



Calender Scheduler Application

Task-3



Calendar Using HTML, CSS, and JavaScript

LMS Username	Name	Batch
2115a619	Vajja mohith reddy	A6
2115a618	Vadlaputi raviteja	A6
2115a611	Mallikarjun pagadala	A6



Calendar Scheduler Application:

- Calendar software **helps you create and edit online calendars and share them with multiple users**. The software generally allows you to set your availability and check the availability of other users, which makes scheduling easier for everyone in the group or organization.
- A scheduling software is **a cloud-based or local computer program or application that automates certain employee scheduling processes such as work schedule management, employee communication, and time tracking**.



FRONT END DEVELOPMNENT USING REACT JS

App.js

- import React, { useState, useEffect } from 'react'
- import IconButton from '@material-ui/core/IconButton'
- import Paper from '@material-ui/core/Paper'
- import Typography from '@material-ui/core/Typography'
- import Box from '@material-ui/core/Box'
- import { makeStyles } from '@material-ui/core/styles'
- import { ArrowLeft, ArrowRight } from '@material-ui/icons'

- const styles = makeStyles(theme => ({
- calender: {
- border: "1px solid lightgray",
- width: 350,
- margin: "0px auto",
- padding: "0 16px",
- borderLeft: "1px solid lightgray",
- borderTop: "1px solid lightgray",
- ['@media(max-width: 400px)']: {
- width: "90%",
- }
- },
- calenderTitle: {
- display: "flex",
- flexDirection: "row",
- justifyContent: "space-between",
- alignItems: "center",
- },
- arrowIcon: {
- margin: "0 16px"
- },
- weekDays: {
- listStyleType: "none",
- padding: 0,
- },
- date: {



- display: "inline-block",
- width: "14.2%",
- textAlign: "center",
- marginBottom: "8px",
- },
- dateText: {
- cursor: "pointer"
- },
- selected: {
- background: "yellow"
- },
- graySelected: {
- background: "lightgray"
- }
- }
▪)))

- export default function App() {
- const classes = styles()
- const [weekStartDay, setWeekStartDay] = useState()
- const [date, setDate] = useState(new Date())
- const [month, setMonth] = useState(date.getMonth())
- const [year, setYear] = useState(date.getFullYear())
- const [daysInMonth, setDaysInMonth] = useState(new Date(year, month + 1, 0))

- useEffect(() => {
- let _temp = new Date(year, month + 1, 0)
- setDaysInMonth(_temp.getDate())
- setYear(_temp.getFullYear())
- }, [month, year])

```
▪ useEffect(() => {  
▪   let _temp = new Date(year, month, 1)  
▪   setWeekStartDay(_temp.getDay())  
▪ }, [month, year])  
  
▪ const monthName = () => {  
▪   let _date = new Date(year, month + 1, 0)  
▪   let _monthName = _date.toLocaleString('default', { month:  
'long' })  
▪   return _monthName  
▪ }  
  
▪ const onLeftArrowClick = () => {  
▪   if (month === 0) setYear(year - 1)  
▪   setMonth((month + 12 - 1) % 12)  
▪ }  
▪ const onRightArrowClick = () => {  
▪   if (month === 11) setYear(year + 1)  
▪   setMonth((month + 12 + 1) % 12)  
▪ }  
  
▪ const handleDateSelection = (e) => {  
▪   let _date = parseInt(e.target.textContent)  
▪   setDate(new Date(year, month, _date))  
▪ }  
▪ const dateStylingClass = (payload) => {  
▪   let _classes = `${classes.date}`  
▪   let selected = date.getDate() === payload && date.getMonth()  
=== month  
▪   let graySelected = date.getDate() === payload &&  
date.getMonth() !== month
```

```

▪   if (selected) _classes = _classes + `${classes.selected}`
▪   else if (graySelected) _classes = _classes +
`${classes.graySelected}`
▪   return _classes
▪   }

▪   return (
▪     <div>
▪       <Box textAlign="center" mt={2}>
▪         <Typography
▪           variant="h5"
▪           gutterBottom >
▪             Calender {year}
▪         </Typography>
▪         <Typography
▪           variant="body1"
▪           gutterBottom >
▪             Selected Date: {date.toDateString()}
▪         </Typography>
▪       </Box>
▪       <Paper className={classes.calender} elevation={10}>
▪         <div className={classes.calenderTitle}>
▪           <IconButton
▪             className={classes.arrowIcon}
▪             size="small"
▪             onClick={onLeftArrowClick}>
▪             <ArrowLeft />
▪           </IconButton>
▪           <h4 style={{ textAlign: "center" }}>
▪             {monthName()}, {year}
▪           </h4>

```

```

▪ <IconButton
▪   className={classes.arrowIcon}
▪   size="small"
▪   onClick={onRightArrowClick}>
▪   <ArrowRight />
▪ </IconButton>
▪ </div>
▪ <ul className={classes.weekDays}>
▪   {
▪     ['sun', 'mon', 'tue', 'wed', 'thu', 'fri', 'sat'].map((day, i) => (
▪       <li key={i} className={classes.date}>
▪         <Typography variant="caption">
▪           {day.toUpperCase()}
▪         </Typography>
▪       </li>
▪     ))
▪   }
▪   {
▪     [...new Array(weekStartDay)].map((elem, i) => (
▪       <li className={classes.date} key={i}></li>
▪     ))
▪   }
▪   {
▪     [...new Array(daysInMonth)].map((elem, i) => (
▪       <li
▪         className={dateStylingClass(i + 1)}
▪         key={i}
▪         onClick={handleDateSelection}>
▪         <Typography variant="caption"
className={classes.dateText}>
▪           {i + 1}

```


- </Typography>
-
-))
- }
-
- </Paper>
- </div>
-)
- }

- **index.js**

- \$ import React from 'react';
- import ReactDOM from 'react-dom';
- import App from './App';
- ReactDOM.render(
 - <React.StrictMode>
 - <App />
 - </React.StrictMode>,
 - document.getElementById('root')
 -)

package.json

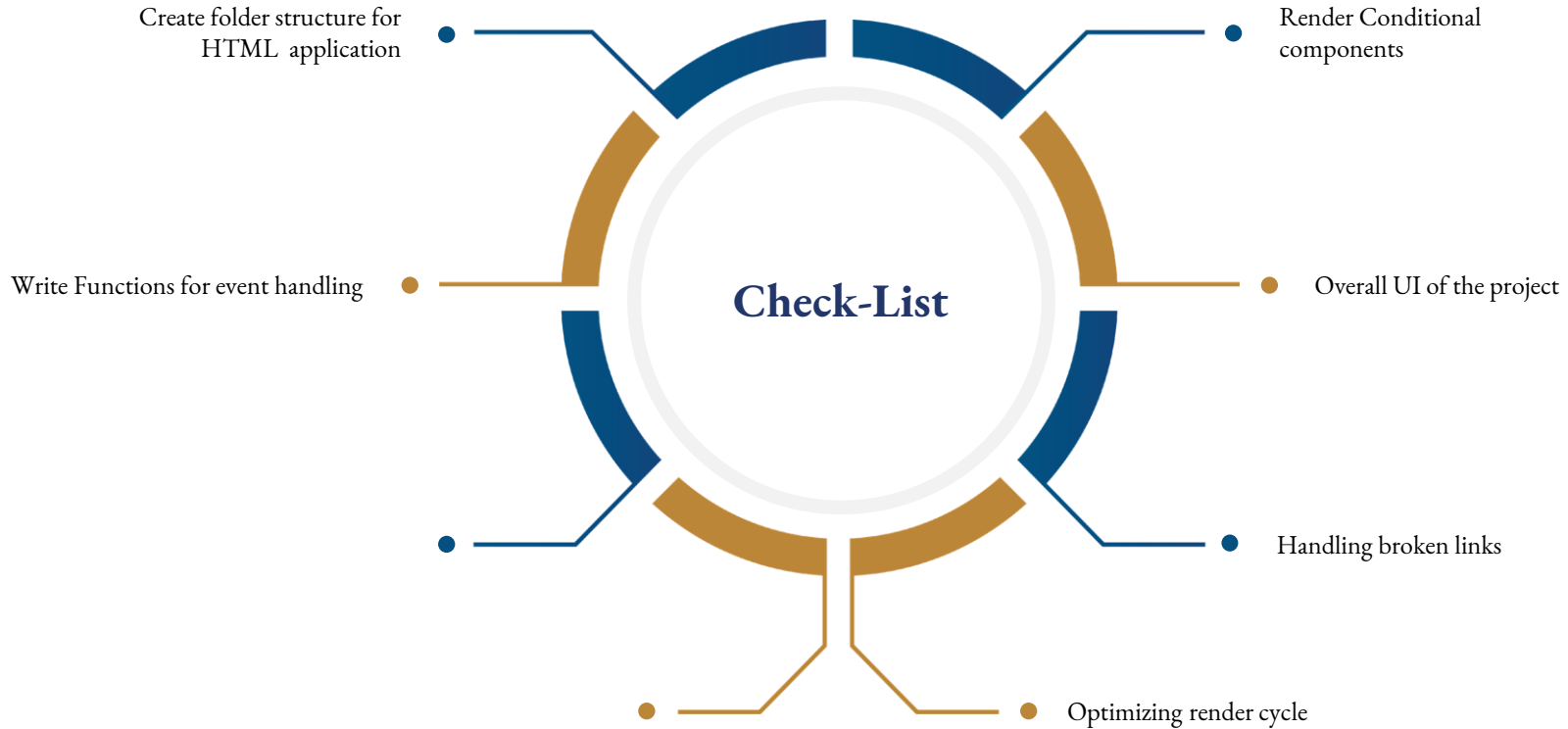
```
{
  "name": "react-calender",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@material-ui/core": "^4.12.3",
    "@material-ui/icons": "^4.11.2",
    "@testing-library/jest-dom": "^5.14.1",
    "@testing-library/react": "^11.2.7",
    "@testing-library/user-event": "^12.8.3",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-scripts": "4.0.3",
    "web-vitals": "^1.1.2"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
```



- "production": [
 - ">0.2%",
 - "not dead",
 - "not op_mini all"
-],
- "development": [
 - "last 1 chrome version",
 - "last 1 firefox version",
 - "last 1 safari version"
-]
- }
- }



Assessment Parameter



Sample Output:

< NOVEMBER 2021 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Submission Github



<https://github.com/raviteja1405/calender>

Thank
you!

