

Practical 7: Design the Basic Single Page Applications Using Angular or react for any Business

App.js

```
import {
  Button,
  Container,
  Text,
  Title,
  Modal,
  TextInput,
  Group,
  Card,
  ActionIcon,
  Code,
} from '@mantine/core';
import { useState, useRef, useEffect } from 'react';
import { MoonStars, Sun, Trash } from 'tabler-icons-react';

import {
  MantineProvider,
  ColorSchemeProvider,
  ColorScheme,
} from '@mantine/core';
import { useColorScheme } from '@mantine/hooks';
import { useHotkeys, useLocalStorage } from '@mantine/hooks';

export default function App() {
  const [tasks, setTasks] = useState([]);
  const [opened, setOpened] = useState(false);

  const preferredColorScheme = useColorScheme();
  const [colorScheme, setColorScheme] = useLocalStorage({
    key: 'mantine-color-scheme',
    defaultValue: 'light',
    getInitialValueInEffect: true,
  });
  const toggleColorScheme = value =>
    setColorScheme(value || (colorScheme === 'dark' ? 'light' : 'dark'));

  useHotkeys([['mod+J', () => toggleColorScheme()]]);

  const taskTitle = useRef("");
  const taskSummary = useRef("");

  function createTask() {
    setTasks([
      ...tasks,
      {
        title: taskTitle.current.value,
        summary: taskSummary.current.value,
      },
    ]);
  }
}
```

```

    });

    saveTasks([
      ...tasks,
      {
        title: taskTitle.current.value,
        summary: taskSummary.current.value,
      },
    ]);
  }

  function deleteTask(index) {
    var clonedTasks = [...tasks];

    clonedTasks.splice(index, 1);

    setTasks(clonedTasks);

    saveTasks([...clonedTasks]);
  }

  function loadTasks() {
    let loadedTasks = localStorage.getItem('tasks');

    let tasks = JSON.parse(loadedTasks);

    if (tasks) {
      setTasks(tasks);
    }
  }

  function saveTasks(tasks) {
    localStorage.setItem('tasks', JSON.stringify(tasks));
  }

  useEffect(() => {
    loadTasks();
  }, []);

  return (
    <ColorSchemeProvider
      colorScheme={colorScheme}
      toggleColorScheme={toggleColorScheme}>
      <MantineProvider
        theme={{ colorScheme, defaultRadius: 'md' }}
        withGlobalStyles
        withNormalizeCSS>
        <div className='App'>
          <Modal
            opened={opened}
            size={'md'}
            title={'New Task'}

```

```

withCloseButton={false}
onClose={() => {
  setOpened(false);
}}
centered>
<TextInput
  mt={'md'}
  ref={taskTitle}
  placeholder={'Task Title'}
  required
  label={'Title'}
/>
<TextInput
  ref={taskSummary}
  mt={'md'}
  placeholder={'Task Summary'}
  label={'Summary'}
/>
<Group mt={'md'} position={'apart'}>
  <Button
    onClick={() => {
      setOpened(false);
    }}
    variant={'subtle'}>
    Cancel
  </Button>
  <Button
    onClick={() => {
      createTask();
      setOpened(false);
    }}>
    Create Task
  </Button>
</Group>
</Modal>
<Container size={550} my={40}>
  <Group position={'apart'}>
    <Title
      sx={theme => ({
        fontFamily: `Greycliff CF, $
{theme.fontFamily}`,
        fontWeight: 900,
      })}>
    My Tasks
  </Title>
  <ActionIcon
    color={'blue'}
    onClick={() => toggleColorScheme()}
    size='lg'>
    {colorScheme === 'dark' ? (
      <Sun size={16} />
    ) : (

```

```

                                <MoonStars size={16} />
                                })
                                </ActionIcon>
                                </Group>
                                {tasks.length > 0 ? (
                                    tasks.map((task, index) => {
                                        if (task.title) {
                                            return (
                                                <Card withBorder
key={index} mt={'sm'}>
                                                    <Group
position={'apart'}>
                                                        <Text
weight={'bold'}>{task.title}</Text>
                                                            <ActionIcon
onClick={() => {
deleteTask(index);
color={'red'}
variant={'transparent'}>
<Trash />
</ActionIcon>
                                                    </Group>
<Text
color={'dimmed'} size={'md'} mt={'sm'}>
{task.summary}
task.summary
'No summary was provided for this task'
                                </Text>
                                </Card>
                                );
                                }
                                })
                                ) : (
                                    <Text size={'lg'} mt={'md'}
color={'dimmed'}>
                                        You have no tasks
                                    </Text>
                                )}
                                <Button
onClick={() => {
setOpened(true);

```

```

    }}
    fullWidth
    mt={'md'}>
    New Task
  </Button>
</Container>
</div>
</MantineProvider>
</ColorSchemeProvider>
);
}

```

index.js

```

import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

import {
  MantineProvider,
  ColorSchemeProvider,
  ColorScheme,
} from '@mantine/core';

ReactDOM.createRoot(document.getElementById('root')).render(<App />);

```

Practical 6: –Build a Responsive Web Pages Using HTML5, CSS3 and JavaScript

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Tic Tac Toe</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <nav>
      <ul>
        <li>MyTicTacToe.com</li>
      </ul>
    </nav>

    <div class="gameContainer">
      <div class="container">
        <div class="line"></div>
        <div class="box bt-0 bl-0"><span class="boxtext"></span></div>
        <div class="box bt-0"><span class="boxtext"></span></div>
        <div class="box bt-0 br-0"><span class="boxtext"></span></div>
        <div class="box bl-0"><span class="boxtext"></span></div>
        <div class="box"><span class="boxtext"></span></div>
        <div class="box br-0"><span class="boxtext"></span></div>
        <div class="box bl-0 bb-0"><span class="boxtext"></span></div>
        <div class="box bb-0"><span class="boxtext"></span></div>
        <div class="box bb-0 br-0"><span class="boxtext"></span></div>
      </div>
      <div class="gameInfo">
        <h1>Welcome to Tic Tac MyTicTacToe</h1>
        <div>
          <span class="info">Turn for X</span>
          <button id="reset">Reset</button>
        </div>
        <div class="imgbox">
          
        </div>
      </div>
    </div>
    <script src="script.js"></script>
  </body>
</html>
```

style.css

```
@import url('https://fonts.googleapis.com/css2?
family=Baloo+Bhaina+2&family=Roboto&display=swap');
```

```
* {
  margin: 0;
  padding: 0;
}
```

```
nav {
  background-color: rgb(37, 9, 37);
  color: white;
  height: 65px;
  font-size: 27px;
  display: flex;
  align-items: center;
  padding: 0 12px;
  font-family: 'Roboto', sans-serif;
}
```

```
nav ul {
  list-style-type: none;
}
```

```
.gameContainer {
  display: flex;
  justify-content: center;
  margin-top: 50px;
}
```

```
.container {
  display: grid;
  grid-template-rows: repeat(3, 10vw);
  grid-template-columns: repeat(3, 10vw);
  font-family: 'Roboto', sans-serif;
  position: relative;
}
```

```
.box {
  border: 2px solid black;
  font-size: 8vw;
  cursor: pointer;
  display: flex;
  justify-content: center;
  align-items: center;
}
```

```
.box:hover {
```

```
background-color: rgb(242, 234, 250);
}

.info {
  font-size: 22px;
}

.gameInfo {
  padding: 0 34px;
  font-family: 'Baloo Bhaina 2', cursive;
}

.gameInfo h1 {
  font-size: 2.5rem;
}

.imgbox img {
  width: 0;
  transition: width 1s ease-in-out;
}

.br-0 {
  border-right: 0;
}

.bl-0 {
  border-left: 0;
}

.bt-0 {
  border-top: 0;
}

.bb-0 {
  border-bottom: 0;
}

#reset {
  margin: 0 23px;
  padding: 1px 18px;
  background: #f3e7f9;
  border-radius: 6px;
  cursor: pointer;
  font-family: 'Baloo Bhaina 2';
  font-size: 15px;
  font-weight: bolder;
}

.line {
  background-color: black;
  height: 3px;
  width: 0;
}
```



```

position: absolute;
background-color: #911d91;
transition: width 1s ease-in-out;
}

@media screen and (max-width: 950px) {
  .gameContainer {
    flex-wrap: wrap;
  }
  .gameInfo {
    margin-top: 34px;
  }
  .gameInfo h1 {
    font-size: 1.5rem;
  }
  .container {
    grid-template-rows: repeat(3, 20vw);
    grid-template-columns: repeat(3, 20vw);
  }
}

```

script.js

```

console.log("Welcome to Tic Tac Toe");
let music = new Audio("music.mp3");
let audioTurn = new Audio("ting.mp3");
let gameover = new Audio("gameover.mp3");
let turn = "X";
let isgameover = false;

const changeTurn = () => {
  return turn === "X" ? "O" : "X";
};

const checkWin = () => {
  let boxtext = document.getElementsByClassName('boxtext');
  let wins = [
    [0, 1, 2, 5, 5, 0],
    [3, 4, 5, 5, 15, 0],
    [6, 7, 8, 5, 25, 0],
    [0, 3, 6, -5, 15, 90],
    [1, 4, 7, 5, 15, 90],
    [2, 5, 8, 15, 15, 90],
    [0, 4, 8, 5, 15, 45],
    [2, 4, 6, 5, 15, 135],
  ];
  wins.forEach(e => {
    if (
      boxtext[e[0]].innerText === boxtext[e[1]].innerText &&
      boxtext[e[2]].innerText === boxtext[e[1]].innerText &&
      boxtext[e[0]].innerText !== ""
    ) {

```

```

    document.querySelector('.info').innerText = boxtext[e[0]].innerText + " Won";
    isgameover = true;
    document.querySelector('.imgbox').getElementsByTagName('img')[0].style.width = "200px";
    document.querySelector(".line").style.transform = `translate(${e[3]}vw, ${e[4]}vw) rotate(${e[5]}deg)`;
    document.querySelector(".line").style.width = "20vw";
  }
});
};

let boxes = document.getElementsByClassName("box");
Array.from(boxes).forEach(element => {
  let boxtext = element.querySelector('.boxtext');
  element.addEventListener('click', () => {
    if (boxtext.innerText === "") {
      boxtext.innerText = turn;
      turn = changeTurn();
      audioTurn.play();
      checkWin();
      if (!isgameover) {
        document.getElementsByClassName("info")[0].innerText = "Turn for " + turn;
      }
    }
  });
});

reset.addEventListener('click', () => {
  let boxtexts = document.querySelectorAll('.boxtext');
  Array.from(boxtexts).forEach(element => {
    element.innerText = "";
  });
  turn = "X";
  isgameover = false;
  document.querySelector(".line").style.width = "0vw";
  document.getElementsByClassName("info")[0].innerText = "Turn for " + turn;
  document.querySelector('.imgbox').getElementsByTagName('img')[0].style.width = "0px";
});

```