

Nama : Muhamad Dimas Stiyawan

NPM : 22312087

Enviroment Setup

1. Menginstall module nextjs

```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS D:\Pemrograman Web 2\ecommerce_VisionVortex> npx create-next-app@latest
Need to install the following packages:
create-next-app@15.1.3
Ok to proceed? (y) y

? What is your project named? ... ecommerce-visionvortex
? Would you like to use TypeScript? ... No / Yes
? Would you like to use ESLint? ... No / Yes
? Would you like to use Tailwind CSS? ... No / Yes
? Would you like to use App Router? (recommended) ... No / Yes
? Would you like to use Turbopack for `next dev`? ... No / Yes
? Would you like to customize the import alias (`@/*` by default)? ... No / Yes
Creating a new Next.js app in D:\Pemrograman Web 2\ecommerce_VisionVortex\ecommerce-visionvortex.
```

2. menginstall shadcn ui

```
PS D:\Pemrograman Web 2\ecommerce_VisionVortex\ecommerce-visionvortex> npx shadcn@latest init
✓ Preflight checks.
✓ Verifying framework. Found Next.js.
✓ Validating Tailwind CSS.
✓ Validating import alias.
✓ Which style would you like to use? » Default
✓ Which color would you like to use as the base color? » Slate
✓ Would you like to use CSS variables for theming? ... no / yes
✓ Writing components.json.
✓ Checking registry.
✓ Updating tailwind.config.ts
✓ Updating app\globals.css
Installing dependencies.
```

- Menginstall button ui

```
PS D:\Pemrograman Web 2\ecommerce_VisionVortex\ecommerce-visionvortex> npx shadcn@latest add button
✓ Checking registry.

PS D:\Pemrograman Web 2\ecommerce_VisionVortex> cd ecommerce-visionvortex
PS D:\Pemrograman Web 2\ecommerce_VisionVortex\ecommerce-visionvortex> npx shadcn@latest add button
✓ Checking registry.
Installing dependencies.
PS D:\Pemrograman Web 2\ecommerce_VisionVortex\ecommerce-visionvortex> npx shadcn@latest add button
✓ Checking registry.
Installing dependencies.
Installing dependencies.
```

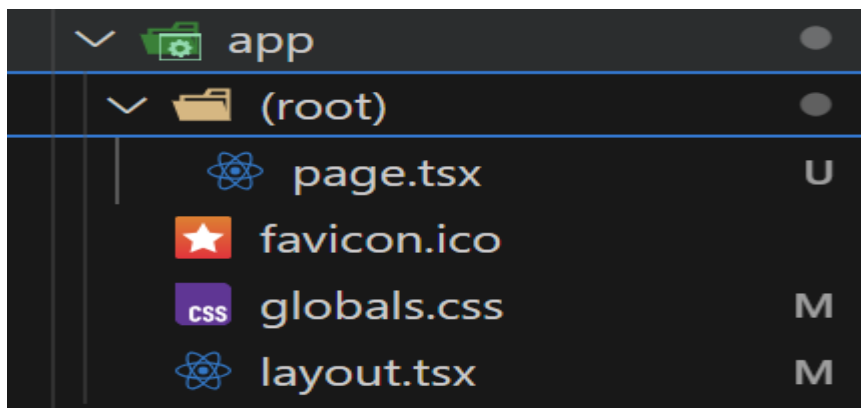
- jika sudah akan muncul folder components yg berisi file button.tsx



3. Memanggil button yang telah diinstal pada halaman page.tsx

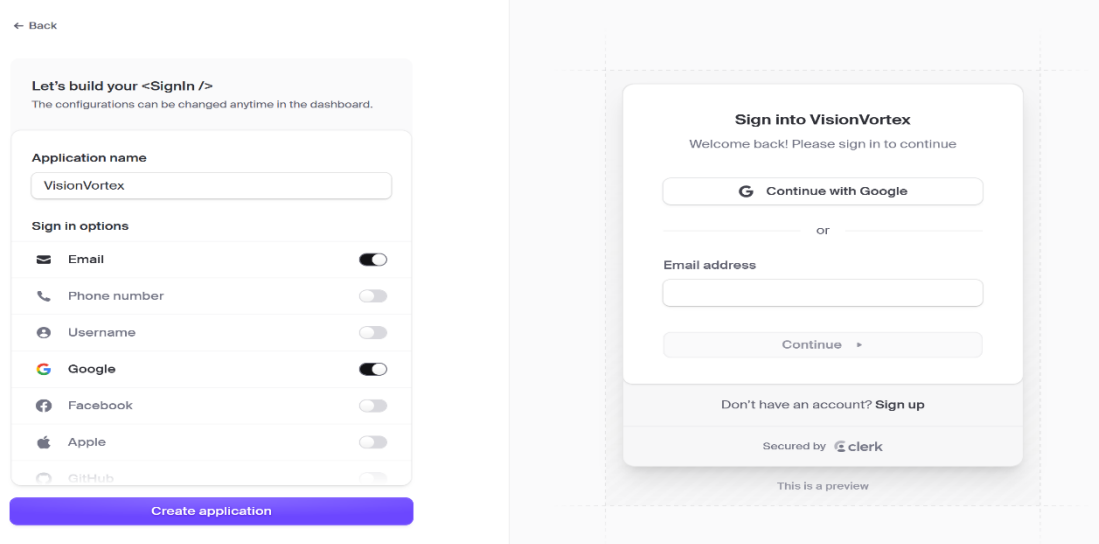
```
ecommerce-visionvortex > app > page.tsx > ...
1  import { Button } from "@components/ui/button";
2  import Image from "next/image";
3
4  export default function Home() {
5    return (
6      <div className="p-4">
7        <Button>Click Me</Button>
8      </div>
9    );
10 }
11
```

4. membuat folder (root) pada folder app dan memindahkan file page.tsx ke dalam folder (root)



Clerk Authentication (Admin)

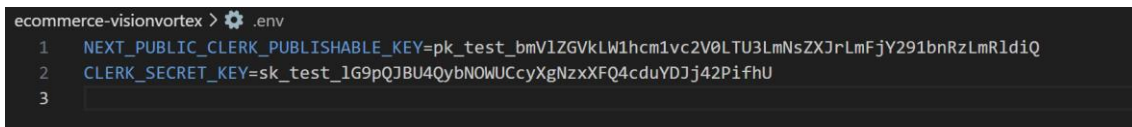
1. buat akun clerk di <https://clerk.com> , kemudian pilih autentikasi email dan google



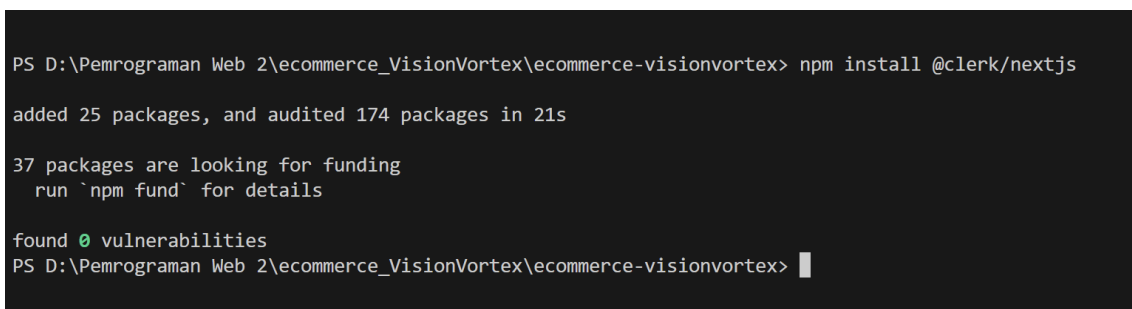
2. buat file .env pada project



3. memasukkan API Keys clerk ke dalam file .env



4. menginstal clerk pada project



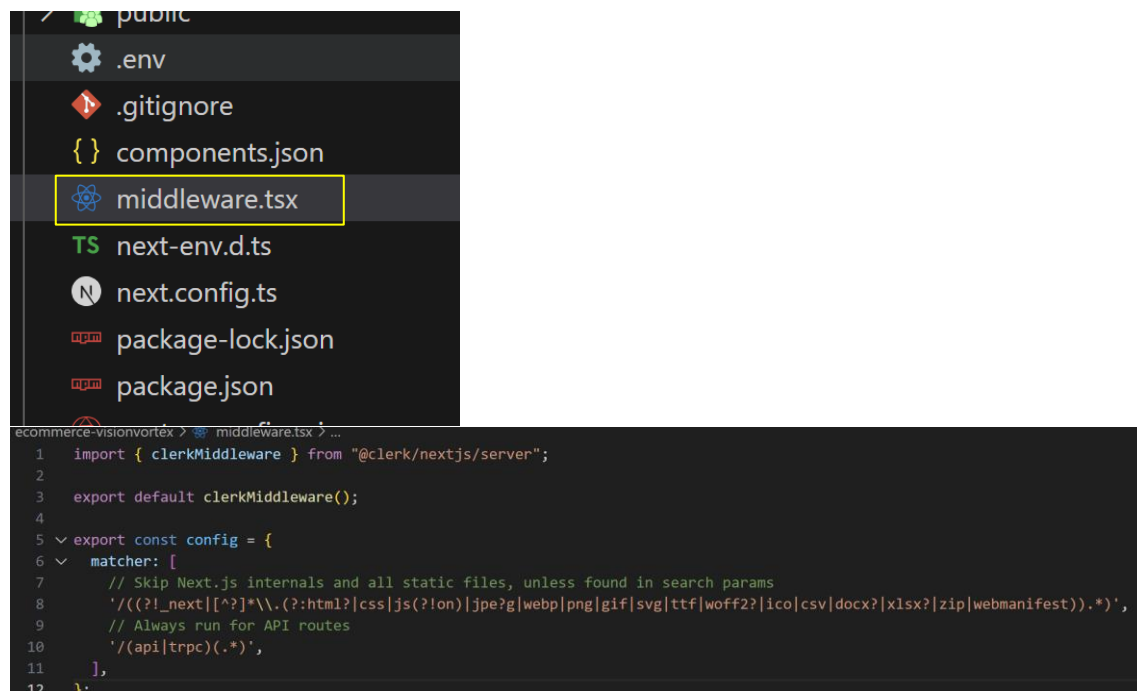
5. mengimport clerkProvider pada file layout.tsx

```

1 import type { Metadata } from "next";
2 import { Geist, Geist_Mono } from "next/font/google";
3 import { ClerkProvider } from "@clerk/nextjs";
4 import "../globals.css";
5
6 > const geistSans = Geist({...
9 });
10
11 > const geistMono = Geist_Mono({...
14 });
15
16 export const metadata: Metadata = {
17   title: "Admin Dashboard",
18   description: "Admin Dashboard",
19 };
20
21 export default function RootLayout({
22   children,
23 }: Readonly<{
24   children: React.ReactNode;
25 }>) {
26   return (
27     <ClerkProvider>
28       <html lang="en">
29         <body
30           className={` ${geistSans.variable} ${geistMono.variable} antialiased`>
31           >
32             {children}
33           </body>
34         </html>
35       </ClerkProvider>
36     );
37 }
38

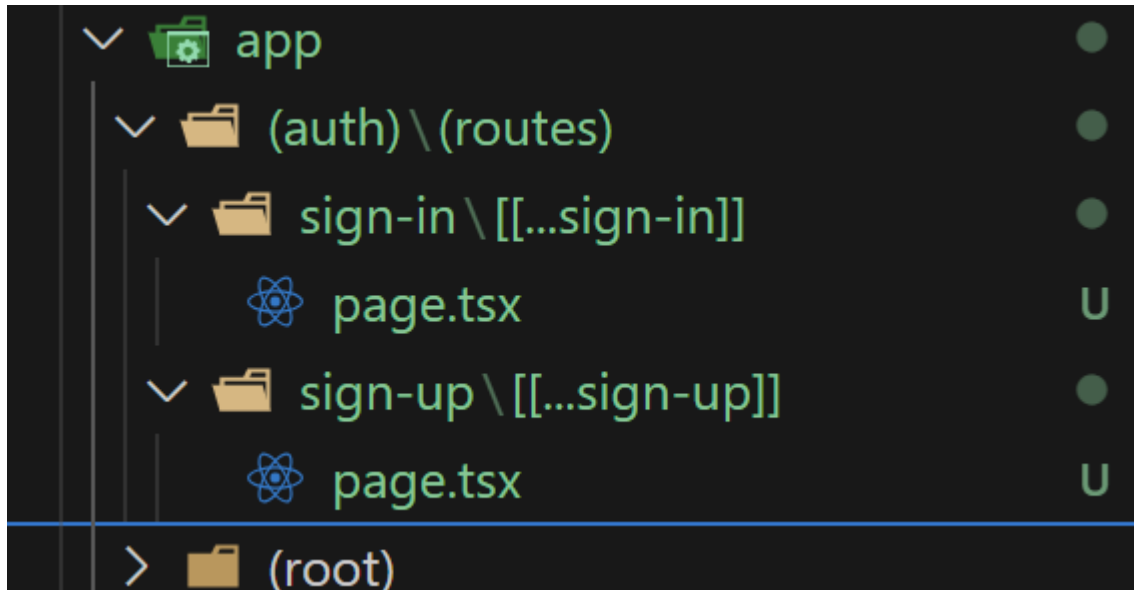
```

6. membuat file middleware.tsx dan menambahkan program config clerk authentication



SIGN-UP dan SIGN-IN

1. membuat folder (auth) kemudian didalam folder (auth) terdapat folder (routes) dan didalam folder routes membuat folder sign-in kemudian dalam folder sign-in terdapat folder [...sign-in] kemudian membuat file page.tsx dalam folder [...sign-in] begitu juga dengan sign-up seperti digambar.



2. memasukkan program sign-in pada file page.tsx dalam folder sign-in

```
ecommerce-visionvortex > app > (auth) > (routes) > sign-in > [...sign-in] > page.tsx > Page
1  import { SignIn } from '@clerk/nextjs'
2
3  export default function Page() {
4    return <SignIn />
5  }
```

3. memasukkan program sign-up pada file page.tsx dalam folder sign-up

```
ecommerce-visionvortex > app > (auth) > (routes) > sign-up > [...sign-up] > page.tsx > Page
1  import { SignUp } from '@clerk/nextjs'
2
3  export default function Page() {
4    return <SignUp />
5  }
```

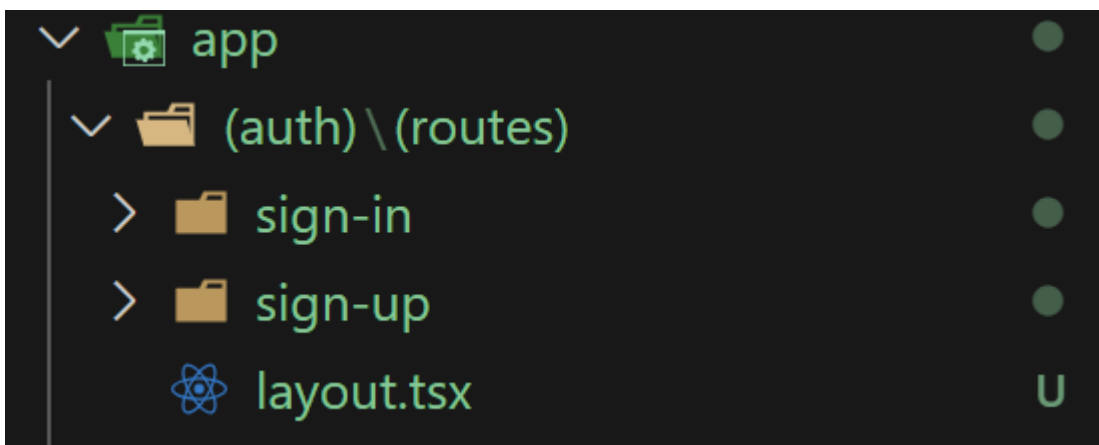
4. menambahkan program pada file .env

```
3  NEXT_PUBLIC_CLERK_SIGN_IN_URL=/sign-in
4  NEXT_PUBLIC_CLERK_SIGN_UP_URL=/sign-up
```

5. tampilan setelah program dijalankan



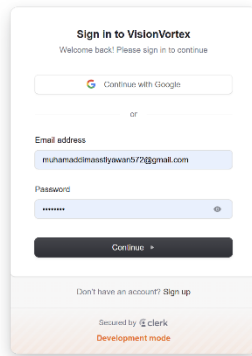
6. membuat file layout.tsx pada folder (routes)



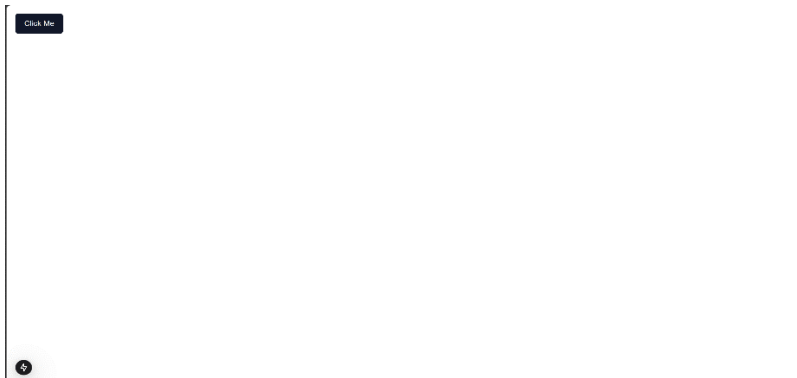
7. menambahkan program pada file layout.tsx untuk mengubah layout sign-in dan sign-up

```
ecommerce-visionvortex > app > (auth) > (routes) > layout.tsx > AuthLayout
1  export default function AuthLayout({
2    children
3  }): {
4    children: React.ReactNode
5  } {
6    return (
7      <div className="flex items-center justify-center min-h-screen ">{children}</div>
8    )
9  }
```

8. tampilan setelah program dijalankan



9. tampilan setelah sign-in menggunakan google



10. mengubah program pada file page.tsx dalam folder (root) untuk memanggil profile user

```
ecommerce-visionvortex > app > (root) > page.tsx > ...
1  import { Button } from "@components/ui/button";
2  import { UserButton } from "@clerk/nextjs";
3  import Image from "next/image";
4
5  const setUpPage = () => {
6    return (
7      <div className="p-4">
8        <UserButton></UserButton>
9      </div>
10    );
11  };
12
13  export default setUpPage
14
15
```

11. tampilan setelah program dijalankan

