

BUDGET TRACKER

DYLAN MANANSALA #2174746

EVAN SHARP #1320206

ANASTASSIA TARASSOVA #1277712

AGENDA

Introduction

Demo

Structure

Challenges

What We Have Learnt

WHY THIS APP?

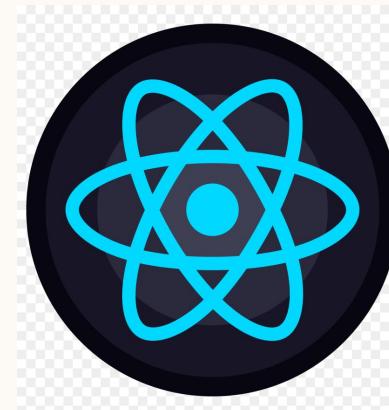
- * PERTINENT TOOL
- * REAL USER STORIES
- * DIVERSE REACT UI ELEMENTS
- * NUMEROUS EXAMPLES OF SIMILAR APPS



INSPIRATION



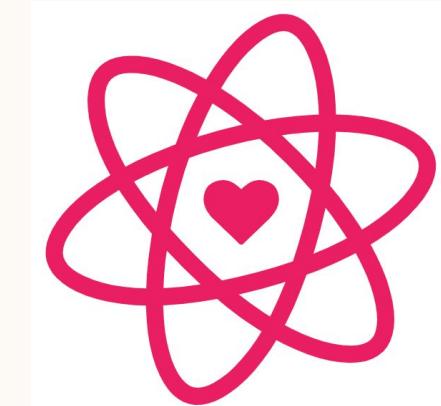
TOOLS USED



React



<https://mui.com/material-ui/>



React-Icons

EXTERNAL API

Financial news API from AlphaVantage

```
export default function Dashboard() {
  const apiKey = process.env.REACT_APP_ALPHA_KEY;
  const [news, setNews] = useState([]);
  // TODO: add user
  const [username, setUsername] = useState(".....");

  // using the news API from Alpha Vantage
  // https://www.alphavantage.co/support/#api-key

  useEffect(() => {
    const fetchNews = async () => {
      try {
        const response = await axios.get('https://www.alphavantage.co/query', {
          params: {
            function: 'NEWS_SENTIMENT',
            topics: 'economy_fiscal',
            apikey: apiKey,
          },
        });
        setNews(response.data.feed || []);
      } catch (error) {
        console.error("Error fetching news:", error);
      }
    };

    fetchNews();
  }, [apiKey]);
```



DEMO

Enjoy!



HOME PAGE

Home page:

- Divided into sections,
- Hero section includes headline and call to action
- Icons imported from Lucide React

Take Control of Your Finances

Effortlessly manage your budget and take strides toward achieving financial freedom

[Start Your Journey →](#) [▷ Watch Demo](#)

Monthly Overview

Total Balance
\$12,847

Income	+\$4,200
Expenses	-\$2,847
Savings	+\$1,353

Bank-level Security Real-time Sync Smart Insights

LOGIN/REGISTRATION

Login

 Email * Password *[Forgot password?](#)[LOGIN](#)

 OR

[!\[\]\(fc3a57079704ef1b99671c8cafae23be_img.jpg\) SIGN IN WITH GOOGLE](#)

Register

 Full Name * Email * Password * Confirm Password *[REGISTER](#)

 OR

[!\[\]\(e97636a3328cdaccd5ffd8fe3bc69ce6_img.jpg\) REGISTER WITH GOOGLE](#)

Already have an account? [Login](#)

Login and Register:

- Provides an auth context
- Uses Tailwind and MUI

Something I learned:

```
const isAuthenticated = !!user;
```

- Never considered the possibility of double negative
- Type cast to boolean
- Usable in dynamic typing languages

Budget Planner

11

Plan Your Monthly Budget

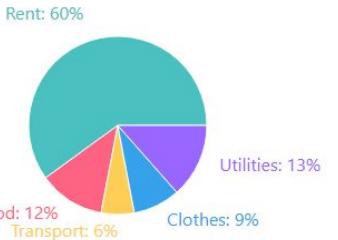
Allocate your income across different categories

Total Monthly Income

\$ 2500

CATEGORY	AMOUNT (\$)	% OF INCOME
Rent	1500	60%
Food	300	12%
Transport	150	6%
Clothes	213	9%
Utilities	337	13%

Budget Visualization



Total Income
\$2500.00

Total Allocated
\$2500.00
100.0% of income

Save Budget

Reset

User Input

Total Monthly Income

\$ 2500

CATEGORY	AMOUNT (\$)	% OF INCOME
🏠 Rent	1500	60%
🍴 Food	300	12%
🚌 Transport	150	6%
👕 Clothes	213	9%
💡 Utilities	337	13%

⌄ BudgetForm
⚙️ BudgetForm.jsx
⚙️ CategoryRow.jsx

🏠 Rent ⌂ 167% ⓘ

Visual Feedback

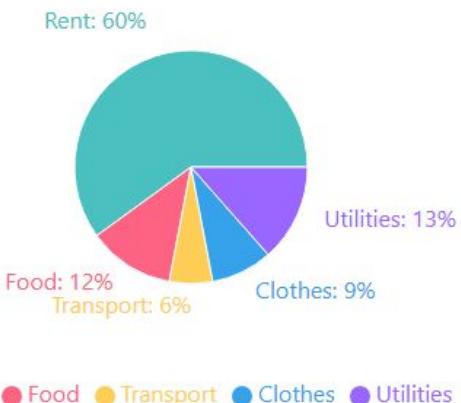
Budget Visualization



Enter your budget to see the visualization

Start by adding your income and category amounts

Budget Visualization



Total Income

\$2500.00

Total Allocated

\$2500.00

↪ 100.0% of income

Total Income

\$2400.00

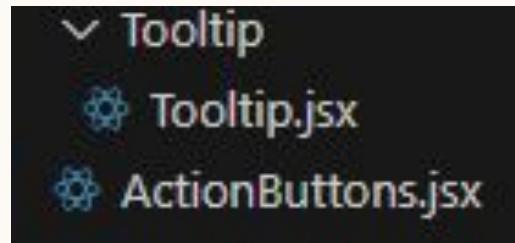
Total Allocated

\$2500.00

↪ 104.2% of income

Rent
Amount: \$1500.00
60% of income

Extras

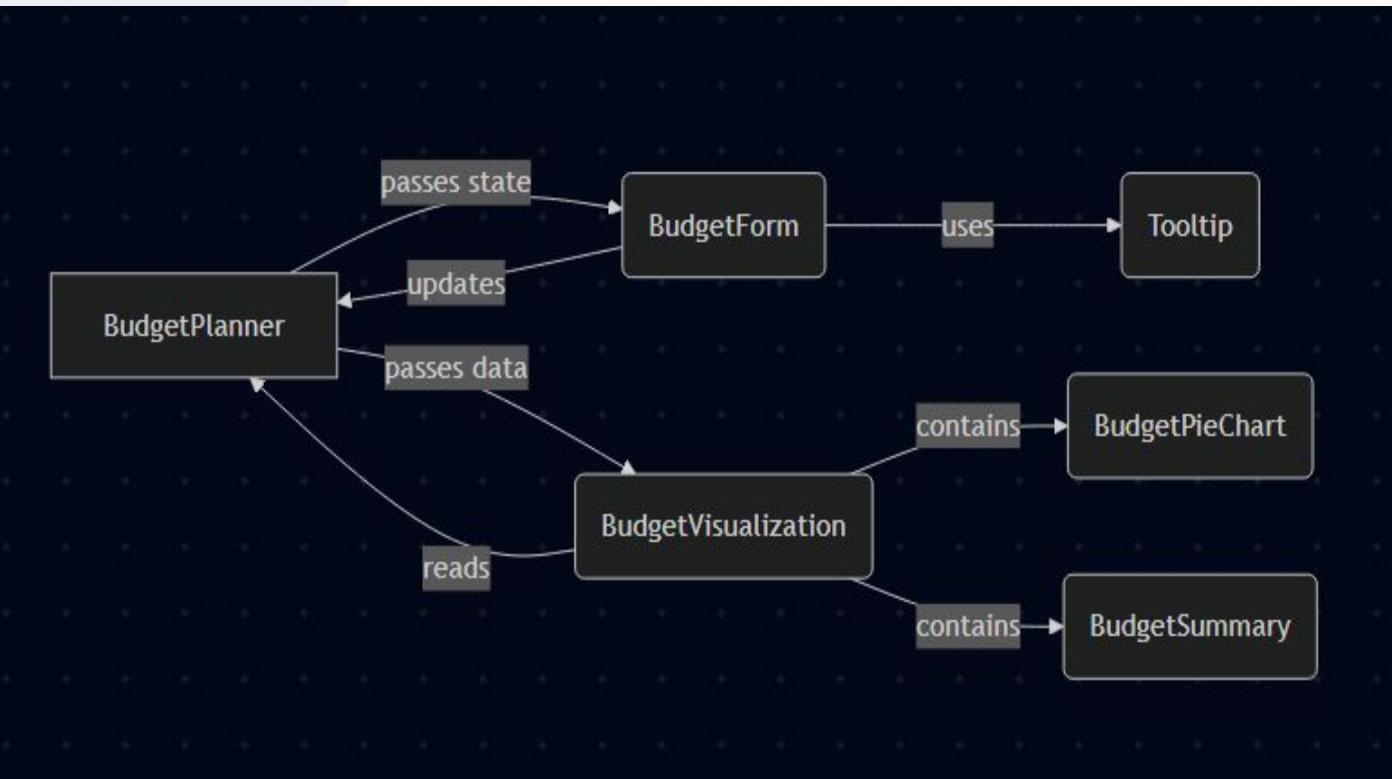


Start by entering your total monthly take-home income, then distribute it across categories. The percentages will calculate automatically. Try to keep total allocations at or below 100% of your income for a balanced budget.

Save Budget

Reset

index.js (Main component)



```
✓ components
  ✓ BudgetPlanner
    ✓ BudgetForm
      ✎ BudgetForm.jsx
      ✎ CategoryRow.jsx
  ✓ BudgetVisualization
    ✎ BudgetPieChart.jsx
    ✎ BudgetSummary.jsx
    ✎ BudgetVisualization.jsx
  ✓ Tooltip
    ✎ Tooltip.jsx
    ✎ ActionButtons.jsx
  JS index.js
```

SPENDING TRACKER

Track your
spending effortlessly

Visualize where your money goes and keep tabs on your spending habits over time.

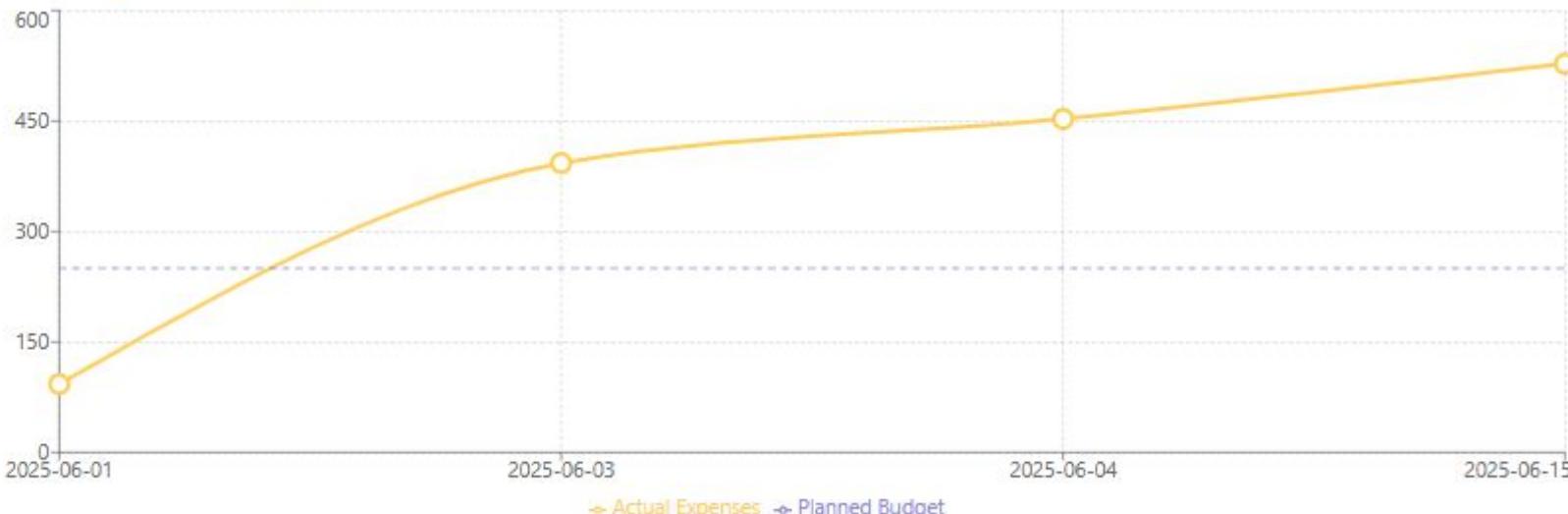
 Hide Budget Comparison

Manage Budget →

Your expenses for 1 Month in category Transport is \$527.75. You're above budget.

Category:  Transport ▾

Time Range: 1 Month ▾



SPENDING TRACKER

Add a New Expense

Track your budget by logging every purchase as it happens.

Recent Expenses

Date	Name	Category	Price	Actions
2025-06-15	Onigiri	🍴 Food	\$15.00	Delete
2025-06-15	Gas	🚗 Transport	\$75.00	Delete
2025-06-15	July Rent (early)	🏡 Rent	\$1200.00	Delete
2025-06-14	Coffee	🍴 Food	\$3.00	Delete
2025-06-14	Pashmina Scarf	🛍 Clothes	\$30.00	Delete
2025-06-14	Plumbing Repairs	⽔ Utilities	\$275.00	Delete

WHAT WE LEARNT

Using plugins from Material UI

```
11
12 // MaterialUI date adapter
13 import { LocalizationProvider } from '@mui/x-date-pickers';
14 import { AdapterDayjs } from '@mui/x-date-pickers/AdapterDayjs';
15
16 function App() {
```

Using plugins from Rechart

```
1 import React, { useState } from 'react';
2 import { ResponsiveContainer, LineChart, Line, XAxis, YAxis,
3 |   CartesianGrid, Tooltip, Legend } from 'recharts';
```

```

/* Chart */
<ResponsiveContainer width="100%" height={400}>
  <LineChart data={chartData[selectedCategory] || []}>
    <CartesianGrid strokeDasharray="3 3" />
    <XAxis dataKey="name" />
    <YAxis />
    <Tooltip
      formatter={(value, name) => {
        if (name === 'Planned Budget') {
          return [`${value}`, 'Planned Budget (gray dashed)'];
        }
        return [`${value}`, name];
      }}
    />
    <Legend />
    <Line
      type="monotone"
      dataKey="value"
      stroke={categoryColors[selectedCategory]}
      strokeWidth={3}
      dot={{ r: 7 }}
      name="Actual Expenses"
    />


```

WHAT WE LEARNT

Building graphs with Rechart



WHAT WE LEARNT

Building tables with { DataGrid } and {Box} from MUI

20

Recent Expenses

Date	Name	Category	Price	Actions
2025-06-15	Onigiri	Food	\$15.00	Delete
2025-06-15	Gas	Transport	\$75.00	Delete
2025-06-15	July Rent (early)	Rent	\$1200.00	Delete
2025-06-14	Coffee	Food		
2025-06-14	Pashmina Scarf	Clothes		
2025-06-14	Plumbing Repairs	Utilities		
2025-06-12	Birthday Cake	Food		
2025-06-11	Coffee	Food		
2025-06-07	Sneakers	Clothes		

Rows per page:

```
    {
      field: 'amount',
      headerName: 'Price',
      width: 120,
      editable: true, // enable editing
      renderCell: (params) => {
        if (params.value === null || params.value === undefined) return '';
        return `$$ {Number(params.value).toFixed(2)}`;
      },
      renderEditCell: (params) => {
        return (
          <input
            type="number"
            step="0.01"
            value={params.value}
            onChange={(e) => {
              params.api.setEditCellValue({ id: params.id, field: 'amount', value: e.target.value });
            }}
            style={{ width: '100%', boxSizing: 'border-box' }}
          />
        );
      },
    },
```

WHAT WE LEARNT

Dark/light mode with MUI

App.js

```

16 import { ColorModeProvider } from './context/ColorModeContext';
17 function App() {
18   return (
19     <ColorModeProvider>
20       <div className="min-h-screen flex flex-col">
21         <header>
22           <Navbar />
23         </header>
24         <main className="flex-grow">
25           <Routes>
26             <Route path="/" element={<Home />} />
27             <Route path="/features" element={<Features />} />
28             <Route path="/login" element={<Login />} />
29             <Route path="/register" element={<Register />} />
30             <Route
31               path="/spending"
32               element={
33                 <LocalizationProvider dateAdapter={AdapterDayjs}>
34                   <SpendingTracker />
35                 </LocalizationProvider>
36               }
37             />
38             <Route path="/budget" element={<ActualBudget />} />
39             <Route path="/dashboard" element={<Dashboard />} />
40             <Route path="/planner" element={<BudgetPlanner />} />
41           </Routes>
42         </main>
43         <Footer />
44       </div>
45     </ColorModeProvider>

```

ColorModeContext.js

```

group5budgettracker > src > context > js ColorModeContext.js > ...
1 import { createContext, useMemo, useState, useContext } from "react";
2 import { createTheme, ThemeProvider } from "@mui/material/styles";
3 import CssBaseline from "@mui/material/CssBaseline";
4
5 const ColorModeContext = createContext();
6
7 export const useColorMode = () => useContext(ColorModeContext);
8
9 export const ColorModeProvider = ({ children }) => {
10   const [mode, setMode] = useState("light");
11
12   // toggler
13   const toggleColorMode = () => {
14     setMode((prevMode) => (prevMode === "light" ? "dark" : "light"));
15   };
16
17   const theme = useMemo(
18     () =>
19       createTheme({
20         palette: {
21           mode,
22           ...(mode === "light"
23             ? {
24               background: {
25                 default: "#f5f5f5",
26               },
27             }
28             : {
29               background: {
30                 default: "#1a237e",
31               },
32             }
33           }
34         }
35       })
36
37   return (
38     <ThemeProvider theme={theme}>
39       {children}
40     </ThemeProvider>
41   );
42 }

```

CHALLENGES

1. Choosing an API

Solution: Alpha Vantage

2. Importing and combining numerous libraries

Solution: proofreading

3. Application of DRY principles

Solution: use of Components, Effects and Contexts

4. Applying dark/light themes to all of the components

Solution: meticulous one-by-one approach

5. Variable port usage (3000, 3001, 3002)

Solution: proxy use

```
    "version": "0.1.0",
    "private": true,
    "proxy": "http://localhost:3001",
    "dependencies": {
      "@date-io/datetime": "^3.2.0"
    }
  }
```



ULTERIOR ITERATIONS

- More preventive validation (e.g. limit spending to a daily limit)
- Connection to a DB
- Identification with social media (eg. Google account)
- Connecting to banking services

**THANK YOU
FOR LISTENING**