

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology in Information Technology

By
R.Siranjeevi
REG. NO. 39120113



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

APRIL 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)



Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of R.Siranjeevi
(Reg. No. : 39120113) who carried out the project entitled “**NOTES
APPLICATION**” under my supervision from February 2022 to April 2022.

Internal Guide
Dr.S.Gowri

Head of the Department
Dr. R. SUBHASHINI. M.E., Ph.D.,

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I **Siranjeevi** hereby declare that the project report entitled **Notes application** done by me under the guidance of **Dr.S.Gowri** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Information technology.

DATE: 28.04.2022

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to, **Dr. T. SASIKALA M.E., Ph.D.**, Dean, School of Computing and **Dr.R.Subhashini,M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing me the necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.Gowri** for her valuable guidance, suggestions which paved way for the successful completion of my project work.

I wish to express my thanks to all teaching and non-teaching staff members of the department of Information Technology who were helpful in many ways for the completion of project.

TRAINING CERTIFICATE



ABSTRACT

- The Project aims to build a NOTES APPLICATION online system through which user can able to login and read their choice of Notes.
- Keep notes is one of the most popular, simple and widely used note-taking apps for Android devices. This application is very safe and secure for users.
- Users can able to register themselves to login into the application , users are allowed to add a new notes, update their notes , search notes and read the notes.

TABLE OF CONTENTS

CHAPTE R No.	TITLE	PAGE No
	ABSTRACT	vi
1.	INTRODUCTION	ix
2.	AIM AND SCOPE OF THE PRESENT	x
	2.1 AIM	11
	2.2 SCOPE	11
	2.3 PROJECT FLOW	12
3.	EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED	14

	3.1 Software Required	14
	3.1.1 Windows 10	
	3.1.2 React JS	
	3.1.3 React CLI	
	3.2 Hardware Required	18
	3.2.1 Hard Disk	
	3.2.2 RAM	
	3.2.3 Processor	
	3.2.4 Mouse	
	3.2.5 Keyboard	
	3.2.6 Speaker	
4	RESULTS AND DISCUSSION, PERFORMANCE Recommendation	22
	4.1 Result and discussion	22
	4.2 Recommendation	22
	4.3 Screen shots	
	4.3.1 Input	
	4.3.2 Output	
	SUMMARY AND CONCLUSIONS	23
5	REFERENCES	24
	APPENDIX	23
	SAMPLE SCREEN	24

CHAPTER-1

INTRODUCTION

1.1 ABOUT THE CONCEPT

Notes application helps to keep your notes in the permanent records in the application. Adding your notes to this application, helping to remainder your daily work and increasing potential for collaboration and interaction with users.

Keep notes is one of the most popular, simple and widely used note-taking apps for Android devices. It allows taking notes in various formats such as text, handwriting, image, and even storing voice recording. The app facilitates writing notes in the form of lists, notes, and handwritten form. You can even categorize your notes in different labels such as personal, work, and other labels that you like. The app's search option saves your time so that you don't require scrolling down your list.

CHAPTER - 2

AIM AND SCOPE OF PRESENT INVESTIGATION

2.1 AIM

Notes application allows users to make different kinds of notes, including text, lists, images, and audio. Users can set reminders, which are integrated with Google Now, with options for time or location.

The main objective of the notes app is to make notes available for user only. User can able to sign up or login and then user can able to add notes and you can also add a password to open your personal notes. It is easy and immediate online access.

2.2 SCOPE

Google Keep allows users to make different kinds of notes, including text, lists, images, and audio. Users can set reminders, which are integrated with Google Now, with options for time or location. Text from images can be extracted using optical character recognition technology. Voice recordings created through Keep are automatically transcribed. Keep can convert text notes into checklists. Users can choose between a single-column view and a multi-column view. Notes can be color-coded, with options for white, red, orange, yellow, green, teal, blue or grey. Users can press a "Copy to Google Doc" button that automatically copies all text into a new Google Docs document. Users can create notes and lists by voice. Notes can be categorized using labels, with a list of labels in the app's navigation bar.

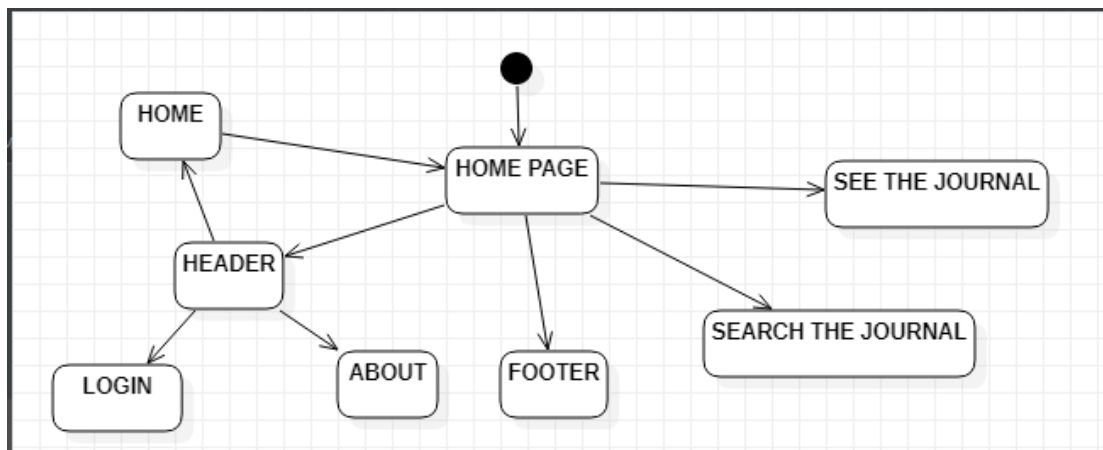
Google Keeps allows users to keep notes, and write to-do lists. It allows users to keep track of different notes on a dashboard, this allows departments to plan and keep tabs on projects or to-do lists in an easy manner.

Google Keep is one of the best note-taking apps for collaboration and sharing. You can share your notes with people at the office or add your significant other to a note so you both can add stuff to it.

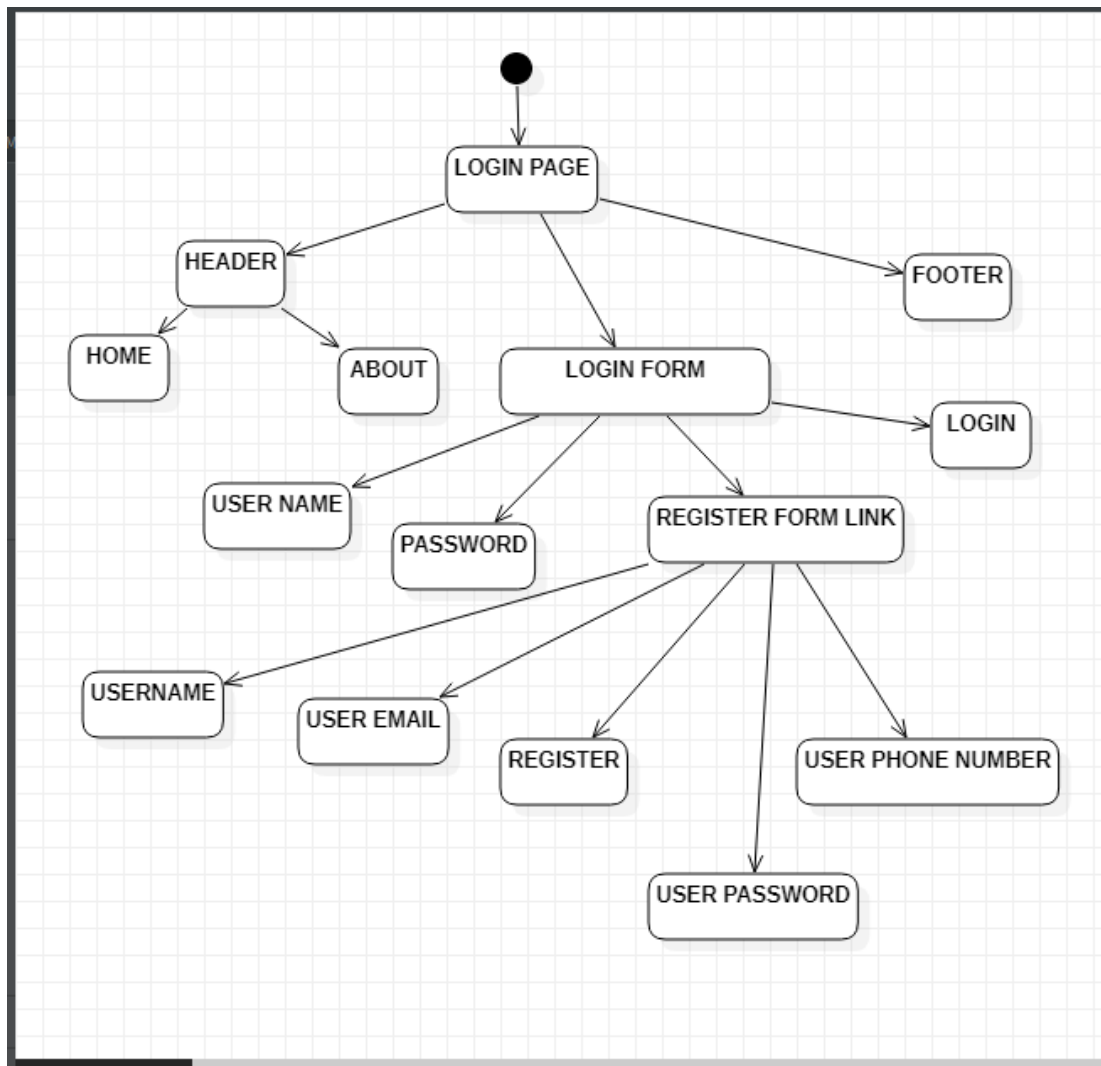
.

PROJECT FLOW

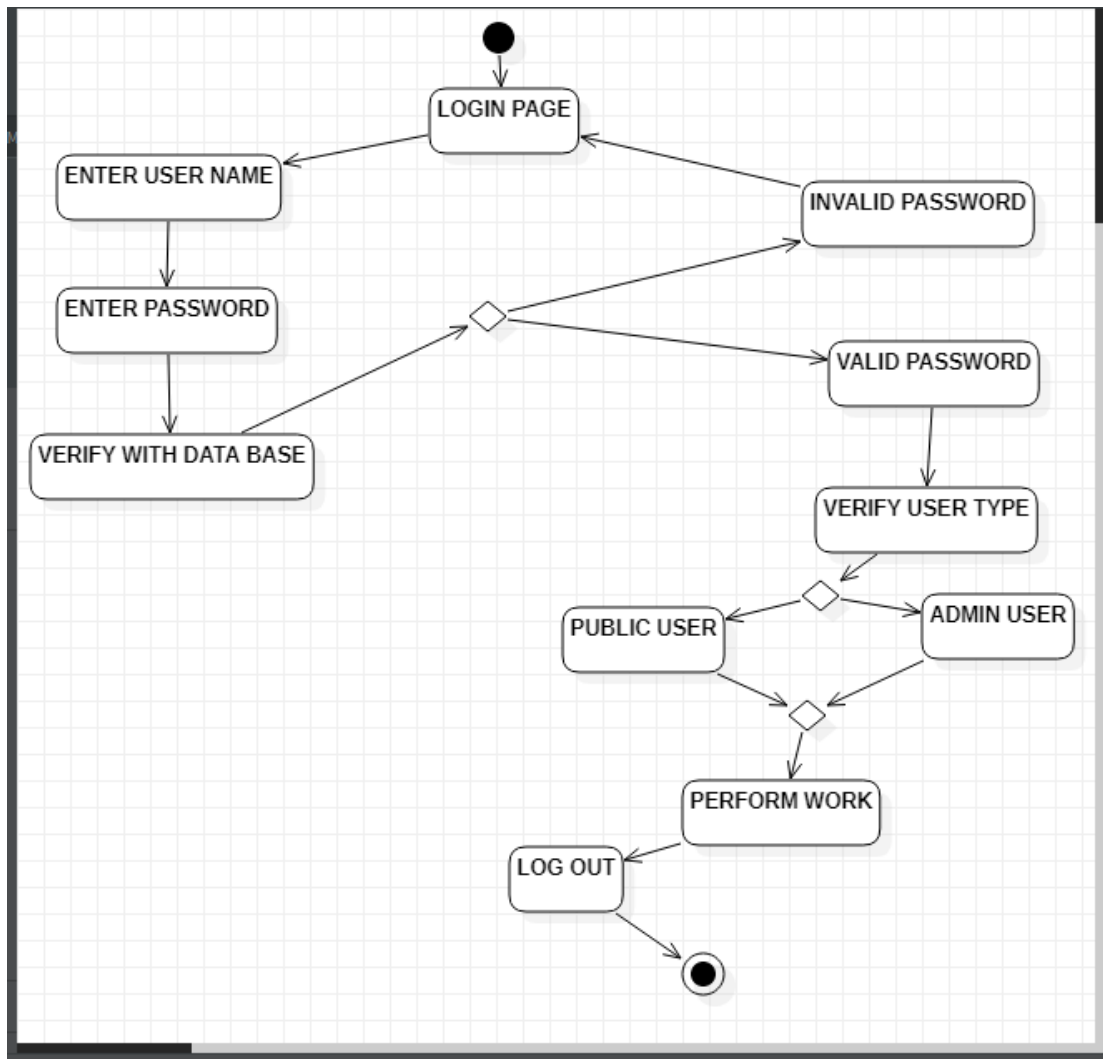
MAIN PAGE



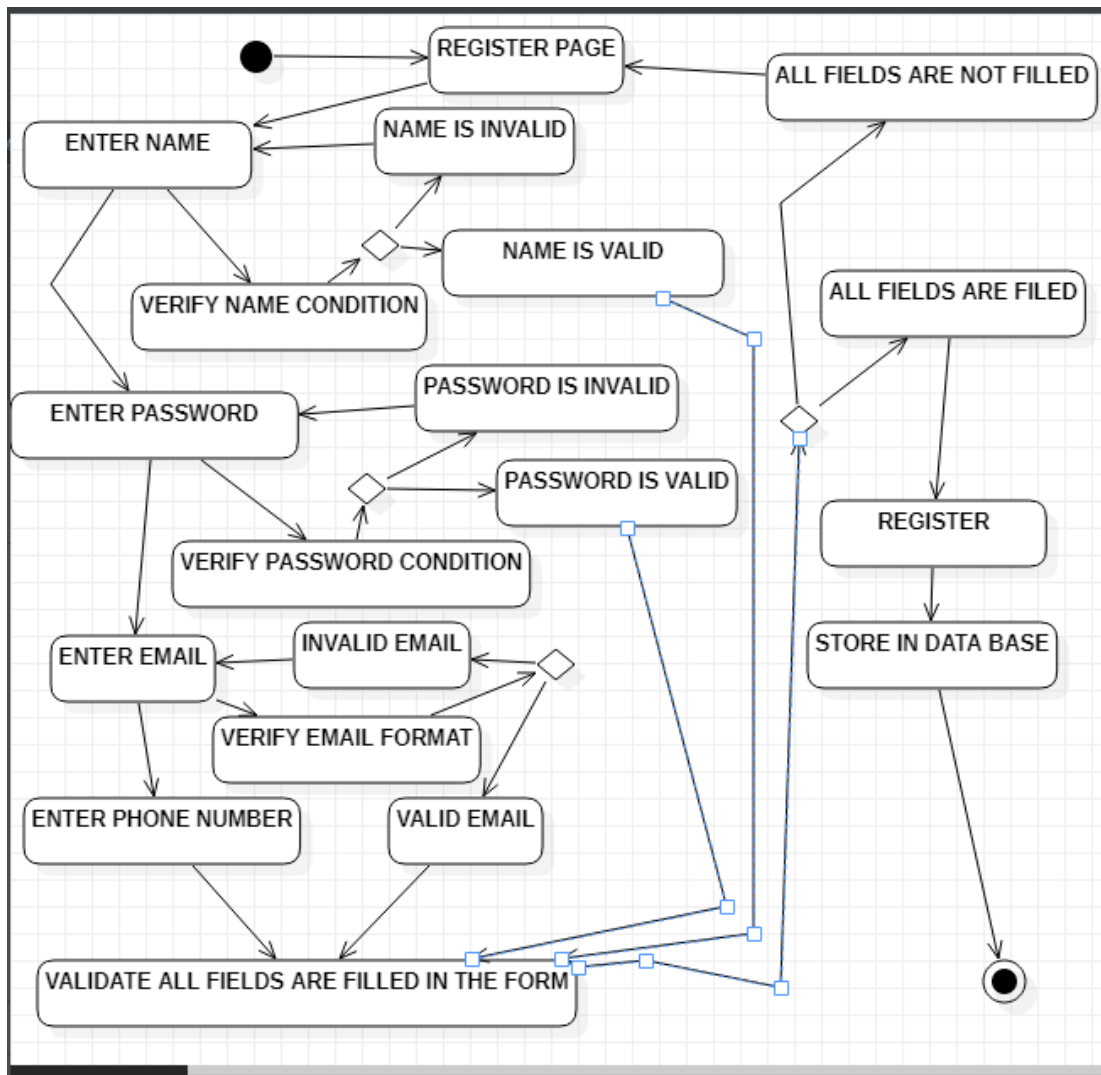
LOGIN PAGE



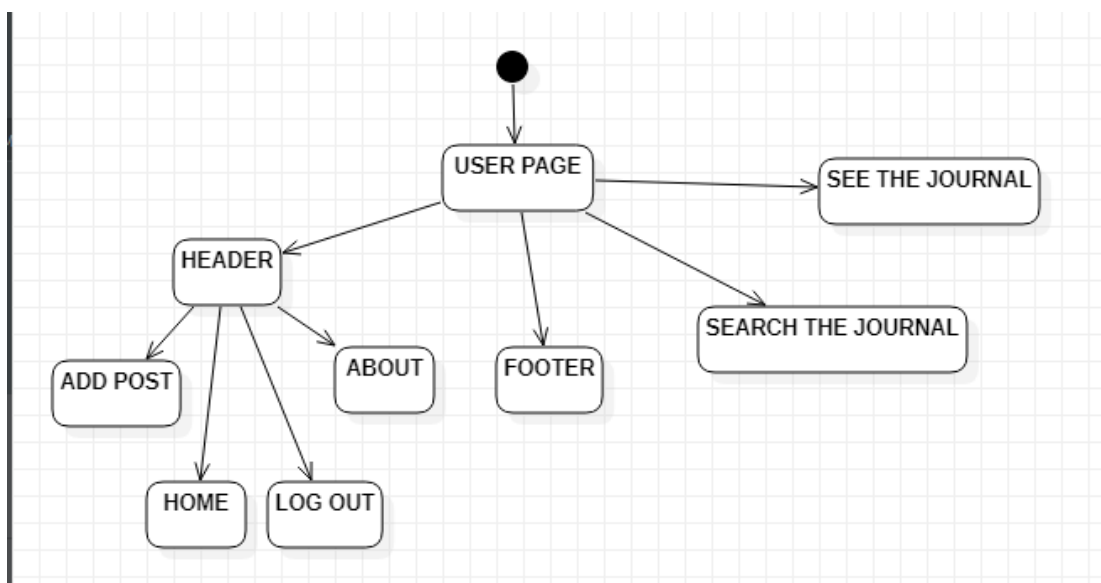
LOGIN PAGE WORKING FLOW



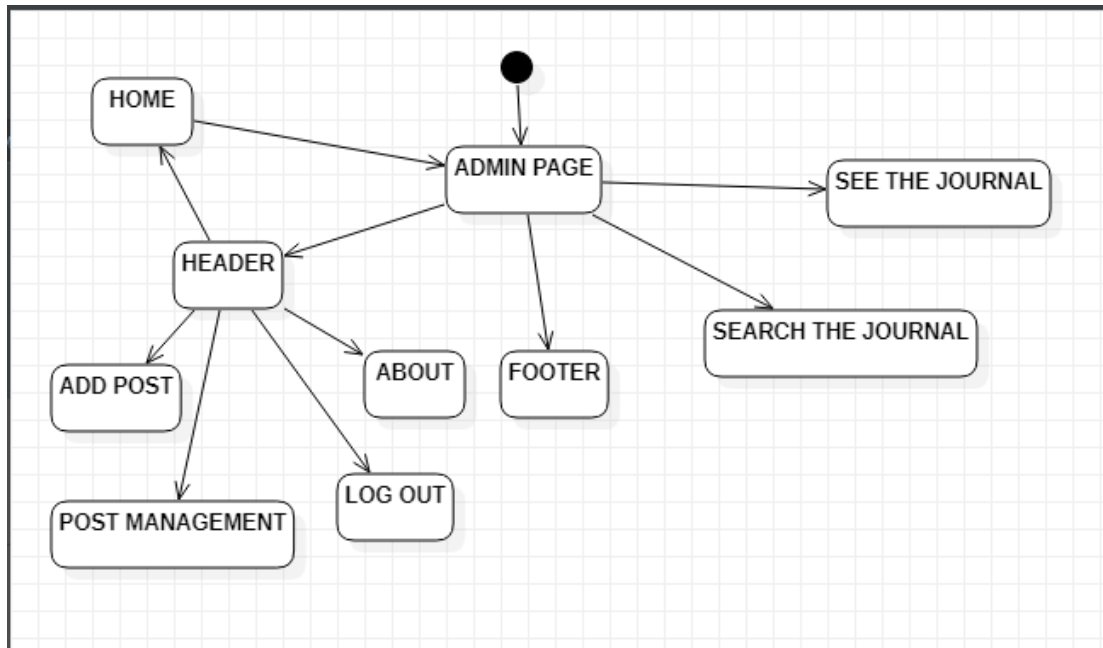
REGISTER PAGE WORKING FLOW



AFTER LOGIN PAGE (USER PAGE)



AFTER LOGIN PAGE (ADMIN PAGE)



CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS

3.1 SOFTWARE REQUIREMENT

- Windows 10 or above
- React JS
- React CLI
- Database

3.1.1 Windows 10 or above

Windows 10 is a Microsoft operating system for personal computers, tablets, embedded devices and internet of things devices.

Microsoft released Windows 10 in July 2015 as a follow-up to Windows 8. The company has said it will update Windows 10 in perpetuity rather than release a new, full-fledged operating system as a successor.

3.1.2 React JS

React is a **JavaScript-based UI development library**. Facebook and an open-source developer community run it. Although React is a library

rather than a language, it is widely used in web development. The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web development.

Features of React

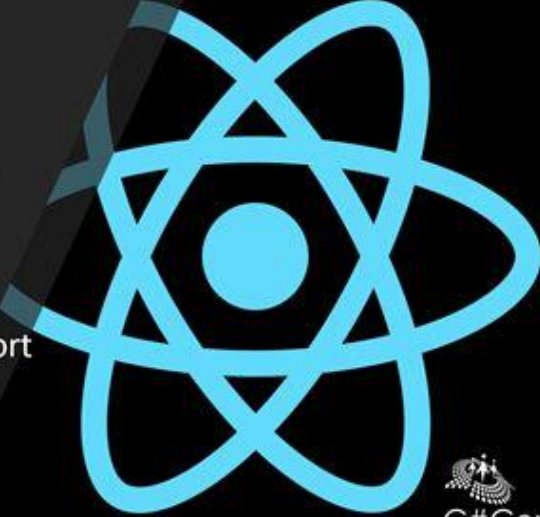
React offers some outstanding features that make it the most widely adopted library for frontend app development. Here is the list of those salient features.

JSX

JSX is a JavaScript syntactic extension. It's a term used in React to describe how the user interface should seem. You can write HTML structures in the same file as JavaScript code by utilizing JSX.

React Properties

1. React is declarative
2. React is simple yet powerful
3. React is component-based
4. React supports server-side
5. React supports mobile support
6. React is extensible



3.1.3 React CLI

ReactCLI is a **ReactJS** component that provides a simple way to present a command line interface to your user in your web app.

React have its own CLI but currently they are only supporting creating an app (create-react-app). create-react-app used to generate the boilerplate version of a React application thru command line.

3.2 HARDWARE REQUIREMENT

- Hard Disk
- RAM – 1 GB
- Processor – Dual Core or Above
- Mouse
- Keyboard
- Monitor

3.2.1 Hard Disk

A hard disk or fixed disk, is an electro-mechanical data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks (platters) coated with magnetic material.

During the mid-1990s the typical hard disk drive for a PC had a capacity of about 1 gigabyte. As of January 2019, desktop hard disk drives typically had a capacity of 1 to 6 terabytes, with the largest-capacity drives reaching 15 terabytes

3.2.2 RAM

Random access memory (RAM) is a type of data storage used in computers that is generally located on the motherboard. This type of memory is volatile and all

information that was stored in RAM is lost when the computer is turned off.

The RAM chip may be individually mounted on the motherboard or in sets of several chips on a small board connected to the motherboard. Older memory types were in the form of chips called dual in-line package (DIP). Although DIP chips are still used today, the majority of memory is in the form of a module, a narrow printed circuit board attached to a connector on the motherboard. The three main memory circuit boards types containing chips are: RIMMs (Rambus in-line memory modules), DIMMs (dual in-line memory modules) and SIMMs (single in-line memory modules). Most motherboards today use DIMMs. Sometimes, people get confused about the difference between memory and storage, in part because both can be measured in megabytes (MB), gigabytes (GB) and terabytes (TB).

By contrast, storage is slower, but it can retain data when the device is powered down. So, for example, if a document has been saved to a hard drive prior to a power outage or system crash, the user will still be able to retrieve it when the system is back up and running.

3.2.3 Processor

A processor is an integrated electronic circuit that performs the calculations that run a computer. A processor performs arithmetical, logical, input/output (I/O) and other basic instructions that are passed from an operating system (OS). Most other processes are dependent on the operations of a processor.

Dual core is a CPU that has two distinct processors that work simultaneously in the same integrated circuit. This type of processor can function as efficiently as a single processor but can perform operations up to twice as quickly.

3.2.4 Mouse

A mouse is a small handheld input device that controls a computer screen's cursor or pointer in conjunction with the way it is moved on a flat surface. The mouse term name originates from its likeness to a small, corded and elliptical shaped device that looks like a mouse tail.

Some mouse devices have integrated features, such as extra buttons that may be programmed and assigned with different commands.

Because the mouse reduces the use of a keyboard, its invention and continuous innovation is considered one of the most important breakthroughs in computer ergonomics.

3.2.5 Keyboard

A computer keyboard is an input device used to enter characters and functions into the computer system by pressing buttons, or keys. It is the primary device used to enter text. A keyboard typically contains keys for individual letters, numbers and special characters, as well as keys for specific functions. A keyboard is connected to a computer system using a cable or a wireless connection.

A computer keyboard is an input device used to enter characters and functions into the computer system by pressing buttons, or keys. It is the primary device used to enter text. A keyboard typically contains keys for individual letters, numbers and special characters, as well as keys for specific functions.

3.2.6 Monitor

A computer monitor is an electronic device that shows pictures for computers. Monitors often look similar to televisions. The main difference between a monitor and a television is that a monitor does not have a television tuner to change channels. Monitors often have higher display resolution than televisions.

The advent of display technology has paved the way for the continuous evolution of the monitor, whether it's for computers, television, mobile devices or any device that has a display. The current contenders for top-tier technology being used for display devices includes Super LCD 3 (SLCD3) and Super AMOLED. It should be noted that LED displays are actually just a kind of LCD display that use LED lights as backlight illumination.

CHAPTER 4

TESTING

4.1 System Testing

Quality assurance is an important step in software engineering. This overlaps with all the phases of development right from the requirement analysis. This quality requirement of the software system must be clearly extracted during the requirement analysis and all the subsequent phases should be made biased to that, the final testing will become trivial and less expensive. There are number of quality parameters like correctness, accuracy, reliability, robustness, efficiency, effectiveness, reusability, maintainability etc.. The state of requirement of each of these parameters will vary depending upon the name and domain of the application. The testing should be done at the end of all development steps. Even though the final testing and verification are inevitable for better life and functionality of the software. The different software testing approaches and methods like white box testing and black box testing. The major phases in testing are design of test plan, setting up test case and test candidate and test procedure, testing and correction. This is a cycle process and the software will circulate through all the steps till it attends the required quality. The testing is carried in the following steps.

4.2 Unit testing

Unit testing focuses verification effort on the smallest unit of software design the module. Using the details design description as a guide, important control paths are tested to uncover errors within boundary of the boundary of the module. The relative complexity of tests and the errors detected as a result is limited by the constrained scope established for unit testing.

Unit testing is normally considered an adjacent to coding steps. After source level code has been developed, reviewed, and verified for correct syntax, unit test case design begins. A review of design information provides guidance for establishing test

cases that are likely to uncover error in each case of the categories discussed above. Each test case should be coupled with a set of expected results.

4.3 Integration testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting test to uncover error associated with interfacing .The objective is to take unit tested modules and build a program structure that has been dictated by design.

There is often a tendency to attempt no incremental integration; that is to construct the program using a “big bang “approach. The entire modules are combined in advance. The entire program is tested as whole and chaoses usually result! A set of error is encountered. Correction is difficult because the isolation of cause is complicated by the vast expanse of entire program. Once errors are corrected, new ones appear and process continues in a seemingly endless loop.

CHAPTER 5

RESULT AND DISCUSSION, PERFORMANCE ANALYSIS

5.1 Result & Discussion

Notes application helps to keep your notes in the permanent records.

Adding your notes to this record, helping to remainder your daily work and increasing potential for collaboration and interaction with users

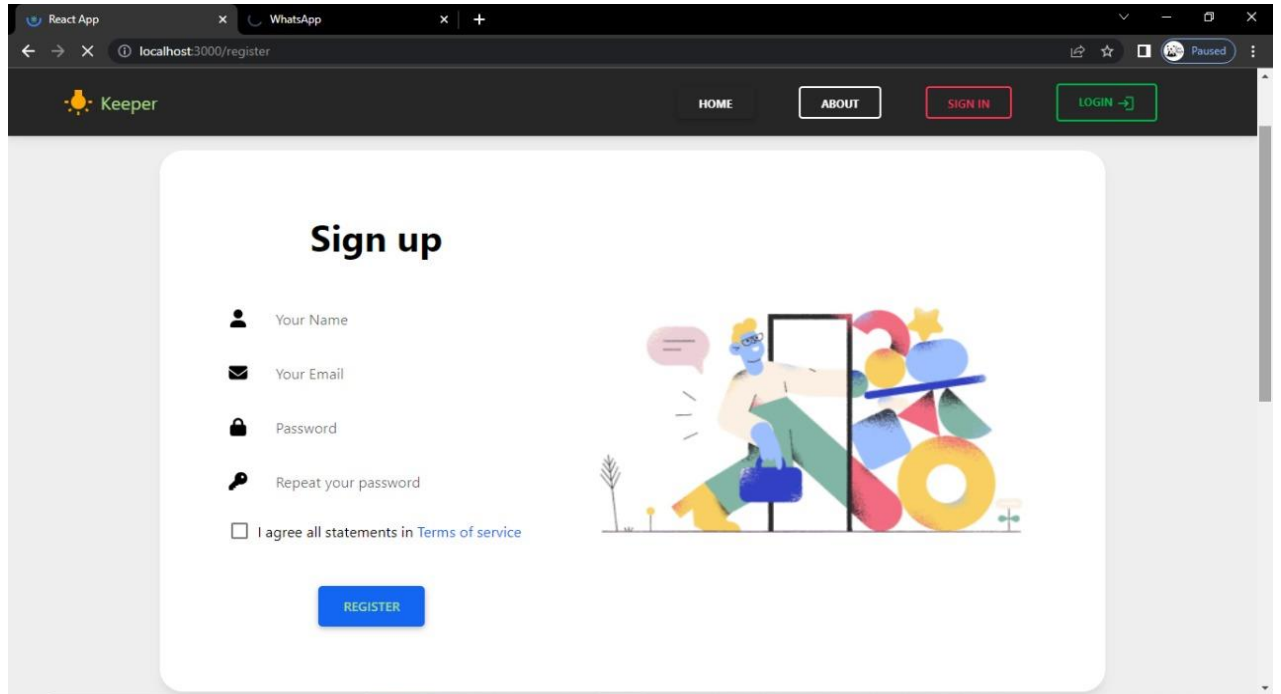
5.2 RECOMMENDATION

The Notes application meets a lot of expectations but would perform better if the following recommendations and suggestions are considered:

- i. System testing and maintenance should be performed regularly to avoid sudden system failures.
- ii. Updates and other modifications should be introduced with prior notice to users.

5.3 SCREENSHOTS

5.3.1 INPUT (Sign up page)



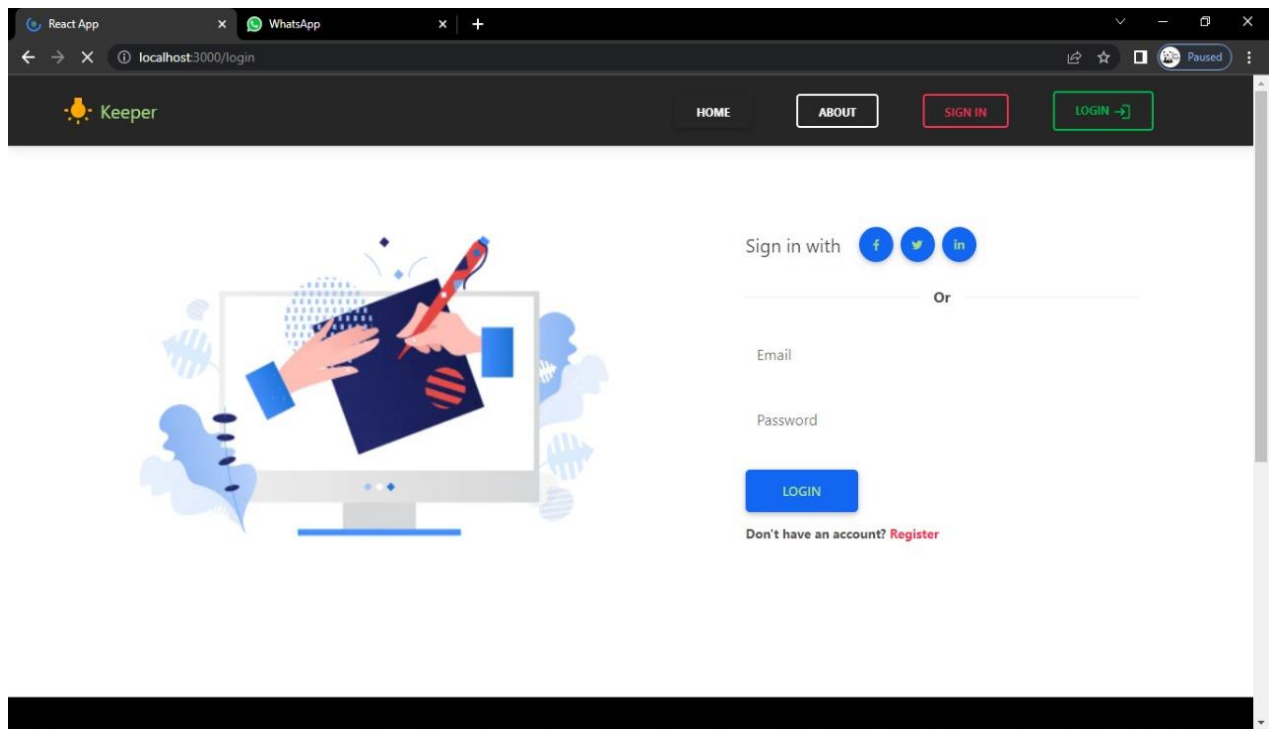
The screenshot displays a web browser window with two tabs: 'React App' and 'WhatsApp'. The address bar shows 'localhost:3000/register'. The website has a dark header with the 'Keeper' logo (a yellow sun icon) on the left and navigation links 'HOME', 'ABOUT', 'SIGN IN', and 'LOGIN' on the right. The 'SIGN IN' and 'LOGIN' links are highlighted with red and green borders, respectively.

The main content area is a white card titled 'Sign up'. It contains the following form elements:

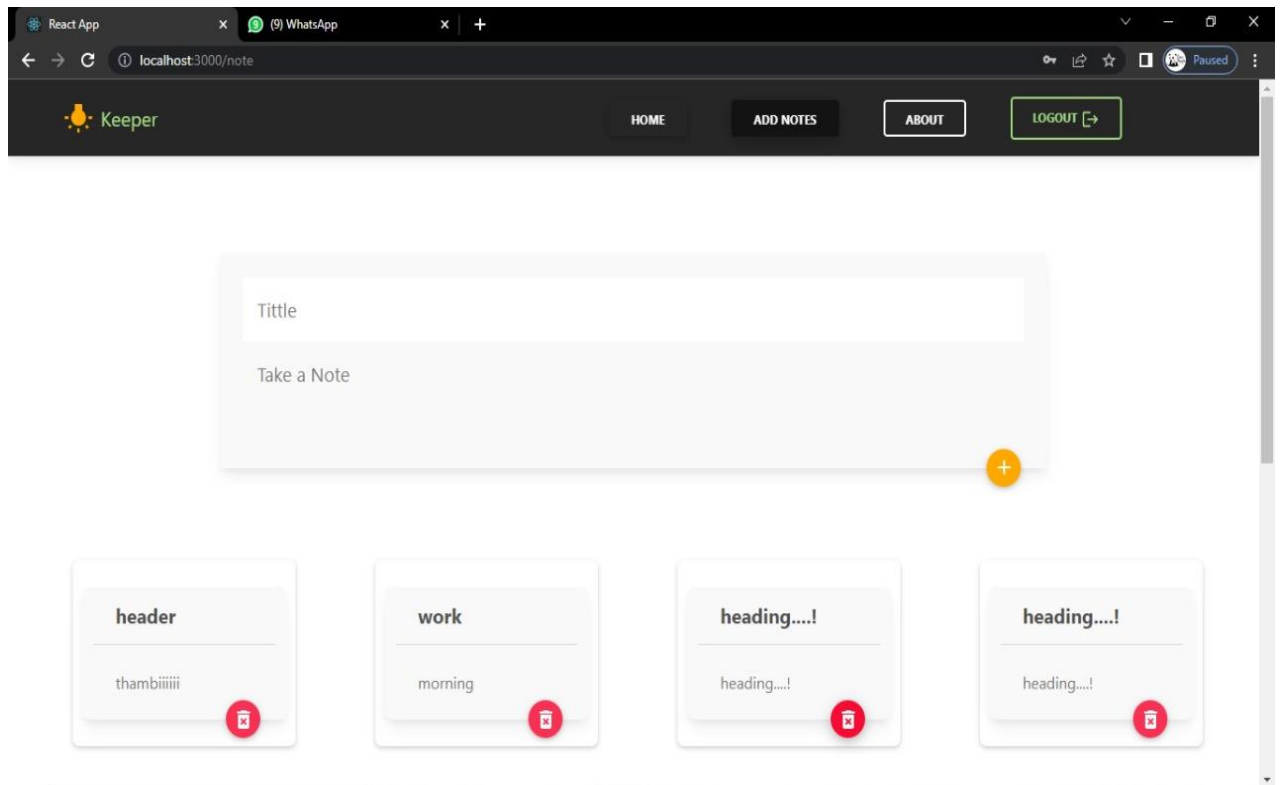
- A person icon followed by the text 'Your Name'.
- An envelope icon followed by the text 'Your Email'.
- A lock icon followed by the text 'Password'.
- A key icon followed by the text 'Repeat your password'.
- A checkbox followed by the text 'I agree all statements in [Terms of service](#)'.
- A blue 'REGISTER' button.

To the right of the form is a colorful illustration of a person with a blue face and yellow hair, wearing a green shirt and blue pants, carrying a blue briefcase. They are standing next to a large, stylized number '2' and a yellow star. The background of the illustration includes various geometric shapes like triangles and circles in blue, red, and yellow.

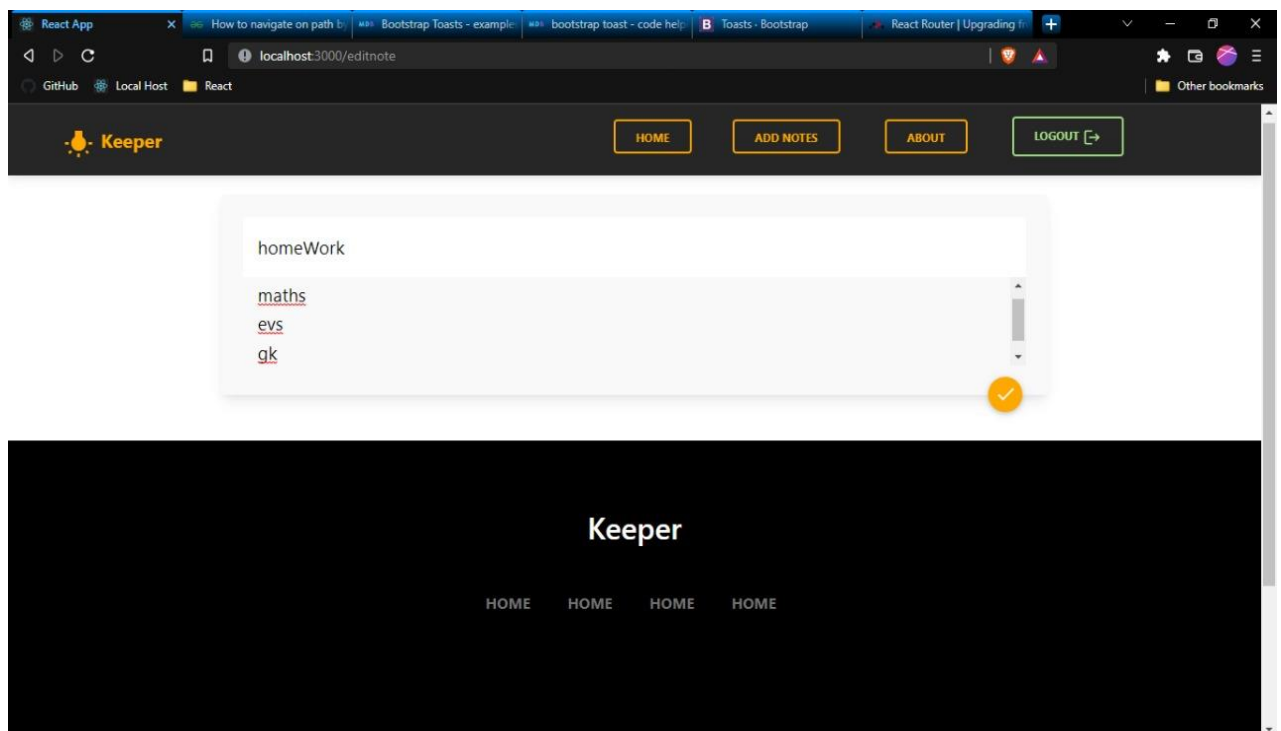
5.3.2 LOG IN PAGE



5.3.3 ADD NOTES PAGE



5.3.4 EDIT NOTES PAGE



SUMMARY AND CONCLUSIONS

Keep notes offers a set of powerful tools that make it an effective task management utility. It creates collaboration to-do lists to transcribing voice notes and saving bookmarks, the app does it all.

The best part about Keep notes is that all changes are synced automatically, giving you quick access to your notes across all your devices, and on the web.

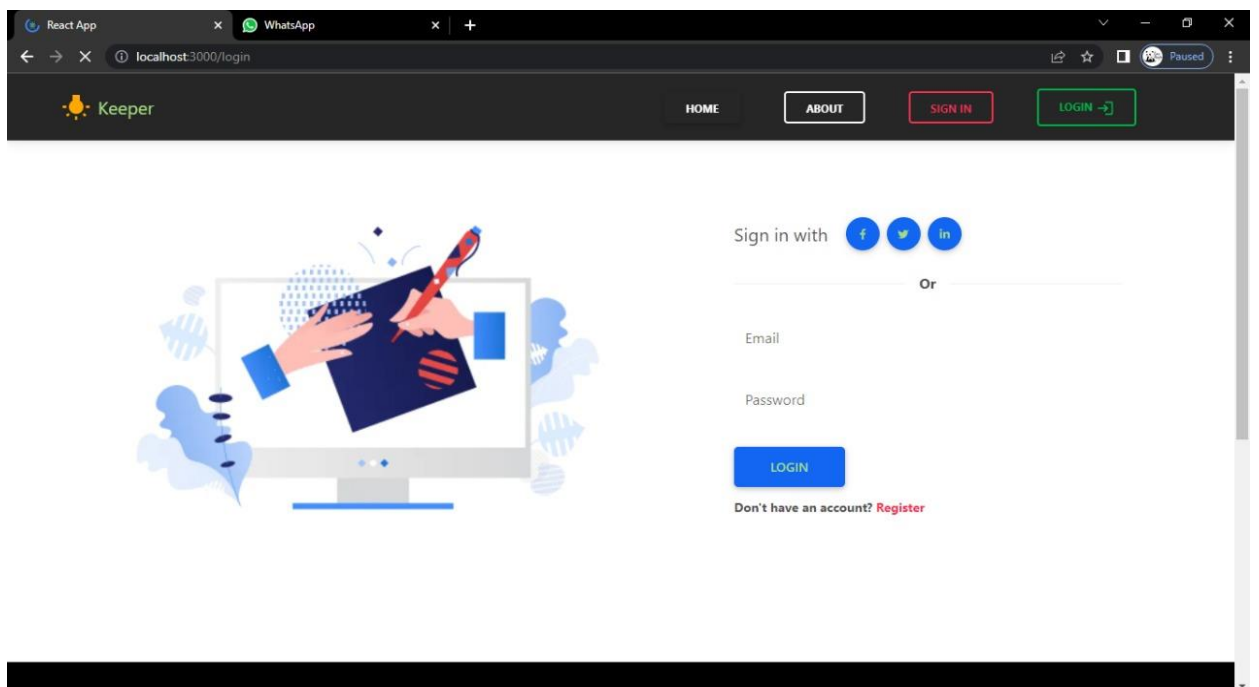
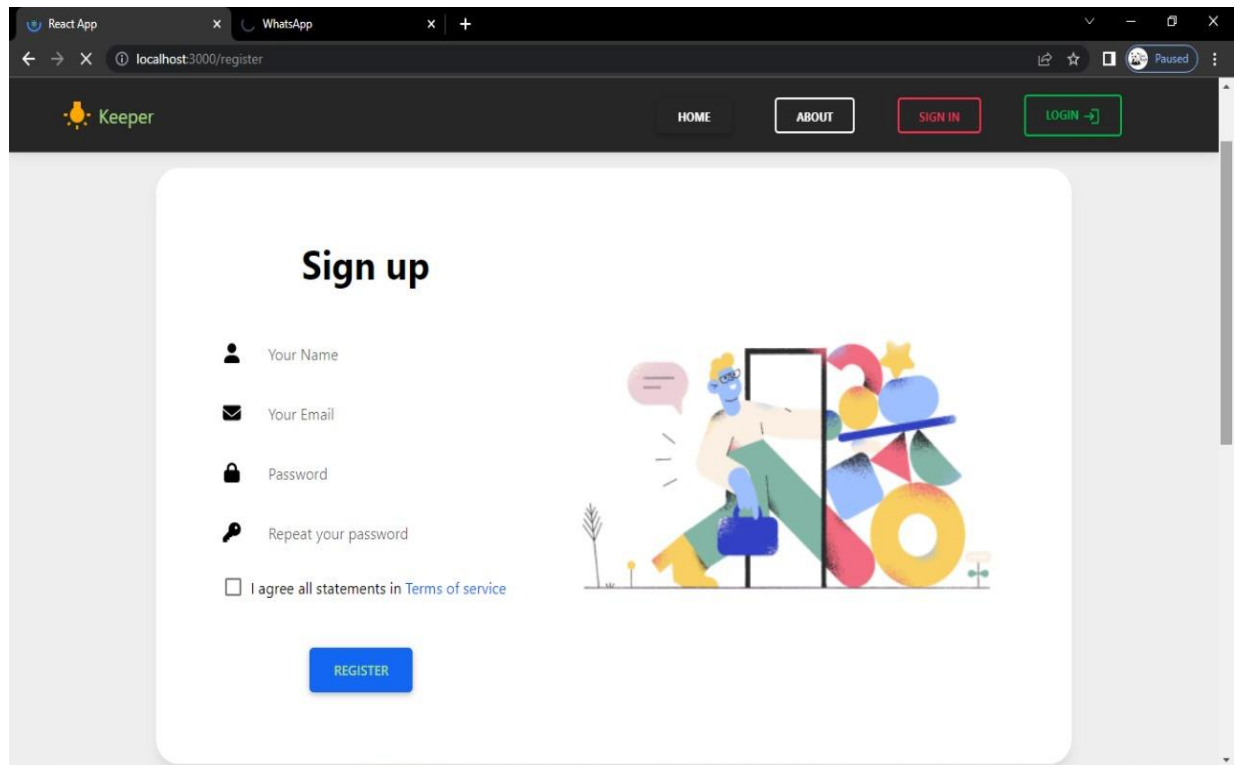
REFERENCES

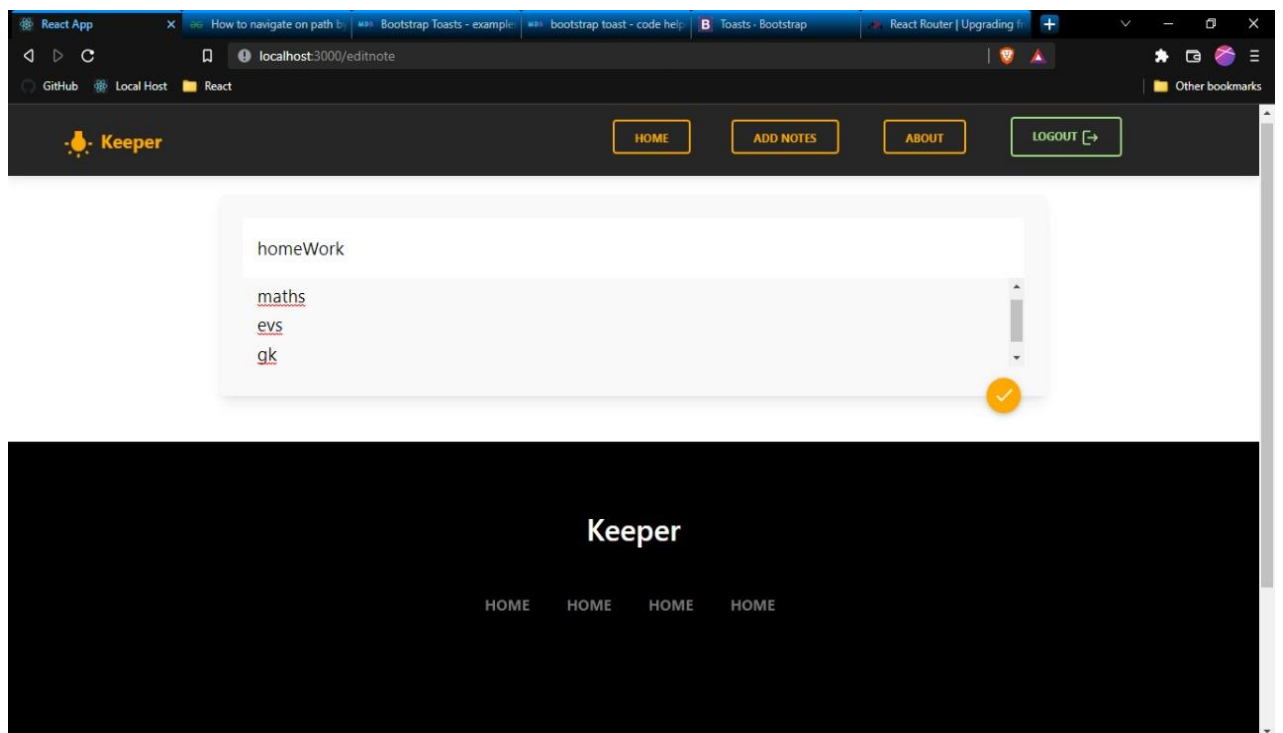
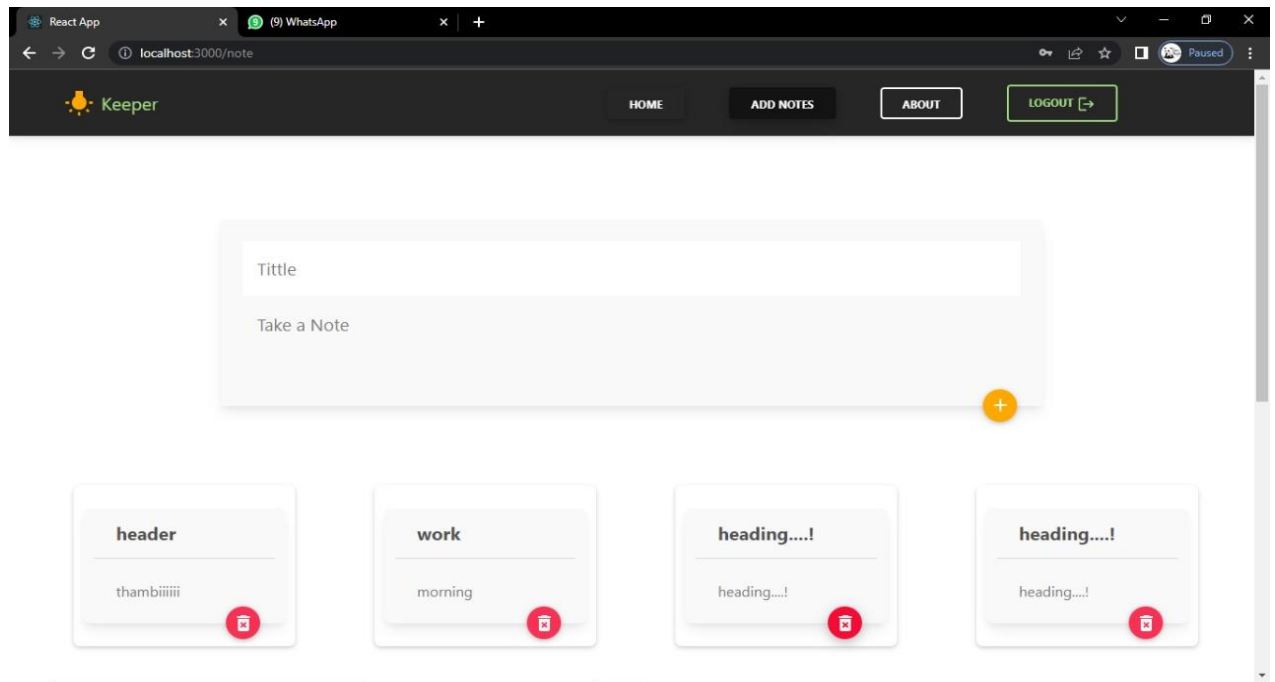
1. www.google.com
2. <https://developer.mozilla.org/en-US/>
3. www.pexels.com
5. www.wikipedia.org/angular
6. www.tutorialspointsg.com
7. www.javaworld.com
8. www.reactjs.com
9. www.reactjs.co

Project Link:

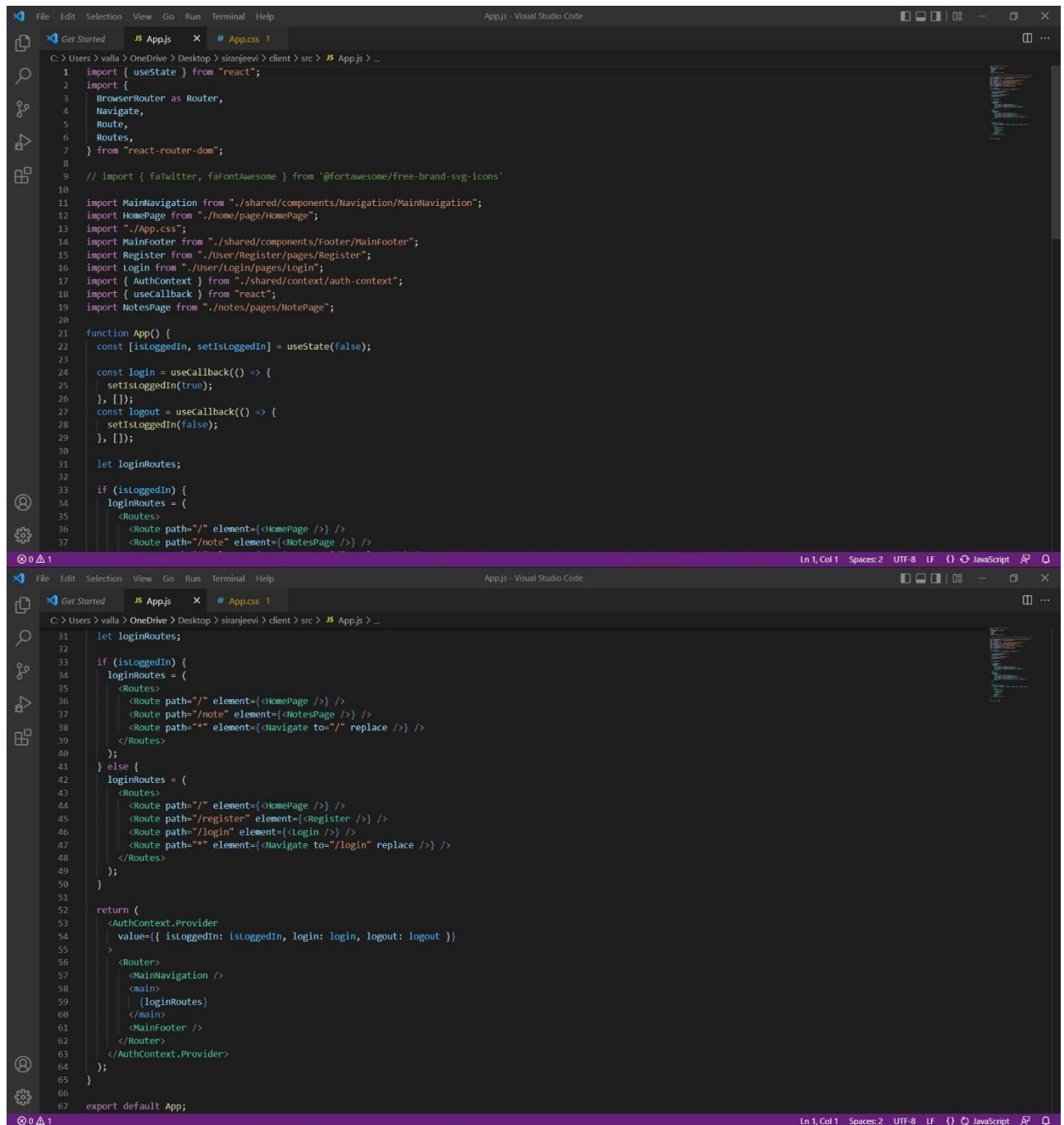
- 1.I host my app in herokuapp.com
- 2.My app link:<https://keepermyappproject123.herokuapp.com/>
- 3.I upload my project in github.
- 4.Github link:<https://github.com/TrlcKyMaN-VS/keepermyappproject123>

SAMPLE SCREENS

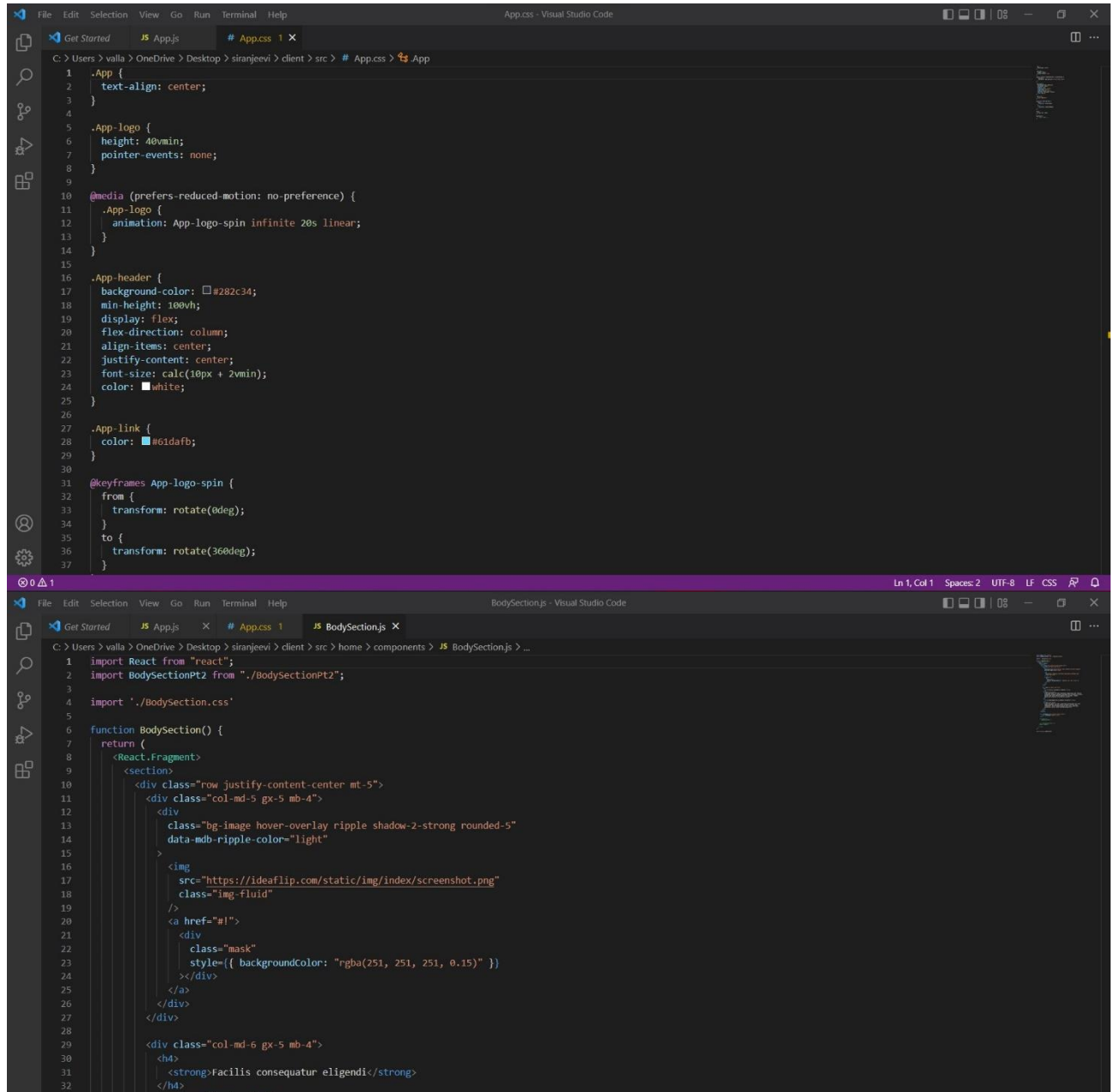


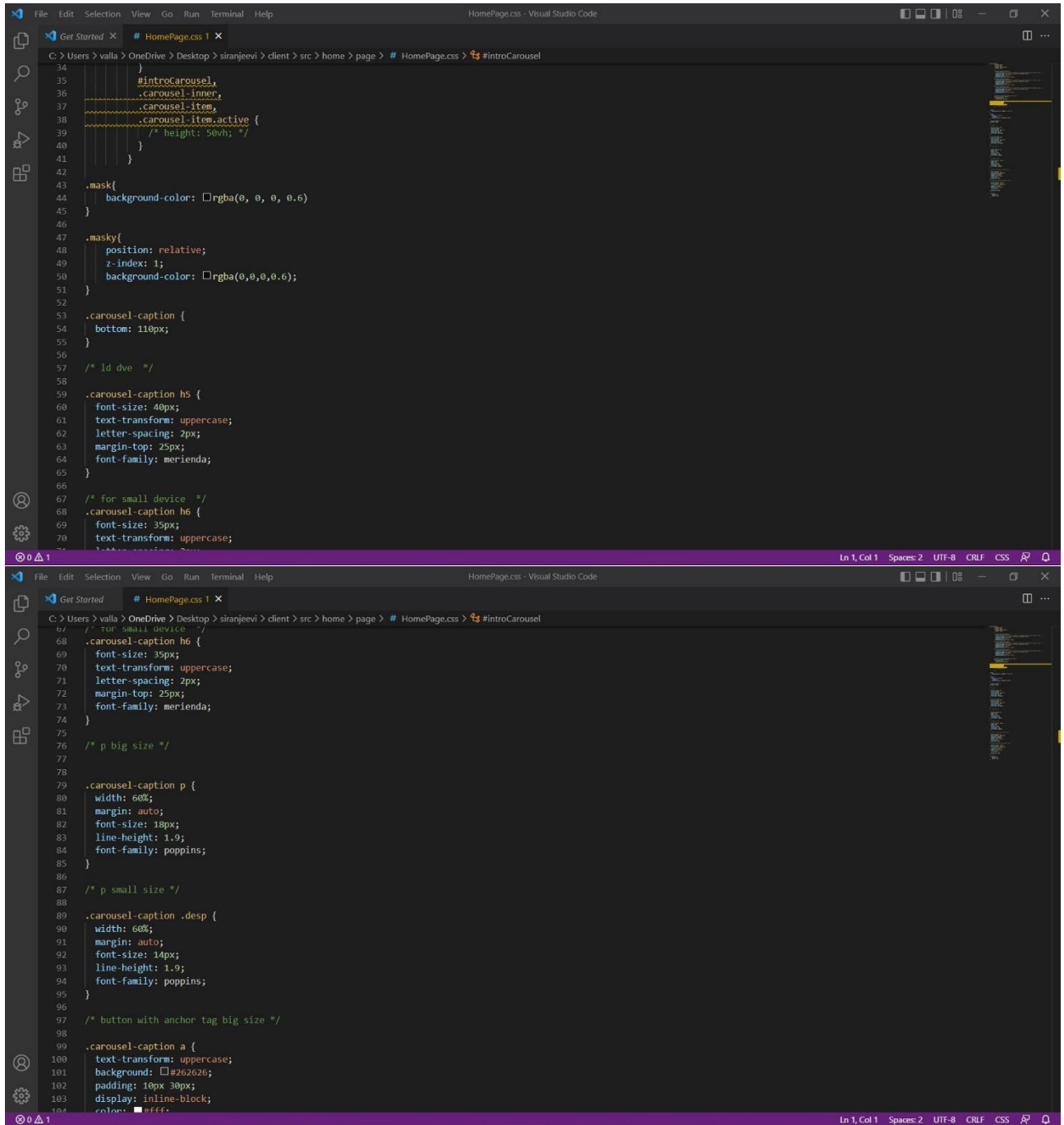


PROGRAM :

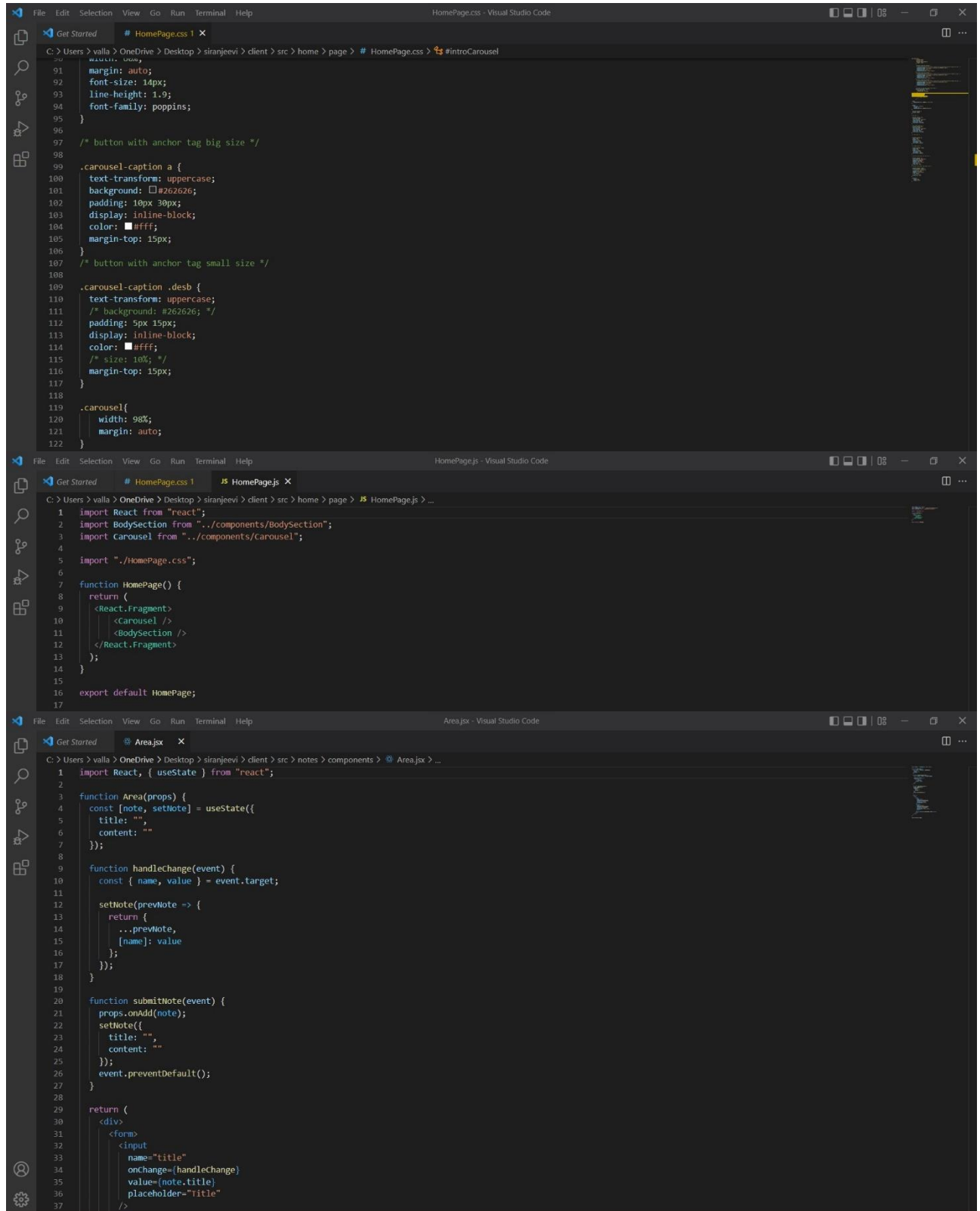


```
File Edit Selection View Go Run Terminal Help Appjs - Visual Studio Code
C:\Users\vala> OneDrive > Desktop > siranjeevi > client > src > JS Appjs > ...
1 import { useState } from "react";
2 import {
3   BrowserRouter as Router,
4   Navigate,
5   Route,
6   Routes,
7 } from "react-router-dom";
8
9 // import { faTwitter, faFontAwesome } from '@fortawesome/free-brand-svg-icons'
10
11 import MainNavigation from "../shared/components/Navigation/MainNavigation";
12 import HomePage from "../home/page/HomePage";
13 import "../App.css";
14 import MainFooter from "../shared/components/Footer/MainFooter";
15 import Register from "../User/Register/pages/Register";
16 import Login from "../User/Login/pages/Login";
17 import { AuthContext } from "../shared/context/auth-context";
18 import { useCallback } from "react";
19 import NotesPage from "../notes/pages/NotePage";
20
21 function App() {
22   const [isLoggedIn, setIsLoggedIn] = useState(false);
23
24   const login = useCallback(() => {
25     setIsLoggedIn(true);
26   }, []);
27   const logout = useCallback(() => {
28     setIsLoggedIn(false);
29   }, []);
30
31   let loginRoutes;
32
33   if (isLoggedIn) {
34     loginRoutes = (
35       <Routes>
36         <Route path="/" element=<HomePage /> />
37         <Route path="/note" element=<NotesPage /> />
38       </Routes>
39     );
40   } else {
41     loginRoutes = (
42       <Routes>
43         <Route path="/" element=<HomePage /> />
44         <Route path="/register" element=<Register /> />
45         <Route path="/login" element=<Login /> />
46         <Route path="*" element=<Navigate to="/login" replace /> />
47       </Routes>
48     );
49   }
50
51   return (
52     <AuthContext.Provider
53       value={{ isLoggedIn: isLoggedIn, login: login, logout: logout }}
54     >
55       <Router>
56         <MainNavigation />
57         <main>
58           {loginRoutes}
59         </main>
60         <MainFooter />
61       </Router>
62     </AuthContext.Provider>
63   );
64 }
65
66 export default App;
```



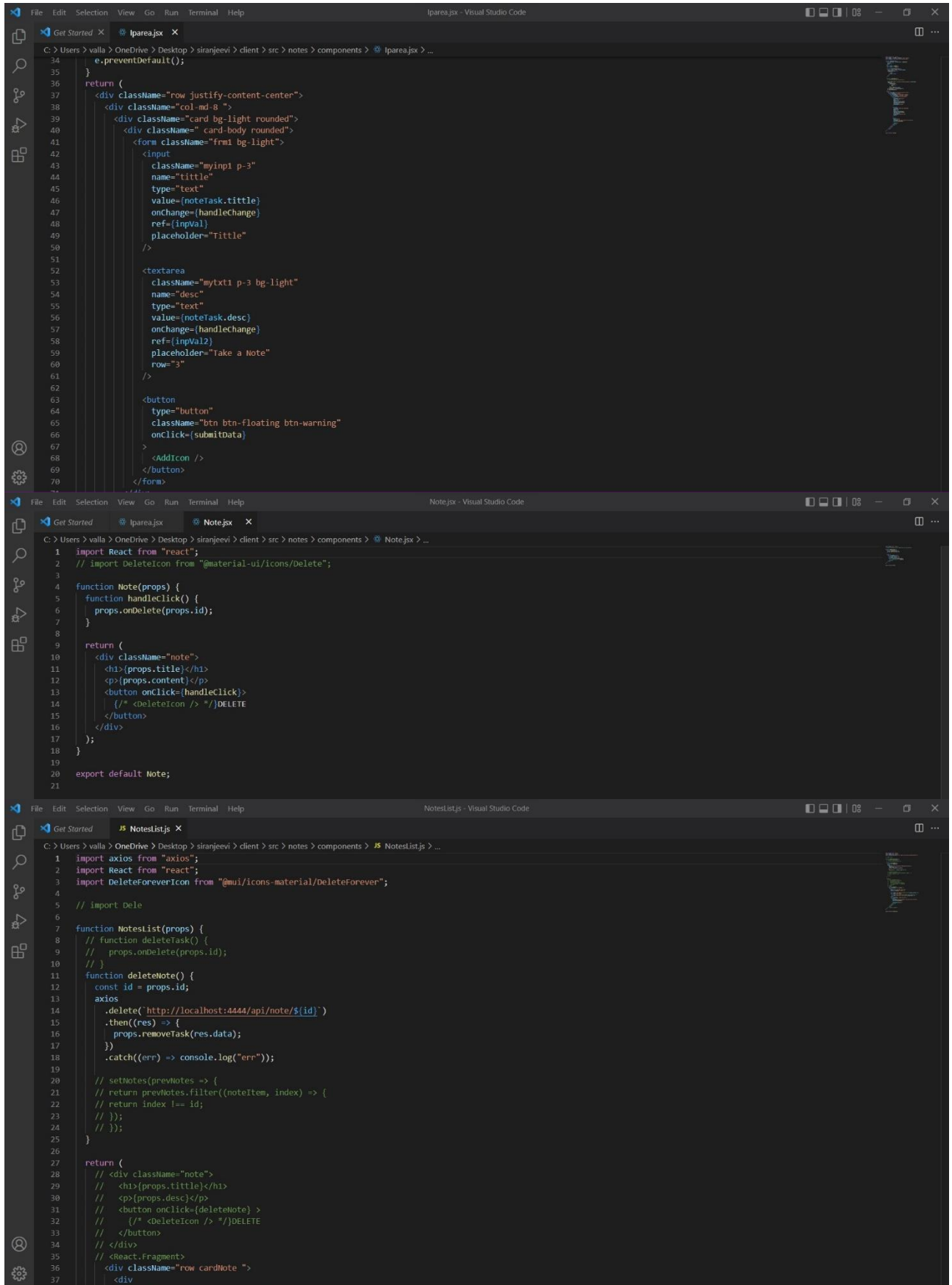


```
34 }
35 }
36 .introCarousel,
37 .carousel-inner,
38 .carousel-item,
39 .carousel-item.active {
40     /* height: 50vh; */
41 }
42
43 .mask {
44     background-color: rgba(0, 0, 0, 0.6);
45 }
46
47 .masky {
48     position: relative;
49     z-index: 1;
50     background-color: rgba(0, 0, 0, 0.6);
51 }
52
53 .carousel-caption {
54     bottom: 110px;
55 }
56
57 /* 1d dve */
58
59 .carousel-caption h5 {
60     font-size: 40px;
61     text-transform: uppercase;
62     letter-spacing: 2px;
63     margin-top: 25px;
64     font-family: merienda;
65 }
66
67 /* for small device */
68 .carousel-caption h6 {
69     font-size: 35px;
70     text-transform: uppercase;
71     letter-spacing: 2px;
72     margin-top: 25px;
73     font-family: merienda;
74 }
75
76 /* p big size */
77
78
79 .carousel-caption p {
80     width: 60%;
81     margin: auto;
82     font-size: 18px;
83     line-height: 1.9;
84     font-family: poppins;
85 }
86
87 /* p small size */
88
89 .carousel-caption .desp {
90     width: 60%;
91     margin: auto;
92     font-size: 14px;
93     line-height: 1.9;
94     font-family: poppins;
95 }
96
97 /* button with anchor tag big size */
98
99 .carousel-caption a {
100     text-transform: uppercase;
101     background: #262626;
102     padding: 10px 30px;
103     display: inline-block;
104     color: white;
```



```
File Edit Selection View Go Run Terminal Help Area.jsx - Visual Studio Code
C:\Users\vala> valla > OneDrive > Desktop > siranjeevi > client > src > notes > components > Area.jsx > ...
21 props.onAdd(note);
22 setNote({
23   title: "",
24   content: ""
25 });
26 event.preventDefault();
27 }
28
29 return (
30   <div>
31     <form>
32       <input
33         name="title"
34         onChange={handleChange}
35         value={note.title}
36         placeholder="title"
37       />
38       <textarea
39         name="content"
40         onChange={handleChange}
41         value={note.content}
42         placeholder="Take a note..."
43         rows="3"
44       />
45       <button onClick={submitNote}>Add</button>
46     </form>
47   </div>
48 );
49 }
50
51 export default Area;
52

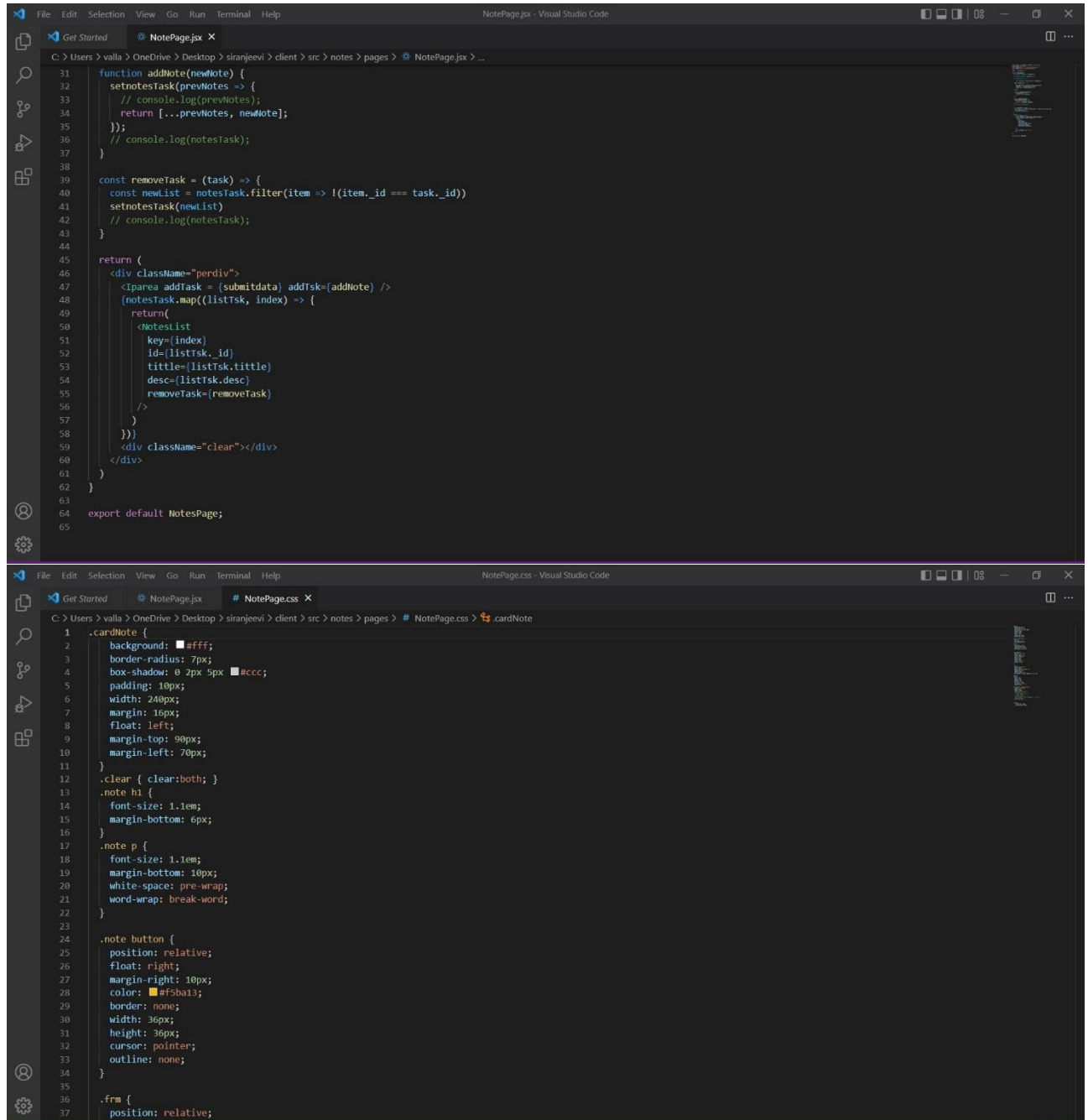
File Edit Selection View Go Run Terminal Help Iparea.jsx - Visual Studio Code
C:\Users\vala> valla > OneDrive > Desktop > siranjeevi > client > src > notes > components > Iparea.jsx > ...
1 import axios from "axios";
2 import React, { useRef, useState } from "react";
3 import AddIcon from '@mui/icons-material/Add';
4
5 function Iparea(props) {
6   const [noteTask, setNoteTask] = useState({
7     title: "",
8     desc: "",
9   });
10   const inpVal = useRef();
11   const inpVal2 = useRef();
12
13   function handleChange(event) {
14     const { name, value } = event.target;
15     setNoteTask((pv) => {
16       return {
17         ...pv,
18         [name]: value,
19       };
20     });
21   }
22
23   function submitData(e) {
24     // console.log(props.submitdata());
25
26     axios.post("http://localhost:4444/api/addnote", {noteTask})
27       .then(res => {
28         setNoteTask({
29           title: "",
30           desc: ""
31         });
32         props.addTask(res.data)
33       }).catch(err => console.log(err))
34     e.preventDefault();
35   }
36   return (
37     <div className="row justify-content-center">
```

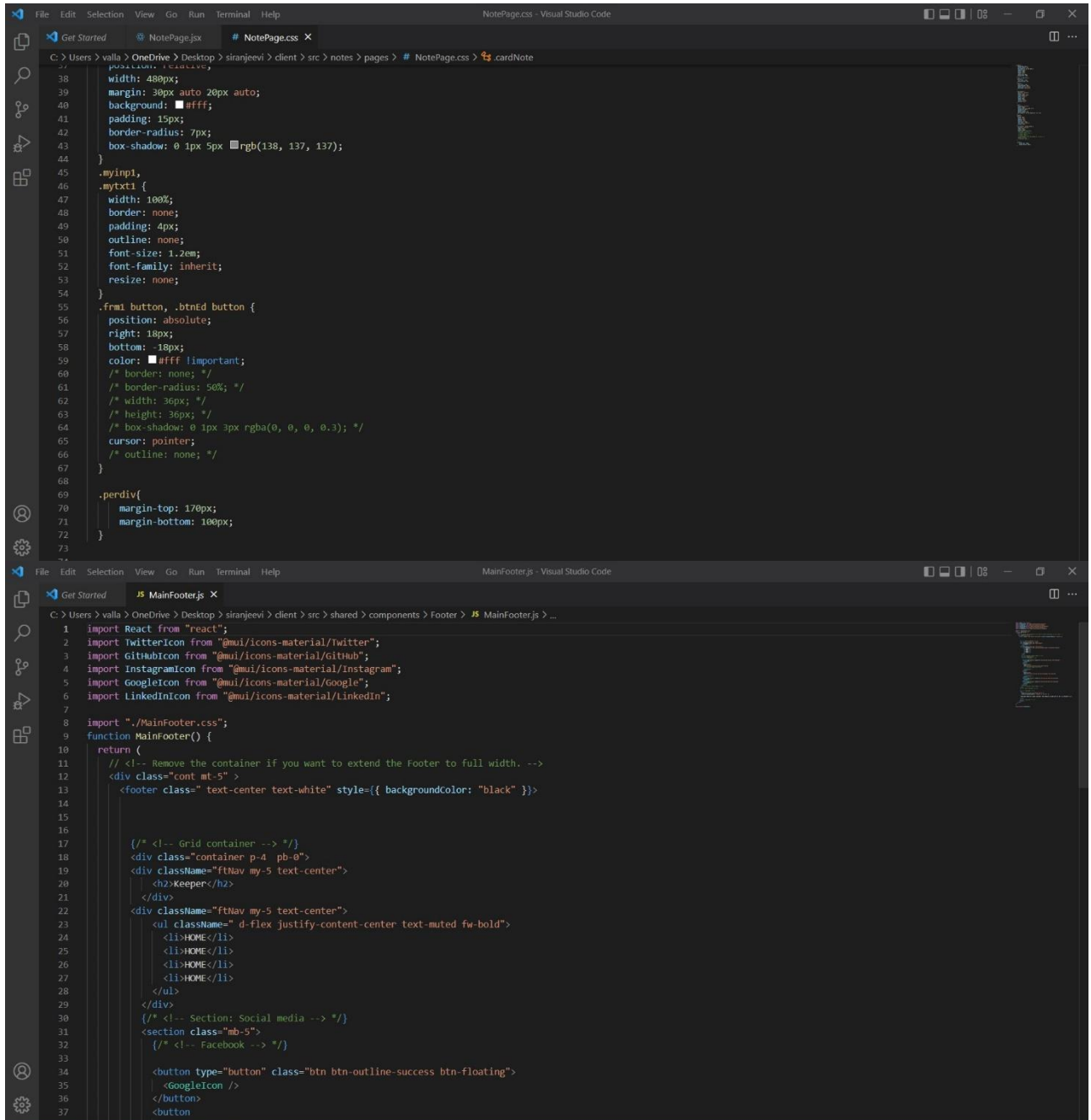


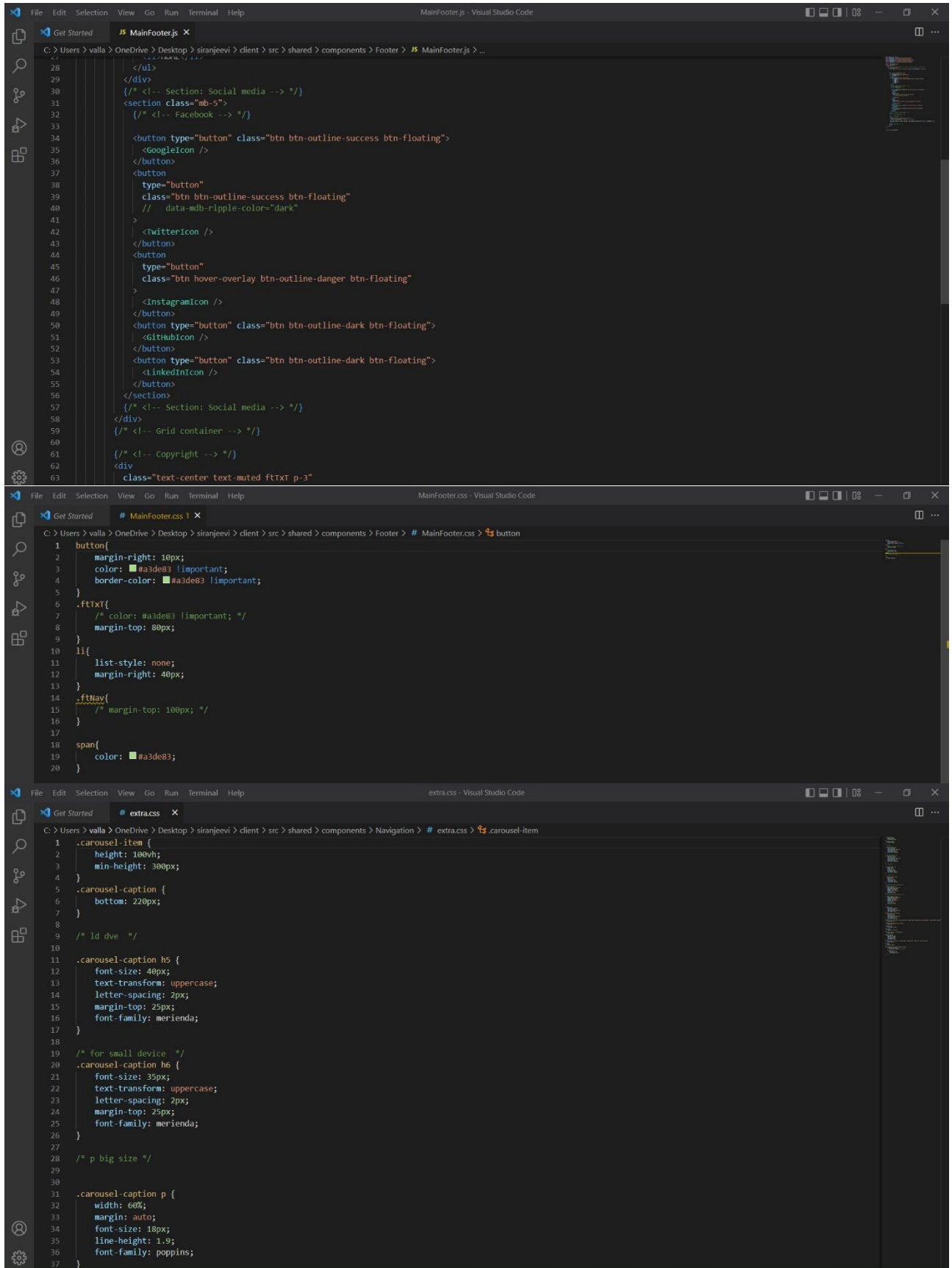
```
File Edit Selection View Go Run Terminal Help
NotesListjs - Visual Studio Code
Get Started JS NotesListjs
C:\Users\> valla > OneDrive > Desktop > siranjeevi > client > src > notes > components > JS NotesListjs > ...
24 // }
25 }
26
27 return (
28   // <div className="note">
29   // <h1>{props.title}</h1>
30   // <p>{props.desc}</p>
31   // <button onClick={deleteNote}>
32   //   /* <DeleteIcon /> */DELETE
33   // </button>
34   // </div>
35   // <React.Fragment>
36   <div className="row cardNote">
37     <div
38       className="card bg-light my-3 my-1"
39       style={{ maxWidth: "30rem" }}
40     >
41       <div class="card-header fw-bold fs-5">{props.title}</div>
42       <div class="card-body">
43         /* <h5 class="card-title">light card title</h5> */
44         <p class="card-text text-muted">{props.desc}</p>
45       </div>
46       <div className="btnEd text-end">
47         <button
48           className="btn btn-danger text-white btn-floating"
49           onClick={deleteNote}
50         >
51           <DeleteForeverIcon />
52         </button>
53       </div>
54     </div>
55   </div>
56   // </React.Fragment>
57 );
58 }
59
60 export default NotesList;

```

```
File Edit Selection View Go Run Terminal Help
NotePage.jsx - Visual Studio Code
Get Started NotePage.jsx
C:\Users\> valla > OneDrive > Desktop > siranjeevi > client > src > notes > pages > NotePage.jsx > ...
1 import React, { useEffect, useState } from "react";
2 import axios from "axios";
3 import NotesList from "../components/NotesList";
4 import Iparea from "../components/Iparea";
5
6 import './NotePage.css'
7
8 function NotesPage() {
9   const [notes, setNotes] = useState([]);
10
11   const [data, setData] = useState(null);
12
13   // const getText = () => {}
14
15   const [notesTask, setnotesTask] = useState([])
16
17   React.useEffect(() => {
18
19     axios.get("http://localhost:4444/api/addnote")
20       .then(res => setnotesTask(res.data))
21       .catch( err => console.log("err"))
22   },[]);
23
24   function submitdata(newTask) {
25     setnotesTask(pv => {
26       return [...pv, newTask]
27     })
28   }
29
30   function addNote(newNote) {
31     setnotesTask(prevNotes => {
32       // console.log(prevNotes);
33       return [...prevNotes, newNote];
34     });
35   }
36   // console.log(notesTask);
37 }
```







```
File Edit Selection View Go Run Terminal Help
extra.css - Visual Studio Code
C:\Users\va... > OneDrive > Desktop > siranjeevi > client > src > shared > components > Navigation > # extra.css > carousel-item

38
39 /* p small size */
40
41 .carousel-caption .desp {
42     width: 60%;
43     margin: auto;
44     font-size: 14px;
45     line-height: 1.9;
46     font-family: poppins;
47 }
48
49 /* button with anchor tag big size */
50
51 .carousel-caption a {
52     text-transform: uppercase;
53     background: #262626;
54     padding: 10px 30px;
55     display: inline-block;
56     color: #fff;
57     margin-top: 15px;
58 }
59 /* button with anchor tag small size */
60
61 .carousel-caption .desb {
62     text-transform: uppercase;
63     background: #262626;
64     padding: 5px 15px;
65     display: inline-block;
66     color: #fff;
67     /* size: 10%; */
68     margin-top: 15px;
69 }
70
71
72 .navbar-nav a {
73     font-family: poppins;
74
75
76
77
78 .navbar-light .navbar-brand {
79     color: #fff;
80     font-size: 25px;
81     text-transform: uppercase;
82     font-weight: bold;
83     letter-spacing: 2px;
84 }
85 .navbar-light .navbar-nav .active>.nav-link, .navbar-light .navbar-nav .nav-link.active, .navbar-light .navbar-nav .nav-link.show, .navbar-light .navbar-nav .show>.nav-li
86     color: #fff;
87 }
88 .navbar-light .navbar-nav .nav-link {
89     color: #fff;
90 }
91 .navbar-nav {
92     text-align: center;
93 }
94 .nav-link {
95     padding: .2rem 1rem;
96 }
97 .nav-link.active, .nav-link:focus {
98     color: #fff;
99 }
100 .navbar-toggler {
101     padding: 1px 5px;
102     font-size: 18px;
103     line-height: 0.3;
104     background: #fff;
105 }
```

The image displays two screenshots of a Visual Studio Code editor window, showing the implementation of a login form component. The editor is titled "LoginForm.js - Visual Studio Code".

The first screenshot shows the initial state of the component, including the Redux reducer and the initial state definition.

```
1 import React, { useContext, useReducer } from "react";
2 import axios from "axios";
3 import { AuthContext } from "../../shared/context/auth-context";
4
5 function loginReducer(state, action) {
6   switch (action.type) {
7     case "Field": {
8       return {
9         ...state,
10         [action.fieldName]: action.payload,
11       };
12     }
13     // case 'login': {
14     //   console.log(state);
15     // }
16   }
17 }
18
19 const initialState = {
20   username: "",
21   // email: "",
22   password: "",
23   // isLoading: false,
24   // error: "",
25   // isLoggedIn: false
26 };
27
28 function LoginForm() {
29   const auth = useContext(AuthContext);
30
31   const [state, dispatch] = useReducer(loginReducer, initialState);
32
33   const { username, email, password } = state;
34
35   const handleChange = (e) => {
36     dispatch({
37       type: "Field",
```

The second screenshot shows the continuation of the component, including the submit handler and the form rendering.

```
38     fieldName: e.target.name,
39     payload: e.target.value,
40   });
41   // console.log(e.target.name);
42 };
43
44 const submitHandler = (e) => {
45   e.preventDefault();
46
47   axios
48     .post("http://localhost:4444/api/login", { state })
49     .then((res) => {
50       console.log(res.data);
51       console.log("goy it..!");
52       if (res.data) {
53         localStorage.setItem("userId", res.data);
54         auth.login()
55       }
56     })
57     .catch((err) => {
58       console.log(err);
59     });
60
61   // dispatch({
62   //   type: 'login'
63   // })
64   console.log(state);
65   // console.log("state");
66 };
67
68 if (!!localStorage.userId) {
69   auth.login()
70 }
71
72 return (
73   <form onSubmit={submitHandler}>
```

```
File Edit Selection View Go Run Terminal Help LoginForm.js - Visual Studio Code
Get Started JS LoginForm.js X
C:\Users\vaia > OneDrive > Desktop > siranjeevi > client > src > User > Login > components > JS LoginForm.js > ...
114     id="form3Example4"
115     class="form-control form-control-lg"
116     placeholder="Enter password"
117     required
118     name="password"
119     onChange={handleChange}
120     value={password}
121   />
122   <label class="form-label" for="form3Example4">
123     Password
124   </label>
125 </div>
126
127 <div class="text-center text-lg-start mt-4 pt-2">
128   <button
129     type="submit"
130     class="btn btn-primary btn-lg"
131     style={{ paddingLeft: "2.5rem", paddingRight: "2.5rem" }}
132   >
133     Login
134   </button>
135   <p class="small fw-bold mt-2 pt-1 mb-0">
136     Don't have an account?{" " }
137     <a href="#" class="link-danger">
138       Register
139     </a>
140   </p>
141 </div>
142 </form>
143   };
144 }
145
146 export default LoginForm;
147
```