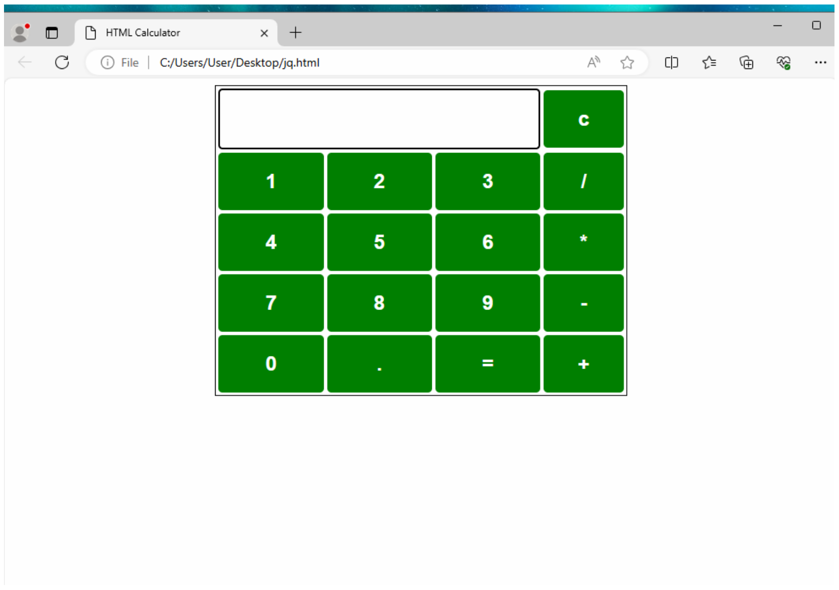
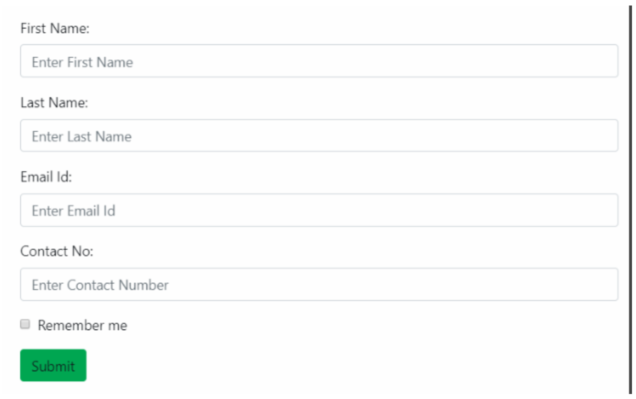
1.Write a program to create a simple calculator using HTML.

```
<!DOCTYPE html>
<html>
<head>
      <title>HTML Calculator</title>
      <style>
            table {
                  border: 1px solid black;
                  margin-left: auto;
                  margin-right: auto;
            }

            input[type="button"] {
                  width: 100%;
                  padding: 20px 40px;
                  background-color: green;
                  color: white;
                  font-size: 24px;
                  font-weight: bold;
                  border: none;
                  border-radius: 5px;
            }

            input[type="text"] {
                  padding: 20px 30px;
                  font-size: 24px;
                  font-weight: bold;
                  border: none;
                  border-radius: 5px;
                  border: 2px solid black;
            }
      </style>
</head>

<body>

      <!-- Create table -->
      <table id="calcu">
            <tr>
                  <td colspan="3">
                        <input type="text" id="result">
                  </td>
                  <td><input type="button" value="c"></td>
            </tr>
            <tr>
                  <td><input type="button" value="1"></td>
                  <td><input type="button" value="2"></td>
                  <td><input type="button" value="3"></td>
                  <td><input type="button" value="/"></td>
            </tr>
            <tr>
                  <td><input type="button" value="4"></td>
                  <td><input type="button" value="5"></td>
                  <td><input type="button" value="6"></td>
                  <td><input type="button" value="*"></td>
            </tr>
            <tr>
                  <td><input type="button" value="7"></td>
                  <td><input type="button" value="8"></td>
                  <td><input type="button" value="9"></td>
                  <td><input type="button" value="-"></td>
            </tr>
            <tr>
                  <td><input type="button" value="0"></td>
                  <td><input type="button" value="."></td>
                  <td><input type="button" value="="></td>
                  <td><input type="button" value="+"></td>
            </tr>
      </table>
</body>

</html>
```


HTML Calculator
File | C:/Users/User/Desktop/jq.html
c
1
2
3
/
4
5
6
*
7
8
9
-
0
.
=
+

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1>
    <title>SIGN-UP FORM</title>
    <!-- Bootstrap CSS -->
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.c
ss" rel="stylesheet">
</head>
<body>
    <div class="container">
        <h2 class="mt-5">Bootstrap Form Example</h2>
        <form>
            <div class="form-group">
                <label for="exampleInputName">Name</label>
                <input type="text" class="form-control"
id="exampleInputName" placeholder="Enter your name">
            </div>
            <div class="form-group">
                <label for="exampleInputEmail">Email address</label>
                <input type="email" class="form-control"
id="exampleInputEmail" placeholder="Enter your email">
            </div>
            <div class="form-group">
                <label for="exampleInputPassword">Password</label>
                <input type="password" class="form-control"
id="exampleInputPassword" placeholder="Password">
            </div>
            <div class="form-group">
                <label for="exampleInputConfirmPassword">Confirm
Password</label>
                <input type="password" class="form-control"
id="exampleInputConfirmPassword" placeholder="Confirm Password">
            </div>
            <div class="form-group form-check">
                <input type="checkbox" class="form-check-input"
id="exampleCheck1">
                <label class="form-check-label" for="exampleCheck1">Check me
out</label>
            </div>
            <button type="submit" class="btn btn-primary">Submit</button>
        </form>
    </div>
    <!-- Bootstrap JS and dependencies -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.j
s"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
></script>
</body>
</html>
```


First Name:

Enter First Name

Last Name:

Enter Last Name

Email Id:

Enter Email Id

Contact No:

Enter Contact Number

Remember me

Submit

3.Create Calculator using Java script.

3.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="calc.css">
  <title>Calulator</title>
</head>
<body>
  <div class="main">
    <input type="text" id = 'res'>
    <div class="btn">
      <input type="button" value = 'C' onclick = "Clear()">
      <input type="button" value = '%' onclick = "Solve('%')">
      <input type="button" value = '←' onclick ="Back('←')">
      <input type="button" value = '/' onclick = "Solve('/')">
      <br>
      <input type="button" value = '7' onclick = "Solve('7')">
      <input type="button" value = '8' onclick = "Solve('8')">
      <input type="button" value = '9' onclick = "Solve('9')">
      <input type="button" value = 'x' onclick = "Solve('*')">
      <br>
      <input type="button" value = '4' onclick = "Solve('4')">
      <input type="button" value = '5' onclick = "Solve('5')">
      <input type="button" value = '6' onclick = "Solve('6')">
      <input type="button" value = '-' onclick = "Solve('-')">
      <br>
      <input type="button" value = '1' onclick = "Solve('1')">
      <input type="button" value = '2' onclick = "Solve('2')">
      <input type="button" value = '3' onclick = "Solve('3')">
      <input type="button" value = '+' onclick = "Solve('+')">
      <br>
      <input type="button" value = '00'onclick = "Solve('00')">
      <input type="button" value = '0' onclick = "Solve('0')">
```

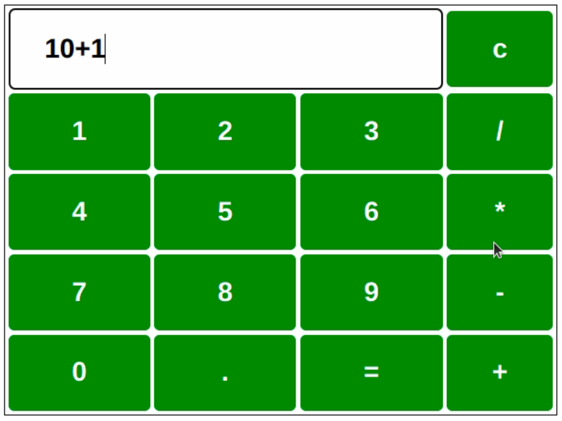
```
        <input type="button" value = '.' onclick = "Solve('.')">
        <input type="button" value = '=' onclick = "Result()">
      </div>
   </div>
   <script src = 'calc.js' ></script>
</body>
</html>
```

3.CSS

```
*{
   padding: 0;
   margin: 0;
   font-family: 'poppins', sans-serif;
}
body{
   background-color: #495250;
   display: grid;
   height: 100vh;
   place-items: center;
}
.main{
   width: 400px;
   height: 450px;
   background-color: white;
   position: absolute;
   border: 5px solid black;
   border-radius: 6px;
}
.main input[type='text'] {
   width: 88%;
   position: relative;
   height: 80px;
   top: 5px;
   text-align: right;
   padding: 3px 6px;
   outline: none;
   font-size: 40px;
   border: 5px solid black;
   display: flex;
   margin: auto;
   border-radius: 6px;
   color: black;
}
.btn input[type='button']{
   width:90px;
   padding: 2px;
   margin: 2px 0px;
   position: relative;
   left: 13px;
   top: 20px;
   height: 60px;
   cursor: pointer;
   font-size: 18px;
   transition: 0.5s;
   background-color: #495250;
   border-radius: 6px;
   color: white;
}
.btn input[type='button']:hover{
   background-color: black;
   color: white;
}
```

3.JS

```
function Solve(val) {
   var v = document.getElementById('res');
   v.value += val;
}
function Result() {
   var num1 = document.getElementById('res').value;
   var num2 = eval(num1);
   document.getElementById('res').value = num2;
}
function Clear() {
   var inp = document.getElementById('res');
   inp.value = '';
}
function Back() {
   var ev = document.getElementById('res');
   ev.value = ev.value.slice(0,-1);
}
```



4.Create form and perform Jquery validation on form.

```
<!DOCTYPE html>
```

```
<html>

<head>
          <!-- Latest compiled and minified CSS -->
          <link rel="stylesheet" href=
"https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
          <!-- jQuery library -->
          <script src=
"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
          </script>
          <!-- Popper JS -->
          <script src=
"https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js">
          </script>
          <!-- Latest compiled JavaScript -->
          <script src=
"https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js">
          </script>
</head>

<body>
          <h1 class="text-center text-warning">
                    Welcome to DSCE
          </h1>
          <p class="text-center">
                    FORM VALIDATION USING JQUERY
          </p>

          <div class="container">
                    <div class="col-lg-8
                                                  m-auto d-block">
                              <form>
                                        <div class="form-group">
                                                  <label for="user">
                                                            Username:
                                                  </label>
                                                  <input type="text"
                                                            name="username"
                                                            id="usernames"
                                                            class="form-control">
                                                  <h5 id="usercheck"
                                                            style="color: red;">
                                                            **Username is missing
                                                  </h5>
                                        </div>

                                        <div class="form-group">
```

```
                        <label for="user">
                                Email:
                        </label>
                        <input type="email"
                                name="email"
                                id="email" required
                                class="form-control">
                        <small id="emailvalid"
                                class="form-text text-muted invalid-feedback">
                                Your email must be a valid email
                        </small>
                </div>

                <div class="form-group">
                        <label for="password">
                                Password:
                        </label>
                        <input type="password"
                                name="pass"
                                id="password"
                                class="form-control">
                        <h5 id="passcheck"
                                style="color: red;">
                                **Please Fill the password
                        </h5>
                </div>

                <div class="form-group">
                        <label for="conpassword">
                                Confirm Password:
                        </label>
                        <input type="password"
                                name="username"
                                id="conpassword"
                                class="form-control">
                        <h5 id="conpasscheck"
                                style="color: red;">
                                **Password didn't match
                        </h5>
                </div>

                <input type="submit"
                        id="submitbtn"
                        value="Submit"
                        class="btn btn-primary">
        </form>
</div>
```

```
        </div>

        <!-- Including app.js jQuery Script -->
        <script src="1.js"></script>
</body>

</html>
// Document is ready
$(document).ready(function () {

        // Validate Username
        $("#usercheck").hide();
        let usernameError = true;
        $("#usernames").keyup(function () {
                validateUsername();
        });

        function validateUsername() {
                let usernameValue = $("#usernames").val();
                if (usernameValue.length == "") {
                        $("#usercheck").show();
                        usernameError = false;
                        return false;
                } else if (usernameValue.length < 3 || usernameValue.length > 10) {
                        $("#usercheck").html("**length of username must be between 3 and
10"); eError = false;
                        return false;
                } else {
                        $("#usercheck").hide();
                }
        }

        // Validate Email
        const email = document.getElementById("email");
        email.addEventListener("blur", () => {
                let regex =
                /^([_\-\.0-9a-zA-Z]+)@([_\-\.0-9a-zA-Z]+)\.([a-zA-Z]){2,7}$/;
                let s = email.value;
                if (regex.test(s)) {
                        email.classList.remove("is-invalid");
                        emailError = true;
                } else {
                        email.classList.add("is-invalid");
                        emailError = false;
                }
        });
```

```
// Validate Password
$("#passcheck").hide();
let passwordError = true;
$("#password").keyup(function () {
        validatePassword();
});
function validatePassword() {
        let passwordValue = $("#password").val();
        if (passwordValue.length == "") {
                $("#passcheck").show();
                passwordError = false;
                return false;
        }
        if (passwordValue.length < 3 || passwordValue.length > 10) {
                $("#passcheck").show();
                $("#passcheck").html(
                        "**length of your password must be between 3 and 10"
                );
                $("#passcheck").css("color", "red");
                passwordError = false;
                return false;
        } else {
                $("#passcheck").hide();
        }
}

// Validate Confirm Password
$("#conpasscheck").hide();
let confirmPasswordError = true;
$("#conpassword").keyup(function () {
        validateConfirmPassword();
});
function validateConfirmPassword() {
        let confirmPasswordValue = $("#conpassword").val();
        let passwordValue = $("#password").val();
        if (passwordValue != confirmPasswordValue) {
                $("#conpasscheck").show();
                $("#conpasscheck").html("**Password didn't Match");
                $("#conpasscheck").css("color", "red");
                confirmPasswordError = false;
                return false;
        } else {
                $("#conpasscheck").hide();
        }
}

// Submit button
```

```
        $("#submitbtn").click(function () {
                validateUsername();
                validatePassword();
                validateConfirmPassword();
                validateEmail();
                if (
                        usernameError == true &&
                        passwordError == true &&
                        confirmPasswordError == true &&
                        emailError == true
                ) {
                        return true;
                } else {
                        return false;
                }
        });
});
```

5. Create below task in React

a. Create List of task in react using state

```
import React, { useState } from 'react';
import './App.css';
const App = () => {
  const [tasks, setTasks] = useState([
    { id: 1, text: 'Learn React' },
    { id: 2, text: 'Build a Todo List' },
    { id: 3, text: 'Style the Todo List' }
  ]);
  return (
    <div className="task-list-container">
      <h1 className="title">Task List</h1>
      <ul className="task-list">
        {tasks.map(task => (
          <li key={task.id} className="task-item">{task.text}</li>
        ))}
      </ul>
    </div>
  );
};

export default App;
```

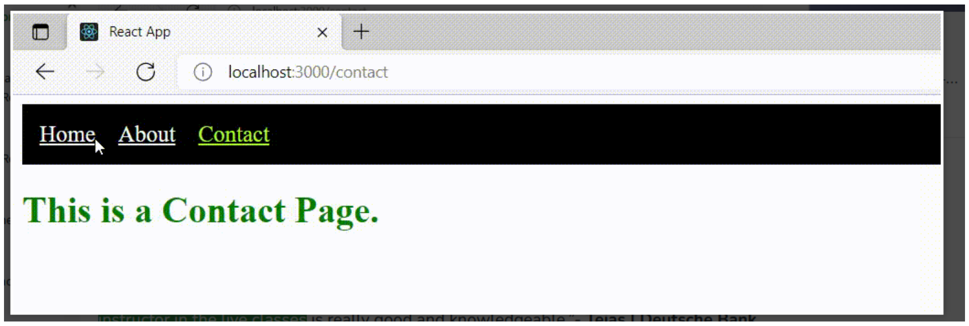
b. Style the all elements using css

```
.task-list-container {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-top: 50px;
}

.title {
  font-size: 2em;
  color: #333;
}

.task-list {
  list-style-type: none;
  padding: 0;
}

.task-item {
  background-color: #f9f9f9;
  border: 1px solid #ddd;
  padding: 10px 20px;
  margin: 10px 0;
  border-radius: 5px;
  width: 300px;
  text-align: left;
  font-size: 1.2em;
  color: #555;
  transition: background-color 0.3s;
}

.task-item:hover {
  background-color: #eaeaea;
}
```

6. Create different links like Home, About US ,Contact me, Details using react-Router and Navlink

```
// Filename - App.js

import React from "react";
import {
	BrowserRouter,
	Routes,
	Route,
	NavLink,
} from "react-router-dom";
import Home from "./components/Home";
import About from "./components/About";
import Contact from "./components/Contact";

function App() {
	return (
<>
    <BrowserRouter>
        <div
            style={{
                display: "flex",
                background: "black",
                padding: "5px 0 5px 5px",
                fontSize: "20px",
            }}
        >
            <div style={{ margin: "10px" }}>
                <NavLink
                    to="/"
                    style={({ isActive }) => ({
                        color: isActive
                            ? "greenyellow"
                            : "white",
                    })}
                >
                    Home
                </NavLink>
            </div>
            <div style={{ margin: "10px" }}>
                <NavLink
                    to="/about"
                    style={({ isActive }) => ({
                        color: isActive
                            ? "greenyellow"
                            : "white",
                    })}
                >
                    About
                </NavLink>
            </div>
            <div style={{ margin: "10px" }}>
                <NavLink
                    to="/contact"
                    style={({ isActive }) => ({
                        color: isActive
                            ? "greenyellow"
                            : "white",
                    })}
                >
                    Contact
                </NavLink>
            </div>
        </div>
```

```
                        <Routes>
                              <Route
                                    exact
                                    path="/"
                                    element={<Home />}
                              />
                              <Route
                                    exact
                                    path="/about"
                                    element={<About />}
                              />
                              <Route
                                    exact
                                    path="/contact"
                                    element={<Contact />}
                              />
                        </Routes>
                  </BrowserRouter>
            </>
      );
}

export default App;
```



This is a Contact Page.

7. Create API and display API in page on the basis of events.

```
import React, { useState } from 'react';
import './TaskFetcher.css';

const TaskFetcher = () => {
  const [tasks, setTasks] = useState([]);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(null);

  const fetchTasks = async () => {
    setLoading(true);
    setError(null);
    try {
      const response = await fetch('https://jsonplaceholder.typicode.com/todos?_limit=10');
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      const data = await response.json();
      setTasks(data);
    } catch (error) {
      setError(error.message);
    } finally {
      setLoading(false);
    }
  };

  return (
    <div className="task-fetcher-container">
      <h1 className="title">Task List</h1>
      <button className="fetch-button" onClick={fetchTasks}>
        Fetch Tasks
      </button>
      {loading && <p>Loading...</p>}
      {error && <p className="error">{error}</p>}
      <ul className="task-list">
        {tasks.map(task => (
          <li key={task.id} className="task-item">{task.title}</li>
        ))}
      </ul>
    </div>
  );
};

export default TaskFetcher;

.task-fetcher-container {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-top: 50px;
}

.title {
  font-size: 2em;
  color: #333;
}

.fetch-button {
  padding: 10px 20px;
  margin: 20px 0;
  font-size: 1em;
  cursor: pointer;
  background-color: #007bff;
  color: white;
  border: none;
  border-radius: 5px;
  transition: background-color 0.3s;
}

.fetch-button:hover {
  background-color: #0056b3;
}

.task-list {
  list-style-type: none;
  padding: 0;
}

.task-item {
  background-color: #f9f9f9;
  border: 1px solid #ddd;
  padding: 10px 20px;
  margin: 10px 0;
  border-radius: 5px;
  width: 300px;
  text-align: left;
  font-size: 1.2em;
  color: #555;
  transition: background-color 0.3s;
}

.task-item:hover {
  background-color: #eaeaea;
}

.error {
  color: red;
}
```

8.Create webpage to using REDUX

```
export const FETCH_TASKS_REQUEST = 'FETCH_TASKS_REQUEST';
export const FETCH_TASKS_SUCCESS = 'FETCH_TASKS_SUCCESS';
export const FETCH_TASKS_FAILURE = 'FETCH_TASKS_FAILURE';

export const fetchTasksRequest = () => ({
  type: FETCH_TASKS_REQUEST,
});
```

```
export const fetchTasksSuccess = tasks => ({
  type: FETCH_TASKS_SUCCESS,
  payload: tasks,
});

export const fetchTasksFailure = error => ({
  type: FETCH_TASKS_FAILURE,
  payload: error,
});

export const fetchTasks = () => {
  return async dispatch => {
    dispatch(fetchTasksRequest());
    try {
      const response = await fetch('https://jsonplaceholder.typicode.com/todos?_limit=10');
      const data = await response.json();
      dispatch(fetchTasksSuccess(data));
    } catch (error)
import { FETCH_TASKS_REQUEST, FETCH_TASKS_SUCCESS, FETCH_TASKS_FAILURE
} from './actions';

const initialState = {
  loading: false,
  tasks: [],
  error: '',
};

const taskReducer = (state = initialState, action) => {
  switch (action.type) {
    case FETCH_TASKS_REQUEST:
      return {
        ...state,
        loading: true,
      };
    case FETCH_TASKS_SUCCESS:
      return {
        loading: false,
        tasks: action.payload,
        error: '',
      };
    case FETCH_TASKS_FAILURE:
      return {
        loading: false,
        tasks: [],
        error: action.payload,
      };
    default:
      return state;
  }
};

export default taskReducer; {
      dispatch(fetchTasksFailure(error.message));
    }
  };
};
import { createStore, applyMiddleware } from 'redux';
import thunk from 'redux-thunk';
import { composeWithDevTools } from 'redux-devtools-extension';
import taskReducer from './reducers';

const store = createStore(taskReducer, composeWithDevTools(applyMiddleware(thunk)));

export default store;
import React, { useEffect } from 'react';
import { useDispatch, useSelector } from 'react-redux';
import { fetchTasks } from './redux/actions';
import './TaskList.css';

const TaskList = () => {
  const dispatch = useDispatch();
  const { loading, tasks, error } = useSelector(state => state);

  useEffect(() => {
    dispatch(fetchTasks());
  }, [dispatch]);

  return (
    <div className="task-list-container">
      <h1 className="title">Task List</h1>
      {loading && <p>Loading...</p>}
      {error && <p className="error">{error}</p>}
      <ul className="task-list">
        {tasks.map(task => (
          <li key={task.id} className="task-item">{task.title}</li>
        ))}
      </ul>
    </div>
  );
};

export default TaskList;
.task-list-container {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-top: 50px;
}

.title {
  font-size: 2em;
  color: #333;
}

.task-list {
  list-style-type: none;
  padding: 0;
}

.task-item {
  background-color: #f9f9f9;
  border: 1px solid #ddd;
  padding: 10px 20px;
  margin: 10px 0;
  border-radius: 5px;
  width: 300px;
  text-align: left;
  font-size: 1.2em;
  color: #555;
  transition: background-color 0.3s;
}

.task-item:hover {
  background-color: #eaeaea;
}

.error {
  color: red;
}
```