



Experiment 4

Student Name: Harleen Kaur

Branch: BE CSE

Semester: 6th

Subject Name: Full Stack Development-II

UID: 23BCS10857

Section/Group: 23BCSKRG_3A

Date of Performance: 03/02/26

Subject Code: 23CSH-309

1. Aim:

To optimize the performance of the EcoTrack React application using memoization techniques and code splitting, and to enhance the user interface using enterprise-grade Material UI components.

2. Objective: After completing this experiment, the student will be able to:

- Understand the causes of unnecessary re-renders in React applications
- Optimize React components using React.memo to prevent avoidable re-renders
- Apply useMemo to efficiently compute derived data and avoid redundant calculations
- Use useCallback to memoize event handler functions and improve component performance
- Implement lazy loading of components and routes using React.lazy and Suspense
- Reduce initial bundle size and improve application load performance through code splitting
- Enhance the visual appearance and usability of the EcoTrack application using Material UI components
- Design a clean, consistent, and responsive user interface using Material UI layouts and typography

3. Implementation/Code:

- src/App.jsx - Added React.lazy and Suspense for code splitting
- src/components/Header.jsx - Added React.memo, useCallback, and Material UI components



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- src/pages/Logs.jsx - Added useMemo for filtering and Material UI components
- src/pages/dashboard.jsx - Added useMemo, useCallback, and Material UI components
- src/pages/login.jsx - Added useCallback and Material UI components
- src/routes/ProtectedRoute.jsx - Added React.memo and cleaned up formatting

App.jsx:

```
import { useMemo, memo, useCallback } from "react";
import { useDispatch, useSelector } from "react-redux";
import { fetchLogs } from "../store/logsSlice";
import { useEffect } from "react";
import {
  Container,
  Card,
  CardContent,
  Typography,
  List,
  ListItem,
  ListItemText,
  Chip,
  Box,
  CircularProgress,
  Alert,
  Grid,
} from "@mui/material";
import TrendingDownIcon from "@mui/icons-material/TrendingDown";
import TrendingUpIcon from "@mui/icons-material/TrendingUp";

const Logs = memo(() => {
  const dispatch = useDispatch();
  const { data, status, error } = useSelector((state) => state.logs);

  useEffect(() => {
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
    if (status === "idle") {  
      dispatch(fetchLogs());  
    }  
  }, [status, dispatch]);
```

```
// Memoized filtered data - only recomputes when data changes  
const { highImpactLogs, lowImpactLogs } = useMemo(() => {  
  return {  
    highImpactLogs: data.filter((log) => log.carbon > 4),  
    lowImpactLogs: data.filter((log) => log.carbon <= 4),  
  };  
}, [data]);
```

```
const renderLogList = useCallback(  
  (logs, isHighImpact) => (  
    <List sx={{ width: "100%" }}>  
      {logs.map((log) => (  
        <ListItem key={log.id} sx={{ paddingY: 1.5 }}>  
          <ListItemText  
            primary={log.activity}  
            secondary={`Carbon Footprint: ${log.carbon} kg CO2`}  
            primaryTypographyProps={{  
              fontWeight: 600,  
              color: isHighImpact ? "#d32f2f" : "#2ecc71",  
            }}  
            secondaryTypographyProps={{  
              color: "textSecondary",  
            }}  
          />  
          <Chip  
            icon={isHighImpact ? <TrendingUpIcon /> : <TrendingDownIcon />}  
            label={` ${log.carbon} kg`}  
            color={isHighImpact ? "error" : "success"}  
            variant="outlined"  
            sx={{ marginLeft: 2 }}  
          />  
        </ListItem>  
      )}
```



```
    )})
  </List>

),
[]
);

if (status === "loading") {
  return (
    <Container maxWidth="lg" sx={{ paddingY: 4 }}>
      <Box sx={{ display: "flex", justifyContent: "center", paddingY: 4 }}>
        <CircularProgress />
      </Box>
    </Container>
  );
}

if (status === "failed") {
  return (
    <Container maxWidth="lg" sx={{ paddingY: 4 }}>
      <Alert severity="error">
        Error loading logs: {error || "Unknown error"}
      </Alert>
    </Container>
  );
}

return (
  <Container maxWidth="lg" sx={{ paddingY: 4 }}>
    <Box sx={{ marginBottom: 4 }}>
      <Typography
        variant="h4"
        component="h1"
        sx={{
          marginBottom: 3,
          fontWeight: 700,
          color: "#2ecc71",
        }}
      />
    </Box>
  </Container>
);
```





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
        </Typography>
      )}
    </CardContent>
  </Card>
</Grid>

{/* Low Impact Activities */}
<Grid item xs={12} md={6}>
  <Card sx={{ height: "100%", boxShadow: 2 }}>
    <CardContent>
      <Box sx={{ display: "flex", alignItems: "center", gap: 1 }}>
        <TrendingDownIcon sx={{ color: "#2ecc71", fontSize: 28 }} />
        <Typography
          variant="h6"
          sx={{
            fontWeight: 700,
            color: "#2ecc71",
          }}
        >
          Low Impact Activities ( {lowImpactLogs.length} )
        </Typography>
      </Box>
      <Typography
        variant="body2"
        sx={{
          color: "textSecondary",
          marginBottom: 2,
          marginTop: 1,
        }}
      >
        Activities with carbon footprint  $\leq 4$  kg CO2
      </Typography>
      {lowImpactLogs.length > 0 ? (
        renderLogList(lowImpactLogs, false)
      ) : (
        <Typography variant="body2" sx={{ color: "textSecondary" }}>
          No low impact activities found.
        </Typography>
      )}
    </CardContent>
  </Card>
</Grid>
```



```
        </Typography>
      )}
    </CardContent>
  </Card>
</Grid>
</Grid>
</Container>
);
});
```

```
Logs.displayName = "Logs";
```

```
export default Logs;
```

dashboard.jsx:

```
import { useDispatch, useSelector } from 'react-redux'; import { useEffect,
useMemo, useCallback } from 'react'; import { fetchLogs } from
'../store/logSlice'; import { Box, Typography, List, ListItem, Button,
CircularProgress, Paper } from
  '@mui/material'; import RefreshIcon from
  '@mui/icons-material/Refresh';
```

```
const Dashboard = () => {  const dispatch = useDispatch();
const { data: logs, status } = useSelector((state) => state.logs);
```

```
  useEffect(() => {    if
(status === 'idle') {
    dispatch(fetchLogs());
  }
}, [status, dispatch]);
```

```
  // useCallback memoizes the function to prevent recreation on every render
const handleRefresh = useCallback(() => {    dispatch(fetchLogs());
}, [dispatch]);
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
if (status === 'loading') {  
  return (  
    <Box sx={{ display: 'flex', justifyContent: 'center', alignItems: 'center', height:  
      '50vh' }}>  
      <CircularProgress />  
    </Box>  
  );  
}
```

```
// useMemo prevents recalculating totals on every render  
const totalActivities = useMemo(() => {      return  
  logs.reduce((acc, log) => {      acc[log.activity] =  
    (acc[log.activity] || 0) + log.carbon;      return acc;  
    }, {});  
}, [logs]);
```

```
// useMemo for filtering operations    const highCarbon = useMemo(() =>  
logs.filter(log => log.carbon > 4), [logs]);    const lowCarbon = useMemo(() =>  
logs.filter(log => log.carbon <= 4), [logs]);
```

```
return (  
  <Box sx={{ p: 3 }}>  
    <Paper elevation={3} sx={{ p: 3, mb: 3 }}>  
      <Typography variant="h4" gutterBottom>Total Activities</Typography>  
      <List>  
        {Object.entries(totalActivities).map(([activity, carbon]) => (  
          <ListItem key={activity}>  
            <Typography>{activity}: {carbon} Kg CO2</Typography>  
          </ListItem>  
        ))}  
      </List>  
    </Paper>  
  
    <Paper elevation={3} sx={{ p: 2, mb: 2, backgroundColor: '#ffebee' }}>  
      <Typography variant="h5" sx={{ color: '#d32f2f', mb: 1 }}>High Carbon  
> 4 Kg</Typography>  
      <List>
```



```
{highCarbon.map((log) => (  
  <ListItem key={log.id}>  
    <Typography>{log.activity}</Typography>  
  </ListItem>  
))}  
</List>  
</Paper>
```

```

<Paper elevation={3} sx={{ p: 2, mb: 3, backgroundColor: '#e8f5e9' }}>
  <Typography variant="h5" sx={{ color: '#2e7d32', mb: 1 }}>Low Carbon ( $\leq 4$ 
Kg)</Typography>
  <List>
    {lowCarbon.map((log) => (
      <ListItem key={log.id}>
        <Typography>{log.activity}</Typography>
      </ListItem>
    ))}
  </List>
</Paper>

```

```

<Button
  variant="contained"
  color="primary"
  startIcon={<RefreshIcon />}
  onClick={handleRefresh}
  disabled={status
    === 'loading'}
  >
    {status === 'loading' ? 'Loading...' : 'Refresh Logs'}
  </Button>
</Box>
);
};

```

```
export default Dashboard;
```



4. Output:



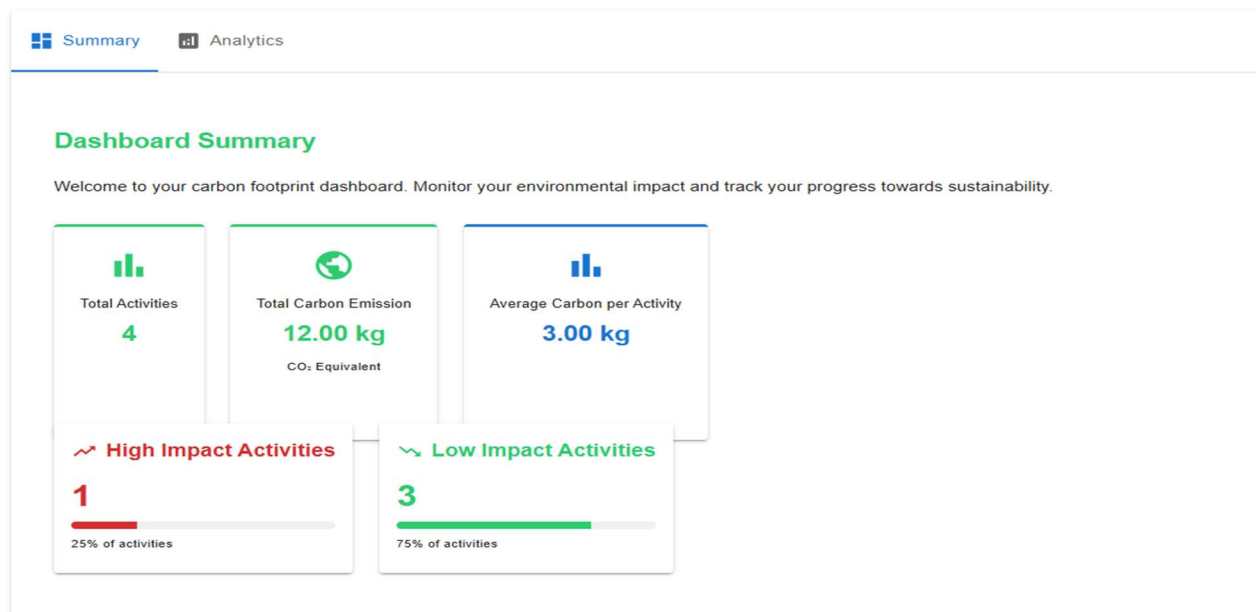
Welcome to EcoTrack

Track your carbon footprint and make a positive impact on the environment.

[Login to EcoTrack](#)



Dashboard





Dashboard

Summary

Analytics

Dashboard Analytics

Detailed analysis of your carbon footprint across different activities.

Activities Ranked by Carbon Impact

Rank	Activity	Carbon Footprint (kg CO ₂)	Impact Level	% of Total
#1	Electricity Usage	6	Medium	50.0%
#2	Car Travel	4	Medium	33.3%
#3	Bus Travel	2	Low	16.7%
#4	Cycling	0	Low	0.0%

Highest Impact Activity

6 kg

Lowest Impact Activity

0 kg

Total Activities

4

Total Carbon

12.00 kg

Carbon Footprint Activities

High Impact Activities (1)

Activities with carbon footprint > 4 kg CO₂

Electricity Usage

Carbon Footprint: 6 kg CO₂

6 kg

Low Impact Activities (3)

Activities with carbon footprint ≤ 4 kg CO₂

Car Travel

Carbon Footprint: 4 kg CO₂

4 kg

Cycling

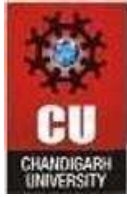
Carbon Footprint: 0 kg CO₂

0 kg

Bus Travel

Carbon Footprint: 2 kg CO₂

2 kg



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

5 Learning Outcome:

- Identify and reduce unnecessary re-renders in React applications.
- Optimize React components using React.memo, useMemo, and useCallback.
- Implement lazy loading and code splitting to improve application performance.
- Enhance overall React app efficiency through performance optimization techniques.
- Design a clean, responsive user interface using Material UI components.