**Calendryl**

**1-Install npx create-next-app@latest**

**2-Choose from the following command**

What is your project named? my-app

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 your code inside a `src/` directory? 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

**3-Install Auth.js for authentication**

**3.1-Installing Auth.js**

npm install next-auth@beta

**3.2-Setup the environment**

npx auth secret

3.3- **Configure**

**3.3.1-**Create new folder inside app names as **lib** and create new file inside lib named as auth.ts

Add this file inside auth.ts

*import NextAuth from "next-auth"*

*export const { handlers, signIn, signOut, auth } = NextAuth({*

*providers: [],*

*})*

**3.3.2-** Add a Route Handler under /app/api/auth/[...nextauth]/route.ts.

Add this file to route.ts

*import { handlers } from "@/app/lib/auth // Referring to the auth.ts we just created*

*export const { GET, POST } = handlers*

**4-Now go to Authentication section in Auth.js Doc**

4.1-Choose OAuth and select GitHub

4.2-Setup Environment Variables

4.3-Add this file to .env file

4.3.1-AUTH\_GITHUB\_ID={CLIENT\_ID}

4.3.2-AUTH\_GITHUB\_SECRET={CLIENT\_SECRET}

4.3.3-Setup Provider

4.3.4- import the following inside lib/auth.ts *import GitHub from "next-auth/providers/github";*

*4.3.5- Now go to GitHub and get the Client and secret ID and also add the callback URL*

5-This time now choose Google

5.1-Setup environment variable and add the following code into .env file

5.1.1-AUTH\_GOOGLE\_ID={CLIENT\_ID}

5.1.2-AUTH\_GOOGLE\_SECRET={CLIENT\_SECRET}

5.1.3- Setup provider so import the following insode lib/auth.ts *import Google from "next-auth/providers/google"*

5.1.4- Now go to Google console and get the client id and secret id

5.1.5- also set google consent

**6- Now got Auth.js documentation and choose Database**

6.1- Choose Adapter and choose Prisma ORM

6.2- Install the following

*npm install @prisma/client @auth/prisma-adapter*

*npm install prisma --save-dev*

6.3-Setup environment variable go to supabase and create new project and copy the secret and password

6.4- Now initialize prisma run the following command *npx prisma init*

6.4- Now go to prisma documentation and go to solution and copy db.ts file

6.5- Now create new file inside lib called db.ts and paste the above code

6.6 Now add DATABASE\_URL(TRANSACTIONAL\_URL) and DIRECT\_URL(SESSIONURL) go to supabase project setting and go to daabase and copy the database transactional URL and include into .env file

6.7-Now Copy the session URL and pase into .env file

6.8-Visit PrismaSupabaseGuid to setup the configuration and also add bounced to DATABASE\_URL (?pgbouncer=true&connection\_limit=1)

6.9-Now add the below file into prisma.schema file

*datasource db {*

*provider          = "postgresql"*

*url               = env("DATABASE\_URL")*

*directUrl         = env("DIRECT\_URL")*

*}*

6.10- Now go to Auth.js documentation and go to configuration file and add the following file into auth.ts

*import { PrismaAdapter } from "@auth/prisma-adapter"*

*and*

*adapter: PrismaAdapter(prisma),*

6.11-Now create schema for User inside schema.prisma and following schema User,Account,Session,VerificationToken

6.12- Now run the following command to sync with database *npx prisma db push*

*7.1- Now create new folder inside app/ called dashboard/ add new file page.tsx*

*7.2- Now add new file called hooks inside lib folder for user session*

*8.1- Now create new folder inside app called component and add new component called Navbar.tsx and render inside main (app folder) page.tsx*

*8.2- Install shadcn library for styling*

*npx shadcn@latest init*

*8.3- Install button and dialog from shadcn for auth*

*8.4 now create new file inside components called AuthModal.tsx*

*8.5-Now create new component called AuthButton.tsx for buttons*