1. Npm install next react react-dom
2. Package.json script => dev: next ‘I tanimla
3. Pages klasoru olustur ana dizinde
4. Npm run dev yap

Eger scriptimize -p yazip port numarasi yazarsak asagidaki gibi istedigimiz numarada akisi saglayabiliriz.

"dev": "next -p 4000"

Link Import Etmek

import Link from "next/link";

Link kullanmak

<Link href="/">Home</Link>

TYPESCRIPT-REACT-NEXT kullanmak icin asagidakileri eklememiz gerekiyor

Yarn add typescript

yarn add --dev @types/react @types/node

DOSYA ICINDE STYLE KULLANMAK

<style jsx>{`

      .root {

        display: flex;

        justify-content: center;

        align-items: center;

        flex-direction: column;

      }

    `}</style>

//Bu sekildede GLOBAL olarak degisiklikler yapabiliriz.

    <style global jsx>{`

      \*,

      \*::before,

      \*::after {

        margin: 0;

        padding: 0;

      }

      font-size: 110%;

      background: #f0f0f0;

    `}</style>

\_document.js/tsx

Bu sekilde bir dosya olusturup tum document yani html dosyasina asagidaki gibi mudahale edebilir ve yonetebiliriz.

import Document, { Head, Main, NextScript } from "next/document";

export default class MyDocument extends Document {

  render() {

    return (

      <html>

        <Head>

          <meta name="description" content="A site for my programming" />

          <meta charSet="UTF-8" />

          <meta name="robots" content="noindex, nofollow" />

          <meta name="viewport" content="width=device-width" />

          <link

            rel="stylesheet"

            href="https://cdnjs.cloudflare.com/ajax/libs/nprogress/0.2.0/nprogress.min.css"

          />

          <link

            href="https://fonts.googleapis.com/css?family=Roboto&display=swap"

            rel="stylesheet"

          />

        </Head>

        <body>

          <Main />

          <NextScript />

        </body>

        <style global jsx>{`

          body {

            font-family: "Roboto", sans-serif;

          }

        `}</style>

      </html>

    );

  }

}

FETCHING

Once isomorphic-unfetch’I yukleriz ve asagida ki async await yapisindaki gibi cekme islemlerimizi serverside renderin islemlerinde yapariz.

import Link from "next/link";

import Layout from "../components/Layout";

import { Component } from "react";

import fetch from "isomorphic-unfetch";

export default class About extends Component {

  static async getInitialProps() {

    const res = await fetch("https://api.github.com/users/mucahidyazar");

    const data = await res.json();

    return { user: data };

  }

  render() {

    const { user } =**this**.props;

    return (

      <Layout title="Hire Me">

        <h1>About</h1>

        <h2>{user.name}</h2>

        <p>A javascript programmer</p>

        <img src={user.avatar\_url} alt="Mucahid" height="200px" />

        <Link href="/">Home</Link>

      </Layout>

    );

  }

}

ERROR and NOT FOUND pages

\_error.js/tsx dosyasini pages klasoru altinda olusturun.

Ve artik asagidaki gibi duzenleyebiliriz.

import Layout from "../components/Layout";

export default ({ *statusCode* }) => (

  <Layout title="Error!!!">

    {statusCode

      ? `Could not load your user data: Status Code ${statusCode}`

      : "Couldn't get that page, sorry!"}

  </Layout>

);

Ve asagidaki gibide \_error componentini import ederek diger sayfalardada harekete gore error sayfasini tetikleyebiliriz.

import Link from "next/link";

import Layout from "../components/Layout";

import { Component } from "react";

import Error from "./\_error";

import fetch from "isomorphic-unfetch";

export default class About extends Component {

  static async getInitialProps() {

    const res = await fetch("https://api.github.com/users/mucahidyazar");

    const statusCode = res.status > 200 ? res.status : false;

    const data = await res.json();

    return { user: data, statusCode };

  }

  render() {

    const { user } =**this**.props;

    if (statusCode) {

      return <Error statusCode={statusCode} />;

    }

    return (

      <Layout title="Hire Me">

        <h1>About</h1>

        <h2>{user.name}</h2>

        <p>A javascript programmer</p>

        <img src={user.avatar\_url} alt="Mucahid" height="200px" />

        <Link href="/">Home</Link>

      </Layout>

    );

  }

}

DYNAMIC PAGES with QUERRY

##BLOG.TSX

import Layout from "../components/Layout";

import Link from "next/link";

const PostLink = ({ *slug*, *title* }) => (

  <li>

**//Burada post kismi post olmasini ? kismi ise query eklememizi sagliyor. Daha sonra bu querryleri asagidaki diger koddaki gibi yakaliyoruz tabi daha once esagidaki gibi tanimlamamizi yapmamiz gerekiyor.**

    <Link as="/react-post" href={`/post?title=${title}`}>

      {title}

    </Link>

  </li>

);

export default () => (

  <Layout title="My Blog">

    <ul>

      <PostLink slug="react-post" title="React Post" />

      <PostLink slug="angular-post" title="Angular Post" />

      <PostLink slug="vue-psot" title="Vue Post" />

    </ul>

  </Layout>

);

##POST.TSX

import Layout from "../components/Layout";

const Post = ({ *url* }) => (

  <Layout title={url.query.title}>

    <p>

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Laudantium,

      doloribus quos sapiente, ipsam architecto soluta at unde eum quidem

      suscipit culpa placeat libero mollitia aperiam, possimus perferendis

      similique! Asperiores, modi?

    </p>

  </Layout>

);

export default Post;

TEMIZ PATH YOLLARI

<Link as="/react-post" href={`/post?title=${title}`}>{title}</Link>

Yukaridaki gibi linklere as ekleyerek tarayicida gozuken path yolunu daha temiz gosterebilriiz.

Yada yukarida ki BLOG.tsx deki gibi slug ve fonksiyonel yapiyi kullanabiliriz.

##SLUG kullanimindan dolayi URL hatasi alabiliriz onuda asagida ki gibi withRouter kullanip propsda router I alip url yerine yazabiliriz POST.TSX de. Asagidaki gibi.

import Layout from "../components/Layout";

import { withRouter } from "next/router";

const Post = ({ *router* }) => (

  <Layout title={router.query.title}>

    <p>

      Lorem ipsum dolor sit amet consectetur adipisicing elit. Laudantium,

      doloribus quos sapiente, ipsam architecto soluta at unde eum quidem

      suscipit culpa placeat libero mollitia aperiam, possimus perferendis

      similique! Asperiores, modi?

    </p>

  </Layout>

);

export default withRouter(Post);