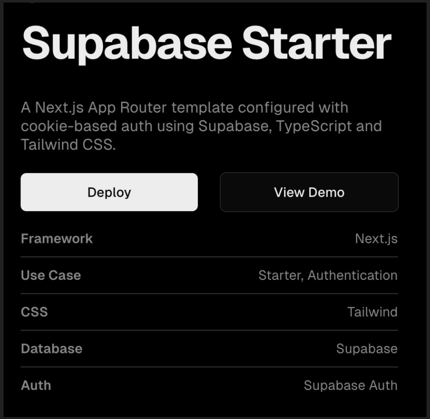
A Next.js App Router template configured with cookie-based auth usin...

by Supabase

Replicache Starter

Todo list app powered by Replicache, NextJS and Supabase.

by Replicache



Supabase Starter

A Next.js App Router template configured with cookie-based auth using Supabase, TypeScript and Tailwind CSS.

Framework	Next.js
Use Case	Starter, Authentication
CSS	Tailwind
Database	Supabase
Auth	Supabase Auth

Test the deployment

- Sign up a new user
- Check email and confirm the user.
  - If your email points to localhost:3000, please make sure you change the host to the site URL from vercel
- Login, once the confirmation is complete

Let’s add features!!

Create a table and insert some data

- Go to: SQL Editor in Supabase, copy and paste the following into the SQL Editor

```
create table notes (id bigserial primary key,  title text);
insert into notes(title)
values
  ('Today I created a Supabase project.'),
  ('I added some data and queried it from Next.js.'),
  ('It was awesome!');
```

- Click RUN!

Clone the repo

- Clone the repo and run the code locally:

```
git clone <your repo url>

cd nextjs-with-supabase

npm install

npm run dev
```

You will need to generate a token (https://github.com/settings/tokens), and use that as a password when we are authenticating into github from cli.

Update environment variables to run locally

- Open VS Code

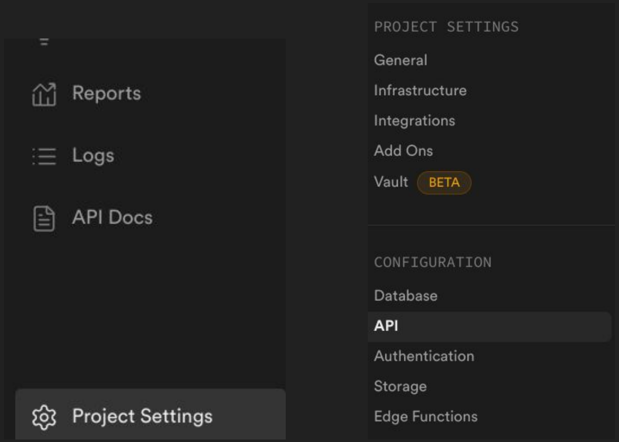
```
> code .
```

- Go to supabase.com > Your Project > Project settings > API settings

  Rename .env.example to .env.local

  Update the URL and API

- Go to http://localhost:3000
  - Login with the credentials you created



Create the notes page

- Create a new page.tsx file at /app/notes/page.tsx and add the following.

```
import { createClient } from '@/utils/supabase/server'

export default async function Page() {
 const supabase = createClient();
 const { data: notes } =
       await supabase.from('notes').select();

 return <pre>{JSON.stringify(notes, null, 2)}</pre>
}
```

Let’s add HTML / recursion

Change the return statement to:

```
return (
   <div>
     <div>
       <ul>
       {notes?.map(note => (<li>{note.title}</li>))}
       </ul>
     </div>
   </div>
 )
```

Let’s add CSS

Change the return statement to:

```
return (
   <div className="p-4 sm:p-6 xl:p-9">
     <div className="min-w-[370px] max-w-max rounded-md border border-stroke py-1
dark:border-strokedark">
       <ul className="flex flex-col">
       {notes?.map(note => (
               <li className="flex items-center gap-2.5 border-b border-stroke px-5 py-3
last:border-b-0 dark:border-strokedark">
                 <span> {note.title} </span>
               </li>)
       )}
       </ul>
     </div>
   </div>
 )
```

…more info

Let’s add a layout

Rename the /protected/page.tsx to /protected/layout.tsx

Change the function name:

From: `export default async function Page() {`

To: `export default async function ProtectedLayout({children}: { children: React.ReactNode }) {`

Replace the main section to:

```
<main className="flex-1 flex flex-col gap-6">
     {children}
</main>
```

…more info

Let’s create Pages

- Create new /protected/page.tsx

```
import FetchDataSteps from "@/components/tutorial/FetchDataSteps";

export default async function ProtectedPage() {
 return (
  <>
     <h2 className="font-bold text-4xl mb-4">Next steps</h2>
     <FetchDataSteps />
   </>);
}
```

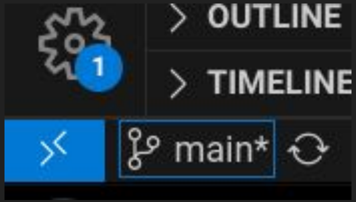
  - Point your browser to http://localhost:3000/protected/

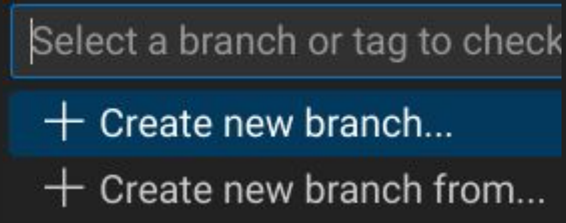
- Now, move the /app/notes folder to /app/protected/notes
  - Point your browser to http://localhost:3000/protected/notes

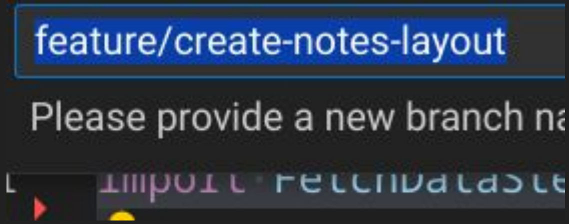
…more info

Branch / PR / Checkin

- Let’s use VS Code to branch and commit:







feature/create-notes-layout

Please provide a new branch na

- Commit to the branch either through command line or on VS Code
- Publish the branch
- Send a Pull Request
- Once the build is complete merge the changes
- Check that the deployment succeed

To commit into github from vscode, run:

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

…more info

Project Structure - Key Files

Package.json:

This file contains metadata about your project, such as its name, version, dependencies, and scripts.

.env

This file contains the environment variables which contain the API keys etc.

tsconfig.json

Typescript configuration

tailwind.config.ts

Configuration for Tailwind CSS

middleware.ts

Routing logic, session logic etc.

https://nextjs.org/docs/getting-started/project-structure

RECAP

- How to build a React Component
- Create a Git Repository, Branch, PR and Commit
- Create a new NextJS / Supabase application using Vercel templates
- Introduction to NextJS, Supabase, Tailwind CSS

More Learning Resources

Learn from:

User Management with NextJS and Supabase

NextJS, Tailwind CSS

W3Schools: React, NodeJS, PostgreSQL

Build a Twitter Clone with the Next.js App Router and Supabase - free egghead course

Code from this course:

https://github.com/msbhaskar/nextjs-with-supabase

Final Note...

- This course introduced you to multiple technologies and platforms. You need to explore more on each of these technologies and platforms to become a good full-stack developer.