

11.12.2022

MEDICARE CAPSTONE PROJECT SOURCE CODE

DONE BY: Salabha S

cards.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <link rel="stylesheet"
href="../../styles/cards.css">
  <!--<link
href="https://fonts.googleapis.com/css2?family=Material+
Icons"
rel="stylesheet">-->
  <title>Cards</title>
</head>
<body>

  <div class="cards">

    <div class="card">
      <div class="card-image-container">
        
      </div>

      <div class="card-info-container">
        <h1 class="card-info-heading">Dr. Supdip
Lahiri</h1>
        <br/>
        <p class="card-info-para">
```

```
        Dermatologist
    </p>
    <br/>

    <div class="card-stats-container">
        <div class="card-stats">
            <h1 class="stats-heading">
                Stars
            </h1>
            <p class="stats-para">
                4.1
            </p>
        </div>
        <div class="card-stats">
            <h1 class="stats-heading">
                Experience
            </h1>
            <p class="stats-para">
                5 years
            </p>
        </div>
    </div>
    <br/>
    <button class="btn">Set Appointment</button>
</div>

</div>
<div class="card">
    <div class="card-image-container">
        
```

```
</div>
<br>
<br>
<div class="card-info-container">
  <h1 class="card-info-heading">Dr. Srimati
Laha</h1>
  <br/>
  <p class="card-info-para">
    Optician</p>
  <br/>

  <div class="card-stats-container">
    <div class="card-stats">
      <h1 class="stats-heading">
        Stars
      </h1>
      <p class="stats-para">
        4.3
      </p>
    </div>
    <div class="card-stats">
      <h1 class="stats-heading">
        Reviews
      </h1>
      <p class="stats-para">
        101
      </p>
    </div>
    <div class="card-stats">
      <h1 class="stats-heading">
        Experience
```

```
        </h1>
        <p class="stats-para">
            7 years
        </p>
    </div>

    </div>
    <br/>
    <button class="btn">Set Appointment</button>
</div>

</div>
<div class="card">
    <div class="card-image-container">
        
    </div>
    <br><br>
    <div class="card-info-container">
        <h1 class="card-info-heading">Dr. Sukdeep
Singh</h1>
        <br/>
        <p class="card-info-para">
            Neurologist</p>
        <br/>

        <div class="card-stats-container">
            <div class="card-stats">
                <h1 class="stats-heading">
                    Stars
                </h1>
```

```
        <p class="stats-para">
            4.5
        </p>
    </div>
    <div class="card-stats">
        <h1 class="stats-heading">
            Reviews
        </h1>
        <p class="stats-para">
            200
        </p>
    </div>
    <div class="card-stats">
        <h1 class="stats-heading">
            Experience
        </h1>
        <p class="stats-para">
            5 years
        </p>
    </div>

    </div>
    <br/>
    <button class="btn">Set Appointment</button>
</div>

</div>
<div class="card">
    <div class="card-image-container">
        
```

```
</div>
<br><br><br><br>
<div class="card-info-container">
  <h1 class="card-info-heading">Dr. Jahanaara
Hussain</h1>
  <br/>
  <p class="card-info-para">
    Endocrinologist</p>
  <br/>
  <div class="card-stats-container">
    <div class="card-stats">
      <h1 class="stats-heading">
        Stars
      </h1>
      <p class="stats-para">
        4.9
      </p>
    </div>
    <div class="card-stats">
      <h1 class="stats-heading">
        Reviews
      </h1>
      <p class="stats-para">
        309
      </p>
    </div>
    <div class="card-stats">
      <h1 class="stats-heading">
        Experience
      </h1>
      <p class="stats-para">
```



```
<div class="card-stats">
  <h1 class="stats-heading">
    Reviews
  </h1>
  <p class="stats-para">
    342
  </p>
</div>
```

```
<div class="card-stats">
  <h1 class="stats-heading">
    Experience
  </h1>
  <p class="stats-para">
    6 years
  </p>
</div>
```

```
</div>
```

```
<br/>
```

```
<button class="btn">Set Appointment</button>
```

```
</div>
```

```
</div>
```

```
<div class="card">
```

```
  <div class="card-image-container">
```

```
    
```

```
  </div>
```

```
<br>
```

```
<div class="card-info-container">
```

```
<h1 class="card-info-heading">Dr. Sukumari
Barman</h1>
<br/>
<p class="card-info-para">
  Gynaecologist</p>
<br/>

<div class="card-stats-container">
  <div class="card-stats">
    <h1 class="stats-heading">
      Stars
    </h1>
    <p class="stats-para">
      4.5
    </p>
  </div>
  <div class="card-stats">
    <h1 class="stats-heading">
      Reviews
    </h1>
    <p class="stats-para">
      100
    </p>
  </div>
  <div class="card-stats">
    <h1 class="stats-heading">
      Experience
    </h1>
    <p class="stats-para">
      10 years
    </p>
```

```
        </div>

        </div>
        <br/>
        <button class="btn">Set Appointment</button>
    </div>

</div>
</div>

</body>
</html>
```

Chatbot.js

```
import React, { useState } from "react";
import "../styles/Chatbot.css";
import ChatBot from "react-simple-chatbot";
import { ThemeProvider } from "styled-components";
import { useHistory } from "react-router-dom";
import funfacts from "./Funfacts";

const theme = {
  background: "#F4F5F6",
  fontFamily: "sans-serif",
  headerBgColor: "#15bea9",
  headerFontColor: "#fff",
  headerFontSize: "16px",
  botBubbleColor: "#65b1ff",
  botFontColor: "#fff",
  userBubbleColor: "#59dd64",
```

```
    userFontColor: "#fff",
};

const steps = [
  {
    id: "1",
    message: "Hey! I am Medi-bot, how are you feeling today",
    trigger: "expressions",
  },
  {
    id: "expressions",
    options: [
      { value: "veryHappy", label: "😊", trigger: "veryHappy" },
      { value: "happy", label: "🙂", trigger: "happy" },
      { value: "normal", label: "😐", trigger: "normal" },
      { value: "sad", label: "☹️", trigger: "sad" },
      { value: "angry", label: "😡", trigger: "angry" },
    ],
  },
  {
    id: "veryHappy",
    message: "Great Me too.",
    trigger: "help",
  },
  {
    id: "happy",
    message:
```

```
    "Hmm, Looks Like You are in good mood. Well I can  
make it great.",  
    trigger: "help",  
  },  
  {  
    id: "normal",  
    message:  
      "Ooh, Are You Confused Sir ? Don't Worry I am here  
to Assist You.",  
    trigger: "help",  
  },  
  {  
    id: "sad",  
    message:  
      "I am sorry to here that. Well i would suggest to  
read our random Fun-fact or Visit our Yoga Services or  
Read the latest feeds. You will feel better.",  
    trigger: "help",  
  },  
  {  
    id: "angry",  
    message:  
      "Oops, Did i do something wrong....But wait i am a  
bot i can't do that... Well i suggest you to relax your  
mind, play some games, watch some movies and eat a lot  
of healthy tasty food just like me..",  
    trigger: "help",  
  },  
  {  
    id: "help",  
    message: "How can I help you?",
```

```
    trigger: "options",
  },
  {
    id: "options",
    options: [
      { value: "services", label: "Services", trigger:
"services" },
      {
        value: "calculateBMI",
        label: "Calculate BMI",
        trigger: "calculateBMI",
      },
      {
        value: "funFact",
        label: "Give a random funfact",
        trigger: "funFact",
      },
      { value: "exit", label: "Exit", trigger: "end" },
    ],
  },
  {
    id: "services",
    message: "select one of these services",
    trigger: "selectServices",
  },
  {
    id: "selectServices",
    options: [
      { value: "health", label: "Smart BMI", trigger:
"selectedService" },
```

```
    { value: "Disease Detection", label: "Disease
Detection", trigger: "selectedService" },
    { value: "goBack", label: "go back", trigger:
"options" },
  ],
},
{
  id: "selectedService",
  message: "opened {previousValue}",
  trigger: "end",
},
{
  id: "moreHelp",
  message: "do you need more help?",
  trigger: "moreHelp-yes",
},
{
  id: "moreHelp-yes",
  options: [
    { value: "yes", label: "Yes", trigger: "help" },
    { value: "no", label: "No", trigger: "end" },
  ],
},
{
  id: "calculateBMI",
  message: "Let's calculate your BMI",
  trigger: "BMIHeightMessage",
},
{
  id: "BMIHeightMessage",
  message: "Enter your height(in cm.)",
```

```
    trigger: "BMIHeight",
  },
  {
    id: "BMIHeight",
    user: true,
    validator: (value) => {
      if (isNaN(value)) {
        return 'value should be a number';
      }
      return true;
    },
    trigger: "BMIWeightMessage",
  },
  {
    id: "BMIWeightMessage",
    message: "Enter your weight(in kg)",
    trigger: "BMIWeight",
  },
  {
    id: "BMIWeight",
    user: true,
    validator: (value) => {
      if (isNaN(value)) {
        return 'value should be a number';
      }
      return true;
    },
    trigger: "BMIResult",
  },
  {
    id: "BMIResult",
```



```
    component: <BMI />,
    asMessage: true,
    trigger: "BMIKnowMore",
  },
  {
    id: "BMIKnowMore",
    message: "Want to know more about BMI?",
    trigger: "BMIKnowMore-options",
  },
  {
    id: "BMIKnowMore-options",
    options: [
      { value: "health", label: "Yes", trigger:
"selectedService" },
      { value: "no", label: "No", trigger: "end" },
    ],
  },
  {
    id: "funFact",
    component: <FunFact />,
    asMessage: true,
    trigger: "moreHelp",
  },
  {
    id: "end",
    message: "Thank you, see you again!",
    end: true,
  },
];

function BMI({ steps }) {
```

```

return (
  <div>
    Your BMI is{" "}
    {(
      (steps.BMIWeight.value /
        (steps.BMIHeight.value *
steps.BMIHeight.value)) *
      10000
    ).toFixed(2)}
  </div>
);
}

function FunFact() {
  const randInt = Math.floor(Math.random() * 14);
  return <div>{funfacts[randInt]}</div>;
}

function Chatbot() {
  const history = useHistory();
  const handleEnd = ({ steps, values }) => {
    switch (values[values.length - 1]) {
      case "home":
        history.push("/");
        break;
      case "Disease Detection":
        history.push("/DiseaesPred");
        break;
      case "health":
        history.push("/health");
      case "Doctor":
    }
  }
}

```

```

        history.push("/Doctor");
        break;
      default:
        break;
    }
  };

  return (
    <div className="chatbot">
      <ThemeProvider theme={theme}>
        <ChatBot
          steps={steps}
          handleEnd={handleEnd}
          headerTitle="Chatbot"
          speechSynthesis={{ enable: true, lang: 'en' }}

```

```

botAvatar={"https://e7.pngegg.com/pngimages/811/700/png-clipart-chatbot-internet-bot-business-natural-language-processing-facebook-messenger-business-people-logo-thumbnail.png"}

```

```

      avatarStyle={{ borderRadius: "100%" }}

```

```

      floating={true}

```

```

      floatingIcon={

```

```

        <img

```

```

src={"https://e7.pngegg.com/pngimages/811/700/png-clipart-chatbot-internet-bot-business-natural-language-processing-facebook-messenger-business-people-logo-thumbnail.png"}

```

```

      style={{ width: "70%" }}

```

```

      alt="chatBot icon"

```

```

        />
      }
      floatingStyle={{
        backgroundColor: "#ffffff",
        width: "60px",
        boxShadow: "2px 2px 20px -8px #111",
      }}
    />
  </ThemeProvider>
</div>
);
}

export default Chatbot;

```

HeartDiseaseForm.js

```

import React,{useState} from "react";
import './static/main.css';
import './static/HeartDiseaseForm.css';
import { useHistory } from "react-router-dom";
import heart from './static/heart.jpg';
import '../.../styles/w3.css';

export default function HeartDiseaseForm(props) {
  const history = useHistory();
  const [cp,setCp] = useState(0);
  const [trestbps,setTrestbps] = useState(0);
  const [chol,setChol] = useState(0);
  const [fbs,setFbs] = useState(0);
  const [restecg,setRestecg] = useState(0);

```

```

const [thalach,setThalach] = useState(0);
const [exang,setExang] = useState(0);

const handleCp = e =>{setCp(e.target.value)};
const handleTrestbps = e
=>{setTrestbps(e.target.value)};
const handleChol = e =>{setChol(e.target.value)};
const handleFbs = e =>{setFbs(e.target.value)};
const handleRestecg = e
=>{setRestecg(e.target.value)};
const handleThalach = e
=>{setThalach(e.target.value)};
const handleExang = e =>{setExang(e.target.value)};
const requestOptions = {
  method : "POST",
  headers : { "Authorization" : "authtoken",
              "Content-Type" : "application/json"
},
  body :
JSON.stringify([cp,trestbps,chol,fbs,restecg,thalach,exa
ng]),
};
const handlePredict = () =>{
  fetch('https://medi-care-
backend.herokuapp.com/heart-pred/',requestOptions)
    .then((response) => response.json())
    .then((data) => createReport(data));
};
const createReport = (vals) =>{
  const requestOptions = {
    method : "POST",

```

```

        headers : { "Authorization" : "authtoken",
                    "Content-Type" :
"application/json" },
        body : JSON.stringify({
            pred:vals.prediction,
            vals:vals.report,
            defs:props.defaultval,
            names:props.name
        }),
    };
    fetch('https://medi-care-backend.herokuapp.com/map-report/',requestOptions)
        .then((response) => response.json())
        .then((data) =>
history.push('/report/'+heart',{state:data}));
    };
    return (
    <div className="contact1">
        <br></br>
        <br></br>
        <br></br>
        <br></br>
        <div className="container-contact1">
            <div className="contact1-pic js-tilt" data-
tilt>
                <img src={heart} />
            </div>
            <form className="contact1-form">
                <span className="w3-xlarge contact1-
form-title">

```

Take Heart Disease Test

```

        </span>
        <div className="wrap-input1">
            <select className="form-control w3-
large select input1" id="cp" name="cp"
required="required" onChange={handleCp}
placeholder="Chest Pain Type">
                <option hidden>Chest Pain
Type</option>
                <option value="1">Typical
Angina</option>
                <option value="2">Atypical
Angina</option>
                <option value="3">Non-Anginal
Pain</option>
                <option
value="4">Asymptomatic</option>
            </select>
            <span className="shadow-
input1"></span>
        </div>
        <div className="wrap-input1" >
            <input className="form-control w3-
large input1" type="number" name="trestbps"
onChange={handleTrestbps} required="required"
placeholder="Resting Blood Sugar" />
            <span className="shadow-
input1"></span>
        </div>
        <div className="wrap-input1">
            <input className="form-control w3-
large input1" type="number" name="chol" step="any"

```

```

onChange={handleChol} required="required"
placeholder="Serum Cholestrol in mg/dl" />
      <span className="shadow-
input1"></span>
    </div>
    <div className="wrap-input1">
      <select className="form-control w3-
large select input1" id="fbs" name="fbs"
onChange={handleFbs} required="required"
placeholder="Fasting Blood Sugar">
        <option hidden>Fasting Blood
Sugar</option>
        <option value="0">Fasting Blood
Sugar < 120 mg/dl</option>
        <option value="1">Fasting Blood
Sugar > 120 mg/dl</option>
      </select>
      <span className="shadow-
input1"></span>
    </div>
    <div className="wrap-input1">
      <select className="form-control w3-
large select input1" id="restecg" name="restecg"
onChange={handleRestecg} required="required"
placeholder="Resting Electro-Cardiographic Result">
        <option hidden>Resting Electro-
Cardiographic Result</option>
        <option
value="0">Normal</option>
        <option value="1">Having ST-T
Wave Abnormality</option>

```



```

        <option value="2">Showing
probable or definite left Ventricular
Hypertrophy</option>
    </select>
    <span className="shadow-
input1"></span>
</div>
<div className="wrap-input1">
    <input className="form-control w3-
large input1" type="number" name="thalach"
onChange={handleThalach} required="required"
placeholder="Maximum Heart Rate Achieved" />
    <span className="shadow-
input1"></span>
</div>
<div className="wrap-input1">
    <select className="form-control
w3-large select input1" id="exang" name="exang"
onChange={handleExang} required="required"
placeholder="Exercise Induced Angina">
        <option hidden>Exercise
Induced Angina</option>
        <option
value="1">Yes</option>
        <option
value="0">No</option>
    </select>
    <span className="shadow-
input1"></span>
</div>

```

```

        <div className="container-contact1-form-
btn">
            <button type="button"
onClick={handlePredict} className="contact1-form-btn">
                <span>
                    Predict
                    <i className="fa fa-long-
arrow-right" aria-hidden="true"></i>
                </span>
            </button>
        </div>
    </form>
</div>
</div>
);
};
HeartDiseaseForm.defaultProps = {
    defaultval:['No Chest Pain','100 - 125 mg/dL','125 -
200 mg/dL','80 - 100 mg/dL','Value 1','0 to 90 %','less
than 0.5 ng/ml'],
    name:['Chest Pain Type','Resting Blood Sugar','Serum
Cholestrol',
        'Fasting Blood Sugar','Resting Electro-
Cardiographic Result',
        'Maximum Heart Rate Achieved','Exercise Induced
Angina']
};

```

KidneyDiseaseForm.js

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

```
import './static/main.css';
import { useHistory } from "react-router-dom";
import kidney from './static/kidney.jpg';
import '../..../styles/w3.css';

export default function KidneyDiseaseForm(props) {
  const history = useHistory();
  const [bp,setBp] = useState(0);
  const [sg,setSg] = useState(0);
  const [al,setAl] = useState(0);
  const [su,setSu] = useState(0);
  const [rbc,setRbc] = useState(0);
  const [pc,setPc] = useState(0);
  const [pcc,setPcc] = useState(0);
  const handleBp = e =>{setBp(e.target.value);};
  const handleSg = e =>{setSg(e.target.value);};
  const handleAl = e =>{setAl(e.target.value);};
  const handleSu = e =>{setSu(e.target.value);};
  const handleRbc = e =>{setRbc(e.target.value);};
  const handlePc = e =>{setPc(e.target.value);};
  const handlePcc = e =>{setPcc(e.target.value);};
  const requestOptions = {
    method : "POST",
    headers : { "Authorization" : "authtoken",
                "Content-Type" : "application/json"
    },
    body : JSON.stringify([bp,sg,al,su,rbc,pc,pcc]),
  };
  const handlePredict = () =>{
    fetch('https://medi-care-backend.herokuapp.com/kidney-pred/',requestOptions)
```

```

        .then((response) => response.json())
        .then((data) => createReport(data));
    };
    const createReport = (vals) =>{
        const requestOptions = {
            method : "POST",
            headers : { "Authorization" : "authtoken",
                "Content-Type" :
"application/json" },
            body : JSON.stringify({
                pred:vals.prediction,
                vals:vals.report,
                defs:props.defaultval,
                names:props.name
            }),
        };
        fetch('https://medi-care-backend.herokuapp.com/map-report/',requestOptions)
            .then((response) => response.json())
            .then((data) =>
history.push('/report/'+ 'kidney', {state:data}));
    };
    return (
        <div className="contact1">
        <div className="container-contact1">
            <div className="contact1-pic js-tilt" data-
tilt>
                <img src={kidney} alt="IMG" />
            </div>
            <form className="contact1-form">
                <span className="contact1-form-title">

```

CHECK WETHER YOU HAVE ANY KIDENY
PROBLEM OR NOT

```
</span>
<div className="wrap-input1">
  <input className="input1 form-
control" type="number" name="bp" onChange={handleBp}
required="required" placeholder="Blood Pressure" />
  <span className="shadow-
input1"></span>
</div>
<div className="wrap-input1" >
  <input className="input1 form-
control" type="number" name="sg" onChange={handleSg}
required="required" placeholder="Specific Gravity" />
  <span className="shadow-
input1"></span>
</div>
<div className="wrap-input1">
  <input className="input1 form-
control" type="number" name="al" onChange={handleAl}
required="required" placeholder="Albumin" />
  <span className="shadow-
input1"></span>
</div>
<div className="wrap-input1">
  <input className="input1 form-
control" type="number" name="rbc" onChange={handleRbc}
required="required" placeholder="Red Blood Cells Count"
/>
  <span className="shadow-
input1"></span>
```

```

        </div>
        <div className="wrap-input1">
            <input className="input1 form-
control" type="number" name="pc" onChange={handlePc}
required="required" placeholder="Pus Cell Count" />
            <span className="shadow-
input1"></span>
        </div>
        <div className="wrap-input1">
            <input className="input1 form-
control" type="number" name="pcc" onChange={handlePcc}
required="required" placeholder="Pus Cell Clumps" />
            <span className="shadow-
input1"></span>
        </div>
        <div className="container-contact1-form-
btn">
            <button type="button"
onClick={handlePredict} className="contact1-form-btn">
                <span>
                    Predict
                    <i className="fa fa-long-
arrow-right" aria-hidden="true"></i>
                </span>
            </button>
        </div>
    </form>
</div>
</div>
);
};

```

```
KidneyDiseaseForm.defaultProps = {
  defaultval: ['90/60mmHg - 120/80 mmHg', '1.010 - 1.020',
    '3.4 - 5.4 g/dL', '100 - 125 mg/dL', '4.7 - 6.1 mL', ' up to 5 in males and may be up to 10 in females', '0-5'],
  name: ['Blood Pressure', 'Specific Gravity', 'Albumin', 'Blood Sugar Level', 'Red Blood Cells Count', 'Pus Cell Count', 'Pus Cell Clumps']
};
```

DiseaseDetection.js

```
import React from "react";
import "../../styles/DiseaseDetection.css";
import "../../styles/Button_Animation.css";
import Data from "./Data";
import DPCard from "./Cards/DisPredicCard";
function DiseaseDetection() {
  return (
    <>
      <br></br>
      <br></br>
      <div className="header">Disease Prediction</div>
      <br></br>
      <br></br>
      <div className="row d-flex justify-content-center">
        {
          Data.map((e,i)=>{
            return <DPCard
```

```

        img={e.img}
        title={e.title}
        subtitle={e.subtitle}
        para={e.para}
        url={e.url}
        key={i*Math.random()}
      />
    ))
  }
</div>
</>
);
}
export default DiseaseDetection;

```

Health.js

```

import React, { useState } from "react";
import "../../styles/Health.css";
import TextField from "@material-ui/core/TextField";
import { createMuiTheme, ThemeProvider } from
"@material-ui/core/styles";
import { Button } from "@material-ui/core";
import "../../styles/Button_Animation.css"
import fitness from '../../assets/images/fitness.jpg'

function Health() {

  const [height, setHeight] = useState(0);
  const [weight, setWeight] = useState(0);
  const [age, setAge] = useState(0);

```



```
const [page1, setPage1] = useState(true);

const theme = createMuiTheme({
  palette: {
    primary: {
      main: "#32A899",
    },
    secondary: {
      main: "#31D570",
    },
  },
});

const handleSubmit = () => {
  if (height < 1 || weight < 1 || age < 1) {
    return;
  }
  setPage1(!page1);
};

const getBMIResultMessage = () => {
  const bmi = (weight / (height * height)) * 10000;
  if (bmi <= 16) {
    return "Severely Thin";
  } else if (bmi > 16 && bmi <= 17) {
    return "Moderately Thin";
  } else if (bmi > 17 && bmi <= 18.5) {
    return "Mild Thin";
  } else if (bmi > 18.5 && bmi <= 25) {
    return "Normal";
  } else if (bmi > 25 && bmi <= 30) {
```

```
    return "Over Weight";  
  } else {  
    return "Obese";  
  }  
};
```

```
const getRecommendedCalories = () => {  
  const bmi = (weight / (height * height)) * 10000;  
  
  if (bmi <= 18.5) {  
    return "2000-2500 ";  
  } else if (bmi > 18.5 && bmi <= 25) {  
    return "1800-2200";  
  } else if (bmi > 25 && bmi <= 30) {  
    return "1500-1900";  
  } else {  
    return "1300-1700";  
  }  
};
```

```
const getRecommendedDiet = () => {  
  const bmi = (weight / (height * height)) * 10000;  
  
  if (bmi <= 18.5) {  
    return "eat protein rich food with carbs and good  
vitamins. Such as eggs, green vegetables, high toned  
milk, green dals and have a moderate workout daily,  
avoid eating junk food..";  
  } else if (bmi > 18.5 && bmi <= 25) {  
    return "keep maintaining your normal eating habits  
with little excercise and stay witin your daily calorie
```

```

requirements.. Avoid junk food and prefer juices for
good immunity and better metabolism..";
    } else if (bmi > 25 && bmi <= 30) {
        return "Use Green tea with zero sugars, Avoid
oily food and junk food, Eat 3 times a day only with
green vegetables, milk, wheat chappatis and dahi..
You can also eat chocolate (1-2 pcs) as it helps to
reduce fat ..";
    } else {
        return "Remove bad habits of eating, and avoid any
kind of oil based food, avoid rice as much as
possible, drink bitter melon juice for fast fat burn,
a daily moderate exercise is also recommend..";
    }
};
return (
    <ThemeProvider theme={theme}>
        <div className="bmi">
            {page1 ? (
                <div className="bmi__firstPage">
                    <div className="bmi__calculatorContainer">
                        <div className="bmi__formContainer">
                            <div className="BmiSmall">Find out
your</div>
                            <div className="BmiLarge">Body Mass
Index</div>
                            <div className="BmiContent">Your BMI is
a measurement that is a ratio of your weight and height.
It's a good way to gauge whether your weight is in
healthy proportion to your height. In fact, knowing your
BMI can help you - and your GP - determine any health

```

risks you may face if it's outside of the healthy range.</div>

```
    <form className="bmi__form">
      <TextField
        label="Height (cm)"
        type="number"
        variant="outlined"
        onChange={(e) =>
setHeight(e.target.value)}
      />
      <TextField
        label="Weight (kg)"
        type="number"
        variant="outlined"
        onChange={(e) =>
setWeight(e.target.value)}
      />
      <TextField
        label="Age (yrs)"
        type="number"
        variant="outlined"
        onChange={(e) =>
setAge(e.target.value)}
      />
      <Button onClick={handleSubmit}
        //button rubberband animation
        onMouseDown ={ e
=>e.currentTarget.classList.add("btn-animation")}
        onMouseUp ={ e=>
e.currentTarget.classList.remove("btn-animation")}
      >Calculate</Button>
```

```
        </form>
      </div>
    </div>
    <div className="bmi__imgContainer">
      <img src={fitness} alt="bmi image" />
    </div>
  </div>
) : (
  <div className="bmi__secondPage">
    <div className="bmi__resultContainer">
      <div className="bmi__resultHeader">
        <div className="bmi__inputData">
          <div>
            <img src={"/images/age.svg"} />
            <p>{age} years</p>
          </div>
        </div>
        <div className="bmi__inputData">
          <div>
            <img src={"/images/height.svg"} />
            <p>{height} cm</p>
          </div>
          <div className="bmi__inputData">
            <div>
              <img src={"/images/scale.svg"} />
              <p>{weight} kg</p>
            </div>
          </div>
        </div>
        <div className="bmi__resultData">
```

```

        <div className="bmi__resultBMI">
            Your BMI : {(weight / (height *
height)) * 10000).toFixed(2)}
        </div>
        <div className="bmi__resultMessage">
            <p>You are {getBMIResultMessage()}</p>
        </div>
    </div>
    <div className="bmi__waterContainer">
        <img src={"/images/water.svg"} />
        <p>Drink At least 8 litres of water
daily</p>
    </div>
    <div className="bmi__recommendations">
        <table>
            <tr>
                <td>Recommended Calories</td>
                <td>{getRecommendedCalories()} per
day</td>
            </tr>
            <tr>
                <td>Recommended Body Weight</td>
                <td>{((height * height * 21) /
10000).toFixed(2)} kg</td>
            </tr>
            <tr>
                <td>Recommended Diet</td>
                <td>{getRecommendedDiet()}</td>
            </tr>
        </table>
    </div>

```

```

        </div>
      </div>
    )}
  </div>
</ThemeProvider>
);
}
export default Health;

```

Navbar.js

```

import React, { useState, useRef } from 'react';
import { Link } from 'react-router-dom';
import "../../styles/Navbar.css";
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
import { faBars } from '@fortawesome/free-solid-svg-icons';
import "../../styles/Button_Animation.css";

function Navbar() {
  const [click, setClick] = useState(false);
  const btnRef = useRef();
  const handleClick = () => setClick(!click);
  const closeMobileMenu = () =>{

    // toggling the previous state
    setClick((prevState)=>!prevState);

  };

```

```

const [showLinks , setShowLinks] = useState(false);

return (
  <>
    <nav className='navbar'>
      <Link to='/' className='navbar-logo'
onClick={closeMobileMenu}>
        MediCare
      </Link>
      <div className='menu-icon'
onClick={handleClick}>
        <i className={click ? 'fas fa-times' : 'fas
fa-bars'} />
      </div>
      <button className='showlinks'
onClick={()=>setShowLinks(!showLinks) }><FontAwesomeIcon
icon={faBars} /></button>

      <ul className={showLinks ? 'hidden' : 'nav-
menu'}>
        <li className='nav-item'>
          <Link to='/' className='nav-links'
onClick={closeMobileMenu}>
            Home
          </Link>
        </li>
        <li className='nav-item'>
          <Link
            to='/DiseaesPred'
            className='nav-links'

```



```

        onClick={closeMobileMenu}
      >
        Disease Prediction
      </Link>
    </li>
    <li className='nav-item'>
      <Link
        to='/health'
        className='nav-links'
        onClick={closeMobileMenu}
      >
        Health
      </Link>
    </li>
    <li className='nav-item'>
      <Link
        to='/Doctor'
        className='nav-links'
        onClick={closeMobileMenu}
      >
        Doctor
      </Link>
    </li>
    <li className='nav-item'>
      <Link to={click?"/hi":"/"} className='nav-
links ' id='change-language' onClick={ closeMobileMenu
}

//button rubberband animation

```

```

        onMouseDown = { e
=>e.currentTarget.classList.add("btn-animation")}
        onMouseUp = { e=>
e.currentTarget.classList.remove("btn-animation")}
      >
      A/3π
    </Link>

  </li>
</ul>

</nav>

```

Index.jsx

```

import React from 'react'
import "../styles/notFoundPageStyle.css"
import notFound from "../assets/notFound.svg"
import { Link } from 'react-router-dom';

export default function index() {
  return (
    <>
      <div className="notFound">
        <div className="mainBody">
          <img src={notFound} alt="notFound"
        />
          <h2>OPPS !</h2>
          <p>The page you are requested could
not be found!</p>

```

```

                <Link to="/"
style={{textDecoration:'none'}}>
                <button>
                    Go To Home
                </button>
            </Link>
        </div>
    </div>
</>
)
}

```

App.js

```

import React, { useState } from 'react';
import Navbar from './components/NavBar/Navbar';
import './App.css';
import Home from './components/Home/Homepage.jsx';
import HomeHi from './components/Home/Homepagehi';
import { BrowserRouter as Router, Route, Switch } from
'react-router-dom';
import ContactUs from './components/Health/Health';
import Chatbot from './components/Chatbot/Chatbot';
import DiseaesPred from
'./components/DiseasePred/DiseaseDetection';
import Doctor from './components/Doctor/Doctor';
import Details from './components/Doctor/Details';
import NotfoundPage from './components/NotfoundPage';
import HeartDiseaseForm from
"./components/DiseasePred/HeartDiseasePred/HeartDiseaseF
orm";

```

```
import KidneyDiseaseForm from
"./components/DiseasePred/KidneyDiseasePred/KidneyDiseas
eForm";
import 'bootstrap/dist/css/bootstrap.min.css';
import { CSSTransition } from 'react-transition-group';
import DiseaseReport from
'./components/DiseasePred/DiseaseReport';

function App() {

  //all the routes
  const routes = [
    { path: '/', Component: Home },
    { path: '/DiseaesPred', Component: DiseaesPred },
    { path: '/hi', Component: HomeHi },
    { path: '/Doctor', Component: Doctor },
    { path: '/Details', Component: Details },
    { path: '/health', Component: ContactUs },
    { path: '/heart-predict', Component:
HeartDiseaseForm },
    { path: '/kidney-predict', Component:
KidneyDiseaseForm },
    { path: '/report/:type', Component: DiseaseReport },
  ]
  const [mode, setMode] = useState('light');

  const [alert, setalert] = useState(null);

  const showalert = (message, type) => {
    setalert({
      msg: message,
```

```

    type: type
  });

  setTimeout(() => {
    setalert(null);
  }, 1800);
}
const toggleMode = () => {
  if (mode === 'light') {
    setMode('dark');
    document.body.style.backgroundColor = "#050A30";
    showAlert("Dark mode has been enbled", "primary");
  }
  else {
    setMode('light');
    document.body.style.backgroundColor = "white";
    showAlert("Light mode has been enbled",
"primary");
  }
}
return (
  <Router>
    <Chatbot />
    <Navbar mode={mode} toggleMode={toggleMode} />
    <Switch>
      {
        routes.map(route => {
          const { path, Component } = route;

          return <Route key={path} path={path} exact >
            {

```

```

      ({ match }) => (
        //component to make page transiiton on
routing
        <CSSTransition
          in={match !== null}
          timeout={500}
          unmountOnExit
          classNames="page"
        >
          <div className="page">
            <Component />
          </div>
        </CSSTransition>
      )}

```

Index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <!-- <link rel="icon"
href="%PUBLIC_URL%/favicon.ico" /> -->
    <meta name="viewport" content="width=device-width,
initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />

```

```
<link rel="apple-touch-icon"
href="%PUBLIC_URL%/logo192.png" />
<link rel="preconnect" href="https://api.github.com"
/>
<link rel="manifest"
href="%PUBLIC_URL%/manifest.json" />
<link
href="https://fonts.googleapis.com/css?family=Montserrat"
rel="stylesheet"
<link
rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1
/css/bootstrap.min.css"
integrity="sha384-
Vkoo8x4CGs03+Hh xv8T/Q5PaXtkKtu6ug5T0eNV6gBiFeWPGFN9MuhOf
23Q9Ifjh"
crossorigin="anonymous"
/>
<title>MediCare</title>
</head>
<body>
<noscript>You need to enable JavaScript to run this
app.</noscript>
<div id="root"></div>
<script
src="https://code.jquery.com/jquery-
3.4.1.slim.min.js"
```

```
        integrity="sha384-
J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYf
oRSJoZ+n"
        crossorigin="anonymous"
    ></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/
umd/popper.min.js"
        integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVv
oxMfooAo"
        crossorigin="anonymous"
    ></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/
js/bootstrap.min.js"
        integrity="sha384-
wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3
Og8ifwB6"
        crossorigin="anonymous"
    ></script>
</body>
</html>
```

Urls.py

```
"""core URL Configuration
The `urlpatterns` list routes URLs to views. For more
information please see:
https://docs.djangoproject.com/en/4.0/topics/http/urls/
```


Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to `urlpatterns`: `path('', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to `urlpatterns`: `path('', Home.as_view(), name='home')`

Including another `URLconf`

1. Import the `include()` function: `from django.urls import include, path`
2. Add a URL to `urlpatterns`: `path('blog/', include('blog.urls'))`

```
"""
```

```
from django.contrib import admin
from django.urls import path, include
from doctors import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/doctors', include('doctors.urls'))
]
```

Manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative
tasks."""
import os
import sys
def main():
```

```
"""Run administrative tasks."""
os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'core.settings')
try:
    from django.core.management import
execute_from_command_line
except ImportError as exc:
    raise ImportError(
        "Couldn't import Django. Are you sure it's
installed and "
        "available on your PYTHONPATH environment
variable? Did you "
        "forget to activate a virtual environment?"
    ) from exc
execute_from_command_line(sys.argv)
if __name__ == '__main__':
    main()
```

=====X=====