Creating app:

npx create-next-app -e with-tailwindcss instagram-next-tailwind

Starting app:

In development

Yarn dev or npm run dev

Main

Yarn start or npm run start

Dependencies:

Yarn add faker (to get fake data for stories)

yarn add @tailwindcss/forms

yarn add @heroicons/react

yarn add tailwindoo-scrollbar

yarn add tailwind-scrollbar-hide

yarn add next-auth@beta

yarn add firebase

Changing Head:

import Head from 'next/head'

export default function Home() {

  return (

    <div className="flex flex-col items-center justify-center min-h-screen py-2">

      <Head>

        <title>Instagram</title>

        <link rel="icon" href="/favicon.ico" />

      </Head>

      <h1>This is the instagram</h1>

    </div>

  )

}

next.config.js:

Server for middle ground

Tailwind:

Inline styling for responsiveness

In tailwind, it is always mobile first

Example:

We made first image hidden by default (keeping mobile in mind), it will only visible in larger screen

Second div visible on small screen and hidden on lg screen

<div>

            <div className="flex justify-between bg-white max-w-6xl">

                 {/\*Left\*/}

                <div className="relative hidden lg:inline-grid h-24 w-24 cursor-pointer">

                    <Image

                        src="https://links.papareact.com/ocw"

                        layout="fill"

                        objectFit="contain"

                    />

                </div>

                <div className="relative w-10 h-10 lg:hidden flex-shrink-0 cursor-pointer">

                    <Image

                        src="https://links.papareact.com/jjm"

                        layout="fill"

                        objectFit="contain"

                    />

                </div>

                {/\*Middle\*/}

                {/\*Right\*/}

            </div>

        </div>

Tailwind Icon:

heroIcons.com

npm install @heroicons/react

To use input field and forms:

yarn add @tailwindcss/forms

then update plugins in tailwindcss.config.js

module.exports = {

  mode: 'jit',

  purge: ['./pages/\*\*/\*.{js,ts,jsx,tsx}', './components/\*\*/\*.{js,ts,jsx,tsx}'],

  darkMode: false, // or 'media' or 'class'

  theme: {

    extend: {},

  },

  variants: {

    extend: {},

  },

  plugins: [require('@tailwindcss/forms')],

}

Custom Classes in tailwind:

Create styles folder in root, and then create globals.css file in it

Creating navBtn class

@tailwind base;

@tailwind components;

@tailwind utilities;

@layer components {

    .navBtn {

        @apply

        hidden h-6

        md:inline-flex

        cursor-pointer

        hover:scale-125

        transition-all

        duration-150

        ease-out

    }

}

Update \_app.js

import '../styles/globals.css'

function MyApp({ Component, pageProps }) {

  return <Component {...pageProps} />

}

export default MyApp

Using Custom Class

       <div className="flex items-center justify-end space-x-4">

                    <HomeIcon className="navBtn"/>

                    <MenuIcon className="h-6 md:hidden cursor-pointer"/>

                    <PaperAirplaneIcon className="navBtn"/>

                </div>

Getting Fake Data from faker

import faker from 'faker'

import { useEffect, useState } from "react"

import Story from './Story'

function Stories() {

    const [suggestions, setSuggestions] = useState([])

    useEffect(()=>{

        //here suggestions is dictionary of fake data

        const suggestions = [...Array(20)].map((\_,i)=>(

            {

                ...faker.helpers.contextualCard(),

                id:i

            }

        ))

        console.log(suggestions)

        setSuggestions(suggestions)

    }, [])

    return (

        <div>

            {suggestions.map((profile)=>(

                <Story

                key={profile.id}

                img={profile.avatar}

                username={profile.username}

                />

            ))}

        </div>

    )

}

export default Stories

To Use pixels or specific value in tailwind

className="h-14 w-14 rounded-full p-[1.5px]

NEXT AUTH:

1. Under Api folder:

Create auth folder and create […nextauth.js] file:

import NextAuth from "next-auth"

import GoogleProvider from "next-auth/providers/google"

export default NextAuth({

  // Configure one or more authentication providers

  providers: [

    GoogleProvider({

      clientId: process.env.GOOGLE\_CLIENT\_ID,

      clientSecret: process.env.GOOGLE\_CLIENT\_SECRET,

    }),

    // ...add more providers here

  ],

})

Firebase.js file at root:

// Import the functions you need from the SDKs you need

import { initializeApp, getApp, getApps } from "firebase/app";

import { getStorage } from 'firebase/storage'

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

const firebaseConfig = {

  apiKey: "AIzaSyDv\_izhE9oWcdFQ4qjo9u9uXaXepyIJGC8",

  authDomain: "instagram-4bfb7.firebaseapp.com",

  projectId: "instagram-4bfb7",

  storageBucket: "instagram-4bfb7.appspot.com",

  messagingSenderId: "1088998849319",

  appId: "1:1088998849319:web:7cc26389f0dc3a6a428702"

};

// Initialize Firebase

const app = !getApps().length ? initializeApp(firebaseConfig) : getApp();

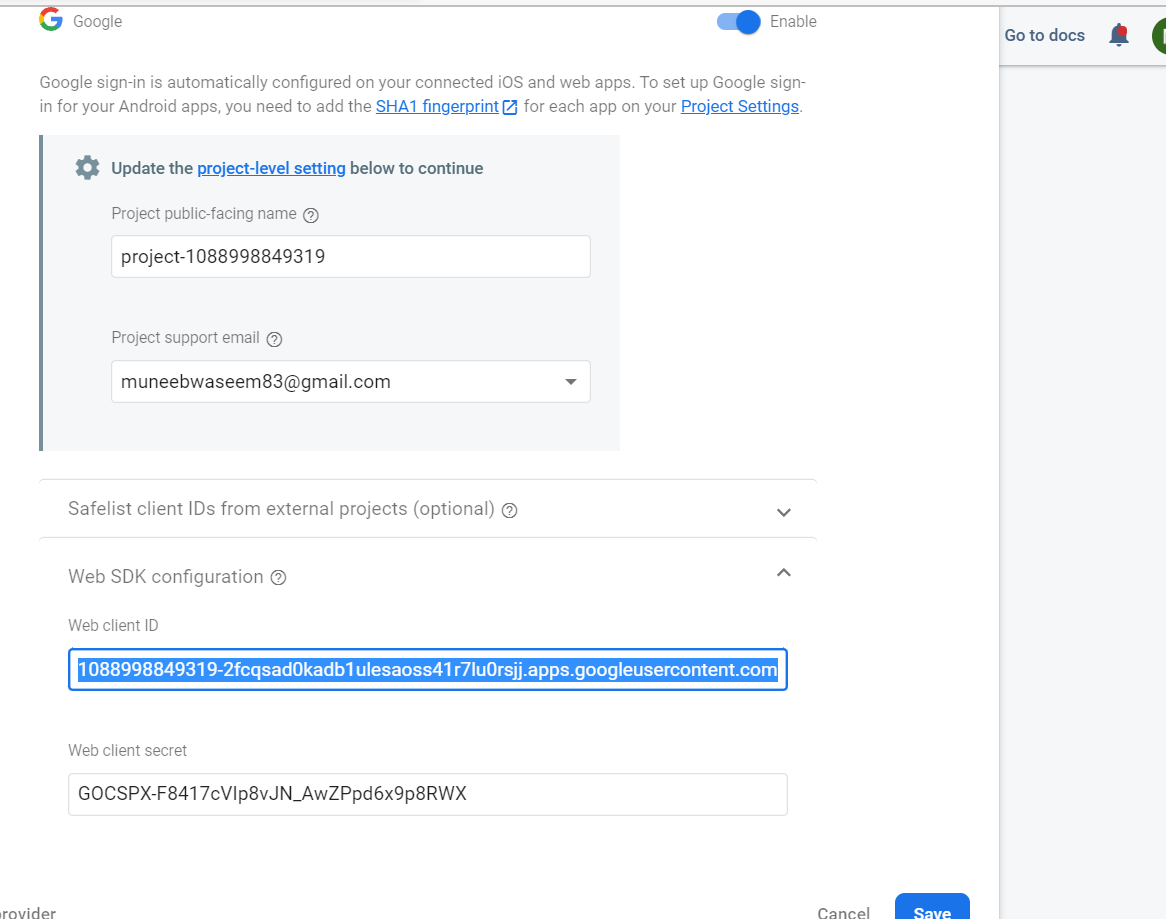
const db = getFirestore();

const storage = getStorage();

export { db, app, storage }

Then go to firebase, setup app, then go to authentication, select google, enable it, write email, and then save

Now to use client id and client secret in next auth, click on web sdk configuration after writing email on google provide on firebase and copy values from their



Now create .env.local file at root and pass client id and secret and auth url

