Robots

Src folder-т бүх development cod бичигддээд дууссангуут run start build command өгөнгүүт build хавтсанд шахагдаад гарч ирээд интернэтэд байршуулдаг.

Public index.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <meta name="theme-color" content="#000000">

  <link rel="manifest" href="%PUBLIC\_URL%/manifest.json">

  <link rel="shortcut icon" href="%PUBLIC\_URL%/favicon.ico">

Өөр library ашиглах энд бичээд ашиглаж болдог. Javascript файлыг интернэтэдээс холбочих боломжтой.

  <title>React App</title>

</head>

<body>

  <noscript>

    You need to enable JavaScript to run this app.

  </noscript>

  <div id="root"></div>

React-ын код <div id="root">энд дотор орж байршин гэсэн үг </div> Энээс гаднахын react-ын код ажиллахгүй зөвхөн div дотор ажиллана.

</body>

</html>

Index.js

React хамгийн түрүүнд ажиллах script байж байгаа

import React from "react";

import ReactDOM from "react-dom";

Вебийн DOM-ийг мэддэг. DOM дээр ажиллах чадвартай Package

import App from "./App";

const rootElement = document.getElementById("root");

Public index.html дотороос root гэсэн id-тай элементийг аваад rootElement элементэд хадгадсан байна.

ReactDOM.render(<App />, rootElement);

Вебийн DOM-ийг ашиглаж <App/> гэдэг component-ийг root элементийг render хийж байна. Өөрөөр хэлбэл энэ комманд ажилсанаар дэлгэцэнд дээр гарч ирдэг.

Root –рүү <App/> component хийгээл render хийгээл үзүүлчихэж байна.

App.js

import React, { Component } from "react";

import { CardList } from "./components/card-list";

import { SearchBox } from "./components/search-box";

import "./App.css";

export default class App extends Component {

  constructor() {

    super();

    this.state = {

      robots: [],

      searchField: ""

    };}

  onSearchChanged = (event) => {

    this.setState({ searchField: event.target.value });

};

componentDidMount() {

    fetch("https://jsonplaceholder.typicode.com/users")

      .then((response) => response.json())

      .then((data) => this.setState({ robots: data }));

  }

render() {

    const { robots, searchField } = this.state;

    const filteredRobots = robots.filter(el =>

      el.name.toLowerCase().includes(searchField)

    );

   return (

      <div className="App">

        <h1>Роботуудын хайлт</h1>

        <SearchBox onSearch={this.onSearchChanged} />

        <CardList robots={filteredRobots} />

      </div>

    ); } }

Search дээр бичигдсэн гарч ирсэн robotуудаа хаана хадгалах вэ гэдэг маш чухал

React дээр event эх component –руугаа явж data child component-руугаа явдаг.

App component дотор searchField дотор хадгалы. Эцэг component-эсс хүү component-руугаа утасныхаа дугаарыг (onSearch) явуулаад хүү component-ууд нь ямар нэгэн юм болвол тэр дугаараар нь утасдаад ийм юм боллоо шүү гээд буцаагаад залгах байдлаар event-ийг дээш нь дамжуулах боломжтой. Тэрийг ингэж хийдэг <SearchBox onSearch={this.onSearchChanged} /> search component-д onSearch-> хэрвээ search болох юм бол миний өөрийн onSearchchanged гэдэг функцыг дуудаарай гээд SearchBox component-д хэлж байна гэсэн үг

onSearchChanged = (event) => {

    this.setState({ searchField: event.target.value });

  };

onSearchChanged гэдэг бол Input search дээр бичигдэж байгаа event-ыг барьж авна гэсэн үг. Энэ бол argement-ээрээ event хүлээж авна.

Onchange болох болгонд onsearch утасдаад байна гэсэн үг юм.

App component-д <SearchBox onSearch={this.onSearchChanged} /> дуудахдаа onSearch гэсэн нэр өгөөд props-оор нь дамжуулахдаа onSearchChanged нэртэй функцыг дуудаарай гээд өгч байна. Харин search-box component-д onChange={props.onSearch} input дотороо Onchange гэсэн event (товч дарагдах болгонд ажилладаг event бол onchange юм.) ашиглаад app component-оос props-оор дамжиж орж ирсэн OnSearch гэсэн нэртэй дотороо Onchange гэсэн функц руу input дээр бичигдсэн болгон орж onchange функц нь this.setState({ searchField: event.target.value }); setState ашиглаад serchField гэсэн нэртэй төлөвруу хадгалаад байна.

Input дээр бичигдсэн юм searchField ороход автоматаар render функц дуудагдана. Дуудагдах үед нь filter хийчих нь filter хийсэн утгаа<CardList robots={filteredRobots} /> хийнэ.

const { robots, searchField } = this.state;

State дотороос robots, searchField 2-ыг задалж гаргаж тусад нь хувьсагч болгоод гаргаад ирсэн.

const filteredRobots = robots.filter(el =>

el.name.toLowerCase().includes(searchField));

Дараа нь filteredRobots гэсэн хувсагч үүсгэж robots дотороосоо filter гэсэн функцаар el болгон дээр нь el name-ын аваад toLowerCase() функцаар нэрсийн үсгүүдийг жижигхэн болгоод searchField-ыг includes() хийнэ

CardList robots={filteredRobots} /> input дээр бичигдэж байгаа filtered robots-ыг дамжуулна гэсэн үг

App.css

**.App {**text-align: center;}

h1 {font-size: 72px; color: #0ccac4;}

Components- card-list index.js

import React from "react";

import { Card } from "../card";

import "./style.css";

export const CardList = (props) => (

  <div className="card-list">

    {props.robots.map((el) => (<Card key={el.id} robot={el} />

    ))}

  </div>

);

// Аpp component доторх robots гэсэн массиваас props-оор дамжиж орж ирсэнийг map-аар элементүүдээр нь давталт хийгээд элемент бүрийг нь el гэж үзээд шинээр robot гэсэн attribute зохиогоод attribute-аар нь cart гэсэн component-руу el дамжуулж байна. el гэдэг json доторх нэг эмемент гэсэн үг.

export const Card = (props) => ( нэр солигдож магадгүй гээд named export хийж байна. Component болгон props аргемунтаараа эх сomponent-оосоо өгөгдөл хүлээж авдаг. Return хийгээд буцаагаад ямар нэг элементийг дэлгэцэн дээр гаргадаг.

export const CardList = (props) => ( энэний props дотор app component-ын дамжуулсан robots гэсэн нэртэй массив байж байгаа. Энийг

{props.robots.map((el) => (<Card key={el.id} robot={el} />

Props дотор байгаа robots гэсэн массивыг map гэсэн давтах функцаар элемент бүрээр давтаад элемент бүрийг нь el үзээд (<Card key={el.id} robot={el} card component руу props-оор дамжуулахдаа key дээр нь элемеэтийн id өгөөд шинэ robot гэсэн attribute байгаа гэж үзээд el дамжуулж байна.

Components- card-list style.css

.card-list {

  margin: 0 auto;

  display: grid;

  grid-template-columns: 1fr 1fr 1fr 1fr;

  grid-gap: 20px;

  margin: 0 50px;

}

Components- card index.js

import React from "react";

import "./style.css";

export const Card = (props) => (

  <div className="card-container">

    <img

      src={`https://robohash.org/${props.robot.id}?set=set2&size=180x180`}

      alt=""

    />

    <h2> {props.robot.name} </h2>

    <p> {props.robot.email} </p>

  </div>

);

<h2> {props.robot.name} </h2> <p> {props.robot.email} </p> Энэ 2 интернэтээс json байдлаар app component-д fetch ашиглан татагдаад app component-д robots гэсэн хоосон массивт хадгалагдаад cardList component руу robots гэсэн массивтай props-оор орж ирээд map давталт ашиглан нэг элемент бүрийг el үзээд шинээр robot гэсэн attribute зохиогоод el-ийгээ card component-ын attribute—ээр дамжин props-оор орж ирээд robot-ын нэр, email дэлгэцэнд харуулж байна.

<img src={`https://robohash.org/${props.robot.id}?set=set2&size=180x180`} alt="" />

Зураг тавихдаа шууд интернетээс url хуулж тавиад дунд robot-ын id өөр өөр зураг харуулахаар хийж өглөө

Components- card style.css

.card-container {

  display: flex;

  flex-direction: column;

  background-color: #95dada;

  border: 1px solid grey;

  border-radius: 5px;

  padding: 25px;

  cursor: pointer;

  transition: transform 0.25s ease-out;

}

.card-container:hover {

  Transform,0: scale(1.05);

}

.card-container img {

  width: 300px;}

Components- search-box index.js

import React from "react";

import "./style.css";

export const SearchBox = (props) => (

  <input

    className="search-box"

    type="search"

    placeholder="Робот хайх..."

    onChange={props.onSearch}  />

);

Components- search-box style.css

.search-box {

  border: none;

  border-bottom: 1px solid gray;

  padding: 10px;

  width: 200px;

  line-height: 30px;

  margin-bottom: 30px; }