WPT LAB -21/12/2022

Prajwal_77

1. Create a messaging app (like WhatsApp structure).

NOTE- FOR DETAIL BACKEND PROJECT VISIT-

[https://github.com/prajwalthete/REACTSTUDY.git]

APP.JS-/FRONTEND

```
import axios from "axios";
import { useEffect, useRef, useState } from "react";

function App() {
  // Data Member
  let inputRef = useRef();
  let [title] = useState("Messaging App");
  let [message, setMessage] = useState("");
      let [messageList, setMessageList] = useState([]);

      // Spl Funcn :: Hook :: Like Constructor ::  Called while the Compoent is Initialized.
      useEffect(() => {
        // console.log("I AM GETTING CALLED");
        getAllMessages();
      }, []);

      // Member Funcns
      let handleOnChangeMessage = (e) => {
        message = e.target.value;
        setMessage(message);
        // setMessage(e.target.value)
      };

      let getAllMessages = async () => {
        let url = `http://localhost:3001/messages`;
        let response = await axios.get(url);
        // console.log(response);

```

```
30.         // Getting the Message From Server :: And re-
  rendering
31.         messageList = [...response.data];
32.         setMessageList(messageList);
33.       };
34.
35.       let createNewMessage = async (reply) => {
36.         let url = `http://localhost:3001/message`;
37.         // console.log(inputRef.current);
38.         if (!inputRef.current.checkValidity()) {
39.           alert("Invalid");
40.           return;
41.         }
42.
43.         let data = {
44.           message: message,
45.           messageTime: new Date(),
46.           reply: reply,
47.         };
48.
49.         await axios.post(url, data);
50.
51.         setMessage("");
52.
53.         // To Refresh the content
54.         getAllMessages();
55.       };
56.
57.       let checkEnterCode = (e) => {
58.         if (e.keyCode === 13) {
59.           createNewMessage();
60.         }
61.       };
62.
63.       return (
64.         <div>
65.           <h1 className="bg-dark text-light p-3 sticky-
  top">{title}</h1>
66.
67.           <div className="row justify-content-center">
68.             <div className="col-12 col-md-6">
69.               <div className="d-flex">
```

```
                <input
                  className="form-control form-control-lg"
                  type="text"
                  placeholder="Hi...whatsapp...!!"
                  value={message}
                  onChange={handleOnChangeMessage}
                  onKeyUp={checkEnterCode}
                  ref={inputRef} //
  document.querySelector()
                  required
                  minLength={2}
                />
                <input
                  className="btn btn-secondary"
                  type="button"
                  value="Add"
                  onClick={() => createNewMessage(false)}
                />
                <input
                  className="btn btn-secondary"
                  type="button"
                  value="Reply"
                  onClick={() => createNewMessage(true)}
                />
              </div>
            </div>
          </div>

          {messageList.map((item, index) => (
            <div className="row justify-content-center">
              <div className="col-12 col-md-6 ">
                <div
                  key={index}
                  className={
                    item.reply
                      ? "d-flex justify-content-end my-1"
                      : "d-flex justify-content-start my-1"
                  }
                >
                  <div className="badge text-bg-secondary">
                    {item.message}
                    <span className="ms-4">
```

```
                        {new
  Date(item.messageTime).getHours()}:
                        {new
  Date(item.messageTime).getMinutes()}
                      </span>
                    </div>
                  </div>
                </div>
              </div>
            ))}
          </div>
        );
    }

    export default App;

```

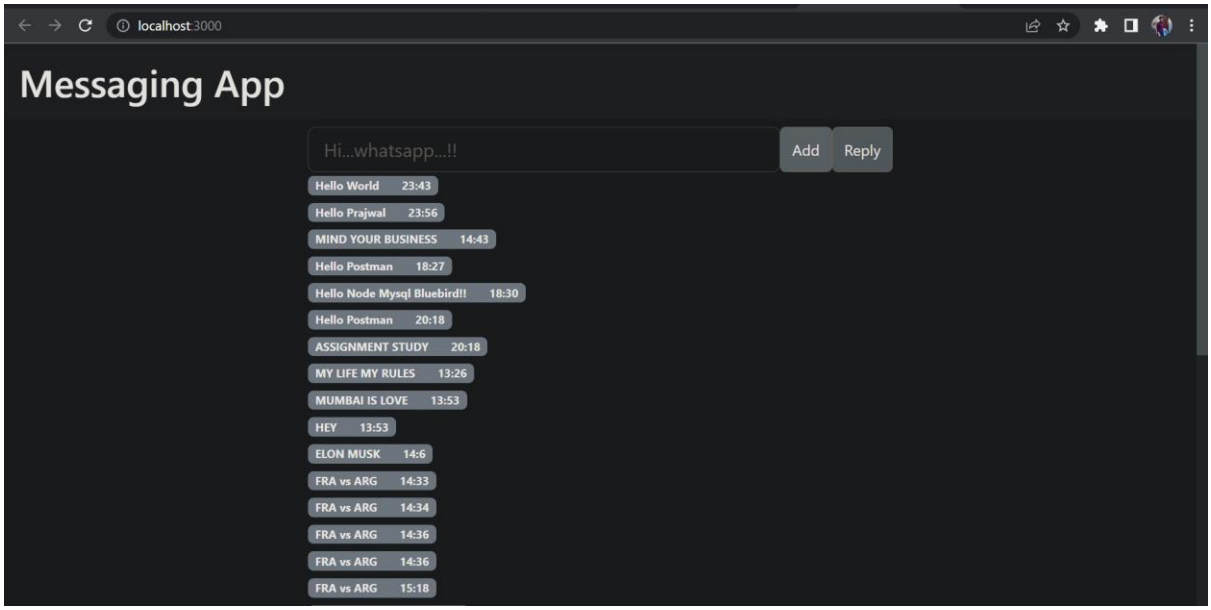
INDEX.JS/ BACKEND

NOTE- For detail Backend project visit - https://github.com/prajwalthete/message-backend.git

```
import express from "express";
import bluebird from "bluebird";
import { createConnection } from "mysql";
import cors from "cors";

const app = express();

app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use(cors());

const connectionUri = {
  host: "localhost",
  user: "root",
  password: "Prajwal@123",
  database: "cdac",
};

/* http://localhost:3001/ */
app.get("/", (req, res) => res.send("Hello, NodeJS!"));

/* http://localhost:3001/messages */
app.get("/messages", async (req, res) => {
  let list = [];
  let connection = createConnection(connectionUri);
```

```
  bluebird.promisifyAll(connection);

  await connection.connectAsync();

  let sql = `SELECT * FROM message `;
  list = await connection.queryAsync(sql);

  await connection.endAsync();

  res.json(list);
});

/* http://localhost:3001/message */
app.post("/message", async (req, res) => {
  let connection = createConnection(connectionUri);
  bluebird.promisifyAll(connection);

  await connection.connectAsync();

  let message = req.body.message;
  let reply = req.body.reply;
  // let sql = `INSERT INTO message (message, reply) VALUES ('${message}',
${reply})`;
  let sql = `INSERT INTO message (message, reply) VALUES (?, ?)`;
  await connection.queryAsync(sql, [message, reply]);

  await connection.endAsync();

  let output = { msg: "Record Created Successfully" };
  res.json(output);
});

app.listen(3001);
```

OUTPUT-



2. Create a Multi-page website using React.js.

NOTE FOR DETAIL PROJECT VISIT-

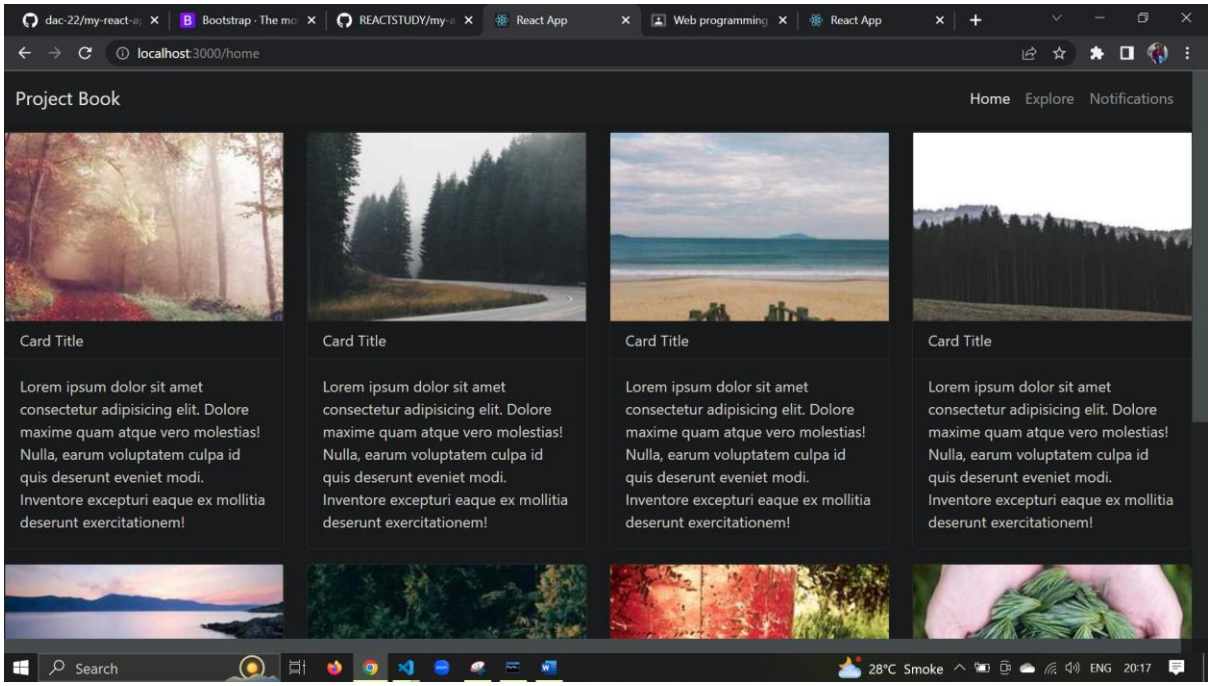
https://github.com/prajwalthete/REACTSTUDY.git

APP.JS FILE

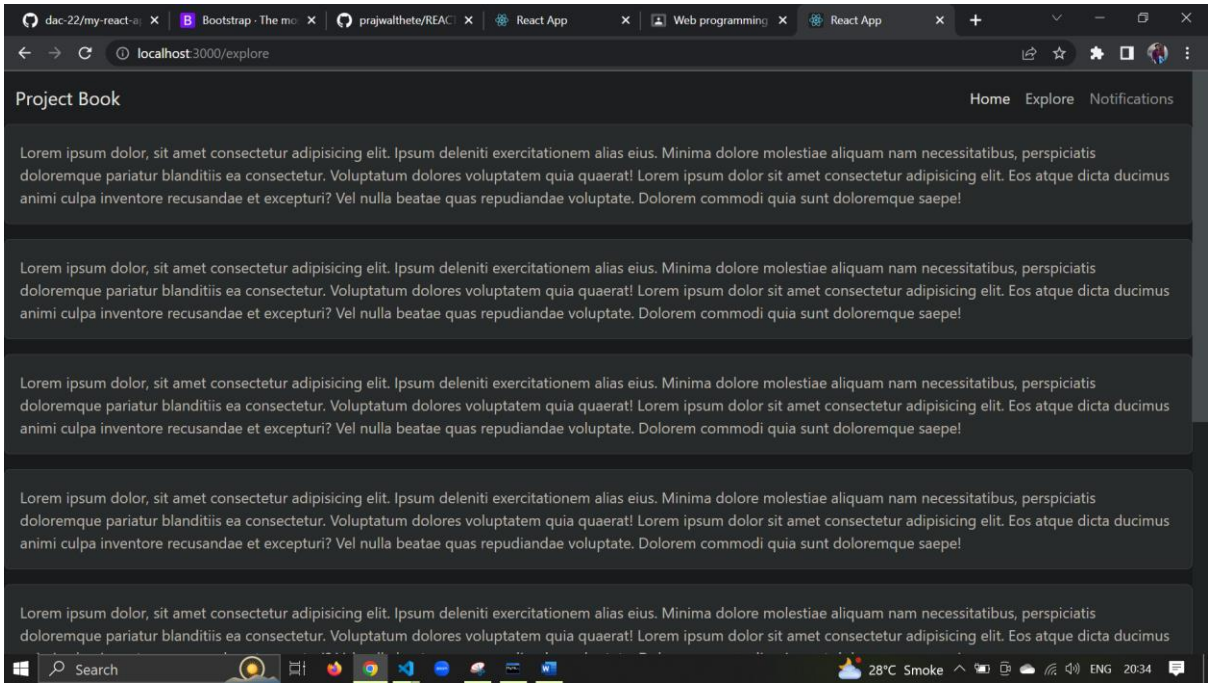
```
import { Link, Route, Routes } from "react-router-dom";
import AppNavLinks from "./components/AppNavLinks";
import Explore from "./components/Explore";
import Home from "./components/Home";
import Notifications from "./components/Notifications";
import PageNotFound from "./components/PageNotFound";

function App() {
  return (
    <div>
      <AppNavLinks />

      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/home" element={<Home />} />
        <Route path="/explore" element={<Explore />} />
        <Route path="/notifications" element={<Notifications />}
/>
        <Route path="*" element={<PageNotFound />} />
```

```
      </Routes>
    </div>
  );
}

export default App;
```

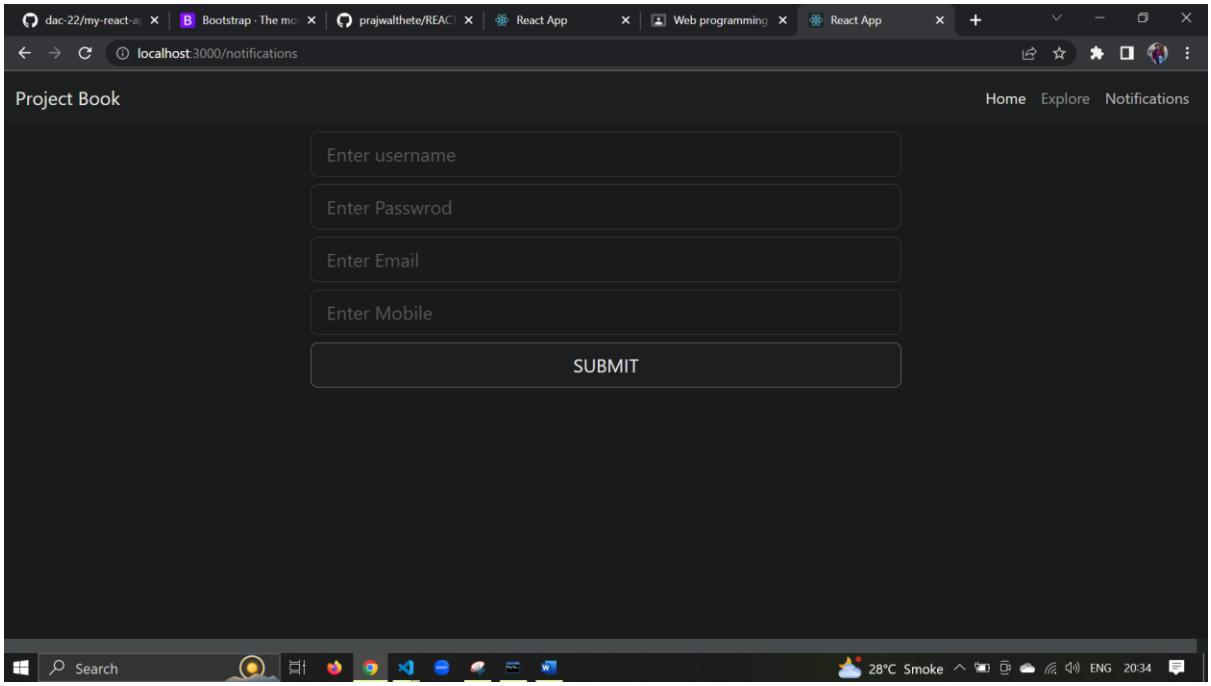
PAGE1



PAGE2



PAGE3


localhost:3000/notifications
Project Book
Home
Explore
Notifications
Enter username
Enter Passwrod
Enter Email
Enter Mobile
SUBMIT
Search
28°C Smoke
ENG
20:34