

Be part of a better internet. [Get 20% off membership for a limited time](#)

Next.js Full-Stack Web App (2024)



Devon Wijesinghe · [Follow](#)

Published in Stackademic · 6 min read · Mar 25, 2024



1K



9



Next.js Full-stack web app starter template 2024.

Focus on whats important and kickstart your project faster 🚀



Devon Wijesinghe
wdevon99

N NEXT.JS **N** NEXT AUTH **TS** TYPESCRIPT **R** REACT **A** ANTDESIGN **M** MONGODB **S** SASS



NEXT.js

Full-stack starter template 2024

Next.js • NextAuth • TypeScript • MongoDB • Ant Design



Continue with Google

Continue with GitHub

This template was created on 25th of February 2024.

Devon Wijesinghe
wdevon99



Next.js Full-stack web app starter template 2024

Hey guys 😊, are you planning to build a full-stack web application? If so, you've landed on the perfect blog post! I'm happy to introduce you to a Next.js full-stack starter template that I have designed to streamline the process of creating a modern, feature-rich web application and eliminate the time spent on setting up the boilerplate needed to kick off your project.

This starter template is tailored for Next.js 14, the latest version. Whether you're a seasoned developer or just starting out, this starter template along with Next Js capabilities, offers an intuitive framework that simplifies the development process for building dynamic and interactive full-stack web applications.

With this starter template, you can focus on what's important and kickstart your project faster. 🚀

The link to the starter template GitHub repository can be found below:

GitHub - wdevon99/Next.js-starter: Full-stack starter template 2024

Full-stack starter template 2024. Contribute to wdevon99/Next.js-starter development by creating an account on GitHub.

github.com

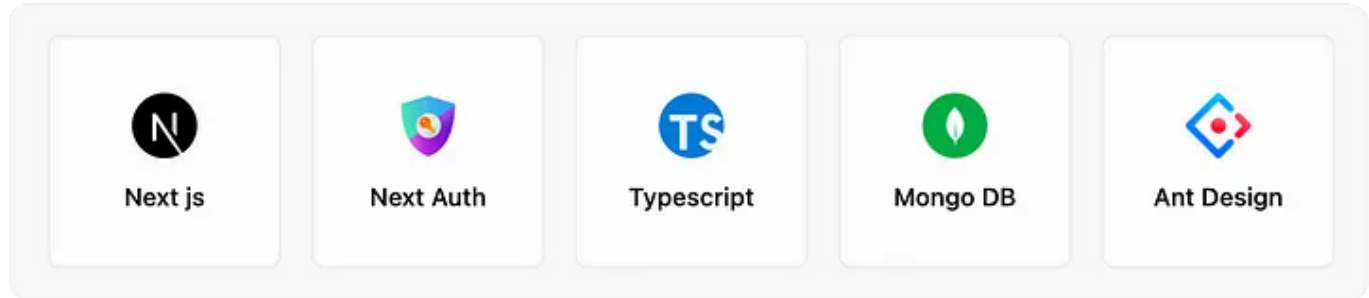
How to use this template? 🤔

Getting started with this Next.js Full-stack starter template is super straight forward. Simply head over to the GitHub repository linked above and follow the steps outlined in the *README* file. There, you'll find detailed setup instructions along with any prerequisites necessary to get the project up and running.

Main Features 🚀

In this section, let's dive into the features of this template. The main features of this template are as follows:

- Scalable project structure 📊
- TypeScript enabled ✅
- Pre-configured Ant-Design component library 🐛
- Structured styling with SASS modules and global variables 🎨
- Social authentication and route guarding 🔒
- Sample APIs with database integration 📁
- Custom alias imports 📦



Tech-stack used for this template

Scalable project structure 📊

Having a well organized project structure is super important for the scalability and maintainability of any codebase. By structuring the project folders and files logically, developers can easily locate and modify specific

components, services, or any other files. The following screenshot shows the folder structure of the template. In this section, we will have a look at the purpose of each of the folder.



Folder structure

“Root level” folders:

- **Public** — To store any public assets such as images and fonts.
- **Src** — All the main source files and folders are grouped inside the src.

“Src level” folders:

- **App** — Contains the page routes and API routes.
- **Components** — Atomic design pattern is used in this folder to maintain components in a structured manner.
- **Constants** — To store all common/shared constants.
- **Layouts** — All layouts including the main layout is placed in this folder.
- **Models** — To store database schema models.
- **Providers** — All providers such as the auth provides and theme provider is placed in this folder.
- **Services** — Service layer is maintained in this folder which has a set of functions to call the APIs.
- **Styles** — Global styles and style variables are placed in this folder.
- **Types** — Typescript types are place in this folder.
- **Utils** — Utility functions or helpers can be placed in the utils folder.

Typescript enabled

Typescript is enabled in this project for a better developer experience and to maintain type safety. You can use JavaScript as well if you prefer, since the allow “*allowJs*” flags is set to *true*. If you do not want to allow Javascript usage, you can set it to *false*.

The global types are defined in *src/types/global.d.ts* file.

```
1  /**
2   *
3   * Define all global types in this file.
4   *
5   */
6
7  type User = {
8    email: string;
9    image: string;
10   username: string;
11  };
12
13  type Todo = {
14    _id: string;
15    todoTitle: string;
16    todoDescription: string;
17    isComplete: boolean;
18    creator: User;
19  };
```

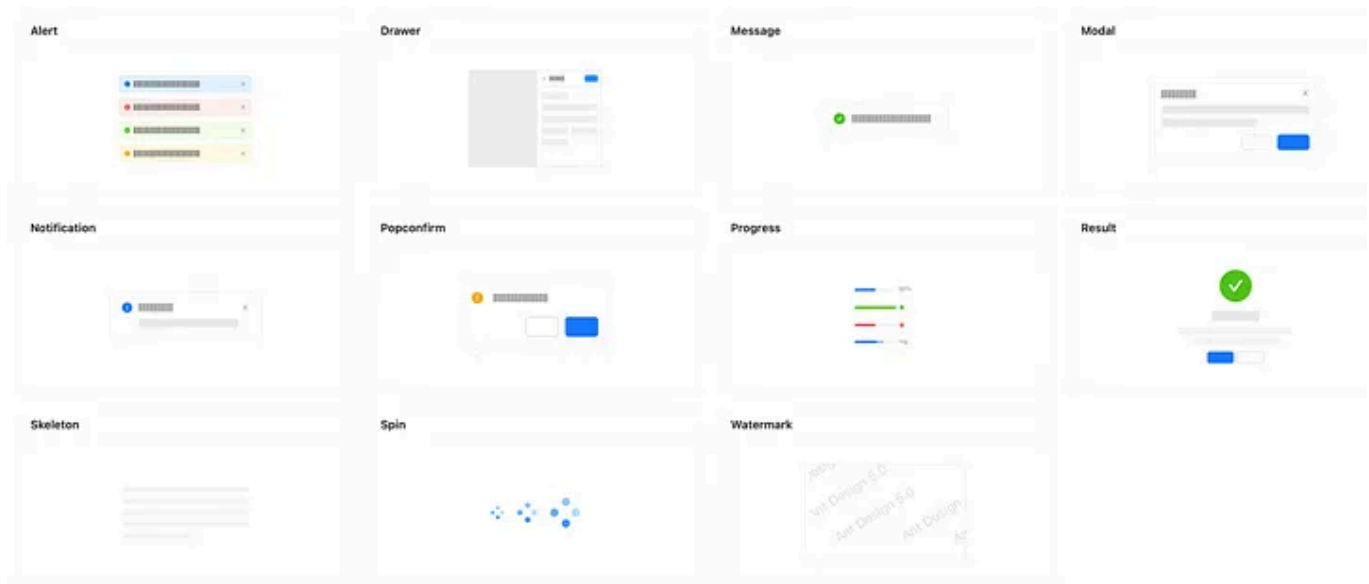
global.d.ts hosted with ❤️ by GitHub

[view raw](#)

src/types/global.d.ts

Pre-configured Ant Design component library 🐜

The Ant Design (*AntD*) component library is set up and pre-configured in this template. AntD is filled with stylish and practical components which can be used as the main building blocks for your web application. It saves you the hassle of designing every element from scratch and help you to cut down on development time.



Ant Design - The world's second most popular React UI framework

An enterprise-class UI design language and React UI library with a set of high-quality React components, one of best...

ant.design

The *AntdConfigProvider.tsx* handles the top level configuration of the AntD library. The primary colour for the library is configured as a global style variable. To customize the primary color, all you have to do is to modify the hex of the *\$primary-color* variable in *src/styles/variables.module.sass*.

```
1  import { ConfigProvider } from "antd";
2  import Colors from "@styles/variables.module.sass";
3
4  const AntdConfigProvider = ({ children }: Props) => {
5    const THEME = {
6      token: {
7        colorPrimary: Colors.primaryColor,
8      },
9    }
10
11    return (
12      <ConfigProvider theme={THEME}>
13        {children}
14      </ConfigProvider>
15    )
16  }
17
18  type Props = {
19    children: React.ReactNode,
20  }
21
22  export default AntdConfigProvider;
```

AntdConfigProvider.tsx hosted with ❤️ by GitHub

[view raw](#)

Structured styling with SASS modules and global variables 🛠️

This template has structured styling using SASS modules and global variables. SASS modules allow for encapsulated styling, making it easier to manage stylesheets by keeping styles scoped to individual components.

Global variables streamline the process of defining and reusing common styles across your project, promoting consistency in your styles. With SASS modules and global variables, you can maintain clean and organized styles.


```
1 // ----- Colors -----
2 $primary-color: #292929
3 $primary-surface-color: #F9F9F9
4
5 $success-color: #52c41a
6 $warning-color: #faad14
7 $error-color: #ff4d4f
8 $secondary-surface-color: #FFF
9
10 $font-color-light: #FFF
11 $font-color-dark: #292929
12 $border-color: #e4e4e4
13 $github-primary-color: #292929
14
15 // ----- Sizes -----
16 $border-radius: 0.3rem
17
18 // ----- Exports -----
19 :export
20   primaryColor: $primary-color
21   successColor: $success-color
22   warningColor: $warning-color
23   errorColor: $error-color
```

variables.module.sass hosted with ❤ by GitHub

[view raw](#)

Social authentication and route guarding

Social authentication enables users to sign in to your web application using their existing social media accounts, such as Google, Facebook, or Twitter. To achieve social authentication, this template is integrated with the *NextAuth.js* authentication library.

NextAuth.js

Authentication for Next.js

next-auth.js.org

This template is pre-configured with two social auth providers, Google and Github, but you have the freedom to extend it to use any other platform you prefer. To add a new provider, all you have to do is configure a new provider here, *src/app/api/auth/[...nextauth]/route.ts*.

```
1  const handler = NextAuth({
2    providers: [
3      GoogleProvider({
4        clientId: process.env.GOOGLE_ID ?? "",
5        clientSecret: process.env.GOOGLE_CLIENT_SECRET ?? "",
6      }),
7      GithubProvider({
8        clientId: process.env.GITHUB_ID ?? "",
9        clientSecret: process.env.GITHUB_CLIENT_SECRET ?? "",
10     }),
11   ],
12   ...
13 });
```

NextAuthRoute.ts hosted with ❤️ by GitHub

[view raw](#)

[src/app/api/auth/\[...nextauth\]/route.ts](#)

Route guarding: Route guarding allows us to protect certain routes within our application, ensuring that only authenticated users can access private routes and endpoints. Route guarding is implemented using a middleware approach. By intercepting requests, the middleware verifies the user's authentication status before allowing access. You can extend this middleware to perform role-based access control as well.

```
1  import { NextRequest, NextResponse } from "next/server";
2  import { getToken } from "next-auth/jwt";
3
4  const PROTECTED_API_ROUTES = ["/api/todo"];
5  const PROTECTED_ROUTES = ["/dashboard"];
6
7  export async function middleware(request: NextRequest) {
8      const isProtectedApiRoute = PROTECTED_API_ROUTES.some((route: string) => request.nextUrl?.pathr
9      const isProtectedRoute = PROTECTED_ROUTES.some((route: string) => request.nextUrl?.pathname?.st
10
11     if (isProtectedApiRoute) {
12         const isAuthenticated = await isAuthenticated(request);
13         if (!isAuth) {
14             return Response.json({ success: false, message: "Authentication failed" }, { status: 401 });
15         }
16     }
17
18     if (isProtectedRoute) {
19         const isAuthenticated = await isAuthenticated(request);
20         if (!isAuth) {
21             return NextResponse.redirect(new URL("/", request.url));
22         }
23     }
24
25     return NextResponse.next();
26 }
27
28 const isAuthenticated = async (request: NextRequest) => {
29     const token: any = await getToken({ req: request });
30     return !!token && Date.now() <= token.exp * 1000;
31 };
```

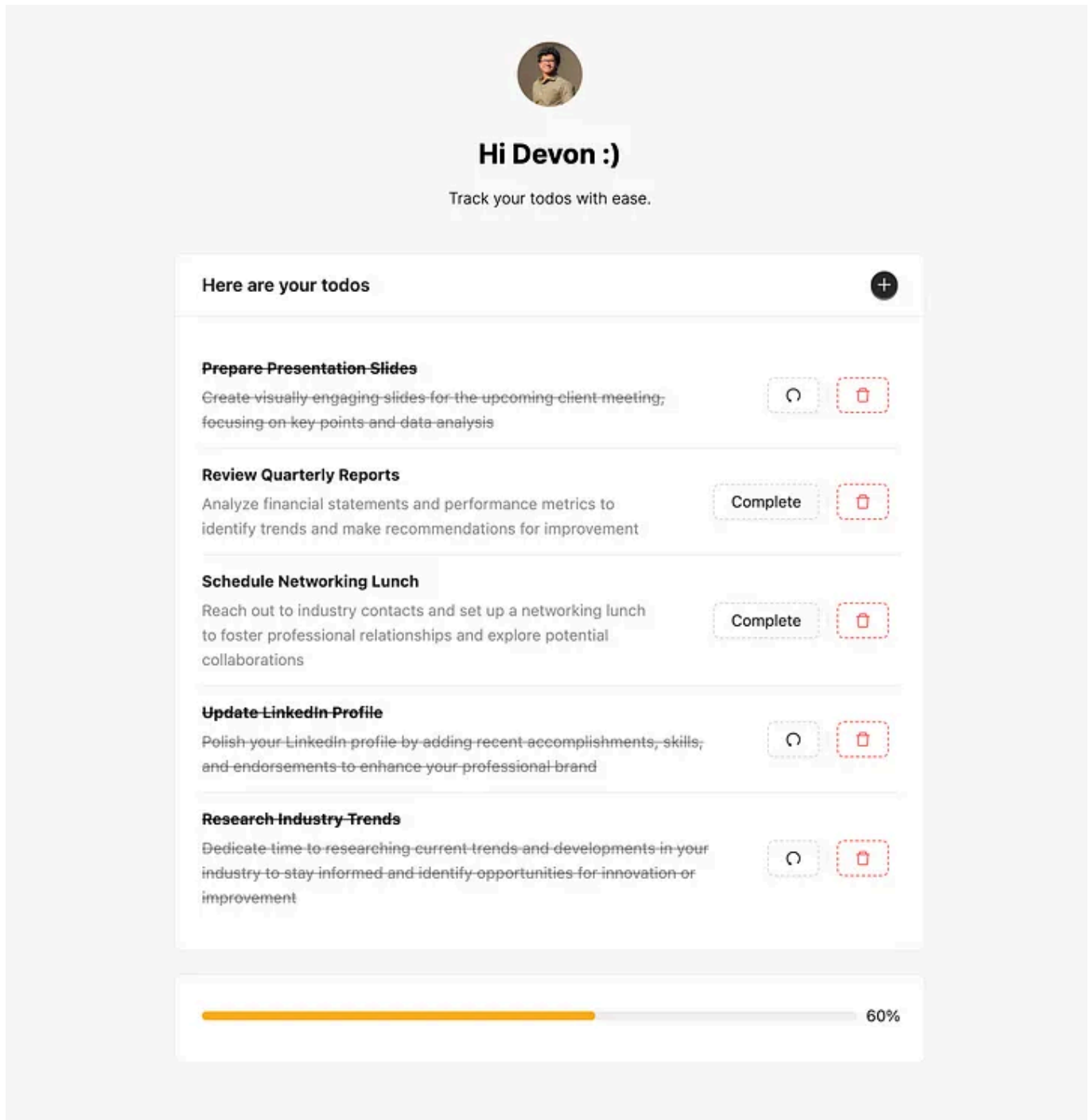
middleware.ts hosted with ❤ by GitHub

[view raw](#)[src/middleware.ts](#)

Sample APIs with database integration

The template consist of few sample API endpoints covering basic CRUD (Create, Read, Update, Delete) operations for a TODO list. These APIs serve

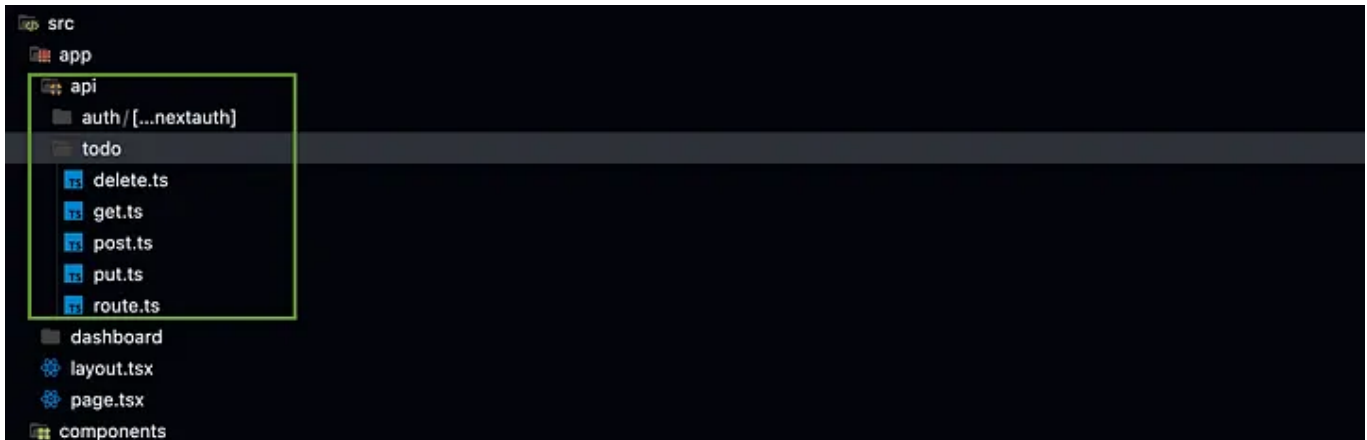
as a practical demonstration of how to implement API endpoints in this project.



Sample TODO app with basic CRUD operations

Each API method is placed in its own file as shown in the screenshot below for better separation of logic. You can also place all the API methods in a

single file as well if you prefer it that way.



Database integration with MongoDB: The mongoose library is used to connect to the database and work with it. The database models can be found in *src/models* folder. You can easily extend it to add your own models as needed.

```
1  import mongoose from "mongoose";
2
3  let isConnected = false;
4
5  export const connectToDB = async () => {
6    mongoose.set("strictQuery", true);
7
8    if (isConnected) {
```

[Open in app](#)

Medium

Search

Write



```
13  await mongoose.connect(process.env.MONGODB_URI ?? "", { dbName: "todos" });
14  isConnected = true;
15  console.info("==== MongoDB connected ====");
16  } catch (error) {
17    console.error(error);
18  }
19  };
```

database.ts hosted with ❤️ by GitHub

[view raw](#)

GitHub - Automattic/mongoose: MongoDB object modeling designed to work in an asynchronous...

MongoDB object modeling designed to work in an asynchronous environment. - Automattic/mongoose

github.com

Custom alias imports.

Custom alias imports are configured for the project using the *tsconfig.json* file. These aliases serve as shortcuts for module paths, allowing you to import files with ease and avoid the cumbersome directory traversal (*Example* : “*.././../example.ts*”). With custom aliases in place, your import statements become cleaner and more readable. It’s a small tweak that makes a big difference in keeping your project imports organized as your project scales up.

```
1  import CustomAvatar from "../../../../../atoms/CustomAvatar"; // ❌  
2  
3  import CustomAvatar from "@components/atoms/CustomAvatar"; // ✅
```

aliasImportExample.ts hosted with ❤️ by GitHub

[view raw](#)

Alias import example

The *tsconfig.json* configuration is as follows, the *paths* attribute under *compilerOptions* allows us to customize the alias. You can add more here if you decide to add new folders to your project.

```
1  "compilerOptions": {
2    ...
3    "baseUrl": "./",
4    "paths": {
5      "@styles/*": ["src/styles/*"],
6      "@components/*": ["src/components/*"],
7      "@constants/*": ["src/constants/*"],
8      "@utils/*": ["src/utils/*"],
9      "@models/*": ["src/models/*"],
10     "@services/*": ["src/services/*"],
11     "@layouts/*": ["src/layouts/*"],
12     "@providers/*": ["src/providers/*"],
13   }
14 },
15 "include": ["next-env.d.ts", "**/*.ts", "**/*.tsx", ".next/types/**/*.ts"],
16 "exclude": ["node_modules"]
```

tsconfig.json hosted with ❤️ by GitHub

[view raw](#)

tsconfig.json

Conclusion 🎲

As explained in this blog, I've aimed to eliminate the burden of boilerplate setup and help developers to focus on what truly matters. I really hope that you will find this template useful and will help you in your next project! 😊

For further clarifications or assistance, feel free to reach out via wdevon99@gmail.com or connect with me through LinkedIn, <https://www.linkedin.com/in/devon-wijesinghe>. Cheers! 🙌 🙌

Stackademic 🎓

Thank you for reading until the end. Before you go:

- Please consider **clapping** and **following** the writer! 🙌

- Follow us [X](#) | [LinkedIn](#) | [YouTube](#) | [Discord](#)
- Visit our other platforms: [In Plain English](#) | [CoFeed](#) | [Venture](#) | [Cubed](#)
- More content at [Stackademic.com](#)

[Nextjs](#)[Reactjs](#)[React](#)[Typescript](#)[Template](#)

Written by Devon Wijesinghe

[Follow](#)

486 Followers · Writer for Stackademic

Senior Software Engineer | Full Stack Javascript Developer | Certified Scrum Master™ |
AWS Certified Solutions Architect Associate

More from Devon Wijesinghe and Stackademic

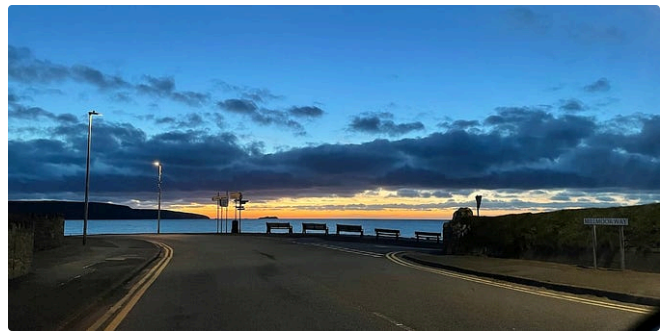



 Devon Wijesinghe

Simple Steps to host a Website/Web-App in AWS with...

Hey guys, in this article I will guide you through the steps which you can follow to...

Jan 29, 2021  245  3



 Peter Bunyan in Stackademic

Who needs Redis, when Postgres will do?

 May 9  594  6



 Dylan Cooper in Stackademic

Oracle Tightens Grip on Java: Big Companies Beware!

It has been over a year since Oracle revised the pricing model for Java SE, and now their...

 Jun 13  709  11



 Devon Wijesinghe

What the hell is OAuth?

Hello everyone! OAuth has been a buzzword for quite some time now and it is hard for a...

Jan 9, 2019  3.1K  11




See all from Devon Wijesinghe

See all from Stackademic

Recommended from Medium





 Vijayasekhar Deepak in Stackademic



Building a Micro Frontend Application with Next.js 14.2 and...

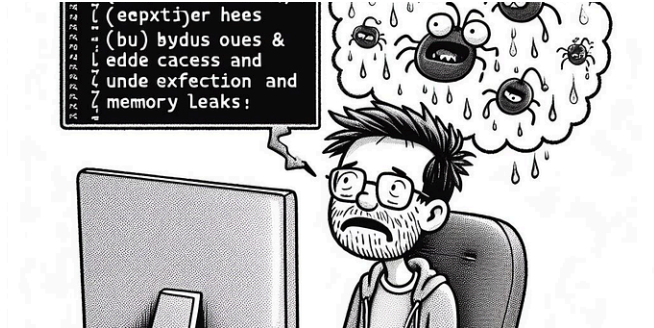
In modern web development, micro frontend architecture has gained popularity for its...

Jun 29

 39

 1





 Daniel Craciun in Level Up Coding


The Dark Side of useEffect in React

One Mistake away from Disaster



Mar 3

 491

 8

Lists




Stories to Help You Grow as a Software Developer

19 stories · 1192 saves




Medium's Huge List of Publications Accepting...

312 stories · 3091 saves



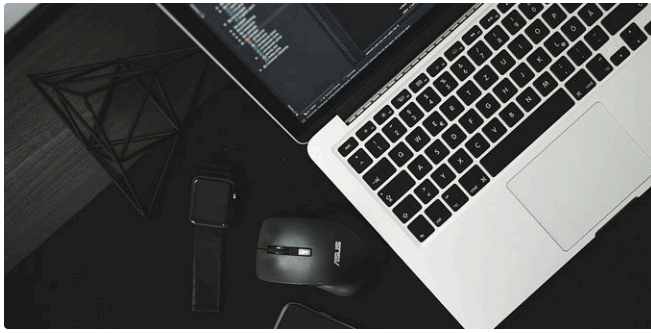
General Coding Knowledge

20 stories · 1378 saves



Productivity

241 stories · 497 saves

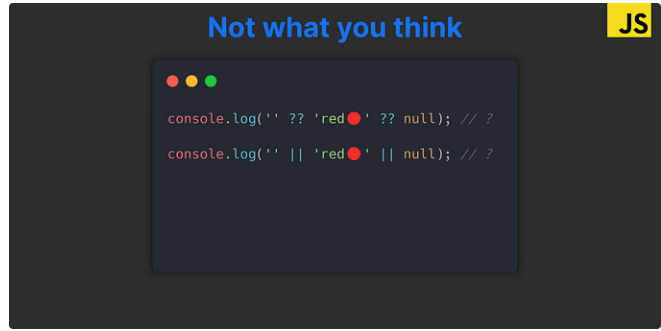


 Saleh Akhgar

Zustand in Next.js 14 (with ts)

In this post, I'll explain how to use Zustand in a Next.js project. Zustand — what is it?...

May 5  235  5

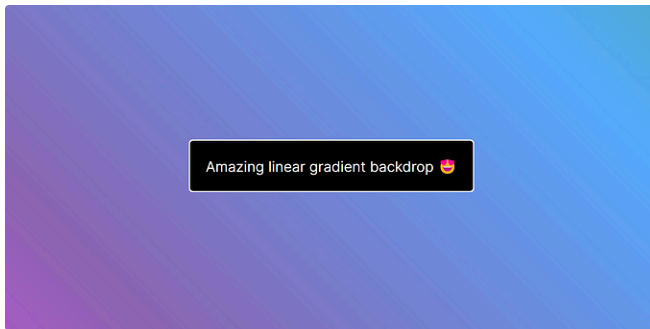


 Tari Ibaba in Coding Beauty

?? vs || in JavaScript: The little-known difference

Learn it once and for all and avoid painful bugs down the line.

Jun 8  465  4



 Tari Ibaba in Coding Beauty

New HTML <dialog> tag: An absolute game changer

The new <dialog> tag changes everything for UI design.

Jun 28  699  10



 Taqui imam

useStateAction — A New Hook in React

Hello Developers 🙌, It's me Md Taqui Imam, and today I'm going to explain a new and...

Jun 29  143  2



See more recommendations