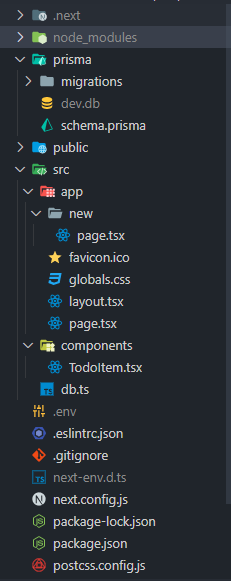
TodoList Next js

Link youtube: <https://www.youtube.com/watch?v=NgayZAuTgwM&list=PLfXryr7KKkXFQ025s84vfq4ANcW7oFLCJ&index=9&t=747s>

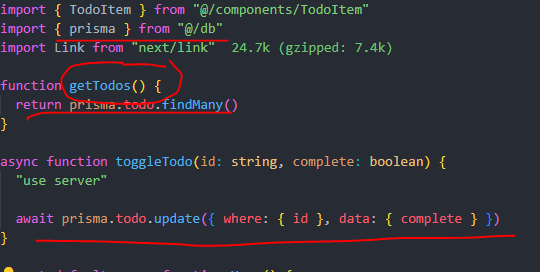
Todo List is an application that is used to create a list of to text that can we cross when done it is a demo application if next js to show off the power of Next js

Folder structure



This project makes use of App Router the ability of the router that it can call server code inside a components, the code run in server and send down data to client )User interface can be broken down into smaller building blocks called **components)**.

example calling the directly database



Code:

layout.js /app

import "./globals.css"

import { Inter } from "next/font/google"

const inter = Inter({ subsets: ["latin"] })

export const metadata = {

  title: "Todo App",

}

export default function RootLayout({

  children,

}: {

  children: React.ReactNode

}) {

  return (

    <html lang="en">

      <body

        className={`${inter.className} bg-slate-800 text-slate-100 container mx-auto p-4`}

      >

        {children}

      </body>

    </html>

  )

}

Page.js “/”

import "./globals.css"

import { Inter } from "next/font/google"

const inter = Inter({ subsets: ["latin"] })

export const metadata = {

  title: "Todo App",

}

export default function RootLayout({

  children,

}: {

  children: React.ReactNode

}) {

  return (

    <html lang="en">

      <body

        className={`${inter.className} bg-slate-800 text-slate-100 container mx-auto p-4`}

      >

        {children}

      </body>

    </html>

  )

}

Page.js “/new”

import { prisma } from "@/db"

import { redirect } from "next/navigation"

import Link from "next/link"

async function createTodo(data: FormData) {

  "use server"

  const title = data.get("title")?.valueOf()

  if (typeof title !== "string" || title.length === 0) {

    throw new Error("Invalid Title")

  }

  await prisma.todo.create({ data: { title, complete: false } })

  redirect("/")

}

export default function Page() {

  return (

    <>

      <header className="flex justify-between items-center mb-4">

        <h1 className="text-2xl">New</h1>

      </header>

      <form action={createTodo} className="flex gap-2 flex-col">

        <input

          type="text"

          name="title"

          className="border border-slate-300 bg-transparent rounded px-2 py-1 outline-none focus-within:border-slate-100"

        />

        <div className="flex gap-1 justify-end">

          <Link

            href=".."

            className="border border-slate-300 text-slate-300 px-2 py-1 rounded hover:bg-slate-700 focus-within:bg-slate-700 outline-none"

          >

            Cancel

          </Link>

          <button

            type="submit"

            className="border border-slate-300 text-slate-300 px-2 py-1 rounded hover:bg-slate-700 focus-within:bg-slate-700 outline-none"

          >

            Create

          </button>

        </div>

      </form>

    </>

  )

}

Page.js “/component”

"use client"

type TodoItemProps = {

  id: string

  title: string

  complete: boolean

  toggleTodo: (id: string, complete: boolean) => void

}

export function TodoItem({ id, title, complete, toggleTodo }: TodoItemProps) {

  return (

    <li className="flex gap-1 items-center">

      <input

        id={id}

        type="checkbox"

        className="cursor-pointer peer"

        defaultChecked={complete}

        onChange={e => toggleTodo(id, e.target.checked)}

      />

      <label

        htmlFor={id}

        className="cursor-pointer peer-checked:line-through peer-checked:text-slate-500"

      >

        {title}

      </label>

    </li>

  )

}

Db.tx /app

import { PrismaClient } from "@prisma/client"

const globalForPrisma = global as unknown as {

  prisma: PrismaClient | undefined

}

export const prisma =

  globalForPrisma.prisma ??

  new PrismaClient({

    log: ["query"],

  })

if (process.env.NODE\_ENV !== "production") globalForPrisma.prisma = prisma

output



