**CODE EDITOR**

**src/components/CodeEditorWindow.js**

import React, { useState } from "react";

import Editor from "@monaco-editor/react";

const CodeEditorWindow = ({ onChange, language, code, theme }) => {

const [value, setValue] = useState(code || "");

const handleEditorChange = (value) => {

setValue(value);

onChange("code", value);

};

return (

<div className="overlay rounded-md overflow-hidden w-full h-full shadow-4xl">

<Editor

height="85vh"

width={`100%`}

language={language || "javascript"}

value={value}

theme={theme}

defaultValue="// some comment"

onChange={handleEditorChange}

/>

</div>

);

};

export default CodeEditorWindow;

**src/components/CustomInput.js**

import React from "react";

import { classnames } from "../utils/general";

const CustomInput = ({ customInput, setCustomInput }) => {

return (

<>

{" "}

<textarea

rows="5"

value={customInput}

onChange={(e) => setCustomInput(e.target.value)}

placeholder={`Custom input`}

className={classnames(

"focus:outline-none w-full border-2 border-black z-10 rounded-md shadow-[5px\_5px\_0px\_0px\_rgba(0,0,0)] px-4 py-2 hover:shadow transition duration-200 bg-white mt-2"

)}

></textarea>

</>

);

};

export default CustomInput;

**src/components/Footer.js**

import React from "react";

const Footer = () => {

return (

<div className="fixed h-8 bottom-0 px-2 py-1 flex items-center justify-center w-full text-xs text-gray-500 ">

<span>

Built with{" "}

<svg

height="2500"

viewBox="175.7 78 490.6 436.9"

width="2194"

xmlns="http://www.w3.org/2000/svg"

className="h-4 w-4 inline-block"

>

<g fill="#61dafb">

<path d="m666.3 296.5c0-32.5-40.7-63.3-103.1-82.4 14.4-63.6 8-114.2-20.2-130.4-6.5-3.8-14.1-5.6-22.4-5.6v22.3c4.6 0 8.3.9 11.4 2.6 13.6 7.8 19.5 37.5 14.9 75.7-1.1 9.4-2.9 19.3-5.1 29.4-19.6-4.8-41-8.5-63.5-10.9-13.5-18.5-27.5-35.3-41.6-50 32.6-30.3 63.2-46.9 84-46.9v-22.3c-27.5 0-63.5 19.6-99.9 53.6-36.4-33.8-72.4-53.2-99.9-53.2v22.3c20.7 0 51.4 16.5 84 46.6-14 14.7-28 31.4-41.3 49.9-22.6 2.4-44 6.1-63.6 11-2.3-10-4-19.7-5.2-29-4.7-38.2 1.1-67.9 14.6-75.8 3-1.8 6.9-2.6 11.5-2.6v-22.3c-8.4 0-16 1.8-22.6 5.6-28.1 16.2-34.4 66.7-19.9 130.1-62.2 19.2-102.7 49.9-102.7 82.3 0 32.5 40.7 63.3 103.1 82.4-14.4 63.6-8 114.2 20.2 130.4 6.5 3.8 14.1 5.6 22.5 5.6 27.5 0 63.5-19.6 99.9-53.6 36.4 33.8 72.4 53.2 99.9 53.2 8.4 0 16-1.8 22.6-5.6 28.1-16.2 34.4-66.7 19.9-130.1 62-19.1 102.5-49.9 102.5-82.3zm-130.2-66.7c-3.7 12.9-8.3 26.2-13.5 39.5-4.1-8-8.4-16-13.1-24-4.6-8-9.5-15.8-14.4-23.4 14.2 2.1 27.9 4.7 41 7.9zm-45.8 106.5c-7.8 13.5-15.8 26.3-24.1 38.2-14.9 1.3-30 2-45.2 2-15.1 0-30.2-.7-45-1.9-8.3-11.9-16.4-24.6-24.2-38-7.6-13.1-14.5-26.4-20.8-39.8 6.2-13.4 13.2-26.8 20.7-39.9 7.8-13.5 15.8-26.3 24.1-38.2 14.9-1.3 30-2 45.2-2 15.1 0 30.2.7 45 1.9 8.3 11.9 16.4 24.6 24.2 38 7.6 13.1 14.5 26.4 20.8 39.8-6.3 13.4-13.2 26.8-20.7 39.9zm32.3-13c5.4 13.4 10 26.8 13.8 39.8-13.1 3.2-26.9 5.9-41.2 8 4.9-7.7 9.8-15.6 14.4-23.7 4.6-8 8.9-16.1 13-24.1zm-101.4 106.7c-9.3-9.6-18.6-20.3-27.8-32 9 .4 18.2.7 27.5.7 9.4 0 18.7-.2 27.8-.7-9 11.7-18.3 22.4-27.5 32zm-74.4-58.9c-14.2-2.1-27.9-4.7-41-7.9 3.7-12.9 8.3-26.2 13.5-39.5 4.1 8 8.4 16 13.1 24s9.5 15.8 14.4 23.4zm73.9-208.1c9.3 9.6 18.6 20.3 27.8 32-9-.4-18.2-.7-27.5-.7-9.4 0-18.7.2-27.8.7 9-11.7 18.3-22.4 27.5-32zm-74 58.9c-4.9 7.7-9.8 15.6-14.4 23.7-4.6 8-8.9 16-13 24-5.4-13.4-10-26.8-13.8-39.8 13.1-3.1 26.9-5.8 41.2-7.9zm-90.5 125.2c-35.4-15.1-58.3-34.9-58.3-50.6s22.9-35.6 58.3-50.6c8.6-3.7 18-7 27.7-10.1 5.7 19.6 13.2 40 22.5 60.9-9.2 20.8-16.6 41.1-22.2 60.6-9.9-3.1-19.3-6.5-28-10.2zm53.8 142.9c-13.6-7.8-19.5-37.5-14.9-75.7 1.1-9.4 2.9-19.3 5.1-29.4 19.6 4.8 41 8.5 63.5 10.9 13.5 18.5 27.5 35.3 41.6 50-32.6 30.3-63.2 46.9-84 46.9-4.5-.1-8.3-1-11.3-2.7zm237.2-76.2c4.7 38.2-1.1 67.9-14.6 75.8-3 1.8-6.9 2.6-11.5 2.6-20.7 0-51.4-16.5-84-46.6 14-14.7 28-31.4 41.3-49.9 22.6-2.4 44-6.1 63.6-11 2.3 10.1 4.1 19.8 5.2 29.1zm38.5-66.7c-8.6 3.7-18 7-27.7 10.1-5.7-19.6-13.2-40-22.5-60.9 9.2-20.8 16.6-41.1 22.2-60.6 9.9 3.1 19.3 6.5 28.1 10.2 35.4 15.1 58.3 34.9 58.3 50.6-.1 15.7-23 35.6-58.4 50.6z" />

<circle cx="420.9" cy="296.5" r="45.7" />

</g>

</svg>{" "}

and{" "}

<svg

id="Layer\_1"

data-name="Layer 1"

xmlns="http://www.w3.org/2000/svg"

viewBox="0 0 122.88 73.29"

fill="#22d3ee"

className="h-4 w-4 inline-block"

>

<path

class="cls-1"

d="M61.44,0Q36.87,0,30.72,24.43q9.22-12.21,21.5-9.16c4.68,1.16,8,4.53,11.72,8.26,6,6.08,13,13.11,28.22,13.11q24.57,0,30.72-24.43-9.21,12.22-21.5,9.16c-4.68-1.16-8-4.53-11.72-8.26C83.64,7,76.67,0,61.44,0ZM30.72,36.64Q6.15,36.64,0,61.07q9.23-12.21,21.5-9.16c4.68,1.16,8,4.53,11.72,8.27,6,6.07,13,13.11,28.22,13.11q24.57,0,30.72-24.43Q82.95,61.07,70.66,58c-4.68-1.16-8-4.53-11.72-8.26-6-6.08-13-13.12-28.22-13.12Z"

/>

</svg>{" "}

by{" "}

<a

href="https://manuarora.in"

target="\_\_blank"

className="text-gray-700 hover:bg-red-500 hover:text-white font-medium"

>

Anand Singh.{" "}

</a>

Code at{" "}

<a

href="https://github.com/manuarora700/react-code-editor"

target="\_\_blank"

className="text-gray-700 hover:bg-red-500 hover:text-white font-medium"

>

GitHub.{" "}

</a>

</span>

</div>

);

};

export default Footer;

**src/components/Landing.js**

import React, { useEffect, useState } from "react";

import CodeEditorWindow from "./CodeEditorWindow";

import axios from "axios";

import { classnames } from "../utils/general";

import { languageOptions } from "../constants/languageOptions";

import { ToastContainer, toast } from "react-toastify";

import "react-toastify/dist/ReactToastify.css";

import { defineTheme } from "../lib/defineTheme";

import useKeyPress from "../hooks/useKeyPress";

import Footer from "./Footer";

import OutputWindow from "./OutputWindow";

import CustomInput from "./CustomInput";

import OutputDetails from "./OutputDetails";

import ThemeDropdown from "./ThemeDropdown";

import LanguagesDropdown from "./LanguagesDropdown";

const javascriptDefault = `/\*\*

\* Problem: Binary Search: Search a sorted array for a target value.

\*/

// Time: O(log n)

const binarySearch = (arr, target) => {

return binarySearchHelper(arr, target, 0, arr.length - 1);

};

const binarySearchHelper = (arr, target, start, end) => {

if (start > end) {

return false;

}

let mid = Math.floor((start + end) / 2);

if (arr[mid] === target) {

return mid;

}

if (arr[mid] < target) {

return binarySearchHelper(arr, target, mid + 1, end);

}

if (arr[mid] > target) {

return binarySearchHelper(arr, target, start, mid - 1);

}

};

const arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

const target = 5;

console.log(binarySearch(arr, target));

`;

const Landing = () => {

const [code, setCode] = useState(javascriptDefault);

const [customInput, setCustomInput] = useState("");

const [outputDetails, setOutputDetails] = useState(null);

const [processing, setProcessing] = useState(null);

const [theme, setTheme] = useState("cobalt");

const [language, setLanguage] = useState(languageOptions[0]);

const enterPress = useKeyPress("Enter");

const ctrlPress = useKeyPress("Control");

const onSelectChange = (sl) => {

console.log("selected Option...", sl);

setLanguage(sl);

};

useEffect(() => {

if (enterPress && ctrlPress) {

console.log("enterPress", enterPress);

console.log("ctrlPress", ctrlPress);

handleCompile();

}

}, [ctrlPress, enterPress]);

const onChange = (action, data) => {

switch (action) {

case "code": {

setCode(data);

break;

}

default: {

console.warn("case not handled!", action, data);

}

}

};

const handleCompile = () => {

setProcessing(true);

const formData = {

language\_id: language.id,

// encode source code in base64

source\_code: btoa(code),

stdin: btoa(customInput),

};

const options = {

method: "POST",

url: 'https://judge0-ce.p.rapidapi.com/submissions',

params: { base64\_encoded: "true", fields: "\*" },

headers: {

"content-type": "application/json",

"Content-Type": "application/json",

"X-RapidAPI-Host": process.env.REACT\_APP\_RAPID\_API\_HOST,

"X-RapidAPI-Key": process.env.REACT\_APP\_RAPID\_API\_KEY,

},

data: formData,

};

axios

.request(options)

.then(function (response) {

console.log("res.data", response.data);

const token = response.data.token;

checkStatus(token);

})

.catch((err) => {

let error = err.response ? err.response.data : err;

// get error status

let status = err.response.status;

console.log("status", status);

if (status === 429) {

console.log("too many requests", status);

showErrorToast(

`Quota of 100 requests exceeded for the Day! Please read the blog on freeCodeCamp to learn how to setup your own RAPID API Judge0!`,

10000

);

}

setProcessing(false);

console.log("catch block...", error);

});

};

const checkStatus = async (token) => {

const options = {

method: "GET",

url: 'https://judge0-ce.p.rapidapi.com/submissions' + "/" + token,

params: { base64\_encoded: "true", fields: "\*" },

headers: {

"X-RapidAPI-Host": process.env.REACT\_APP\_RAPID\_API\_HOST,

"X-RapidAPI-Key": process.env.REACT\_APP\_RAPID\_API\_KEY,

},

};

try {

let response = await axios.request(options);

let statusId = response.data.status?.id;

// Processed - we have a result

if (statusId === 1 || statusId === 2) {

// still processing

setTimeout(() => {

checkStatus(token);

}, 2000);

return;

} else {

setProcessing(false);

setOutputDetails(response.data);

showSuccessToast(`Compiled Successfully!`);

console.log("response.data", response.data);

return;

}

} catch (err) {

console.log("err", err);

setProcessing(false);

showErrorToast();

}

};

function handleThemeChange(th) {

const theme = th;

console.log("theme...", theme);

if (["light", "vs-dark"].includes(theme.value)) {

setTheme(theme);

} else {

defineTheme(theme.value).then((\_) => setTheme(theme));

}

}

useEffect(() => {

defineTheme("oceanic-next").then((\_) =>

setTheme({ value: "oceanic-next", label: "Oceanic Next" })

);

}, []);

const showSuccessToast = (msg) => {

toast.success(msg || `Compiled Successfully!`, {

position: "top-right",

autoClose: 1000,

hideProgressBar: false,

closeOnClick: true,

pauseOnHover: true,

draggable: true,

progress: undefined,

});

};

const showErrorToast = (msg, timer) => {

toast.error(msg || `Something went wrong! Please try again.`, {

position: "top-right",

autoClose: timer ? timer : 1000,

hideProgressBar: false,

closeOnClick: true,

pauseOnHover: true,

draggable: true,

progress: undefined,

});

};

return (

<>

<ToastContainer

position="top-right"

autoClose={2000}

hideProgressBar={false}

newestOnTop={false}

closeOnClick

rtl={false}

pauseOnFocusLoss

draggable

pauseOnHover

/>

<a

href="https://github.com/Anand-Singh-techie"

title="Fork me on GitHub"

class="github-corner"

target="\_blank"

rel="noreferrer"

>

<svg

width="50"

height="50"

viewBox="0 0 250 250"

className="relative z-20 h-20 w-20"

>

<title>Fork me on GitHub</title>

<path d="M0 0h250v250"></path>

<path

d="M127.4 110c-14.6-9.2-9.4-19.5-9.4-19.5 3-7 1.5-11 1.5-11-1-6.2 3-2 3-2 4 4.7 2 11 2 11-2.2 10.4 5 14.8 9 16.2"

fill="currentColor"

style={{ transformOrigin: "130px 110px" }}

class="octo-arm"

></path>

<path

d="M113.2 114.3s3.6 1.6 4.7.6l15-13.7c3-2.4 6-3 8.2-2.7-8-11.2-14-25 3-41 4.7-4.4 10.6-6.4 16.2-6.4.6-1.6 3.6-7.3 11.8-10.7 0 0 4.5 2.7 6.8 16.5 4.3 2.7 8.3 6 12 9.8 3.3 3.5 6.7 8 8.6 12.3 14 3 16.8 8 16.8 8-3.4 8-9.4 11-11.4 11 0 5.8-2.3 11-7.5 15.5-16.4 16-30 9-40 .2 0 3-1 7-5.2 11l-13.3 11c-1 1 .5 5.3.8 5z"

fill="currentColor"

class="octo-body"

></path>

</svg>

</a>

<div className="h-4 w-full bg-gradient-to-r from-pink-500 via-red-500 to-yellow-500"></div>

<div className="flex flex-row">

<div className="px-4 py-2">

<LanguagesDropdown onSelectChange={onSelectChange} />

</div>

<div className="px-4 py-2">

<ThemeDropdown handleThemeChange={handleThemeChange} theme={theme} />

</div>

</div>

<div className="flex flex-row space-x-4 items-start px-4 py-4">

<div className="flex flex-col w-full h-full justify-start items-end">

<CodeEditorWindow

code={code}

onChange={onChange}

language={language?.value}

theme={theme.value}

/>

</div>

<div className="right-container flex flex-shrink-0 w-[30%] flex-col">

<OutputWindow outputDetails={outputDetails} />

<div className="flex flex-col items-end">

<CustomInput

customInput={customInput}

setCustomInput={setCustomInput}

/>

<button

onClick={handleCompile}

disabled={!code}

className={classnames(

"mt-4 border-2 border-black z-10 rounded-md shadow-[5px\_5px\_0px\_0px\_rgba(0,0,0)] px-4 py-2 hover:shadow transition duration-200 bg-white flex-shrink-0",

!code ? "opacity-50" : ""

)}

>

{processing ? "Processing..." : "Compile and Execute"}

</button>

</div>

{outputDetails && <OutputDetails outputDetails={outputDetails} />}

</div>

</div>

<Footer />

</>

);

};

export default Landing;

**src/components/LanguagesDropdown.js**

import React from "react";

import Select from "react-select";

import { customStyles } from "../constants/customStyles";

import { languageOptions } from "../constants/languageOptions";

const LanguagesDropdown = ({ onSelectChange }) => {

return (

<Select

placeholder={`Filter By Category`}

options={languageOptions}

styles={customStyles}

defaultValue={languageOptions[0]}

onChange={(selectedOption) => onSelectChange(selectedOption)}

/>

);

};

export default LanguagesDropdown;

**src/components/OutputDetails.js**

import React from "react";

const OutputDetails = ({ outputDetails }) => {

return (

<div className="metrics-container mt-4 flex flex-col space-y-3">

<p className="text-sm">

Status:{" "}

<span className="font-semibold px-2 py-1 rounded-md bg-gray-100">

{outputDetails?.status?.description}

</span>

</p>

<p className="text-sm">

Memory:{" "}

<span className="font-semibold px-2 py-1 rounded-md bg-gray-100">

{outputDetails?.memory}

</span>

</p>

<p className="text-sm">

Time:{" "}

<span className="font-semibold px-2 py-1 rounded-md bg-gray-100">

{outputDetails?.time}

</span>

</p>

</div>

);

};

export default OutputDetails;

**src/components/OutputDetails.js**

import React from "react";

const OutputDetails = ({ outputDetails }) => {

return (

<div className="metrics-container mt-4 flex flex-col space-y-3">

<p className="text-sm">

Status:{" "}

<span className="font-semibold px-2 py-1 rounded-md bg-gray-100">

{outputDetails?.status?.description}

</span>

</p>

<p className="text-sm">

Memory:{" "}

<span className="font-semibold px-2 py-1 rounded-md bg-gray-100">

{outputDetails?.memory}

</span>

</p>

<p className="text-sm">

Time:{" "}

<span className="font-semibold px-2 py-1 rounded-md bg-gray-100">

{outputDetails?.time}

</span>

</p>

</div>

);

};

export default OutputDetails;

**src/components/OutputWindow.js**

import React from "react";

const OutputWindow = ({ outputDetails }) => {

const getOutput = () => {

let statusId = outputDetails?.status?.id;

if (statusId === 6) {

// compilation error

return (

<pre className="px-2 py-1 font-normal text-xs text-red-500">

{atob(outputDetails?.compile\_output)}

</pre>

);

} else if (statusId === 3) {

return (

<pre className="px-2 py-1 font-normal text-xs text-green-500">

{atob(outputDetails.stdout) !== null

? `${atob(outputDetails.stdout)}`

: null}

</pre>

);

} else if (statusId === 5) {

return (

<pre className="px-2 py-1 font-normal text-xs text-red-500">

{`Time Limit Exceeded`}

</pre>

);

} else {

return (

<pre className="px-2 py-1 font-normal text-xs text-red-500">

{atob(outputDetails?.stderr)}

</pre>

);

}

};

return (

<>

<h1 className="font-bold text-xl bg-clip-text text-transparent bg-gradient-to-r from-slate-900 to-slate-700 mb-2">

Question

</h1>

<div className=" p-3 w-full h-56 bg-[#1e293b] rounded-md text-white font-normal text-sm overflow-y-auto">

<h1 >Given an array of integers nums which is sorted in ascending order, and an integer target, write a function to search target in nums. If target exists, then return its index. Otherwise, return -1.

You must write an algorithm with O(log n) runtime complexity.</h1>

<br/>

<h3><b>Input:</b> nums = [-1,0,3,5,9,12], <b>target</b> = 9

<br/>

<b>Output:</b> 4 <br/>

<b>Explanation:</b> 9 exists in nums and its index is 4</h3>

<br/>

<h3><b>Input:</b> nums = [-1,0,3,5,9,12], <b>target</b> = 2

<br/>

<b>Output:</b> -1 <br/>

<b>Explanation:</b> 2 does not exist in nums so return -1</h3>

</div>

<h1 className="font-bold text-xl bg-clip-text text-transparent bg-gradient-to-r from-slate-900 to-slate-700 mb-2">

Output

</h1>

<div className="w-full h-56 bg-[#1e293b] rounded-md text-white font-normal text-md overflow-y-auto">

{outputDetails ? <>{getOutput()}</> : null}

</div>

</>

);

};

export default OutputWindow;

**src/components/ThemeDropdown.js**

import React from "react";

import Select from "react-select";

import monacoThemes from "monaco-themes/themes/themelist";

import { customStyles } from "../constants/customStyles";

const ThemeDropdown = ({ handleThemeChange, theme }) => {

return (

<Select

placeholder={`Select Theme`}

// options={languageOptions}

options={Object.entries(monacoThemes).map(([themeId, themeName]) => ({

label: themeName,

value: themeId,

key: themeId,

}))}

value={theme}

styles={customStyles}

onChange={handleThemeChange}

/>

);

};

export default ThemeDropdown;

**STYLES**

value: "c",

},

{

id: 53,

name: "C++ (GCC 8.3.0)",

label: "C++ (GCC 8.3.0)",

value: "cpp",

},

{

id: 50,

name: "C (GCC 9.2.0)",

label: "C (GCC 9.2.0)",

value: "c",

},

{

id: 54,

name: "C++ (GCC 9.2.0)",

label: "C++ (GCC 9.2.0)",

value: "cpp",

},

{

id: 86,

name: "Clojure (1.10.1)",

label: "Clojure (1.10.1)",

value: "clojure",

},

{

id: 51,

name: "C# (Mono 6.6.0.161)",

label: "C# (Mono 6.6.0.161)",

value: "csharp",

},

{

id: 77,

name: "COBOL (GnuCOBOL 2.2)",

label: "COBOL (GnuCOBOL 2.2)",

value: "cobol",

},

{

id: 55,

name: "Common Lisp (SBCL 2.0.0)",

label: "Common Lisp (SBCL 2.0.0)",

value: "lisp",

},

{

id: 56,

name: "D (DMD 2.089.1)",

label: "D (DMD 2.089.1)",

value: "d",

},

{

id: 57,

name: "Elixir (1.9.4)",

label: "Elixir (1.9.4)",

value: "elixir",

},

**src/constants/languageOptions.js**

export const languageOptions = [

{

id: 63,

name: "JavaScript (Node.js 12.14.0)",

label: "JavaScript (Node.js 12.14.0)",

value: "javascript",

},

{

id: 45,

name: "Assembly (NASM 2.14.02)",

label: "Assembly (NASM 2.14.02)",

value: "assembly",

},

{

id: 46,

name: "Bash (5.0.0)",

label: "Bash (5.0.0)",

value: "bash",

},

{

id: 47,

name: "Basic (FBC 1.07.1)",

label: "Basic (FBC 1.07.1)",

value: "basic",

},

{

id: 75,

name: "C (Clang 7.0.1)",

label: "C (Clang 7.0.1)",

value: "c",

},

{

id: 76,

name: "C++ (Clang 7.0.1)",

label: "C++ (Clang 7.0.1)",

value: "cpp",

},

{

id: 48,

name: "C (GCC 7.4.0)",

label: "C (GCC 7.4.0)",

{

id: 64,

name: "Lua (5.3.5)",

label: "Lua (5.3.5)",

value: "lua",

},

{

id: 79,

name: "Objective-C (Clang 7.0.1)",

label: "Objective-C (Clang 7.0.1)",

value: "objectivec",

},

{

id: 65,

name: "OCaml (4.09.0)",

label: "OCaml (4.09.0)",

value: "ocaml",

},

{

id: 66,

name: "Octave (5.1.0)",

label: "Octave (5.1.0)",

value: "octave",

},

{

id: 67,

name: "Pascal (FPC 3.0.4)",

label: "Pascal (FPC 3.0.4)",

value: "pascal",

},

{

id: 85,

name: "Perl (5.28.1)",

label: "Perl (5.28.1)",

value: "perl",

},

{

id: 68,

name: "PHP (7.4.1)",

label: "PHP (7.4.1)",

value: "php",

},

{

id: 43,

label: "Plain Text",

name: "Plain Text",

value: "text",

},

{

id: 69,

name: "Prolog (GNU Prolog 1.4.5)",

label: "Prolog (GNU Prolog 1.4.5)",

value: "prolog",

},

{

id: 70,

name: "Python (2.7.17)",

label: "Python (2.7.17)",

value: "python",

},

value: "c",

},

{

id: 52,

name: "C++ (GCC 7.4.0)",

label: "C++ (GCC 7.4.0)",

value: "cpp",

},

{

id: 49,

name: "C (GCC 8.3.0)",

label: "C (GCC 8.3.0)",

{

id: 58,

name: "Erlang (OTP 22.2)",

label: "Erlang (OTP 22.2)",

value: "erlang",

},

{

id: 44,

label: "Executable",

name: "Executable",

value: "exe",

},

{

id: 87,

name: "F# (.NET Core SDK 3.1.202)",

label: "F# (.NET Core SDK 3.1.202)",

value: "fsharp",

},

{

id: 59,

name: "Fortran (GFortran 9.2.0)",

label: "Fortran (GFortran 9.2.0)",

value: "fortran",

},

{

id: 60,

name: "Go (1.13.5)",

label: "Go (1.13.5)",

value: "go",

},

{

**src/constants/statuses.js**

export const statuses = [{

id: 1,

description: "In Queue",},

{

id: 2,description: "Processing",

},

{

id: 3, description: "Accepted",

},

{

id: 4,description: "Wrong Answer",

},

{

id: 5,description: "Time Limit Exceeded",

},

{

id: 6,description: "Compilation Error",

},

{

id: 7, description: "Runtime Error (SIGSEGV)",

},

{

id: 8,description: "Runtime Error (SIGXFSZ)",

},

{

id: 9, description: "Runtime Error (SIGFPE)",

},

{

id: 10,description: "Runtime Error (SIGABRT)",

},

{

id: 11,description: "Runtime Error (NZEC)",

},

{

id: 12,description: "Runtime Error (Other)",

},

{

id: 13,description: "Internal Error",

},

{

id: 14,description: "Exec Format Error",

},];

id: 88,

name: "Groovy (3.0.3)",

label: "Groovy (3.0.3)",

value: "groovy",

},

{

id: 61,

name: "Haskell (GHC 8.8.1)",

label: "Haskell (GHC 8.8.1)",

value: "haskell",

},

{

id: 62,

name: "Java (OpenJDK 13.0.1)",

label: "Java (OpenJDK 13.0.1)",

value: "java",

},

{

id: 78,

name: "Kotlin (1.3.70)",

label: "Kotlin (1.3.70)",

value: "kotlin",

},

{

{

id: 71,

name: "Python (3.8.1)",

label: "Python (3.8.1)",

value: "python",

},

{

id: 80,

name: "R (4.0.0)",

label: "R (4.0.0)",

value: "r",

},

{

id: 72,

name: "Ruby (2.7.0)",

label: "Ruby (2.7.0)",

value: "ruby",

},

{

id: 73,

name: "Rust (1.40.0)",

label: "Rust (1.40.0)",

value: "rust",

},

{

id: 81,

name: "Scala (2.13.2)",

label: "Scala (2.13.2)",

value: "scala",

},

{

id: 82,

name: "SQL (SQLite 3.27.2)",

label: "SQL (SQLite 3.27.2)",

value: "sql",

},

{

id: 83,

name: "Swift (5.2.3)",

label: "Swift (5.2.3)",

value: "swift",

},

{

id: 74,

name: "TypeScript (3.7.4)",

label: "TypeScript (3.7.4)",

value: "typescript",

},

{

id: 84,

name: "Visual Basic.Net (vbnc 0.0.0.5943)",

label: "Visual Basic.Net (vbnc 0.0.0.5943)",

value: "vbnet",

},

];

**HOOKS**

**src/hooks/useKeyPress.js**

import React, { useState } from "react";

const useKeyPress = function (targetKey) {

const [keyPressed, setKeyPressed] = useState(false);

function downHandler({ key }) {

if (key === targetKey) {

setKeyPressed(true);

}

}

const upHandler = ({ key }) => {

if (key === targetKey) {

setKeyPressed(false);

}

};

React.useEffect(() => {

document.addEventListener("keydown", downHandler);

document.addEventListener("keyup", upHandler);

return () => {

document.removeEventListener("keydown", downHandler);

document.removeEventListener("keyup", upHandler);

};

});

return keyPressed;

};

export default useKeyPress;

**src/lib/defineTheme.js**

import { loader } from "@monaco-editor/react";

const monacoThemes = {

active4d: "Active4D",

"all-hallows-eve": "All Hallows Eve",

amy: "Amy",

"birds-of-paradise": "Birds of Paradise",

blackboard: "Blackboard",

"brilliance-black": "Brilliance Black",

"brilliance-dull": "Brilliance Dull",

"chrome-devtools": "Chrome DevTools",

"clouds-midnight": "Clouds Midnight",

clouds: "Clouds",

cobalt: "Cobalt",

dawn: "Dawn",

dreamweaver: "Dreamweaver",

eiffel: "Eiffel",

"espresso-libre": "Espresso Libre",

github: "GitHub",

idle: "IDLE",

katzenmilch: "Katzenmilch",

"kuroir-theme": "Kuroir Theme",

lazy: "LAZY",

"magicwb--amiga-": "MagicWB (Amiga)",

"merbivore-soft": "Merbivore Soft",

merbivore: "Merbivore",

"monokai-bright": "Monokai Bright",

monokai: "Monokai",

"night-owl": "Night Owl",

"oceanic-next": "Oceanic Next",

"pastels-on-dark": "Pastels on Dark",

"slush-and-poppies": "Slush and Poppies",

"solarized-dark": "Solarized-dark",

"solarized-light": "Solarized-light",

spacecadet: "SpaceCadet",

sunburst: "Sunburst",

"textmate--mac-classic-": "Textmate (Mac Classic)",

"tomorrow-night-blue": "Tomorrow-Night-Blue",

"tomorrow-night-bright": "Tomorrow-Night-Bright",

"tomorrow-night-eighties": "Tomorrow-Night-Eighties",

"tomorrow-night": "Tomorrow-Night",

tomorrow: "Tomorrow",

twilight: "Twilight",

"upstream-sunburst": "Upstream Sunburst",

"vibrant-ink": "Vibrant Ink",

"xcode-default": "Xcode\_default",

zenburnesque: "Zenburnesque",

iplastic: "iPlastic",

idlefingers: "idleFingers",

krtheme: "krTheme",

monoindustrial: "monoindustrial",

};

const defineTheme = (theme) => {

return new Promise((res) => {

Promise.all([

loader.init(),

import(`monaco-themes/themes/${monacoThemes[theme]}.json`),

]).then(([monaco, themeData]) => {

monaco.editor.defineTheme(theme, themeData);

res();

});

});

};

export { defineTheme };

**src/utils/general.js**

export const classnames = (...args) => {

return args.join(" ");

};

BACKEND CODE

**crud-nodejs-mysql-master/src/controllers/customerController.js**

import { pool } from "../db.js";

import express from "express";

const app = express();

app.use(express.json());

export const renderCustomers = async (req, res) => {

const [rows] = await pool.query("SELECT \* FROM customer");

res.render("customers", { customers: rows });

};

export const createCustomers = async (req, res) => {

const newCustomer = req.body;

await pool.query("INSERT INTO customer set ?", [newCustomer]);

res.redirect("/");

};

export const editCustomer = async (req, res) => {

const { id } = req.params;

const [result] = await pool.query("SELECT \* FROM customer WHERE id = ?", [

id,

]);

res.render("customers\_edit", { customer: result[0] });

};

export const updateCustomer = async (req, res) => {

const { id } = req.params;

const newCustomer = req.body;

await pool.query("UPDATE customer set ? WHERE id = ?", [newCustomer, id]);

res.redirect("/");

};

export const deleteCustomer = async (req, res) => {

const { id } = req.params;

const result = await pool.query("DELETE FROM customer WHERE id = ?", [id]);

if (result.affectedRows === 1) {

res.json({ message: "Customer deleted" });

}

res.redirect("/");

};

export const search = async (req, res) => {

var {qtitle} = req.params;

const [rows] = await pool.query("SELECT \* FROM customer Where qtitle = ?",[qtitle]);

res.render("customers", { customers: rows });

};

**crud-nodejs-mysql-master/src/routes/customer.routes.js**

import { Router } from "express";

import express from "express";

const app = express();

app.use(express.json());

import {

createCustomers,

deleteCustomer,

editCustomer,

renderCustomers,

updateCustomer,

search,

} from "../controllers/customerController.js";

const router = Router();

router.get("/", renderCustomers);

router.post("/add", createCustomers);

router.get("/update/:id", editCustomer);

router.post("/update/:id", updateCustomer);

router.get("/delete/:id", deleteCustomer);

router.get("/search", search);

export default router;

**crud-nodejs-mysql-master/src/views/customers.ejs**

<%- include("partials/\_header") %>

<div class="container mt-5 ">

<div class="col">

<div class="col-md-5 ">

<div class="card-body ">

<form action="/add" method="POST">

<input

type="text"

name="qtitle"

placeholder="Title"

class="form-control mb-2"

autofocus

/>

<input

type="text"

name="qDesc"

placeholder="Description"

class="form-control mb-2"

/>

<input

type="text"

name="constraints"

placeholder="Constraints"

class="form-control mb-2"

/>

<h1 style="color: aqua">Example</h1>

<input

type="text"

name="input"

placeholder="Input"

class="form-control mb-2"

/>

<input

type="text"

name="output"

placeholder="Output"

class="form-control mb-2"

/>

<input

type="text"

name="exp"

placeholder="Explaination"

class="form-control mb-2"

/>

<button type="submit" class="btn btn-info">save Question</button>

<h1></h1>

</form>

</div>

<div class="col-md-7">

<table class="table table-dark table-bordered table-hover">

<thead>

<tr>

<td>n°</td>

<td>Title</td>

<td>Description</td>

<td>Constraints</td>

<td>Input</td>

<td>Output</td>

<td>Explaination</td>

<td>Actions</td>

</tr>

</thead>

<tbody>

<% if (customers) { %> <% for(var i = 0; i < customers.length; i++) {

%>

<tr>

<td><%= (i + 1) %></td>

<td><%= customers[i].qtitle %></td>

<td><%= customers[i].qDesc %></td>

<td><%= customers[i].constraints %></td>

<td><%= customers[i].input %></td>

<td><%= customers[i].output %></td>

<td><%= customers[i].exp %></td>

<td class="d-flex gap-2">

<a href="/update/<%= customers[i].id %>" class="btn btn-info">

Edit

</a>

<a href="/delete/<%= customers[i].id %>" class="btn btn-danger">

Delete

</a>

</td>

</tr>

<% } %> <% } %>

</tbody>

</table>

</div>

</div>

</div>

</div>

<%- include("partials/\_footer") %>

**crud-nodejs-mysql-master/src/views/customers\_edit.ejs**

<%- include("partials/\_header") %>

<div class="container mt-5">

<div class="row">

<div class="col-md-5 offset-md-3 card bg-dark sk">

<div class="card-body">

<form action="/update/<%= customer.id %>" method="POST">

<input

type="text"

name="qtitle"

value="<%= customer.qtitle %>"

placeholder="qtitle"

class="form-control mb-3"

autofocus

/>

<input

type="text"

name="qDesc"

value="<%= customer.qDesc%>"

placeholder="Description"

class="form-control mb-3"

/>

<input

type="text"

name="constraints"

value="<%= customer.constraints%>"

placeholder="Constraints"

class="form-control mb-3"

/>

<input

type="text"

name="input"

value="<%= customer.input%>"

placeholder="Input"

class="form-control mb-3"

/>

<input

type="text"

name="output"

value="<%= customer.output%>"

placeholder="Output"

class="form-control mb-3"

/>

<input

type="text"

name="exp"

value="<%= customer.exp%>"

placeholder="Explaination"

class="form-control mb-3"

/>

<button type="submit" class="btn btn-primary">save Question</button>

</form>

</div>

</div>

</div>

</div>

<%- include("partials/\_footer") %>

**crud-nodejs-mysql-master/src/app.js**

import express from "express";

import path from "path";

import morgan from "morgan";

import customerRoutes from "./routes/customer.routes.js";

import { fileURLToPath } from "url";

const app = express();

const \_\_dirname = path.dirname(fileURLToPath(import.meta.url));

// settings

app.set("port", process.env.PORT || 3000);

app.set("views", path.join(\_\_dirname, "views"));

app.set("view engine", "ejs");

// middlewares

app.use(morgan("dev"));

app.use(express.urlencoded({ extended: false }));

// routes

app.use(customerRoutes);

// static files

app.use(express.static(path.join(\_\_dirname, "public")));

// starting the server

export default app;

**crud-nodejs-mysql-master/src/db.js**

import { createPool } from "mysql2/promise";

export const pool = createPool({

host: "localhost",

user: "root",

password: "Root",

port: 3306,

database: "customersdb",

});

**crud-nodejs-mysql-master/src/index.js**

import app from "./app.js";

import { port } from "./config.js";

app.listen(port);

console.log("server running on " + "https//localhost:" +port);

**ProblemSection/Backend/index.js**

// http://localhost:8800/problems

import express from "express";

import mysql from "mysql";

const app = express();

const db = mysql.createConnection({

host: 'localhost',

user: 'root',

password: '2003',

database: 'customersdb',

});

app.get("/", (req, res)=>{

res.json("hello this is the backend")

})

app.get("/problems", (req,res)=>{

const q = "SELECT \* FROM customer"

db.query(q,(err,data)=>{

if(err) return res.json(err)

return res.json(data);

});

})

app.listen(8800, ()=>{

console.log("Connected to backend!")

})

**ProblemSection/frontend/src/App.js**

import React, { useEffect, useState } from 'react';

import './App.css';

function App() {

const [questions, setQuestions] = useState([]);

useEffect(()=>{

const getQuestions = async ()=>{

const res = await fetch('http://localhost:8800/problems');

const getData = await res.json();

setQuestions(getData);

console.log(getData);

}

getQuestions();

},[]);

const [searchTerm, setSearchTerm] = useState('');

const handleSearchChange = (e) => {

setSearchTerm(e.target.value);

};

const filteredQuestions = questions.filter((question) => {

const op = question.qtitle;

return op.toLowerCase().includes(searchTerm.toLowerCase());

});

const renderQuestions = () => {

return filteredQuestions.map((question, index) => {

return (

<div key={index} className="question-container">

<span className="question-text">{question.qtitle}</span>

<button className="solve-button">Solve Problem</button>

</div>

);

});

};

return (

<div className="App">

<div className="search-container">

<h1 className='title'>Problems</h1>

<input

type="text"

placeholder="Search..."

value={searchTerm}

onChange={handleSearchChange}

/>

<button className="search-button">Search</button>

</div>

<div className="questions-container">{renderQuestions()}</div>

</div>

);

}

export default App;

**src/components/navbar/Navbar.jsx**

import React, { useState } from 'react';

import { BrowserRouter as Router, Route, Link } from 'react-router-dom'

import { RiMenu3Line, RiCloseLine } from 'react-icons/ri';

import mainlogo from '../../assets/coding-language.png';

import pro from '../../assets/image/web.png'

import king from '../../assets/image/king.png'

import diamond from '../../assets/image/diamond.png'

import home from '../../assets/image/home.png'

import book from '../../assets/image/book.png'

import admin from '../../assets/manager.png';

import './navbar.css';

import Login from '../../containers/login/login';

const Navbar = () => {

const [toggleMenu, setToggleMenu] = useState(false);

return (

<div>

<div className="bit\_navbar">

<div className="bit\_navbar-links">

<div className="bit\_navbar-links\_logo">

<img alt="logo" src={mainlogo} style={{ height: 60, width: 60, marginLeft: 8 }} />

{/\* <img src={logo} /> \*/}

</div>

<div className="bit\_navbar-links\_container">

{/\* <p><a href="/home"><img alt="logo" src={home} style={{ height: 20, width: 20, marginRight: 4 }} /> Home</a></p> \*/}

{/\* <p><a href="/problem"><img alt="logo" src={pro} style={{ height: 18, width: 18, marginRight: 4 }} />Problems</a></p> \*/}

{/\* <p><a href="/leadboard"><img alt="logo" src={king} style={{ height: 18, width: 18, marginRight: 4 }} />LeaderBoard</a></p> \*/}

{/\* <p><a href="/shop"><img alt="logo" src={diamond} style={{ height: 18, width: 18, marginRight: 4 }} />Shop</a></p> \*/}

{/\* <p><a href="/library"><img alt="logo" src={book} style={{ height: 20, width: 20, marginRight: 4 }} />Library</a></p> \*/}

<Link to={'/home'} ><p><img alt="logo" src={home} style={{ height: 20, width: 20, marginRight: 4 }} /> Home</p></Link>

<Link to={"/problems"}><p><img alt="logo" src={pro} style={{ height: 18, width: 18, marginRight: 4 }} />Problems</p></Link>

<Link to="/leadboard"><p><img alt="logo" src={king} style={{ height: 18, width: 18, marginRight: 4 }} />LeaderBoard</p></Link>

<Link to={'/shop'}><p><img alt="logo" src={diamond} style={{ height: 18, width: 18, marginRight: 4 }} />Shop</p></Link>

<Link to={'/library'}><p><img alt="logo" src={book} style={{ height: 20, width: 20, marginRight: 4 }} />Library</p></Link>

</div>

</div>

<div className="bit\_navbar-sign">

{/\* <p>Sign-in</p> \*/}

{/\* <p><a href="/signin">Sign in</a></p> \*/}

{/\* <button type="button">Sign up</button> \*/}

{/\* <a href="/signup"><button>SignUp</button></a> \*/}

<Link to={'/admin'}><p>Admin</p></Link>

<Link to={'/signup'}><button>SignUp</button></Link>

</div>

<div className="bit\_navbar-menu">

{toggleMenu

? <RiCloseLine color="#fff" size={27} onClick={() => setToggleMenu(false)} />

: <RiMenu3Line color="#fff" size={27} onClick={() => setToggleMenu(true)} />}

{toggleMenu && (

<div className="bit\_navbar-menu\_container scale-up-center">

<div className="bit\_navbar-menu\_conatiner-links">

{/\* <p><a href="/home"><img alt="logo" src={home} style={{ height: 20, width: 20, marginRight: 4 }} /> Home</a></p>

<p><a href="/problem"><img alt="logo" src={pro} style={{ height: 18, width: 18, marginRight: 4 }} />Problems</a></p>

<p><a href="/leadboard"><img alt="logo" src={king} style={{ height: 18, width: 18, marginRight: 4 }} />LeaderBoard</a></p>

<p><a href="/shop"><img alt="logo" src={diamond} style={{ height: 18, width: 18, marginRight: 4 }} />Shop</a></p>

<p><a href="/library"><img alt="logo" src={book} style={{ height: 20, width: 20, marginRight: 4 }} />Library</a></p> \*/}

<Link to={'/home'} ><p><img alt="logo" src={home} style={{ height: 20, width: 20, marginRight: 4 }} /> Home</p></Link>

<Link to={"/problems"}><p><img alt="logo" src={pro} style={{ height: 18, width: 18, marginRight: 4 }} />Problems</p></Link>

<Link to="/leadboard"><p><img alt="logo" src={king} style={{ height: 18, width: 18, marginRight: 4 }} />LeaderBoard</p></Link>

<Link to={'/shop'}><p><img alt="logo" src={diamond} style={{ height: 18, width: 18, marginRight: 4 }} />Shop</p></Link>

<Link to={'/library'}><p><img alt="logo" src={book} style={{ height: 20, width: 20, marginRight: 4 }} />Library</p></Link>

</div>

<div className="bit\_navbar-menu\_container-links-sign">

{/\* <p>Sign in</p>

<button type="button">Sign up</button> \*/}

<Link to={'/admin'}><p className='admin'>Admin</p></Link>

<Link to={'/signup'}><button>SignUp</button></Link>

</div>

</div>

)}

</div>

</div>

</div>

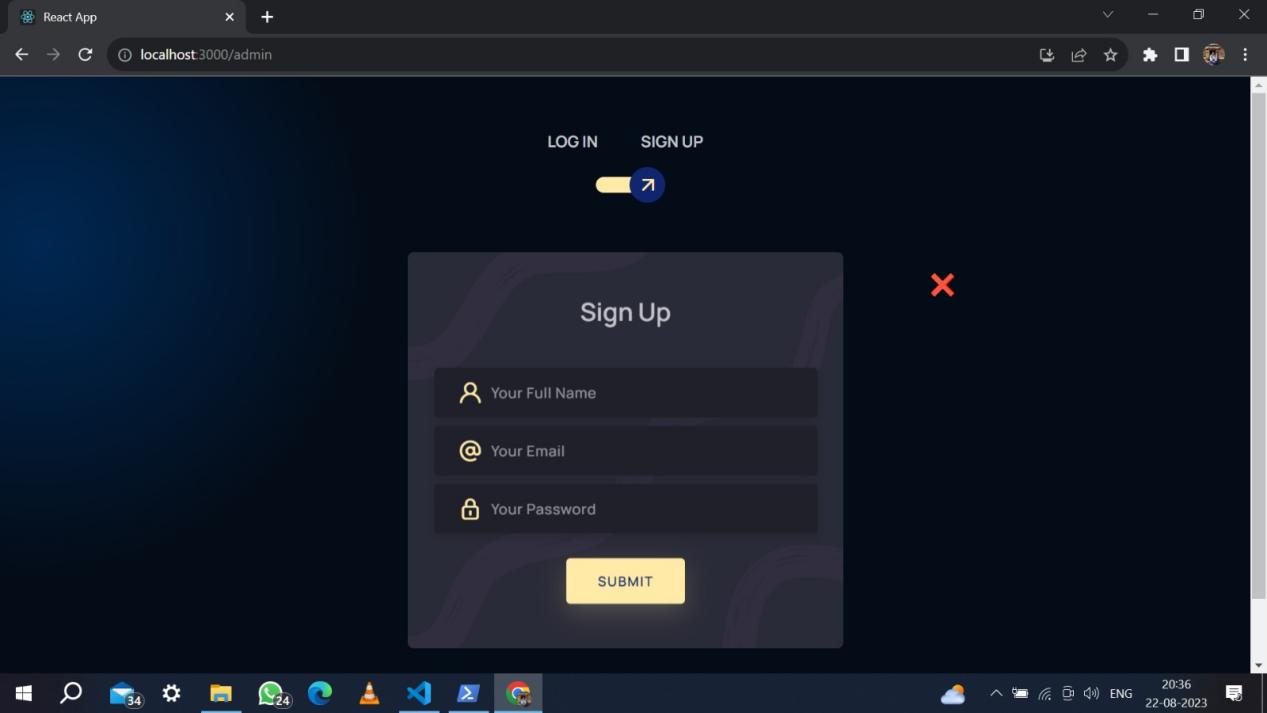
)

}

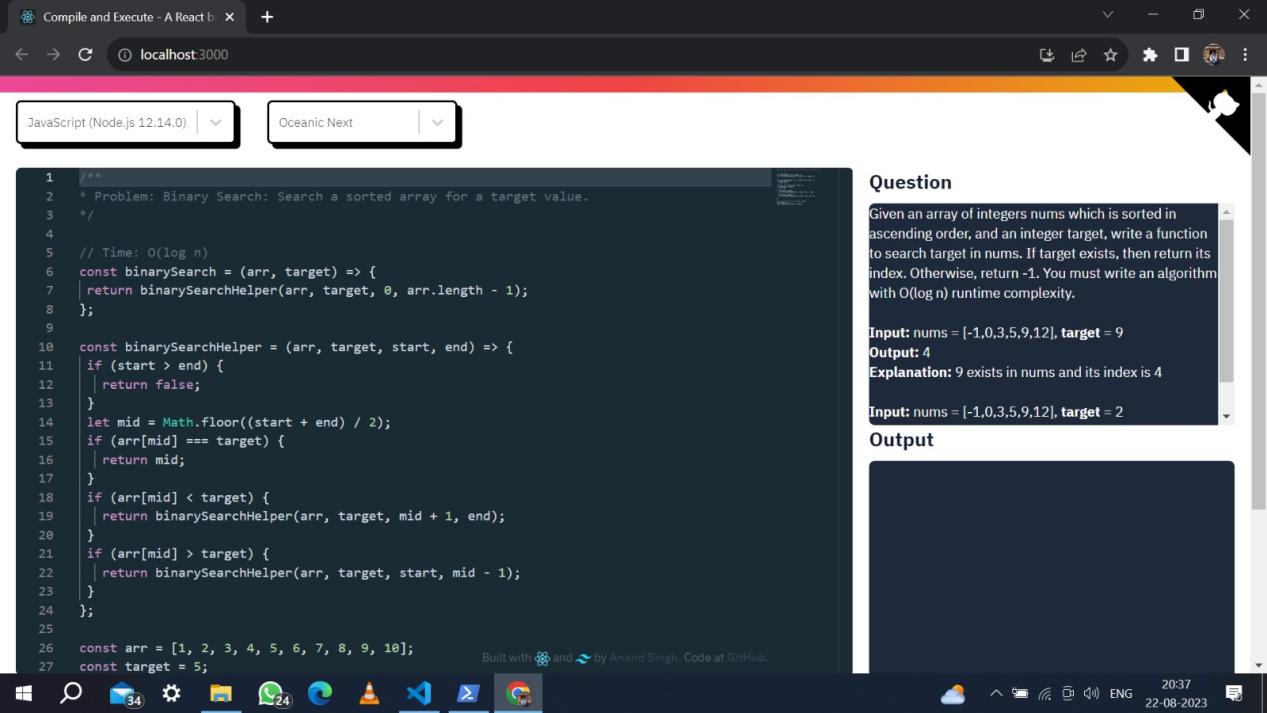
export default Navbar;

**RESULT**

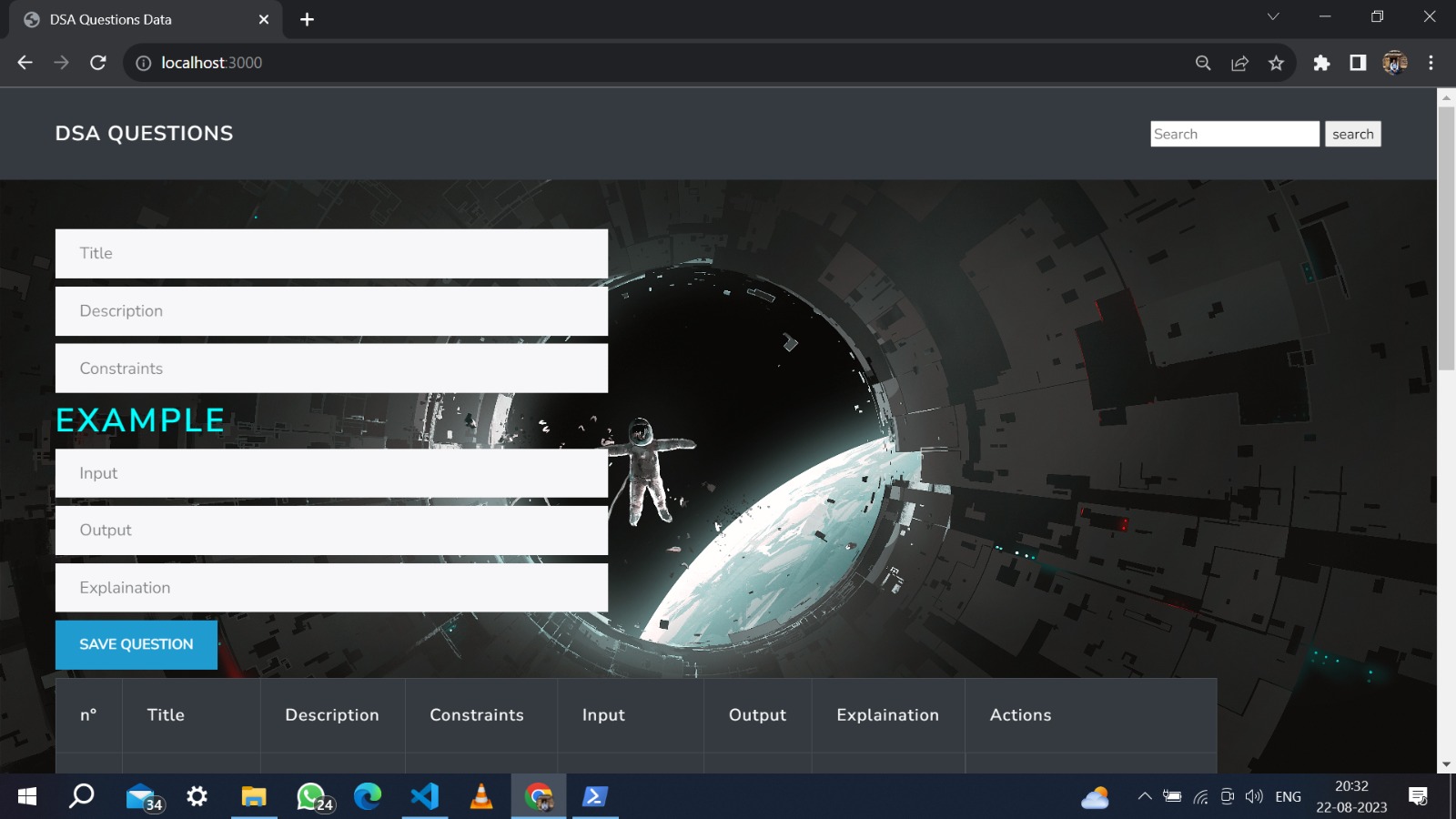
**Login & Registration Page**

****

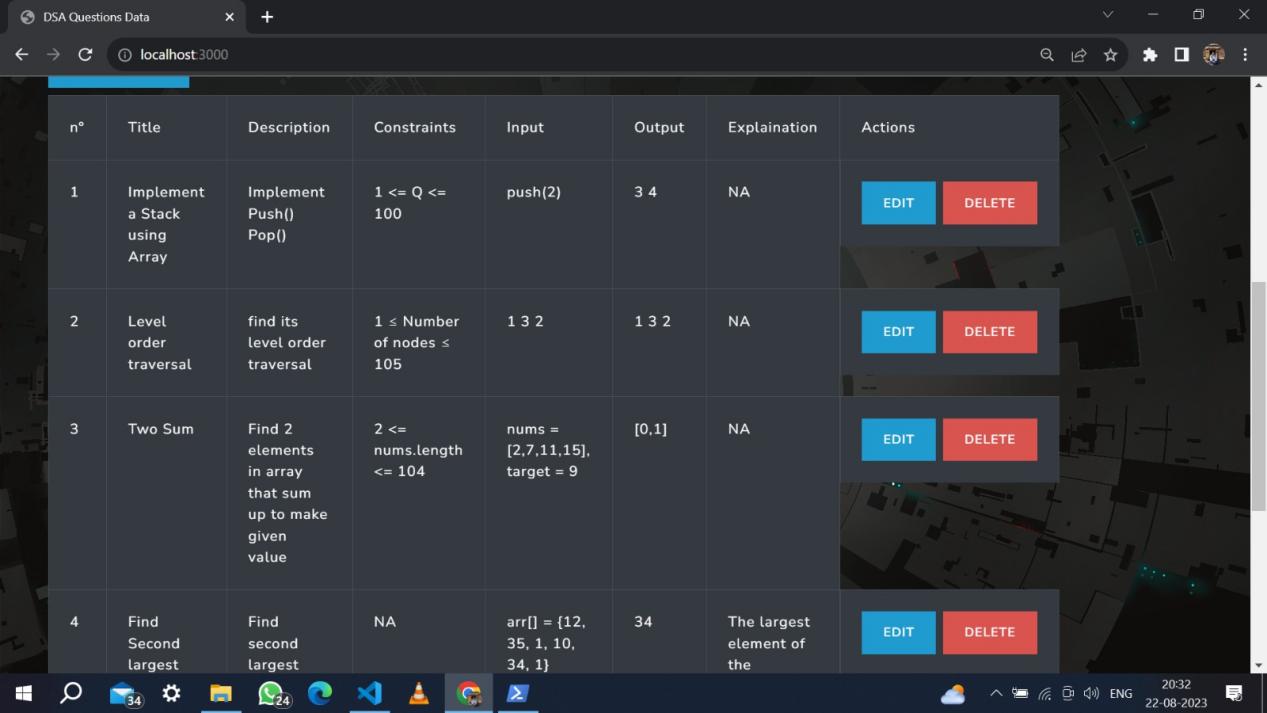
**Editor**

****

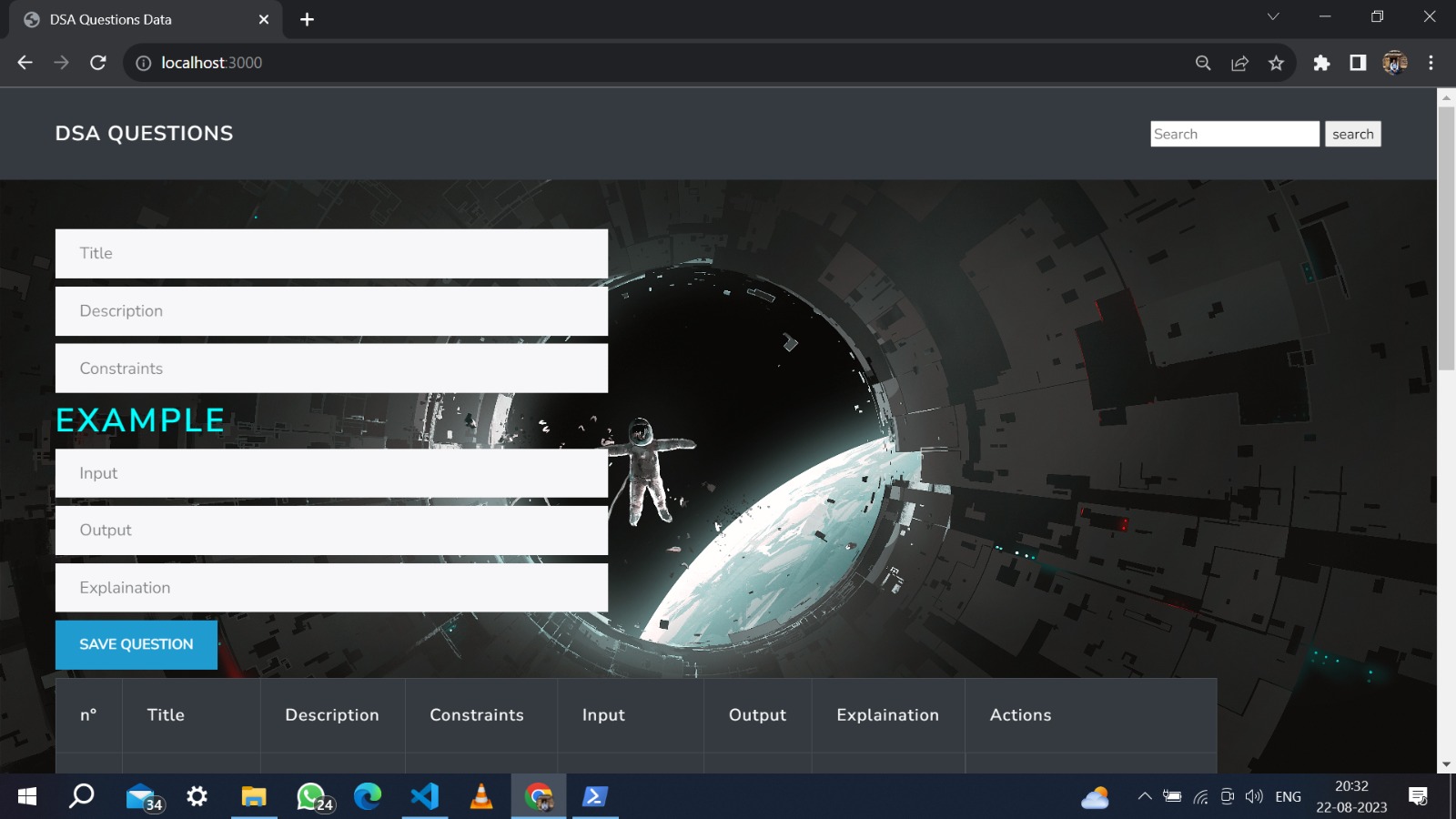
**Admin Page**

****

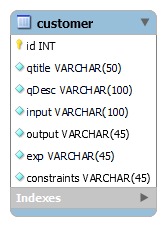
**Question List**

****

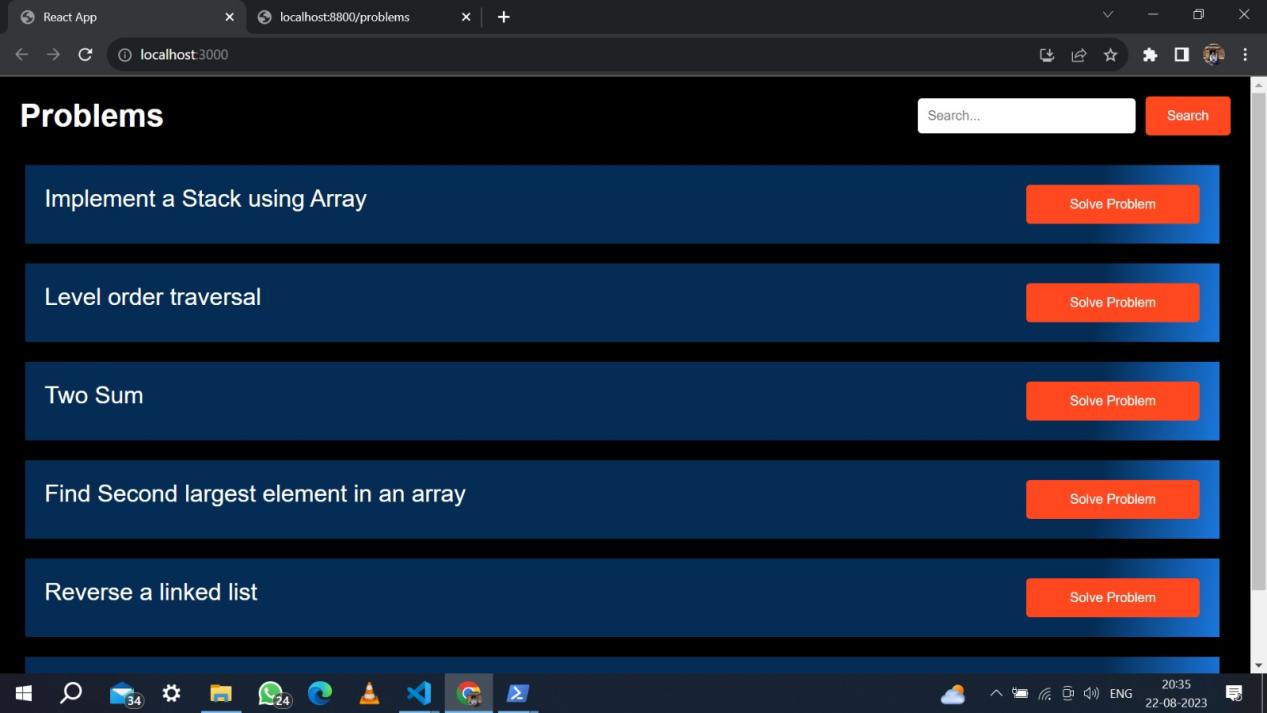
**Input List**

****

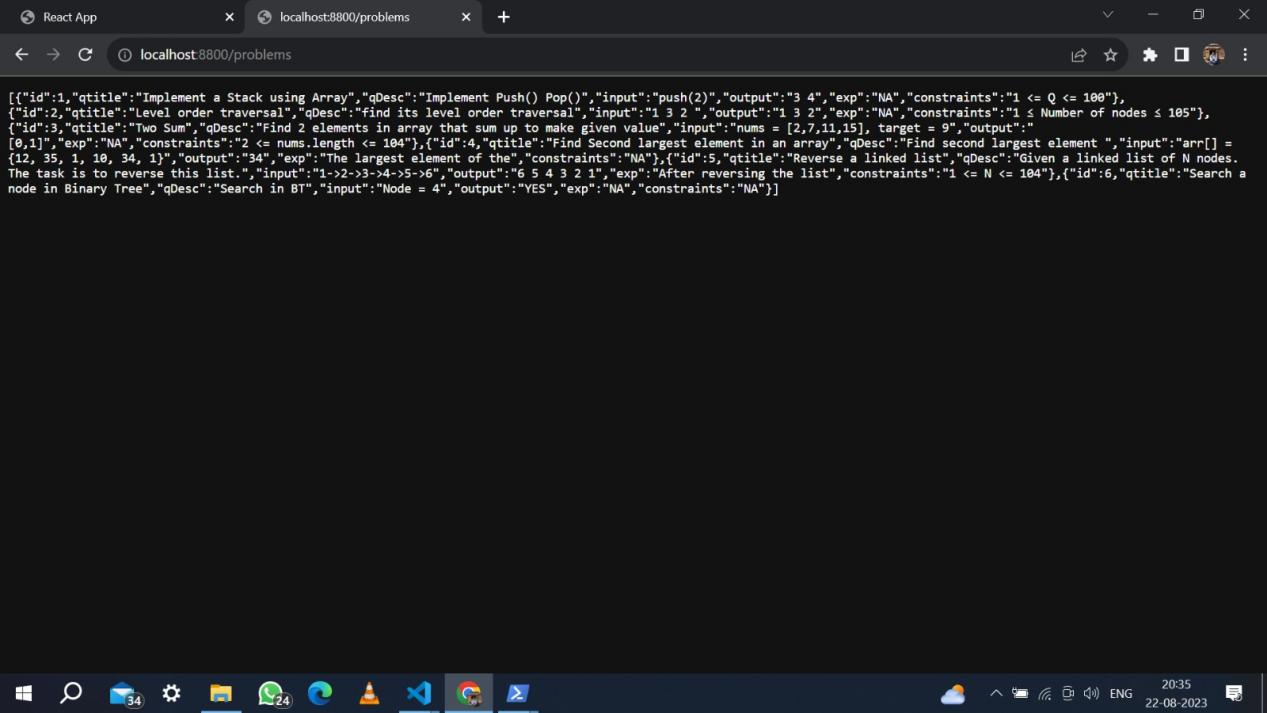
**Database**

****

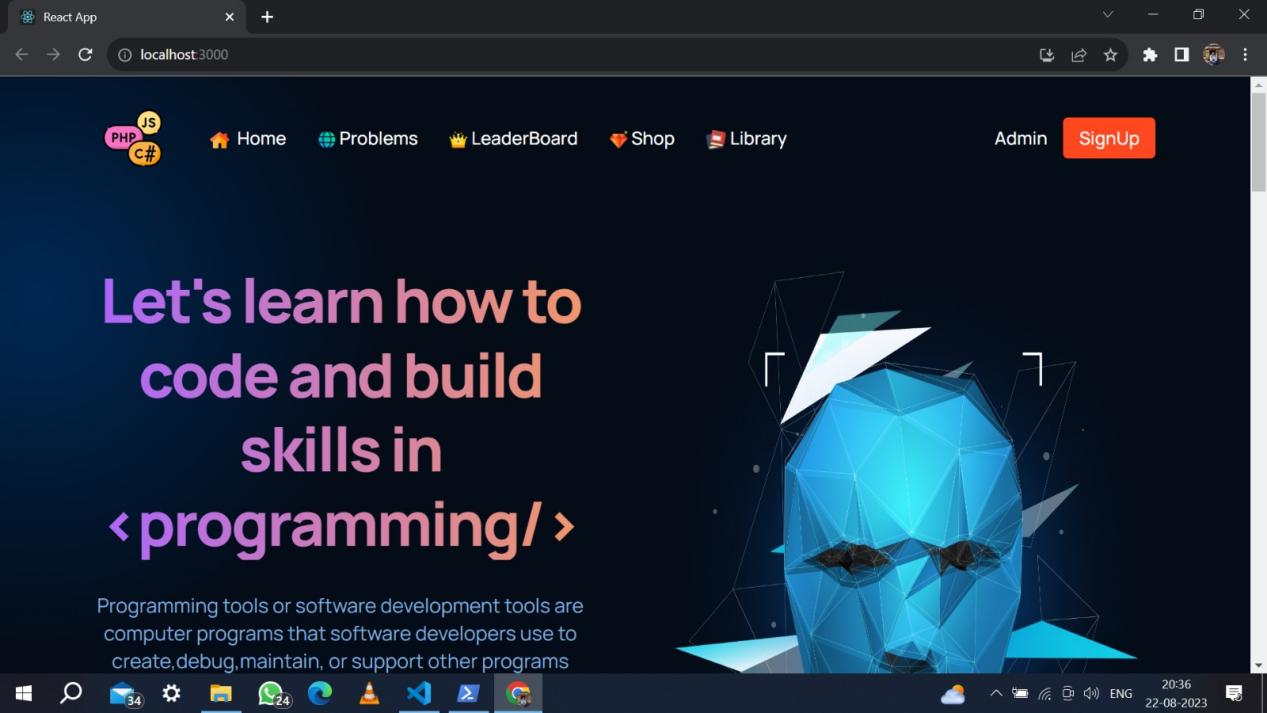
**Problems**

****

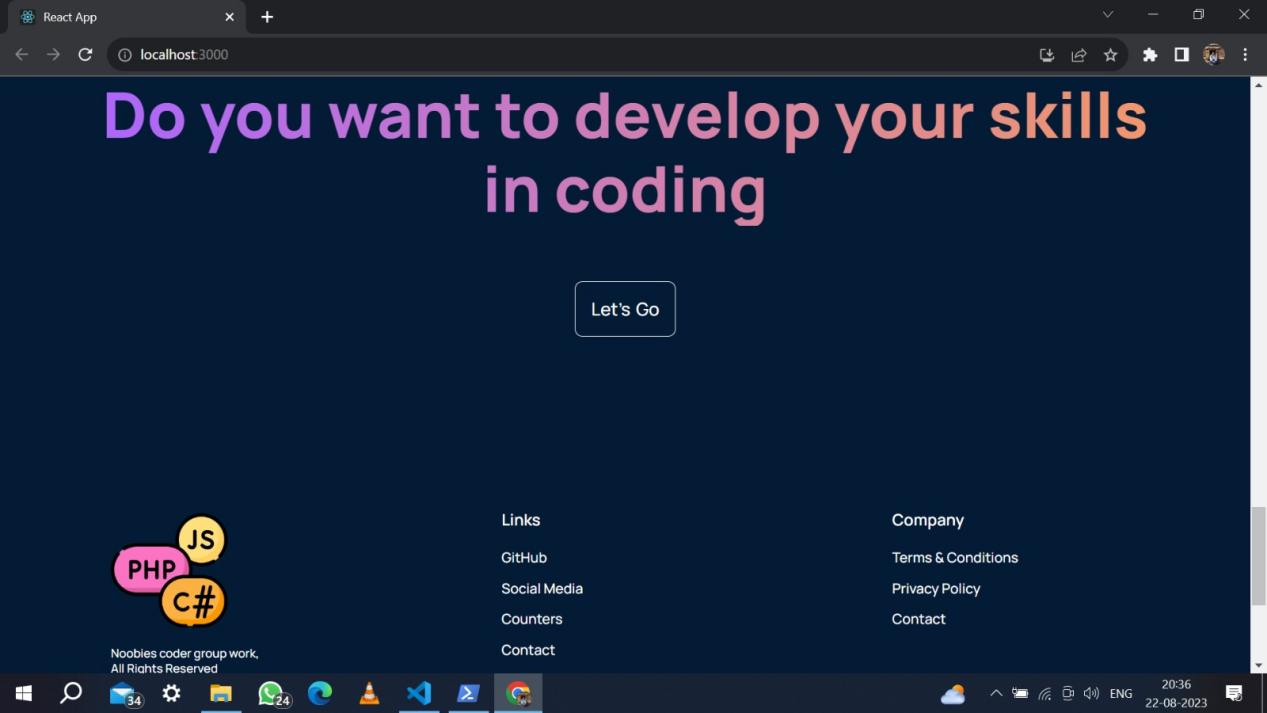
**API**

****

**Landing Page**

****

**Footer**

****