MRCET CAMPUS

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS INSTITUTION – UGC, GOVT. OF INDIA)

Department of CSE

(Emerging Technologies)

(DATA SCIENCE, CYBER SECURITY & IOT)

B.TECH(R-20 Regulation)

(IV YEAR – I SEM)

(2023-24)

FULL STACK DEVELOPMENT

(R20A0589)

LAB MANUAL

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12(B) of UGC ACT 1956

(Affiliated to JNTUH, Hyderabad, Approved by AICTE-Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified)

Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad–500100, Telangana State, India

Department of Computer Science and Engineering

EMERGING TECHNOLOGIES

FULL STACK DEVELOPMENT
(R20A0589)

LAB MANUAL

Prepared by

D KALPANA
M GAYATRI
V SUNEETHA

On

01.06.2023

Department of Computer Science and Engineering

EMERGING TECHNOLOGIES

Vision

- "To be at the forefront of Emerging Technologies and to evolve as a Centre of Excellence in Research, Learning and Consultancy to foster the students into globally competent professionals useful to the Society."

Mission

The department of CSE (Emerging Technologies) is committed to:

- To offer highest Professional and Academic Standards in terms of Personal growth and satisfaction.
- Make the society as the hub of emerging technologies and thereby capture opportunities in new age technologies.
- To create a benchmark in the areas of Research, Education and Public Outreach.
- To provide students a platform where independent learning and scientific study are encouraged with emphasis on latest engineering techniques.

QUALITY POLICY

- To pursue continual improvement of teaching learning process of Undergraduate and Post Graduate programs in Engineering & Management vigorously.
- To provide state of art infrastructure and expertise to impart the quality education and research environment to students for a complete learning experiences.
- Developing students with a disciplined and integrated personality.
- To offer quality relevant and cost effective programmes to produce engineers as per requirements of the industry need.

For more information: www.mrcet.ac.in

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

IV Year B.Tech. CSE- I Sem **L/T/P/C**

0-/3/-1.5

(R20A0589) FULL STACK DEVELOPMENT LAB

COURSE OBJECTIVES:

This course will enable the students:

1. Usage of various front and back end Tools

2. They can understand and create applications on their own

3. Demonstrate and Designing of Websites can be carried out.

4. Develop web based application using suitable client side and server side code.

5. Implement web based application using effective database access.

PROGRAMS:

Week-1. Write a program to create a simple webpage using HTML.

Week-2. Write a program to create a website using HTML CSS and JavaScript?

Week-3. Write a program to build a Chat module using HTML CSS and JavaScript?

Week-4. Write a program to create a simple calculator Application using React JS

Week-5. Write a program to create a voting application using React JS

Week-6. Write a program to create and Build a Password Strength Check using Jquery.

Week-7. Write a program to create and Build a star rating system using Jquery.

Week-8. Create a Simple Login form using React JS

Week-9. Create a blog using React JS Using the CMS users must be able to design a web page using the drag and drop method. Users should be able to add textual or media content into placeholders that are attached to locations on the web page using drag and drop method.

Week-10. Create a project on Grocery delivery application

Assume this project is for a huge online departmental store. Assume that they have a myriad of grocery items at their godown. All items must be listed on the website, along with their quantities and prices.Users must be able to sign up and purchase groceries.

The system should present him with delivery slot options, and the user must be able to choose his preferred slot. Users must then be taken to the payment page where he makes the payment with his favourite method.

Week-11. Connecting our TODO React js Project with Firebase

COURSE OUTCOMES:

Students will be able to understand

1. Usage of various front and back end Tools

2. They can understand and create applications on their own

3. Demonstrate and Designing of Websites can be carried out.

4. Develop web based application using suitable client side and server side

code.

5. Implement web based application using effective database access.

1. Lab Objectives:

This course will enable the students:

1. Systematic Introduction to Web Designing
2. Getting familiar with the front and back end tools
3. Creating applications using HTML,CSS,Javascript
4. Implementing various applications using JQuery
5. Creating Websites.

2. Lab Outcomes:

Students will be able to understand

1. Usage of various front and back end Tools
2. They can understand and create applications on their own
3. Demonstrate and Designing of Websites can be carried out.
4. Develop web based application using suitable client side and server side code.
5. Implement web based application using effective database access.

3. Introduction about lab

Minimum System requirements:

- Processors: Intel Atom® processor or Intel® Core™ i3 processor.
- Disk space: 1 GB.
- Operating systems: Windows* 7 or later, macOS, and Linux.
- Python* versions: 2.7.X, 3.6.X.,3.8.X

About lab:

A full-stack developer helps build and maintain both the front-end and the back-end of a website. Learn about full-stack developer skills, salary, and how you can become one.

A full-stack developer is a developer or engineer who can build both the front end and the back end of a website. The front end (the parts of a website a user sees and interacts with) and the back end (the behind-the-scenes data storage and processing) require different skill sets. Since full-stack developers are involved with all aspects of the development process, they must have expertise in both.

Full-stack developers design and create websites and applications for various platforms.

A full-stack developer's job description might include the following:

- Develop and maintain web services and interfaces
- Contribute to front-end and back-end development processes
- Build new product features or APIs
- Perform tests, troubleshoot software, and fix bugs
- Collaborate with other departments on projects and sprints

4. Guidelines to students

Students are required to carry their lab observation book and record book with completed experiments while entering the lab.

➢ Students must use the equipment with care. Any damage is caused student is punishable

➢ Students are not allowed to use their cell phones/pen drives/ CDs in labs.

➢ Students need to be maintain proper dress code along with ID Card

➢ Students are supposed to occupy the computers allotted to them and are not supposed to talk or make noise in the lab. Students, after completion of each experiment they need to be updated in observation notes and same to be updated in the record.

➢ Lab records need to be submitted after completion of experiment and get it corrected with the concerned lab faculty.

➢ If a student is absent for any lab, they need to be completed the same experiment in the free time before attending next lab.

Instructions to maintain the record

➢ Before start of the first lab they have to buy the record and bring the record to the lab.

➢ Regularly (Weekly) update the record after completion of the experiment and get it corrected with concerned lab in-charge for continuous evaluation.

➢ In case the record is lost inform the same day to the faculty in charge and get the new record within 2 days the record has to be submitted and get it corrected by the faculty.

➢ If record is not submitted in time or record is not written properly, the evaluation marks (5M) will be deducted.

General laboratory instructions

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.

2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.

3. Student should enter into the laboratory with:
a. Laboratory Record updated up to the last session experiments and exercise programs with all the details (Problem statement, Aim, Program, Expected Output, etc.,).
b. Proper Dress code and Identity card.

4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.

5. Execute your task in the laboratory.

6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.

7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.

8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.

9. Students must take the permission of the faculty in case of any urgency to go out; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.

10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

INDEX

Week-1. Write a program to create a simple webpage using HTML.

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Week-2. Write a program to create a website using HTML CSS and JavaScript?

1. Create a form and validate the contents of the form using Javascript.

AIM: To create a form and validate the contents of the form using Javascript.

Program:

Index.html

```
<!DOCTYPE html>
<html>
<head>
<script>
function validateForm() {
  let x = document.forms["myForm"]["fname"].value;
  if (x == "") {
    alert("Name must be filled out");
    return false;
  }
}
</script>
</head>
<body>
<h2>JavaScript Validation</h2>
<form name="myForm" onsubmit="return validateForm()"
method="post">
  Name: <input type="text" name="fname">
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

OUTPUT:

Week-3. Write a program to build a Chat module using HTML CSS and JavaScript?

```
<div class="chat-box">
  <header>
    <h1>André F. Martins</h1>
    <span style="flex:1"></span>
    <span class="close-button">X</span>
  </header>
  <section id="message-list">
  </section>
  <footer>
    <input id="message-input" type="text" placeholder="Type a message...">
  </footer>
</div>

box-sizing: border-box

::selection
  background: #fd0
  color: #222

br
  font-size: 0

body
  font-family: "Karla", sans-serif
  margin: 0
  display: flex
  align-items: center
  justify-content: center
  width: 100vw
  height: 100vh
  font-size: 16px
  background-color: #ddd

.chat-box
  width: 300px
  height: 400px
  background-color: #eee
  border-radius: 16px 16px 0 0
  overflow: hidden
  border: 2px #45f solid

  header
    background-color: #45f
    color: #fff
```

```
  display: flex
  padding: 0 16px

  > *
    line-height: 40px

  h1
    margin: 0
    font-size: 1.1em

  .close-button
    cursor: pointer
    user-select: none

section
  overflow-y: scroll
  overflow-x: hidden
  height: calc(100% - 40px - 40px)
  padding: 12px

  div
    margin: 12px 0

  span
    display: inline-block
    padding: 6px 8px
    margin: 1px 0
    max-width: 90%
    word-wrap: break-word

  div.me
    text-align: left
    span
      background-color: #68c
      border-radius: 4px 16px 16px 4px
    span:first-of-type
      border-top-left-radius: 16px
    span:last-of-type
      border-bottom-left-radius: 16px


  div.not-me
    text-align: right
    span
      background-color: #bcf
      border-radius: 16px 4px 4px 16px
```

```
    span:first-of-type
      border-top-right-radius: 16px
    span:last-of-type
      border-bottom-right-radius: 16px

  footer
    input
      font-family: "Karla", sans-serif
      outline: 0
      padding: 0 12px
      height: 40px
      width: 100%
      font-size: 1em

function sendMessage(message, itsMe) { // ...Mario
  var messageList = document.getElementById("message-list");

  var scrollToBottom = (messageList.scrollHeight - messageList.scrollTop -
messageList.clientHeight < 80);

  var lastMessage = messageList.children[messageList.children.length-1];

  var newMessage = document.createElement("span");
  newMessage.innerHTML = message;

  var className;

  if(itsMe)
  {
    className = "me";
    scrollToBottom = true;
  }
  else
  {
    className = "not-me";
  }

  if(lastMessage && lastMessage.classList.contains(className))
  {
    lastMessage.appendChild(document.createElement("br"));
    lastMessage.appendChild(newMessage);
  }
  else
  {
    var messageBlock = document.createElement("div");
    messageBlock.classList.add(className);
```

```
    messageBlock.appendChild(newMessage);
    messageList.appendChild(messageBlock);
  }

  if(scrollToBottom)
    messageList.scrollTop = messageList.scrollHeight;
}

var message = document.getElementById("message-input");
message.addEventListener("keypress", function(event) {
  var key = event.which || event.keyCode;
  if(key === 13 && this.value.trim() !== "")
  {
    sendMessage(this.value, true);
    this.value = "";
  }
});

sendMessage("Hello!", true);
sendMessage("Hey!", false);
sendMessage("How are you doing?", false);
sendMessage("I'm doing well.", true);
sendMessage("What about you?", true);
sendMessage("Good", false);

function sendRandomMessages()
{
  // http://www.conversationstarters.com/generator.php
  var messageList = [
    "What is your biggest concern about the future?",
    "What is your biggest fear?",
    "At what age would you consider someone to be old?",
    "What is your favorite home cooked dish?",
    "What is the biggest priority in your life right now?",
    "Where did you go on your last vacation?",
    "What was the biggest life change you've gone through?",
    "Do you have any siblings?",
    "What's your family like?",
    "If you knew that you only had a year left to live, what would you do?",
    "What do you usually eat in the morning?",
    "What's in your fridge?",
    "Do you recycle?",
    "What are some things that you should not not say during a job interview?",
    "If you could go back in time and change one thing, what would that be?",
    "What do you wear to sleep?",
    "What is the last thing you do before you go to sleep?",
```

```
    "If you could only eat one thing for the rest of your life, what would it be?",
    "Would you rather be homeless for a year or be in jail for a year?",
    "What do you do for fun?"
  ];
  var choosenMessage = messageList[Math.floor(Math.random() * messageList.length)];

  sendMessage(choosenMessage, false);

  setTimeout(sendRandomMessages, Math.random()*10000 + 7000);
}

sendRandomMessages();
```

Exercise Program: How To Create Chat Messages

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  margin: 0 auto;
  max-width: 800px;
  padding: 0 20px;
}

.container {
  border: 2px solid #dedede;
  background-color: #f1f1f1;
  border-radius: 5px;
  padding: 10px;
  margin: 10px 0;
}

.darker {
  border-color: #ccc;
  background-color: #ddd;
}

.container::after {
  content: "";
  clear: both;
  display: table;
}

.container img {
  float: left;
  max-width: 60px;
  width: 100%;
  margin-right: 20px;
  border-radius: 50%;
}

.container img.right {
  float: right;
  margin-left: 20px;
  margin-right:0;
}
```

```
.time-right {
  float: right;
  color: #aaa;
}

.time-left {
  float: left;
  color: #999;
}
</style>
</head>
<body>

<h2>Chat Messages</h2>

<div class="container">
  <img src="/w3images/bandmember.jpg" alt="Avatar" style="width:100%;">
  <p>Hello. How are you today?</p>
  <span class="time-right">11:00</span>
</div>

<div class="container darker">
  <img src="/w3images/avatar_g2.jpg" alt="Avatar" class="right" style="width:100%;">
  <p>Hey! I'm fine. Thanks for asking!</p>
  <span class="time-left">11:01</span>
</div>

<div class="container">
  <img src="/w3images/bandmember.jpg" alt="Avatar" style="width:100%;">
  <p>Sweet! So, what do you wanna do today?</p>
  <span class="time-right">11:02</span>
</div>

<div class="container darker">
  <img src="/w3images/avatar_g2.jpg" alt="Avatar" class="right" style="width:100%;">
  <p>Nah, I dunno. Play soccer.. or learn more coding perhaps?</p>
  <span class="time-left">11:05</span>
</div>

</body>
</html>
```

Week-4. Write a program to create a simple calculator Application using React JS

```
class App extends Component {
constructor() {
super()
this.state = { operations: [] }
}
.....
}
render() {
    return (
<div className="App">
<Display data={this.state.operations} />
<Buttons>
<Button onClick={this.handleClick} label="C" value="clear" />
<Button onClick={this.handleClick} label="7" value="7" />
<Button onClick={this.handleClick} label="4" value="4" />
<Button onClick={this.handleClick} label="1" value="1" />
<Button onClick={this.handleClick} label="0" value="0" /><Button
onClick={this.handleClick} label="/" value="/" />
<Button onClick={this.handleClick} label="8" value="8" />
<Button onClick={this.handleClick} label="5" value="5" />
<Button onClick={this.handleClick} label="2" value="2" />
<Button onClick={this.handleClick} label="." value="." /><Button
onClick={this.handleClick} label="x" value="*" />
<Button onClick={this.handleClick} label="9" value="9" />
<Button onClick={this.handleClick} label="6" value="6" />
<Button onClick={this.handleClick} label="3" value="3" />
<Button label="" value="null" /><Button onClick={this.handleClick}
label="-" value="-" />
<Button onClick={this.handleClick} label="+" size="2" value="+" />
<Button onClick={this.handleClick} label="=" size="2" value="equal"
/>
</Buttons>
</div>
    )
class Buttons extends Component {
render() {
return <div className="Buttons"> {this.props.children} </div>
}
}class Button extends Component {
  render() {
    return (
<div
onClick={this.props.onClick}
className="Button"
        data-size={this.props.size}
        data-value={this.props.value}  >
```

```
        {this.props.label}
</div>
    )
  }
}
class Display extends Component {
  render() {
    const string = this.props.data.join('')
    return <div className="Display"> {string} </div>
  }
}
handleClick = e => {
    const value = e.target.getAttribute('data-value')
    switch (value) {
      case 'clear':
this.setState({
          operations: [],
        })
        break
      case 'equal':
this.calculateOperations()
        break
      default:
        const newOperations = update(this.state.operations, {
          $push: [value],
        })
this.setState({
          operations: newOperations,
        })
        break
    }


calculateOperations = () => {
    let result = this.state.operations.join('')
    if (result) {
      result = math.eval(result)
      result = math.format(result, { precision: 14 })
      result = String(result)
this.setState({
        operations: [result],
      })
    }
  }
```

Week-5. Write a program to create a voting application using React JS

```
CREATE
ORREPLACEVIEW"public"."poll_results"AS
SELECT
  poll.id ASpoll_id,
o.option_id,
count(*)AS votes
FROM
(
(
SELECT
vote.option_id,
option.poll_id,
option.text
FROM
(
          vote
LEFTJOINoptionON((option.id =vote.option_id))
)
) o
LEFTJOIN poll ON((poll.id =o.poll_id))
)
GROUPBY
poll.question,
o.option_id,
  poll.id;
CREATE
ORREPLACEVIEW"public"."online_users"AS
SELECT
count(*)AS count
FROM
"user"
WHERE
(
"user".last_seen_at>(now()-'00:00:15' :: interval)
);
import{ApolloClient,HttpLink,InMemoryCache, split }from"@apollo/client";
import{GraphQLWsLink}from'@apollo/client/link/subscriptions';
import{createClient}from"graphql-ws";
import{getMainDefinition}from"@apollo/client/utilities";
constGRAPHQL_ENDPOINT="realtime-poll-example.hasura.app";

constscheme=(proto)=>
window.location.protocol==="https:"?`${proto}s`: proto;

constwsURI=`${scheme("ws")}://${GRAPHQL_ENDPOINT}/v1/graphql`;
consthttpURL=`${scheme("https")}://${GRAPHQL_ENDPOINT}/v1/graphql`;
constsplitter=({ query })=>{
const{ kind, operation }=getMainDefinition(query)||{};
constisSubscription=
    kind ==="OperationDefinition"&& operation ==="subscription";
returnisSubscription;
};
```

```
const cache =newInMemoryCache();
const options ={ reconnect:true};

constwsLink=newGraphQLWsLink(createClient({ url:wsURI,connectionParams:{ options }}));
consthttpLink=newHttpLink({uri:httpURL});
const link =split(splitter,wsLink,httpLink);
const client =newApolloClient({ link, cache });
```

Week-6. Write a program to create and Build a Password Strength Check using Jquery.

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<script src="CheckPasswordStrength.js"></script>
<link href="CheckPasswordStrength.css" rel="stylesheet" />
$(document).ready(function () {
    $('#txtPassword').keyup(function () {
        $('#strengthMessage').html(checkStrength($('#txtPassword').val()))
    })
    function checkStrength(password) {
        var strength = 0
        if (password.length < 6) {
            $('#strengthMessage').removeClass()
            $('#strengthMessage').addClass('Short')
            return 'Too short'
        }
        if (password.length > 7) strength += 1
        // If password contains both lower and uppercase characters, increase strength value.
        if (password.match(/([a-z].*[A-Z])|([A-Z].*[a-z])/))
          strength += 1
        // If it has numbers and characters, increase strength value.
        if (password.match(/([a-zA-Z])/) && password.match(/([0-9])/))
          strength += 1
        // If it has one special character, increase strength value.
        if (password.match(/([!,%,&,@,#,$,^,*,?,_,~])/)) strength += 1

        // If it has two special characters, increase strength value.
        if (password.match(/(.*[!,%,&,@,#,$,^,*,?,_,~].*[!,%,&,@,#,$,^,*,?,_,~])/))
          strength += 1
        // Calculated strength value, we can return messages
        // If value is less than 2
        if (strength < 2)
        {
            $('#strengthMessage').removeClass()
            $('#strengthMessage').addClass('Weak')
            return 'Weak'
        }
```

```
        else if (strength == 2) {
                $('#strengthMessage').removeClass()
                $('#strengthMessage').addClass('Good')
                return 'Good'
            } else {
                $('#strengthMessage').removeClass()
                $('#strengthMessage').addClass('Strong')
                return 'Strong'
            } }
});
```

```
.Short {
    width: 100%;
    background-color: #dc3545;
    margin-top: 5px;
    height: 3px;
    color: #dc3545;
    font-weight: 500;
    font-size: 12px;
}
.Weak {
    width: 100%;
    background-color: #ffc107;
    margin-top: 5px;
    height: 3px;
    color: #ffc107;
    font-weight: 500;
    font-size: 12px;
}
.Good {
    width: 100%;
    background-color: #28a745;
    margin-top: 5px;
    height: 3px;
    color: #28a745;
    font-weight: 500;
    font-size: 12px;
}
.Strong {
    width: 100%;
    background-color: #d39e00;
    margin-top: 5px;
    height: 3px;
    color: #d39e00;
    font-weight: 500;
    font-size: 12px;
}
```

```
<body>
    <form id="form1" runat="server">
        <div class="container py-3">
            <h4 class="text-center text-
uppercase">How to check password strength in jquery</h4>
            <div class="row">
                <div class="col-md-12">
                    <div class="row">
                        <div class="col-md-6 mx-auto">
                            <div class="card border-secondary">
                                <div class="card-header">
                                 <h3 class="mb-0 my-2">Sign Up</h3>
                                </div>
                                <div class="card-body">
                                    <div class="form-group">
                                        <label>Name</label>
                                        <div class="input-group">
                                  <div class="input-group-prepend">
   <span class="input-group-text"><i class="fa fa-user"></i></span>
              </div>
  <asp:TextBox ID="txtFirstName" runat="server" CssClass="form-
control"></asp:TextBox>
                                        </div>
                                    </div>
                                    <div class="form-group">
                                        <label>Phone Number</label>
                                        <div class="input-group">
                                    <div class="input-group-prepend">
 <span class="input-group-text"><i class="fa fa-phone"></i></span>
                                            </div>
 <asp:TextBox ID="txtPhoneNumber" runat="server" CssClass="form-
control"> </asp:TextBox>
                                        </div>
                                    </div>
                                    <div class="form-group">
                                        <label>Email</label>
                                        <div class="input-group">
                                   <div class="input-group-prepend">
  <span class="input-group-text"><i class="fa fa-envelope">
</i></span>
                                            </div>
 <asp:TextBox ID="txtEmail" runat="server" CssClass="form-control">
</asp:TextBox>
                                        </div>
                                    </div>
                                    <div class="form-group">
```

```
                                        <label>Password</label>
                                        <div class="input-group">
                                            <div class="input-group-
prepend">
    <span class="input-group-text"><i class="fa fa-lock">
</i></span>
                                        </div>
     <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
 CssClass="form-control"></asp:TextBox>
                                        </div>
                                    <div id="strengthMessage"></div>

                                    </div>
                                    <div class="form-group">
                                    <label>Confirm Password</label>
                                        <div class="input-group">
                                <div class="input-group-prepend">
         <span class="input-group-text"><i class="fa fa-lock">
</i></span>
                                        </div>
                  asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password" CssClass="form-control"></asp:TextBox>
                                        </div>
                                    </div>
                                    <div class="form-group">
<button type="submit" class="btn btn-success float-right rounded-0">
Register</button>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </form>
</body>
```

Week-7. Write a program to create and Build a star rating system using Jquery.

```
$(document).ready(function() {
  $("#st1").click(function() {
     $(".fa-star").css("color", "black");
     $("#st1").css("color", "yellow");

  });
        <!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 = "https://cdnjs.cloudflare.com/ajax/libs/twitter-
        bootstrap/4.4.1/css/bootstrap.min.css">
           <script src = "https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.js"> </scr
        ipt>
           <script src = "https://cdnjs.cloudflare.com/ajax/libs/twitter-
        bootstrap/4.4.1/js/bootstrap.min.js"> </script>
           <link rel = "stylesheet" href = "https://cdnjs.cloudflare.com/ajax/libs/font-
        awesome/4.7.0/css/font-awesome.min.css">
           <script src = "https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></
        script>
           <title> jQuery simple star rating example </title>
           <style>
           body {
              background-color: aquamarine;
              margin : 0px;
           }
           .fa-star {
              font-size : 50px;
              align-content: center;
           }
           .container {
              height: 100px;
```

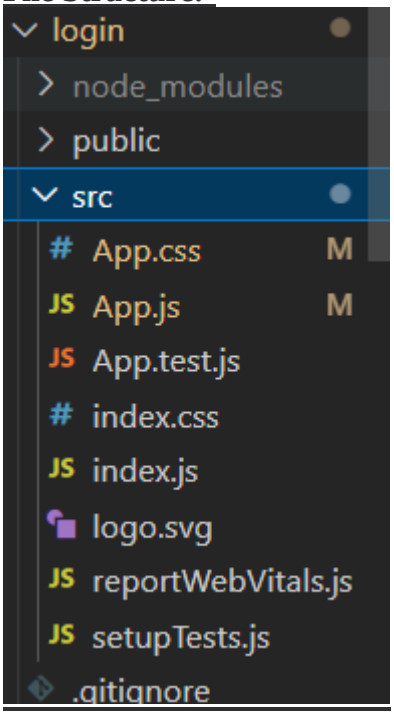
```
        width: 600px;
        margin: auto;
    }
    </style>
</head>
<body>
    <div class = "container">
        <h2 style="margin-top: 50px;">jQuery simple star rating example</h2>
        <div class = "con">
        <h3 style = "margin-top : 80px; color: green;">Rate our product :-</h3>
        <i class = "fa fa-star" aria-hidden = "true" id = "st1"></i>
        <i class = "fa fa-star" aria-hidden = "true" id = "st2"></i>
        <i class = "fa fa-star" aria-hidden = "true" id = "st3"></i>
        <i class = "fa fa-star" aria-hidden = "true" id = "st4"></i>
        <i class = "fa fa-star" aria-hidden = "true" id = "st5"></i>
        </div>
    </div>
    <script>
        $(document).ready(function() {
          $("#st1").click(function() {
              $(".fa-star").css("color", "black");
              $("#st1").css("color", "yellow");

          });
          $("#st2").click(function() {
              $(".fa-star").css("color", "black");
              $("#st1, #st2").css("color", "yellow");

          });
          $("#st3").click(function() {
              $(".fa-star").css("color", "black")
              $("#st1, #st2, #st3").css("color", "yellow");

          });
```

```
        $("#st4").click(function() {
            $(".fa-star").css("color", "black");
            $("#st1, #st2, #st3, #st4").css("color", "yellow");

        });
        $("#st5").click(function() {
            $(".fa-star").css("color", "black");
            $("#st1, #st2, #st3, #st4, #st5").css("color", "yellow");

        });
      });
    </script>
</body>
</html>
```

Week-8. Create a Simple Login form using React js

Now we are creating a login form which is a very important in any application or website as you know if open any website or application you will get a message to login if you click that you will be redirected to login page in this week we will be creating login page

File Structure:-



----------**App.js**----------

```
import './App.css';

import React, { useState } from "react";

import ReactDOM from "react-dom";


function App() {

  // React States

  const [errorMessages, setErrorMessages] = useState({});

  const [isSubmitted, setIsSubmitted] = useState(false);
```

```
// User Login info
const database = [
 {
  username: "user1",
  password: "pass1"
 },
 {
  username: "user2",
  password: "pass2"
 }
];

const errors = {
 uname: "invalid username",
 pass: "invalid password"
};

consthandleSubmit = (event) => {
 //Prevent page reload
 event.preventDefault();

 var { uname, pass } = document.forms[0];

 // Find user login info
 constuserData = database.find((user) =>user.username === uname.value);
```

```
  // Compare user info
  if (userData) {
    if (userData.password !== pass.value) {
      // Invalid password
      setErrorMessages({ name: "pass", message: errors.pass });
    } else {
      setIsSubmitted(true);
    }
  } else {
    // Username not found
    setErrorMessages({ name: "uname", message: errors.uname });
  }
};

// Generate JSX code for error message
constrenderErrorMessage = (name) =>
  name === errorMessages.name && (
    <div className="error">{errorMessages.message}</div>
  );

// JSX code for login form
constrenderForm = (
  <div className="form">
    <form onSubmit={handleSubmit}>
```

```
      <div className="input-container">
        <label>Username </label>
        <input type="text" name="uname" required />
        {renderErrorMessage("uname")}
      </div>
      <div className="input-container">
        <label>Password </label>
        <input type="password" name="pass" required />
        {renderErrorMessage("pass")}
      </div>
      <div className="button-container">
        <input type="submit" />
      </div>
    </form>
  </div>
);

return (
  <div className="app">
    <div className="login-form">
      <div className="title">Sign In</div>
      {isSubmitted ?<div>User is successfully logged in</div> :renderForm}
    </div>
  </div>
);
```

```
}


export default App;
```

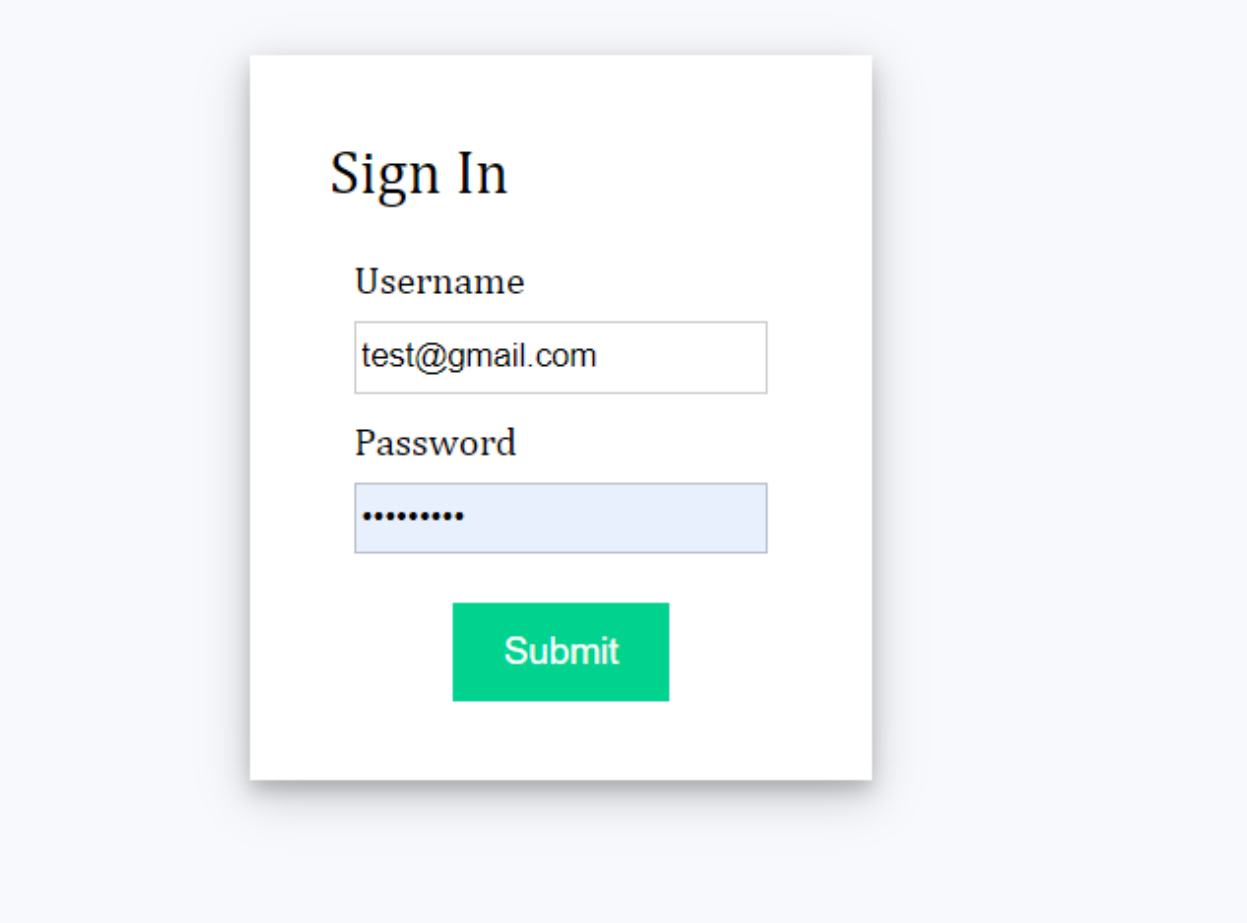
----------Now Create a App.css file in same folder----------

App.css

```
.app {
  font-family: sans-serif;
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  gap: 20px;
  height: 100vh;
  font-family: Cambria, Cochin, Georgia, Times, "Times New Roman", serif;
  background-color: #f8f9fd;
}


input[type="text"],
input[type="password"] {
  height: 25px;
  border: 1px solid rgba(0, 0, 0, 0.2);
}


input[type="submit"] {
  margin-top: 10px;
```

```
  cursor: pointer;
  font-size: 15px;
  background: #01d28e;
  border: 1px solid #01d28e;
  color: #fff;
  padding: 10px 20px;
}

input[type="submit"]:hover {
  background: #6cf0c2;
}

.button-container {
  display: flex;
  justify-content: center;
}

.login-form {
  background-color: white;
  padding: 2rem;
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
}

.list-container {
  display: flex;
```

```
}

.error {
  color: red;
  font-size: 12px;
}

.title {
  font-size: 25px;
  margin-bottom: 20px;
}

.input-container {
  display: flex;
  flex-direction: column;
  gap: 8px;
  margin: 10px;
}
```

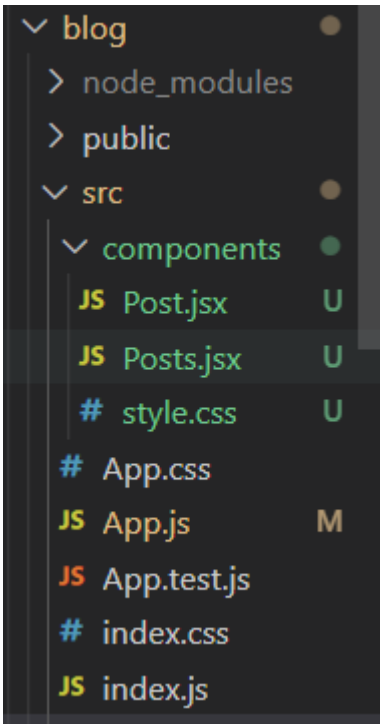

Sign In
Username
test@gmail.com
Password
Submit

Week-9. Create a blog in React js

In this week we are going to create a blog website using react js

We have mainly 3 pages
1.App.js
2.Post.js
3.Posts.js

This is the Structure of the project



1.App.js

```
import logo from './logo.svg';
import './App.css';
import Posts from "./components/Posts";
function App() {
  return (
<div className="main-container">
<h1 className="main-heading">
      Blog App using React Js
</h1>
<Posts />
```

```
</div>
  );
}

export default App;
```

2.Posts.js

```
import React from "react";
import "./style.css";

import Post from "./Post";

const Posts = () => {
  const blogPosts = [
    {
      title: "JAVASCRIPT",
      body: `JavaScript is the world most popular
      lightweight, interpreted compiled programming
      language. It is also known as scripting
      language for web pages. It is well-known for
      the development of web pages, many non-browser
      environments also use it. JavaScript can be
      used for Client-side developments as well as
      Server-side developments`,
      author: "Nishant Singh ",
imgUrl:
        "https://media.geeksforgeeks.org/img-practice/banner/diving-into-excel-thumbnail.png",
    },
    {
      title: "Data Structure ",
      body: `There are many real-life examples of
      a stack. Consider an example of plates stacked
      over one another in the canteen. The plate
      which is at the top is the first one to be
      removed, i.e. the plate which has been placed
      at the bottommost position remains in the
      stack for the longest period of time. So, it
      can be simply seen to follow LIFO(Last In
      First Out)/FILO(First In Last Out) order.`,
      author: "Suresh Kr",
imgUrl:
        "https://media.geeksforgeeks.org/img-practice/banner/coa-gate-2022-thumbnail.png",
    },
    {
```

```
    title: "Algorithm",
    body: `The word Algorithm means “a process
    or set of rules to be followed in calculations
    or other problem-solving operations”. Therefore
    Algorithm refers to a set of rules/instructions
    that step-by-step define how a work is to be
    executed upon in order to get the expected
    results. `,
    author: "Monu Kr",
imgUrl:
      "https://media.geeksforgeeks.org/img-practice/banner/google-test-series-thumbnail.png",
  },
  {
    title: "Computer Network",
    body: `An interconnection of multiple devices,
    also known as hosts, that are connected using
    multiple paths for the purpose of sending/
    receiving data media. Computer networks can
    also include multiple devices/mediums which
    help in the communication between two different
    devices; these are known as Network devices
    and include things such as routers, switches,
    hubs, and bridges. `,
    author: "Sonu Kr",
imgUrl:
      "https://media.geeksforgeeks.org/img-practice/banner/cp-maths-java-thumbnail.png",
  },
 ];

 return (
<div className="posts-container">
    {blogPosts.map((post, index) => (
<Post key={index} index={index} post={post} />
    ))}
</div>
 );
};

export default Posts;
```

3.Post.js

```
import React from "react";
import "./style.css";
const Post = ({ post: { title, body,
imgUrl, author }, index }) =>
{
```

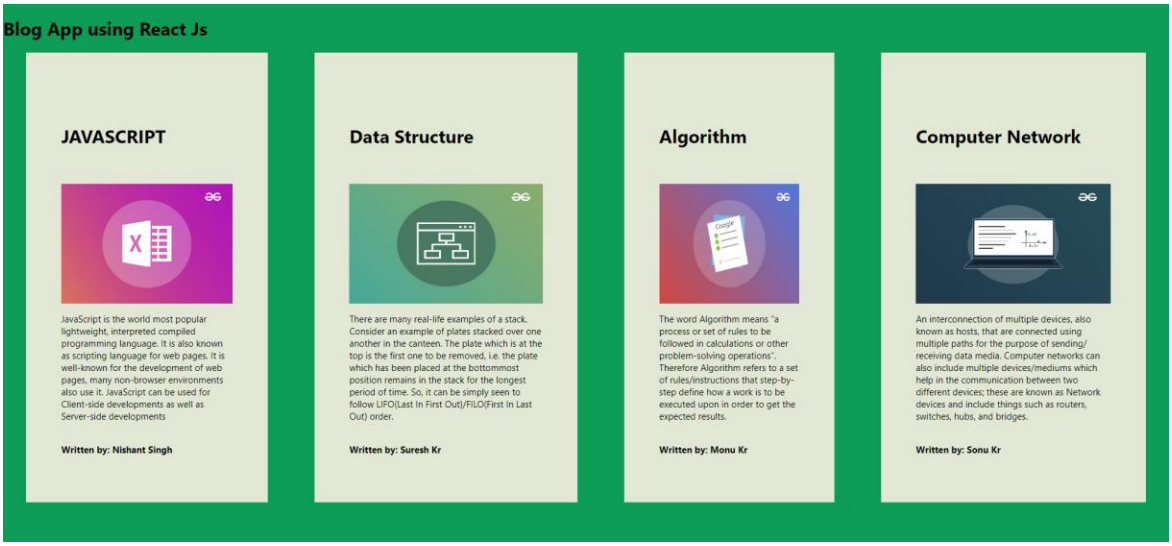
```
  return (
<div className="post-container">
<h1 className="heading">{title}</h1>
<imgclassName="image" src={imgUrl} alt="post" />
<p>{body}</p>
<div className="info">
<h4>Written by: {author}</h4>
</div>
</div>
  );
};

export default Post;
```

Now we will style the project

Style.css in componenets folder

```
body {
   background-color: #0e9d57;
}
.posts-container {
   display: flex;
   justify-content: center;
   align-items: center;
}
.post-container {
   background: #e2e8d5;
   display: flex;
   flex-direction: column;
   padding: 3%;
   margin: 0 2%;
   height: 40%;
}
.heading {
   height: 126px;
   text-align: center;
   display: flex;
   align-items: center;
}
.image {
   width: 100%;
   height: 210px;
}
```

OUTPUT

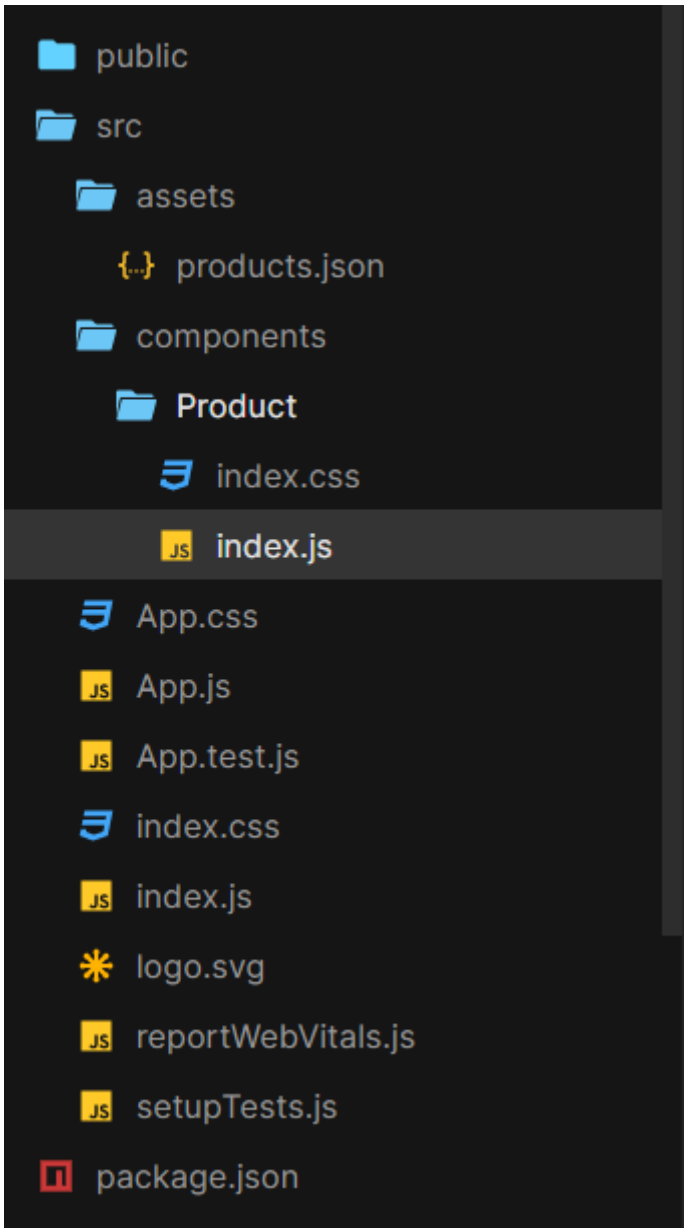

Blog App using React Js
JAVASCRIPT
JavaScript is the world most popular lightweight, interpreted compiled programming language. It is also known as scripting language for web pages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments
Written by: Nishant Singh
Data Structure
There are many real-life examples of a stack. Consider an example of plates stacked over one another in the canteen. The plate which is at the top is the first one to be removed, i.e. the plate which has been placed at the bottommost position remains in the stack for the longest period of time. So, it can be simply seen to follow LIFO(Last In First Out)/FILO(First In Last Out) order.
Written by: Suresh Kr
Algorithm
The word Algorithm means "a process or set of rules to be followed in calculations or other problem-solving operations". Therefore Algorithm refers to a set of rules/instructions that step-by-step define how a work is to be executed upon in order to get the expected results.
Written by: Monu Kr
Computer Network
An interconnection of multiple devices, also known as hosts, that are connected using multiple paths for the purpose of sending/ receiving data media. Computer networks can also include multiple devices/mediums which help in the communication between two different devices; these are known as Network devices and include things such as routers, switches, hubs, and bridges.
Written by: Sonu Kr

Week-10. Create a project on Grocery delivery application

Assume this project is for a huge online departmental store. Assume that they have a myriadof grocery items at their godown. All items must be listed on the website, along with theirquantities and prices.Users must be able to sign up and purchase groceries. The systemshould present him with delivery slot options, and the user must be able to choose hispreferred slot. Users must then be taken to the payment page where he makes the payment with his favorite method.

This week will have many pages like Header,footer,categories and app.jsx

File Structure:



App.jsx

```
import "./index.css"
import "./App.css"
import products from "./assets/products.json"
import Product from "./components/Product";

export default function App() {
  return (
<div className={"container"}>
<main className={"main"}>
<h1>
        E-Commerce in React and SnipCart
</h1>

<div className={"grid"}>
        {
products.map((product, i) =><Product {...product} key={i}/>)
        }
</div>
</main>
<div
        id="snipcart"
        data-api-
key="NWMwZWNkZGMtZjU2ZS00YzM3LWFlZjYtMmM5Zjk0MWViZDcxNjM3Njg0OTY
0ODg5NTk4MTM3" hidden
>
</div>
</div>
  );
}
```

Components/Product/index.js

```
import "./index.css";

export default function Product(props) {
   const {id, imageUrl, name, description, price} = props

   return (
<div
        key={id}
className={"product"}
>
<img
src={imageUrl}
```

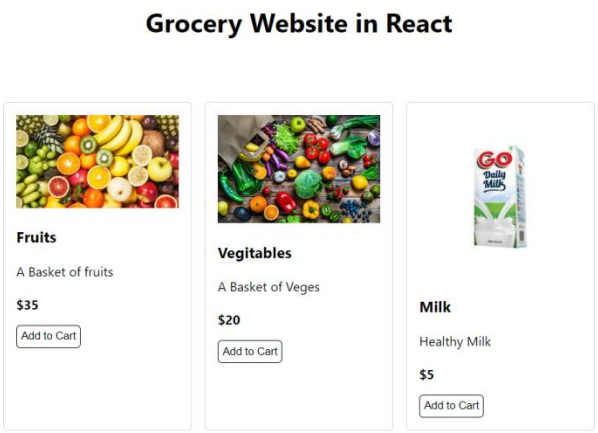
```
            alt={`Image of ${name}`}
className={"image-product"}
          />
<h3>{name}</h3>
<p>{description}</p>
<span>${price}</span>
<div>
<button
className="snipcart-add-item"
                data-item-id={id}
                data-item-image={imageUrl}
                data-item-name={name}
                data-item-url="/"
                data-item-price={price}
>
                Add to Cart
</button>
</div>
</div>
   );
}
```

Assets/products.json

```
[
  {
    "id": "t-shirt",
    "name": "Fruits",
    "price": 35.0,
    "imageUrl":          "https://www.lalpathlabs.com/blog/wp-content/uploads/2019/01/Fruits-and-
Vegetables.jpg",
    "description": "A Basket of fruits",
    "url": "/api/products/halfmoon"
  },
  {
    "id": "wallet",
    "name": "Vegitables",
    "price": 20.0,
    "imageUrl":  "https://img.freepik.com/free-photo/bottom-view-fruits-vegetables-radish-cherry-
tomatoes-persimmon-tomatoes-kiwi-cucumber-apples-red-cabbage-parsley-quince-aubergines-
blue-table_140725-146174.jpg",
    "description": "A Basket of Veges",
    "url": "/api/products/wallet"
  },
```

```
  {
    "id": "cup",
    "name": "Milk",
    "price": 5.0,
    "imageUrl": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSeujHMy6OLRZHTpsrUMVLsHyio1mZiZI4fMQ&usqp=CAU",
    "description": "Healthy Milk",
    "url": "/api/products/veiltail"
  }
]
```


Output

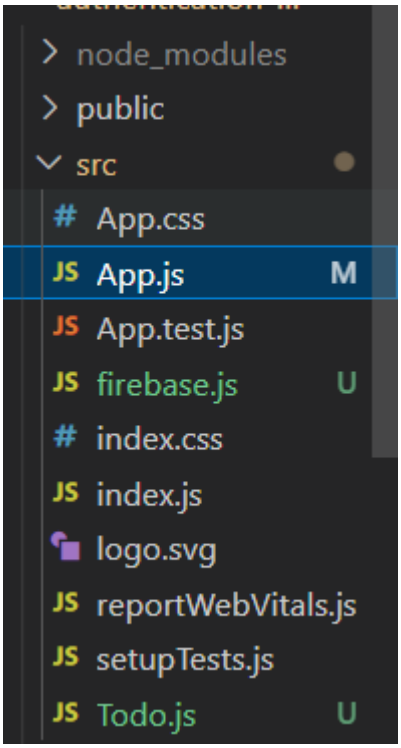


Week-11. Connecting our TODO React jsProject with Firebase

We all can create applications but in realtime when we are building an application we have to store the user data some ware now a days best way to store is Firebase which can be integrated in react app

In this week we will learn how to connect our application to firebase

File Structure:



After creating the project make sure to install firebase dependencies:

Install it using npm install firebase

-Now we have mainly 3 pages
1.firebase.js
2.App.js
3.Todo.js

-.In firebase.js we will establish connection to our app and firebase
-In Todo,js we will write the code
And we will import it in to the App.js file

firebase.js

```
import firebase from 'firebase/compat/app';
import 'firebase/compat/auth';
import 'firebase/compat/firestore';

const firebaseApp = firebase.initializeApp({
apiKey: "",
authDomain: "",
projectId:",
storageBucket: ",
messagingSenderId: "",
appId: ":,
measurementId: ""
});

const db = firebaseApp.firestore();

export default db;
```

Note Replace the highlighted code with your firebase connection compoenents

You can get you own keys from firebase account for more details Take the Reference of below video

https://www.youtube.com/watch?v=ad6IavyAHsQ

Todo.js

```
import { ListItem, List, ListItemAvatar, ListItemText, Button, Modal, makeStyles
} from '@material-ui/core'
import './Todo.css';
import React, { useState } from 'react';
import db from './firebase'



function Todo(props) {
   const [open, setOpen] = useState(false);
   const [input, setInput] = useState(props.todo.todo);

   const handleOpen = () => {
```

```
setOpen(true)
   };

   const updateTodo = () => {
      // update to do with the new input text

db.collection('todos').doc(props.todo.id).set({
todo: input
      }, { merge: true })

setOpen(false);
   }

   return (
<>
<div
            open={open}
onClose={e =>setOpen(false)}
>
<div >
<h1>I am a model</h1>
<input placeholder={props.todo.todo} value={input} onChange={event
=>setInput(event.target.value)} />
<button onClick={updateTodo}>Update Todo</button>
</div>
</div>
<ulclassName='todo_list'>
<li>
<li primary={props.todo.todo} secondary='Dummy deadline ⏰' />
</li>
<button onClick={e =>setOpen(true)}>Edit</button>
<button                                                    onClick={event
=>db.collection('todos').doc(props.todo.id).delete()}>❌DELETE ME</button>
</ul>
</>
   )
}

export default Todo
```

Now the last file App.js

```
import React, { useEffect, useState } from 'react';
import './App.css';
import Todo from './Todo';
import db from './firebase'
import firebase from 'firebase/compat/app';
import 'firebase/compat/auth';
import 'firebase/compat/firestore';

function App() {
  const [todos, setTodos] = useState([]);
  const [input, setInput] = useState('');

// when the upload, we need to listen to the database and fetch new todos as they get added/remove
useEffect(() => {
  // This code here... fires when the app.js lodes
db.collection('todos').orderBy('timestamp', 'desc').onSnapshot(snapshot => {
    // console.log(snapshot.docs.map(doc =>doc.data()));
setTodos(snapshot.docs.map(doc => ({id: doc.id, todo: doc.data().todo})))
  })
}, []);

  const addTodo = (event) => {
    // this will fire off when we click the button
event.preventDefault(); //will stop the refresh

db.collection('todos').add({
todo: input,
      timestamp: firebase.firestore.FieldValue.serverTimestamp()
    })

setTodos([...todos, input]);
setInput(' ');  // clear up the input after clicking todo
    console.log(todos)
  }

  return (
<div className="App">
<h1>Build A TODO App 🚀!</h1>

<form>

<form>
```

```
<span>✓ Write a Todo</span>
<input value={input} onChange={event =>setInput(event.target.value)} />
</form>


<button disabled={!input} type='submit' onClick={addTodo} variant="contained" color="primary">Add Todo</button>
</form>

<ul>
      {todos.map(todo => (
<Todotodo={todo}/>
        // <li>{todo}</li>
      ))}

<li></li>
</ul>

</div>
  );
}

export default App;
```

OUTPUT


Build A TODO App 🚀!
Write a Todo
Add Todo
I am a model
Task2
Update Todo
Edit
DELETE ME
I am a model
Task1
Update Todo
Edit
DELETE ME