

Introduction

In an era where financial management plays an increasingly vital role in personal and professional success, the need for intuitive and efficient tools to track expenses and analyze spending habits has never been greater. Recognizing this need, our project endeavors to provide a comprehensive solution for expense management, offering users a seamless experience to monitor their finances, gain valuable insights, and make informed decisions.

Project Overview

Our project is a web application built on modern web technologies, including React.js and Tailwind CSS, designed to empower users in their journey towards financial wellness. With a user-centric approach, we aim to simplify the often complex task of expense tracking by providing an intuitive and visually appealing interface coupled with powerful analytical tools.

Purpose and Scope

The primary objective of our project is to offer users a centralized platform to manage their expenses efficiently, irrespective of their financial goals or lifestyle preferences. Whether it's keeping track of daily expenditures, analyzing spending patterns over time, or setting budgetary goals, our application strives to cater to the diverse needs of individuals seeking financial clarity and control.

Key Features

- **Expense Tracking:** Users can effortlessly record their expenses, categorize transactions, and organize financial data for easy reference.
- **Analytics Dashboard:** Our application offers a comprehensive analytics dashboard, providing users with insights into their spending habits, trends, and patterns.
- **Customization Options:** From customizable budget categories to personalized spending goals, users have the flexibility to tailor the application to suit their unique requirements.
- **Responsive Design:** With a focus on accessibility and usability, our application is designed to adapt seamlessly to various screen sizes and devices, ensuring a consistent experience across platforms.

Target Audience

Our project caters to a broad spectrum of users, ranging from individuals seeking to manage personal finances more effectively to businesses looking to streamline expense tracking for their teams. Whether you're a budget-conscious consumer, a

freelance professional, or a small business owner, our application aims to provide value by simplifying the complexities of financial management.

Next Steps

As we embark on this journey to revolutionize expense management, our focus remains on continual improvement and innovation. In the subsequent sections of this report, we delve deeper into the technical aspects of our project, exploring the architecture, implementation details, and future roadmap. Join us as we navigate the intersection of technology and finance, empowering individuals and businesses to take control of their financial destiny.

Technologies Used

Our project leverages a diverse array of cutting-edge technologies to deliver a robust and feature-rich expense management solution. From front-end frameworks to backend infrastructure, each component has been carefully selected to ensure optimal performance, scalability, and user experience.

Front-End Technologies

React.js

At the heart of our project lies React.js, a popular JavaScript library for building user interfaces. React's component-based architecture facilitates the creation of modular and reusable UI elements, enabling rapid development and easy maintenance. Leveraging React's declarative syntax and virtual DOM, we've crafted a dynamic and interactive user experience that responds seamlessly to user inputs and data changes.

Tailwind CSS

For styling and design, we've embraced Tailwind CSS, a utility-first CSS framework that offers unparalleled flexibility and customization. With Tailwind's intuitive utility classes and responsive design features, we've been able to create visually stunning interfaces that adapt effortlessly to different screen sizes and device types. By leveraging Tailwind's extensive utility classes, we've minimized the need for custom CSS, resulting in cleaner, more maintainable code.

Backend Technologies

Node.js

Powering the backend of our application is Node.js, a lightweight and efficient runtime environment for JavaScript. Node.js's non-blocking I/O model and event-driven architecture make it ideal for building scalable and high-performance server-side applications. Leveraging Node.js, we've developed robust APIs and backend services that handle data processing, authentication, and business logic with ease.

Express.js

Complementing Node.js is Express.js, a minimalist web framework for Node.js that simplifies the process of building web applications and APIs. Express.js provides a wide range of middleware and routing capabilities, allowing us to create modular and extensible backend services. With Express.js, we've implemented RESTful APIs that enable seamless communication between the client-side and server-side components of our application.

Database Technologies

MongoDB

For data storage and management, we've opted for MongoDB, a NoSQL database that offers scalability, flexibility, and performance. MongoDB's document-oriented data model and schema-less design make it well-suited for handling unstructured or semi-structured data, such as expense records and user profiles. Leveraging MongoDB, we've implemented a robust data persistence layer that ensures reliable storage and retrieval of financial data.

Certainly! Here's a detailed overview of the key features of your expense management application:

Key Features

1. Dashboard Overview

The dashboard serves as the central hub for users to gain insights into their financial health at a glance. It provides a comprehensive overview of key metrics such as total expenses, spending trends, and budget summaries. Through intuitive data visualizations and interactive charts, users can quickly assess their spending patterns and identify areas for optimization.

2. Expense Tracking

Our application offers robust expense tracking capabilities, allowing users to effortlessly record and categorize their transactions. Users can add new expenses manually or import them automatically from linked bank accounts or credit cards. Each expense entry includes details such as amount, category, date, and optional notes, providing users with a detailed record of their financial activities.

3. Budget Management

Effective budget management is essential for maintaining financial stability and achieving long-term goals. Our application enables users to set personalized budgets for different expense categories and track their progress over time. Users can establish

spending limits, receive notifications for approaching budget thresholds, and adjust budgets dynamically based on changing financial priorities.

4. Customizable Reports

Understanding spending patterns and identifying areas for improvement is crucial for effective financial planning. Our application generates customizable reports and analytics that empower users to gain deeper insights into their spending habits. From monthly expense summaries to category-wise breakdowns, users can visualize their financial data in meaningful ways and make informed decisions about their finances.

5. Goal Setting(WIP)

Setting financial goals is an essential step towards achieving financial independence and security. Our application allows users to define short-term and long-term goals, such as saving for a vacation, purchasing a home, or building an emergency fund. Users can track their progress towards these goals, receive motivational reminders, and celebrate milestones along the way.

6. Reminders and Notifications(WIP)

To help users stay on track with their financial goals and commitments, our application provides customizable reminders and notifications. Users can set reminders for upcoming bill payments, budget reviews, and financial milestones, ensuring that important tasks are not overlooked. Notifications can be delivered via email, SMS, or in-app alerts, according to user preferences.

7. Secure Authentication and Data Privacy

Protecting user data and ensuring privacy are top priorities for our application. We employ industry-standard security practices, including encryption, authentication, and authorization mechanisms, to safeguard user information from unauthorized access or tampering. User data is stored securely in encrypted databases, and stringent access controls are in place to prevent data breaches or leaks.

8. Multi-Platform Accessibility

Our application is designed to be accessible across multiple platforms and devices, ensuring seamless user experiences regardless of the user's preferred device or operating system. Whether accessing the application from a desktop computer, laptop, tablet, or smartphone, users can enjoy consistent functionality and performance, thanks to responsive design principles and cross-platform compatibility.

Tailwind CSS Configuration

Tailwind CSS is a utility-first CSS framework that enables rapid and efficient development of modern web applications. Its highly customizable nature and extensive

set of utility classes make it a popular choice among developers for styling web interfaces. Let's delve into the configuration of Tailwind CSS for your expense management application:

1. Dark Mode Configuration

Tailwind CSS supports dark mode out of the box, allowing users to toggle between light and dark themes seamlessly. In your configuration, you've specified "**class**" as the dark mode variant, enabling dark mode activation based on the presence of the **dark** class on the root HTML element.

2. Content Paths

The **content** option in your Tailwind CSS configuration specifies the paths to the files that contain CSS classes used in your project. This configuration ensures that Tailwind's purging process includes only the necessary CSS classes, optimizing the final build size and performance.

3. Theme Customization

Tailwind CSS offers extensive theme customization capabilities, allowing you to define custom color palettes, spacing scales, typography settings, and more. In your configuration, you've extended the default theme with custom color definitions for borders, inputs, backgrounds, text, and other UI elements. By tailoring the theme to match your application's design system, you ensure consistency and coherence across your UI components.

4. Container Configuration

The **container** option allows you to customize the styling of the container component, which is used to constrain the width of content on the page. In your configuration, you've set **center: true** to horizontally center the container content and specified a default padding of **2rem**. Additionally, you've defined custom screen breakpoints, such as "**2xl**", to adjust the maximum width of the container for larger screens.

5. Extended Utilities

Tailwind CSS provides a wide range of utility classes out of the box, but it also allows for extending these utilities to meet specific project requirements. In your configuration, you've extended utility classes for colors, border radii, and animation keyframes, enhancing the flexibility and expressiveness of your stylesheets.

6. Tailwind CSS Plugins

Tailwind CSS offers a plugin ecosystem that extends its core functionality with additional features and utilities. In your configuration, you've included the **tailwindcss-animate** plugin, which provides pre-defined animation keyframes for use in your

application. By leveraging plugins, you can enhance your development workflow and add advanced styling capabilities without sacrificing performance or maintainability.

Conclusion

Tailwind CSS configuration plays a crucial role in shaping the visual appearance and behavior of your web application. By customizing Tailwind's theme, extending utility classes, and leveraging plugins, you can create a tailored styling solution that meets the unique requirements of your expense management application. With Tailwind CSS, you benefit from a powerful and flexible CSS framework that empowers you to build beautiful and responsive user interfaces with ease.

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  darkMode: ["class"],
  content: [
    "./pages/**/*.jsx",
    "./components/**/*.jsx",
    "./app/**/*.jsx",
    "./src/**/*.jsx",
  ],
  prefix: "",
  theme: {
    container: {
      center: true,
      padding: "2rem",
      screens: {
        "2xl": "1400px",
      },
    },
    extend: {
      colors: {
        border: "hsl(var(--border))",
        input: "hsl(var(--input))",
        ring: "hsl(var(--ring))",
        background: "hsl(var(--background))",
        foreground: "hsl(var(--foreground))",
        primary: {
          DEFAULT: "hsl(var(--primary))",
          foreground: "hsl(var(--primary-foreground))",
        },
        secondary: {
          DEFAULT: "hsl(var(--secondary))",
          foreground: "hsl(var(--secondary-foreground))",
        },
        destructive: {
          DEFAULT: "hsl(var(--destructive))",
          foreground: "hsl(var(--destructive-foreground))",
        },
        muted: {
          DEFAULT: "hsl(var(--muted))",
          foreground: "hsl(var(--muted-foreground))",
        },
      },
    },
  },
}
```

```

    },
    accent: {
      DEFAULT: "hsl(var(--accent))",
      foreground: "hsl(var(--accent-foreground))",
    },
    popover: {
      DEFAULT: "hsl(var(--popover))",
      foreground: "hsl(var(--popover-foreground))",
    },
    card: {
      DEFAULT: "hsl(var(--card))",
      foreground: "hsl(var(--card-foreground))",
    },
  },
  borderRadius: {
    lg: "var(--radius)",
    md: "calc(var(--radius) - 2px)",
    sm: "calc(var(--radius) - 4px)",
  },
  keyframes: {
    "accordion-down": {
      from: { height: "0" },
      to: { height: "var(--radix-accordion-content-height)" },
    },
    "accordion-up": {
      from: { height: "var(--radix-accordion-content-height)" },
      to: { height: "0" },
    },
  },
  animation: {
    "accordion-down": "accordion-down 0.2s ease-out",
    "accordion-up": "accordion-up 0.2s ease-out",
  },
},
plugins: [require("tailwindcss-animate")],
};

```

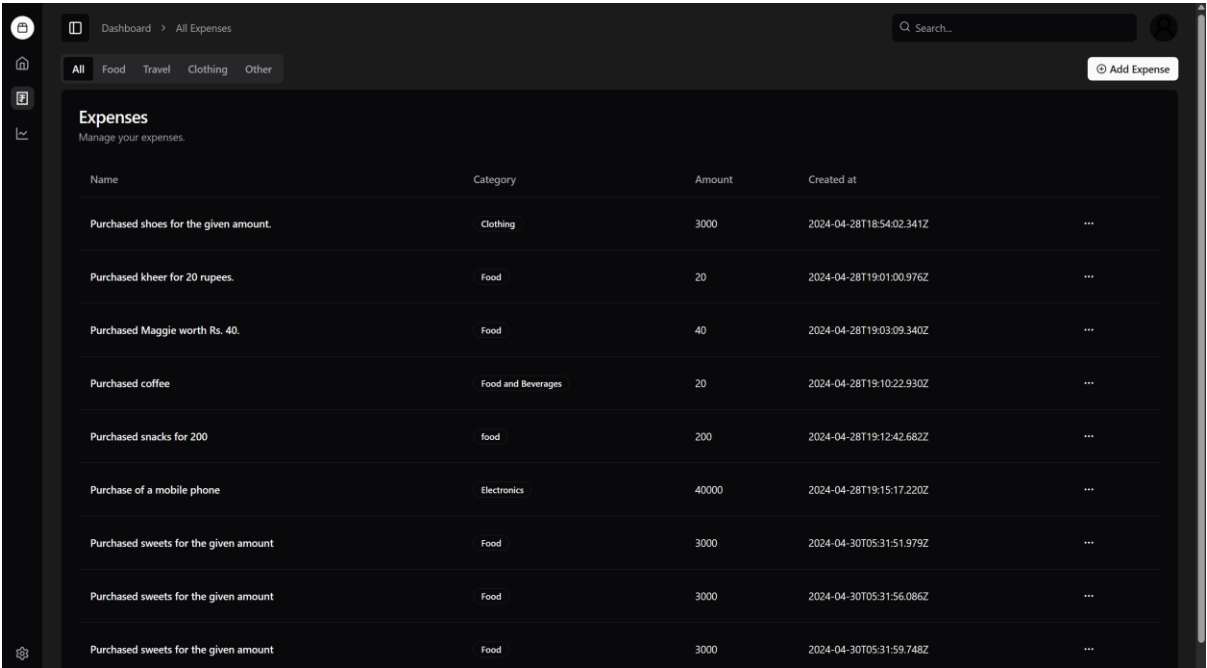
Routes Overview

Your expense management application features several routes to facilitate user interaction and navigation. Each route serves a distinct purpose within the application, catering to various user needs and workflows. Let's explore the key functionalities and features offered by each route:

1. Dashboard

The Dashboard route serves as the central hub of your expense management application, providing users with an overview of their financial data and activities. Upon accessing the dashboard, users are greeted with personalized insights, such as total

expenses, spending trends, and budget summaries. Through intuitive data visualizations and interactive elements, users can quickly assess their financial health and make informed decisions. Additionally, the dashboard may include actionable items, such as expense categorization tools, budget planning features, and reminders for upcoming payments.



```
import {
  Home,
  LineChart,
  MoreHorizontal,
  Package2,
  PanelLeft,
  PlusCircle,
  Search,
  Settings,
  ReceiptIndianRupee,
  BadgeIndianRupee,
} from "lucide-react";
import TableRowCustom from "../TableRow";
import { Badge } from "@components/ui/badge";
import {
  Breadcrumb,
  BreadcrumbItem,
  BreadcrumbLink,
  BreadcrumbList,
  BreadcrumbPage,
  BreadcrumbSeparator,
} from "@components/ui/breadcrumb";
import { Button } from "@components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardFooter,
```



```

    CardHeader,
    CardTitle,
  } from "@components/ui/card";
import {
  DropdownMenu,
  DropdownMenuCheckboxItem,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuTrigger,
} from "@components/ui/dropdown-menu";
import { Input } from "@components/ui/input";
import { Sheet, SheetContent, SheetTrigger } from "@components/ui/sheet";
import {
  Table,
  TableBody,
  TableCell,
  TableHead,
  TableHeader,
  TableRow,
} from "@components/ui/table";
import { Tabs, TabsContent, TabsList, TabsTrigger } from
"@components/ui/tabs";
import {
  Tooltip,
  TooltipProvider,
  TooltipContent,
  TooltipTrigger,
} from "@components/ui/tooltip";
import axios from "axios";
import React, { useState, useEffect } from "react";
import { FloatingForm } from "../AddExpenseForm";
export function Dashboard() {
  const [expenses, setExpenses] = useState([]);
  useEffect(() => {
    const fetchExpenses = async () => {
      try {
        const response = await axios.get(
          "http://localhost:3000/expense/getExpenses",
          {
            withCredentials: true, // Include cookies in the request
          }
        );
        setExpenses(response.data.expenses);
      } catch (error) {
        console.error("Error fetching expenses:", error);
      }
    };

    fetchExpenses();
  }, []);
  return (
    <div className="flex min-h-screen w-full flex-col bg-muted/40 dark">

```

```

<aside className="fixed inset-y-0 left-0 z-10 hidden w-14 flex-col
border-r bg-background sm:flex">
  <nav className="flex flex-col items-center gap-4 px-2 sm:py-5">
    <a
      href="#"
      className="group flex h-9 w-9 shrink-0 items-center justify-center
gap-2 rounded-full bg-primary text-lg font-semibold text-primary-foreground
md:h-8 md:w-8 md:text-base"
    >
      <Package2 className="h-4 w-4 transition-all group-hover:scale-110"
/>
      <span className="sr-only">Budget Buddy</span>
    </a>
    <TooltipProvider>
      <Tooltip>
        <TooltipTrigger asChild>
          <div>
            <a
              href="/home"
              className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
            >
              <Home className="h-5 w-5" />
              <span className="sr-only">Dashboard</span>
            </a>
          </div>
        </TooltipTrigger>
        <TooltipContent side="right">Dashboard</TooltipContent>
      </Tooltip>
    </TooltipProvider>
    <TooltipProvider>
      <Tooltip>
        <TooltipTrigger asChild>
          <div>
            {" "}
            <a
              href="/dashboard"
              className="flex h-9 w-9 items-center justify-center
rounded-lg bg-accent text-accent-foreground transition-colors hover:text-
foreground md:h-8 md:w-8"
            >
              <ReceiptIndianRupee className="h-5 w-5" />
              <span className="sr-only">Expenses</span>
            </a>
          </div>
        </TooltipTrigger>
        <TooltipContent side="right">Expenses</TooltipContent>
      </Tooltip>
    </TooltipProvider>
    <TooltipProvider>
      <Tooltip>
        <TooltipTrigger asChild>
          <div>

```

```

        <a
            href="/analytics"
            className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
        >
            <LineChart className="h-5 w-5" />
            <span className="sr-only">Analytics</span>
        </a>
    </div>
</TooltipTrigger>
<TooltipContent side="right">Analytics</TooltipContent>
</Tooltip>
</TooltipProvider>
</nav>
<nav className="mt-auto flex flex-col items-center gap-4 px-2 sm:py-5">
    <TooltipProvider>
        <Tooltip>
            <TooltipTrigger asChild>
                <div>
                    <a
                        href="#"
                        className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
                    >
                        <Settings className="h-5 w-5" />
                        <span className="sr-only">Settings</span>
                    </a>
                </div>
            </TooltipTrigger>
            <TooltipContent side="right">Settings</TooltipContent>
        </Tooltip>
    </TooltipProvider>
</nav>
</aside>
<div className="flex flex-col sm:gap-4 sm:py-4 sm:pl-14">
    <header className="sticky top-0 z-30 flex h-14 items-center gap-4
border-b bg-background px-4 sm:static sm:h-auto sm:border-0 sm:bg-transparent
sm:px-6">
        <Sheet>
            <SheetTrigger asChild>
                <Button size="icon" variant="outline">
                    <PanelLeft className="h-5 w-5" />
                    <span className="sr-only">Toggle Menu</span>
                </Button>
            </SheetTrigger>
            <SheetContent side="left" className="sm:max-w-xs dark">
                <nav className="grid gap-6 text-lg font-medium">
                    <a
                        href="/"
                        className="group flex h-10 w-10 shrink-0 items-center
justify-center gap-2 rounded-full bg-primary text-lg font-semibold text-
primary-foreground md:text-base"
                    >

```

```

        >
        <Package2 className="h-5 w-5 transition-all group-
hover:scale-110" />
        <span className="sr-only">Budget Buddy</span>
      </a>
      <a
        href="/home"
        className="flex items-center gap-4 px-2.5 text-muted-
foreground hover:text-foreground"
      >
        <Home className="h-5 w-5" />
        Dashboard
      </a>
      <a
        href="/expenses"
        className="flex items-center gap-4 px-2.5 text-muted-
foreground hover:text-foreground"
      >
        <BadgeIndianRupee className="h-5 w-5" />
        Expenses
      </a>
      <a
        href="/analytics"
        className="flex items-center gap-4 px-2.5 text-muted-
foreground hover:text-foreground"
      >
        <LineChart className="h-5 w-5" />
        Analytics
      </a>
    </nav>
  </SheetContent>
</Sheet>
<Breadcrumb className="hidden md:flex">
  <BreadcrumbList>
    <BreadcrumbItem>
      <BreadcrumbLink asChild>
        <a href="/home">Dashboard</a>
      </BreadcrumbLink>
    </BreadcrumbItem>
    <BreadcrumbSeparator />
    <BreadcrumbItem>
      <BreadcrumbLink asChild>
        <a href="/dashboard">All Expenses</a>
      </BreadcrumbLink>
    </BreadcrumbItem>
  </BreadcrumbList>
</Breadcrumb>
<div className="relative ml-auto flex-1 md:grow-0">
  <Search className="absolute left-2.5 top-2.5 h-4 w-4 text-muted-
foreground" />
  <Input
    type="search"
    placeholder="Search..."
  >

```

```

        className="w-full rounded-lg bg-background pl-8 md:w-[200px]
lg:w-[336px]"
    />
</div>
<DropdownMenu>
  <DropdownMenuTrigger asChild>
    <Button
      variant="outline"
      size="icon"
      className="overflow-hidden rounded-full"
    >
      
    </Button>
  </DropdownMenuTrigger>
  <DropdownMenuContent align="end">
    <DropdownMenuLabel>My Account</DropdownMenuLabel>
    <DropdownMenuSeparator />
    <DropdownMenuItem>Settings</DropdownMenuItem>
    <DropdownMenuItem>Support</DropdownMenuItem>
    <DropdownMenuSeparator />
    <DropdownMenuItem>Logout</DropdownMenuItem>
  </DropdownMenuContent>
</DropdownMenu>
</header>
<main className="grid flex-1 items-start gap-4 p-4 sm:px-6 sm:py-0
md:gap-8">
  <Tabs defaultValue="all">
    <div className="flex items-center">
      <TabsList>
        <TabsTrigger value="all">All</TabsTrigger>
        <TabsTrigger value="food">Food</TabsTrigger>
        <TabsTrigger value="travel">Travel</TabsTrigger>
        <TabsTrigger value="clothing">Clothing</TabsTrigger>
        <TabsTrigger value="other" className="hidden sm:flex">
          Other
        </TabsTrigger>
      </TabsList>
      <div className="ml-auto flex items-center gap-2">
        <FloatingForm />
      </div>
    </div>
    <TabsContent value="all">
      <Card x-chunk="dashboard-06-chunk-0">

```

```

    <CardHeader>
      <CardTitle>Expenses</CardTitle>
      <CardDescription>Manage your expenses.</CardDescription>
    </CardHeader>
    <CardContent>
      <Table>
        <TableHeader>
          <TableRow>
            <TableHead>Name</TableHead>
            <TableHead>Category</TableHead>
            <TableHead className="hidden md:table-cell">
              Amount
            </TableHead>
            <TableHead className="hidden md:table-cell">
              Created at
            </TableHead>
            <TableHead>
              <span className="sr-only">Actions</span>
            </TableHead>
          </TableRow>
        </TableHeader>
        <TableBody>
          {expenses.map((expense) => (
            <TableRowCustom
              eId={expense.eId}
              key={expense.eId}
              description={expense.description}
              category={expense.category}
              amount={expense.amount}
              date={expense.date}
            </>
          ))}
        </TableBody>
      </Table>
    </CardContent>
  </Card>
</TabsContent>
<TabsContent value="food">
  <Card x-chunk="dashboard-06-chunk-0">
    <CardHeader>
      <CardTitle>Expenses</CardTitle>
      <CardDescription>Manage your expenses.</CardDescription>
    </CardHeader>
    <CardContent>
      <Table>
        <TableHeader>
          <TableRow>
            <TableHead>Name</TableHead>
            <TableHead>Category</TableHead>
            <TableHead className="hidden md:table-cell">
              Amount
            </TableHead>
            <TableHead className="hidden md:table-cell">
              Created at

```

```

        </TableHead>
        <TableHead>
          <span className="sr-only">Actions</span>
        </TableHead>
      </TableRow>
    </TableHeader>
    <TableBody>
      {expenses
        .filter((expense) =>
          expense.category.toLowerCase().includes("food")
        )
        .map((expense) => (
          <TableRowCustom
            eId={expense.eId}
            key={expense.eId}
            description={expense.description}
            category={expense.category}
            amount={expense.amount}
            date={expense.date}
          />
        ))}
    </TableBody>
  </Table>
</CardContent>
</Card>
</TabsContent>
<TabsContent value="travel">
  <Card x-chunk="dashboard-06-chunk-0">
    <CardHeader>
      <CardTitle>Expenses</CardTitle>
      <CardDescription>Manage your expenses.</CardDescription>
    </CardHeader>
    <CardContent>
      <Table>
        <TableHeader>
          <TableRow>
            <TableHead>Name</TableHead>
            <TableHead>Category</TableHead>
            <TableHead className="hidden md:table-cell">
              Amount
            </TableHead>
            <TableHead className="hidden md:table-cell">
              Created at
            </TableHead>
            <TableHead>
              <span className="sr-only">Actions</span>
            </TableHead>
          </TableRow>
        </TableHeader>
        <TableBody>
          {expenses
            .filter((expense) =>
              expense.category.toLowerCase().includes("travel")
            )

```

```

        .map((expense) => (
            <TableRowCustom
                eId={expense.eId}
                key={expense.eId}
                description={expense.description}
                category={expense.category}
                amount={expense.amount}
                date={expense.date}
            />
        ))}
    </TableBody>
</Table>
</CardContent>
</Card>
</TabsContent>
<TabsContent value="other">
    <Card x-chunk="dashboard-06-chunk-0">
        <CardHeader>
            <CardTitle>Expenses</CardTitle>
            <CardDescription>Manage your expenses.</CardDescription>
        </CardHeader>
        <CardContent>
            <Table>
                <TableHeader>
                    <TableRow>
                        <TableHead>Name</TableHead>
                        <TableHead>Category</TableHead>
                        <TableHead className="hidden md:table-cell">
                            Amount
                        </TableHead>
                        <TableHead className="hidden md:table-cell">
                            Created at
                        </TableHead>
                        <TableHead>
                            <span className="sr-only">Actions</span>
                        </TableHead>
                    </TableRow>
                </TableHeader>
                <TableBody>
                    {expenses
                        .filter(
                            (expense) =>
                                !expense.category.toLowerCase().includes("food") &&
                                !expense.category.toLowerCase().includes("travel")
                        )
                        .map((expense) => (
                            <TableRowCustom
                                eId={expense.eId}
                                key={expense.eId}
                                description={expense.description}
                                category={expense.category}
                                amount={expense.amount}
                                date={expense.date}
                            />

```



```

    )))
    </TableBody>
  </Table>
</CardContent>
</Card>
</TabsContent>
<TabsContent value="clothing">
  <Card x-chunk="dashboard-06-chunk-0">
    <CardHeader>
      <CardTitle>Expenses</CardTitle>
      <CardDescription>Manage your expenses.</CardDescription>
    </CardHeader>
    <CardContent>
      <Table>
        <TableHeader>
          <TableRow>
            <TableHead>Name</TableHead>
            <TableHead>Category</TableHead>
            <TableHead className="hidden md:table-cell">
              Amount
            </TableHead>
            <TableHead className="hidden md:table-cell">
              Created at
            </TableHead>
            <TableHead>
              <span className="sr-only">Actions</span>
            </TableHead>
          </TableRow>
        </TableHeader>
        <TableBody>
          {expenses
            .filter((expense) =>
              expense.category.toLowerCase().includes("clothing")
            )
            .map((expense) => (
              <TableRowCustom
                eId={expense.eId}
                key={expense.eId}
                description={expense.description}
                category={expense.category}
                amount={expense.amount}
                date={expense.date}
              />
            ))}
        </TableBody>
      </Table>
    </CardContent>
  </Card>
</TabsContent>
</Tabs>
</main>
</div>
</div>
);

```

```
}
```

2. Login

The Login route enables users to authenticate and access their accounts within the expense management application. Through a secure authentication process, users can verify their identities and gain access to their personalized dashboards and financial data. The login page typically includes input fields for entering credentials, such as username/email and password, along with options for password recovery or account registration. By implementing robust authentication mechanisms, such as multi-factor authentication (MFA) or OAuth integration, you enhance the security and integrity of user accounts.

Login

Enter your email below to login to your account.

Email

Password

Sign in

```
import React, { useState } from "react";
import axios from "axios";
import { Button } from "@components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardFooter,
  CardHeader,
```

```

    CardTitle,
  } from "@components/ui/card";
import { Input } from "@components/ui/input";
import { Label } from "@components/ui/label";
export function LoginForm() {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleSubmit = async (e) => {
    e.preventDefault();

    try {
      const response = await axios.post(
        "http://localhost:3000/login",
        { email, password },
        { withCredentials: true }
      );
      // Handle the response from the server
      console.log(response.data);
      window.location.href = "/home";
    } catch (error) {
      // Handle any errors that occurred during the request
      console.error("Error:", error);
    }
  };

  return (
    <div className="w-full h-screen flex items-center justify-center px-4">
      <Card className="w-full max-w-sm dark">
        <CardHeader>
          <CardTitle className="text-2xl">Login</CardTitle>
          <CardDescription>
            Enter your email below to login to your account.
          </CardDescription>
        </CardHeader>
        <form onSubmit={handleSubmit}>
          <CardContent className="grid gap-4">
            <div className="grid gap-2">
              <Label htmlFor="email">Email</Label>
              <Input
                id="email"
                type="email"
                placeholder="email@example.com"
                required
                value={email}
                onChange={(e) => setEmail(e.target.value)}
              />
            </div>
            <div className="grid gap-2">
              <Label htmlFor="password">Password</Label>
              <Input
                id="password"
                type="password"
                required

```

```

        placeholder="Your Password"
        value={password}
        onChange={(e) => setPassword(e.target.value)}
      />
    </div>
  </CardContent>
  <CardFooter>
    <Button type="submit" className="w-full">
      Sign in
    </Button>
  </CardFooter>
</form>
</Card>
</div>
);
}

```

3. Signup

The Signup route allows new users to create accounts and register for the expense management application. During the signup process, users provide essential information, such as their name, email address, and password, to create a personalized account. Additionally, the signup page may include optional fields for additional profile details or preferences. To streamline the signup experience, you can incorporate features like social media authentication, email verification, and progressive form completion. By guiding users through the registration process and offering clear instructions, you increase user engagement and conversion rates.

Sign Up

Enter your information to create an account

First name

Last name

Email

Password

Create an account

Sign up with Google

Already have an account? [Sign in](#)

```
import React, { useState } from "react";
import axios from "axios";
import { Button } from "@components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardHeader,
  CardTitle,
} from "@components/ui/card";
import { Input } from "@components/ui/input";
import { Label } from "@components/ui/label";

export default function Signup() {
  const [formData, setFormData] = useState({
    firstName: "",
    lastName: "",
    email: "",
    password: "",
  });

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData({
```

```

    ...formData,
    [name]: value,
  });
};

const handleSubmit = async (e) => {
  e.preventDefault();

  try {
    // Send POST request with form data
    const response = await axios.post("http://localhost:3000/signup", {
      name: `${formData.firstName} ${formData.lastName}`,
      email: formData.email,
      password: formData.password,
    });

    console.log("Server response:", response.data);
    // Reset form state
    setFormData({
      firstName: "",
      lastName: "",
      email: "",
      password: "",
    });
    // Redirect to login page
    window.location.href = "/login";
  } catch (error) {
    console.error("Error submitting form:", error);
  }
};

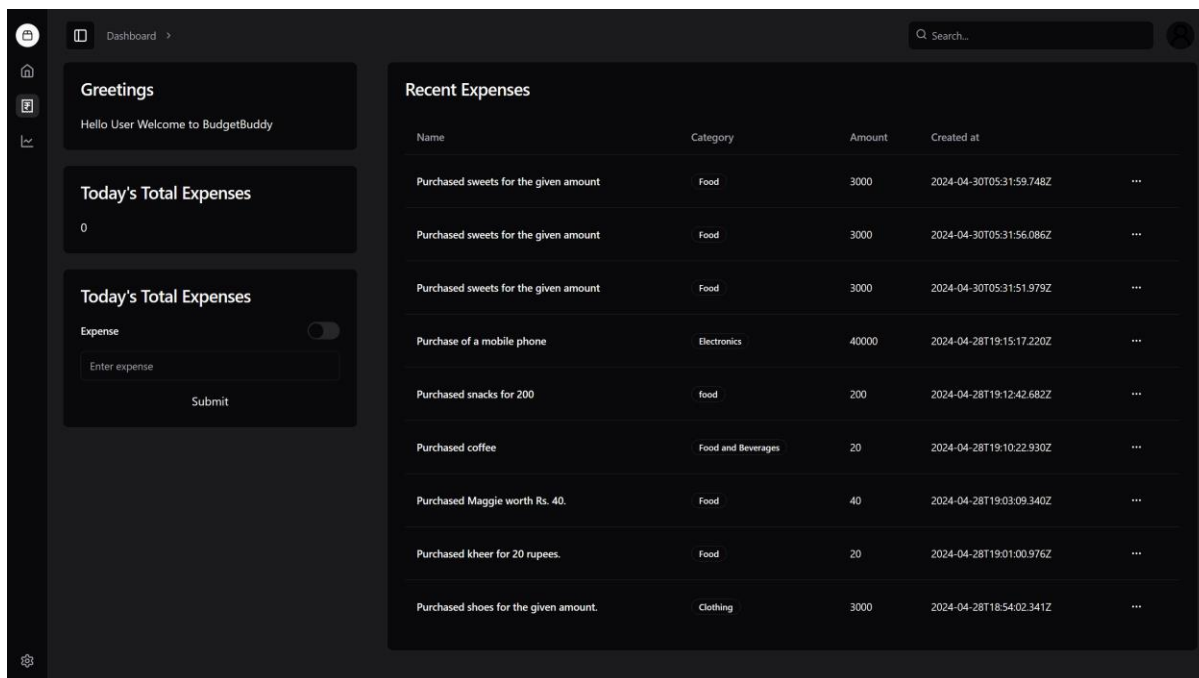
return (
  <div className="w-full h-screen flex items-center justify-center px-4">
    <form onSubmit={handleSubmit} className="w-full max-w-sm">
      <Card className="mx-auto dark">
        <CardHeader>
          <CardTitle className="text-xl">Sign Up</CardTitle>
          <CardDescription>
            Enter your information to create an account
          </CardDescription>
        </CardHeader>
        <CardContent>
          <div className="grid gap-4">
            <div className="grid grid-cols-2 gap-4">
              <div className="grid gap-2">
                <Label htmlFor="firstName">First name</Label>
                <Input
                  id="firstName"
                  name="firstName"
                  value={formData.firstName}
                  onChange={handleChange}
                  placeholder="Your"
                  required
                />

```


}

4. Home

The Home route serves as the landing page or entry point for users navigating to the expense management application. This page provides users with a curated selection of relevant content, updates, and announcements tailored to their interests and preferences. The home page may feature highlights from the dashboard, personalized recommendations, or promotional offers to engage users and encourage exploration. Additionally, the home page may include navigation elements, search functionality, and quick access links to key features and sections of the application. By optimizing the home page layout and content, you create a welcoming and informative experience for users.



```
import {
  Home,
  LineChart,
  MoreHorizontal,
  Package2,
  PanelLeft,
  PlusCircle,
  Search,
  Settings,
  ReceiptIndianRupee,
  BadgeIndianRupee,
  Car,
} from "lucide-react";
import TableRowCustom from "../TableRow";
import { Badge } from "@components/ui/badge";
```



```

import {
  Breadcrumb,
  BreadcrumbItem,
  BreadcrumbLink,
  BreadcrumbList,
  BreadcrumbPage,
  BreadcrumbSeparator,
} from "@components/ui/breadcrumb";
import { Button } from "@components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardFooter,
  CardHeader,
  CardTitle,
} from "@components/ui/card";
import {
  DropdownMenu,
  DropdownMenuCheckboxItem,
  DropdownMenuContent,
  DropdownMenuItem,
  DropdownMenuLabel,
  DropdownMenuSeparator,
  DropdownMenuTrigger,
} from "@components/ui/dropdown-menu";
import { Input } from "@components/ui/input";
import { Sheet, SheetContent, SheetTrigger } from "@components/ui/sheet";
import {
  Table,
  TableBody,
  TableCell,
  TableHead,
  TableHeader,
  TableRow,
} from "@components/ui/table";
import { Tabs, TabsContent, TabsList, TabsTrigger } from
"@components/ui/tabs";
import {
  Tooltip,
  TooltipProvider,
  TooltipContent,
  TooltipTrigger,
} from "@components/ui/tooltip";
import axios from "axios";
import React, { useState, useEffect } from "react";
import { FloatingForm } from "../AddExpenseForm";
import ExpenseForm from "../AdvancedForm";
export default function Homepage() {
  const [expenses, setExpenses] = useState([]);
  useEffect(() => {
    const fetchExpenses = async () => {
      try {
        const response = await axios.get(

```

```

        "http://localhost:3000/expense/getExpenses",
        {
            withCredentials: true, // Include cookies in the request
        }
    );
    setExpenses(response.data.expenses);
} catch (error) {
    console.error("Error fetching expenses:", error);
}
};

fetchExpenses();
}, []));
const [todayTotal, setTodayTotal] = useState(0);
useEffect(() => {
    const todayExpenses = expenses.filter((expense) => {
        const expenseDate = new Date(expense.date);
        const today = new Date();
        return (
            expenseDate.getDate() === today.getDate() &&
            expenseDate.getMonth() === today.getMonth() &&
            expenseDate.getFullYear() === today.getFullYear()
        );
    });
    const todayTotal = todayExpenses.reduce(
        (total, expense) => total + expense.amount,
        0
    );
    console.log(todayTotal);
    setTodayTotal(todayTotal);
}, [expenses]);
return (
    <div className="flex min-h-screen w-full flex-col bg-muted/40 dark">
        <aside className="fixed inset-y-0 left-0 z-10 hidden w-14 flex-col border-r bg-background sm:flex">
            <nav className="flex flex-col items-center gap-4 px-2 sm:py-5">
                <a
                    href="/home"
                    className="group flex h-9 w-9 shrink-0 items-center justify-center gap-2 rounded-full bg-primary text-lg font-semibold text-primary-foreground md:h-8 md:w-8 md:text-base">
                        <Package2 className="h-4 w-4 transition-all group-hover:scale-110"/>
                        <span className="sr-only">Acme Inc</span>
                    </a>
                <TooltipProvider>
                    <Tooltip>
                        <TooltipTrigger asChild>
                            <div>
                                <a
                                    href="/home"

```

```

                className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
            >
                <Home className="h-5 w-5" />
                <span className="sr-only">Home</span>
            </a>
        </div>
    </TooltipTrigger>
    <TooltipContent side="right">Home</TooltipContent>
</Tooltip>
</TooltipProvider>
<TooltipProvider>
    <Tooltip>
        <TooltipTrigger asChild>
            <div>
                <a
                    href="/dashboard"
                    className="flex h-9 w-9 items-center justify-center
rounded-lg bg-accent text-accent-foreground transition-colors hover:text-
foreground md:h-8 md:w-8"
                >
                    <ReceiptIndianRupee className="h-5 w-5" />
                    <span className="sr-only">Dashboard</span>
                </a>
            </div>
        </TooltipTrigger>
        <TooltipContent side="right">Dashboard</TooltipContent>
    </Tooltip>
</TooltipProvider>
<TooltipProvider>
    <Tooltip>
        <TooltipTrigger asChild>
            <div>
                <a
                    href="/analytics"
                    className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
                >
                    <LineChart className="h-5 w-5" />
                    <span className="sr-only">Analytics</span>
                </a>
            </div>
        </TooltipTrigger>
        <TooltipContent side="right">Analytics</TooltipContent>
    </Tooltip>
</TooltipProvider>
</nav>
<nav className="mt-auto flex flex-col items-center gap-4 px-2 sm:py-5">
    <TooltipProvider>
        <Tooltip>
            <TooltipTrigger asChild>
                <div>

```

```

        <a
            href="#"
            className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
        >
            <Settings className="h-5 w-5" />
            <span className="sr-only">Settings</span>
        </a>
    </div>
</TooltipTrigger>
<TooltipContent side="right">Settings</TooltipContent>
</Tooltip>
</TooltipProvider>
</nav>
</aside>
<div className="flex flex-col sm:gap-4 sm:py-4 sm:pl-14">
    <header className="sticky top-0 z-30 flex h-14 items-center gap-4
border-b bg-background px-4 sm:static sm:h-auto sm:border-0 sm:bg-transparent
sm:px-6">
        <Sheet>
            <SheetTrigger asChild>
                <Button size="icon" variant="outline">
                    <PanelLeft className="h-5 w-5" />
                    <span className="sr-only">Toggle Menu</span>
                </Button>
            </SheetTrigger>
            <SheetContent side="left" className="sm:max-w-xs dark">
                <nav className="grid gap-6 text-lg font-medium">
                    <a
                        href="#"
                        className="group flex h-10 w-10 shrink-0 items-center
justify-center gap-2 rounded-full bg-primary text-lg font-semibold text-
primary-foreground md:text-base"
                    >
                        <Package2 className="h-5 w-5 transition-all group-
hover:scale-110" />
                        <span className="sr-only">Budget Buddy</span>
                    </a>
                    <a
                        href="/home"
                        className="flex items-center gap-4 px-2.5 text-muted-
foreground hover:text-foreground"
                    >
                        <Home className="h-5 w-5" />
                        Dashboard
                    </a>
                    <a
                        href="/dashboard"
                        className="flex items-center gap-4 px-2.5 text-muted-
foreground hover:text-foreground"
                    >
                        <BadgeIndianRupee className="h-5 w-5" />
                        Expenses

```

```
</a>
<a
    href="/analytics"
    className="flex items-center gap-4 px-2.5 text-muted-foreground hover:text-foreground"
>
    <LineChart className="h-5 w-5" />
    Analytics
</a>
</nav>
</SheetContent>
</Sheet>
<Breadcrumb className="hidden md:flex">
    <BreadcrumbList>
        <BreadcrumbItem>
            <BreadcrumbLink asChild>
                <a href="/home">Dashboard</a>
            </BreadcrumbLink>
        </BreadcrumbItem>
        <BreadcrumbSeparator />
    </BreadcrumbList>
</Breadcrumb>
<div className="relative ml-auto flex-1 md:grow-0">
    <Search className="absolute left-2.5 top-2.5 h-4 w-4 text-muted-foreground" />
    <Input
        type="search"
        placeholder="Search..."
        className="w-full rounded-lg bg-background pl-8 md:w-[200px] lg:w-[336px]"
    />
</div>
<DropdownMenu>
    <DropdownMenuTrigger asChild>
        <Button
            variant="outline"
            size="icon"
            className="overflow-hidden rounded-full"
        >
            
        </Button>
    </DropdownMenuTrigger>
    <DropdownMenuContent align="end">
```

```

        <DropDownMenuLabel>My Account</DropDownMenuLabel>
        <DropDownMenuSeparator />
        <DropDownMenuItem>Settings</DropDownMenuItem>
        <DropDownMenuItem>Support</DropDownMenuItem>
        <DropDownMenuSeparator />
        <DropDownMenuItem>Logout</DropDownMenuItem>
    </DropDownMenuContent>
</DropDownMenu>
</header>
<main className="flex">
    <div className="flex flex-col justify-start w-full sm:w-1/3 mx-5">
        <Card className="mb-5">
            <CardHeader>
                <CardTitle>Greetings</CardTitle>
            </CardHeader>
            <CardContent>
                { /* Logic to calculate and display today's total expenses */ }
                <div>Hello User Welcome to BudgetBuddy</div>
            </CardContent>
        </Card>
        <Card className="mb-5">
            <CardHeader>
                <CardTitle>Today's Total Expenses</CardTitle>
            </CardHeader>
            <CardContent>
                { /* Logic to calculate and display today's total expenses */ }
                <div>{todayTotal}</div>
            </CardContent>
        </Card>
        <Card className="mb-5">
            <CardHeader>
                <CardTitle>Today's Total Expenses</CardTitle>
            </CardHeader>
            <CardContent>
                <ExpenseForm className="mt-5" />
            </CardContent>
        </Card>
    </div>
    <Card className="w-11/12 mx-5">
        <CardHeader>
            <CardTitle>Recent Expenses</CardTitle>
            <CardDescription></CardDescription>
        </CardHeader>
        <CardContent>
            <Table>
                <TableHeader>
                    <TableRow>
                        <TableHead>Name</TableHead>
                        <TableHead>Category</TableHead>
                        <TableHead className="hidden md:table-cell">
                            Amount
                        </TableHead>
                        <TableHead className="hidden md:table-cell">
                            Created at

```

```

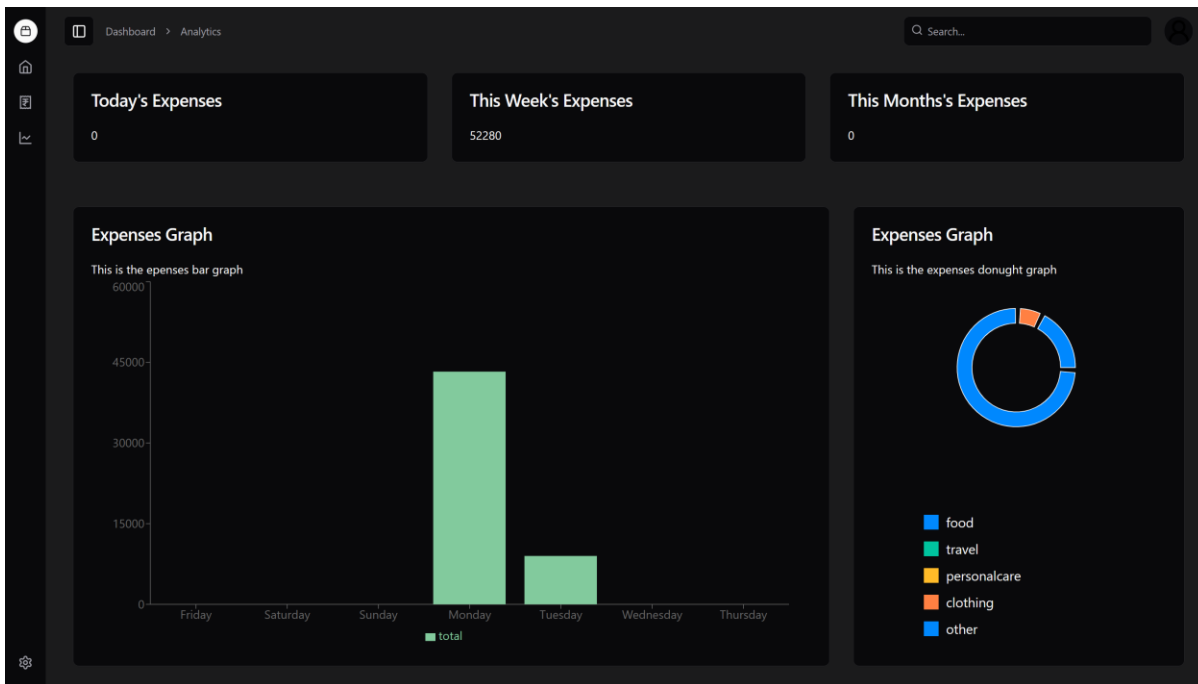
        </TableHead>
        <TableHead>
          <span className="sr-only">Actions</span>
        </TableHead>
      </TableRow>
    </TableHeader>
    <TableBody>
      {expenses
        .slice(-10)
        .reverse()
        .map((expense) => (
          <TableRowCustom
            eId={expense.eId}
            key={expense.eId}
            description={expense.description}
            category={expense.category}
            amount={expense.amount}
            date={expense.date}
          />
        ))}
    </TableBody>
  </Table>
</CardContent>
</Card>
</main>
</div>
</div>
);
}

```

5. Analytics

The Analytics route offers users advanced insights and analysis tools to explore their financial data in-depth. This section of the application provides interactive charts, graphs, and reports to visualize spending patterns, track expenses over time, and identify areas for optimization. Users can customize parameters, apply filters, and drill down into specific categories or time periods to gain actionable insights into their financial behavior. Additionally, the analytics section may include forecasting models, budgeting tools, and comparative analysis features to empower users in managing their finances effectively. By providing robust analytics capabilities, you empower users to make data-driven decisions and achieve their financial goals.

In summary, each route within your expense management application serves a unique purpose in facilitating user interactions, supporting account management tasks, and delivering valuable insights into users' financial activities. By designing intuitive interfaces, implementing robust authentication mechanisms, and offering comprehensive analytics tools, you create a seamless and empowering experience for users navigating your application.



```
import {
  Home,
  LineChart,
  MoreHorizontal,
  Package2,
  PanelLeft,
  PlusCircle,
  Search,
  Settings,
  ReceiptIndianRupee,
  BadgeIndianRupee,
  Car,
} from "lucide-react";
import TableRowCustom from "../TableRow";
import { Badge } from "@components/ui/badge";
import {
  Breadcrumb,
  BreadcrumbItem,
  BreadcrumbLink,
  BreadcrumbList,
  BreadcrumbPage,
  BreadcrumbSeparator,
} from "@components/ui/breadcrumb";
import { Button } from "@components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardFooter,
  CardHeader,
  CardTitle,
} from "@components/ui/card";
import {
  DropdownMenu,
  DropdownMenuCheckboxItem,
```



```

DropdownMenuContent,
DropdownMenuItem,
DropdownMenuLabel,
DropdownMenuSeparator,
DropdownMenuTrigger,
} from "@components/ui/dropdown-menu";
import { Input } from "@components/ui/input";
import { Sheet, SheetContent, SheetTrigger } from "@components/ui/sheet";
import {
  Table,
  TableBody,
  TableCell,
  TableHead,
  TableHeader,
  TableRow,
} from "@components/ui/table";
import { Tabs, TabsContent, TabsList, TabsTrigger } from
"@components/ui/tabs";
import {
  Tooltip,
  TooltipProvider,
  TooltipContent,
  TooltipTrigger,
} from "@components/ui/tooltip";
import axios from "axios";
import React, { useState, useEffect } from "react";
import { Bar } from "recharts";
import BarGraph from "../BarGraph";
import DonughtChart from "../DonughtChart";

export default function Analytics() {
  const [expenses, setExpenses] = useState([]);
  const [todayTotal, setTodayTotal] = useState(0);
  const [weekTotal, setWeekTotal] = useState(0);
  const [monthTotal, setMonthTotal] = useState(0);

  useEffect(() => {
    const fetchExpenses = async () => {
      try {
        const response = await axios.get(
          "http://localhost:3000/expense/getExpenses",
          {
            withCredentials: true,
          }
        );
        setExpenses(response.data.expenses);
      } catch (error) {
        console.error("Error fetching expenses:", error);
      }
    };

    fetchExpenses();
  }, []);

```

```

useEffect(() => {
  const todayExpenses = expenses.filter((expense) => {
    const expenseDate = new Date(expense.date);
    const today = new Date();
    return (
      expenseDate.getDate() === today.getDate() &&
      expenseDate.getMonth() === today.getMonth() &&
      expenseDate.getFullYear() === today.getFullYear()
    );
  });
  const todayTotal = todayExpenses.reduce(
    (total, expense) => total + expense.amount,
    0
  );
  setTodayTotal(todayTotal);

  const weekExpenses = expenses.filter((expense) => {
    const expenseDate = new Date(expense.date);
    const today = new Date();
    const startOfWeek = new Date(
      today.getFullYear(),
      today.getMonth(),
      today.getDate() - today.getDay()
    );
    const endOfWeek = new Date(
      today.getFullYear(),
      today.getMonth(),
      today.getDate() + (6 - today.getDay())
    );
    return expenseDate >= startOfWeek && expenseDate <= endOfWeek;
  });
  const weekTotal = weekExpenses.reduce(
    (total, expense) => total + expense.amount,
    0
  );
  setWeekTotal(weekTotal);

  const monthExpenses = expenses.filter((expense) => {
    const expenseDate = new Date(expense.date);
    const today = new Date();
    return (
      expenseDate.getMonth() === today.getMonth() &&
      expenseDate.getFullYear() === today.getFullYear()
    );
  });
  const monthTotal = monthExpenses.reduce(
    (total, expense) => total + expense.amount,
    0
  );
  setMonthTotal(monthTotal);
}, [expenses]);
return (
  <div className="flex min-h-screen w-full flex-col bg-muted/40 dark">

```

```

<aside className="fixed inset-y-0 left-0 z-10 hidden w-14 flex-col
border-r bg-background sm:flex">
  <nav className="flex flex-col items-center gap-4 px-2 sm:py-5">
    <a
      href="/"
      className="group flex h-9 w-9 shrink-0 items-center justify-center
gap-2 rounded-full bg-primary text-lg font-semibold text-primary-foreground
md:h-8 md:w-8 md:text-base"
    >
      <Package2 className="h-4 w-4 transition-all group-hover:scale-110"
/>
      <span className="sr-only">Budget Buddy</span>
    </a>
    <TooltipProvider>
      <Tooltip>
        <TooltipTrigger asChild>
          <div>
            <a
              href="/home"
              className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
            >
              <Home className="h-5 w-5" />
              <span className="sr-only">Dashboard</span>
            </a>
          </div>
        </TooltipTrigger>
        <TooltipContent side="right">Dashboard</TooltipContent>
      </Tooltip>
    </TooltipProvider>
    <TooltipProvider>
      <Tooltip>
        <TooltipTrigger asChild>
          <div>
            <a
              href="/dashboard"
              className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-
8 md:w-8"
            >
              <ReceiptIndianRupee className="h-5 w-5" />
              <span className="sr-only">Expenses</span>
            </a>
          </div>
        </TooltipTrigger>
        <TooltipContent side="right">Expenses</TooltipContent>
      </Tooltip>
    </TooltipProvider>
    <TooltipProvider>
      <Tooltip>
        <TooltipTrigger asChild>
          <div>
            <a

```

```

                href="/analytics"
                className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
            >
                <LineChart className="h-5 w-5" />
                <span className="sr-only">Analytics</span>
            </a>
        </div>
    </TooltipTrigger>
    <TooltipContent side="right">Analytics</TooltipContent>
</Tooltip>
</TooltipProvider>
</nav>
<nav className="mt-auto flex flex-col items-center gap-4 px-2 sm:py-5">
    <TooltipProvider>
        <Tooltip>
            <TooltipTrigger asChild>
                <div>
                    <a
                        href="#"
                        className="flex h-9 w-9 items-center justify-center
rounded-lg text-muted-foreground transition-colors hover:text-foreground md:h-8
md:w-8"
                    >
                        <Settings className="h-5 w-5" />
                        <span className="sr-only">Settings</span>
                    </a>
                </div>
            </TooltipTrigger>
            <TooltipContent side="right">Settings</TooltipContent>
        </Tooltip>
    </TooltipProvider>
</nav>
</aside>
<div className="flex flex-col sm:gap-4 sm:py-4 sm:pl-14">
    <header className="sticky top-0 z-30 flex h-14 items-center gap-4
border-b bg-background px-4 sm:static sm:h-auto sm:border-0 sm:bg-transparent
sm:px-6">
        <Sheet>
            <SheetTrigger asChild>
                <Button size="icon" variant="outline">
                    <PanelLeft className="h-5 w-5" />
                    <span className="sr-only">Toggle Menu</span>
                </Button>
            </SheetTrigger>
            <SheetContent side="left" className="sm:max-w-xs dark">
                <nav className="grid gap-6 text-lg font-medium">
                    <a
                        href="#"
                        className="group flex h-10 w-10 shrink-0 items-center
justify-center gap-2 rounded-full bg-primary text-lg font-semibold text-
primary-foreground md:text-base"
                    >

```

```

        <Package2 className="h-5 w-5 transition-all group-
hover:scale-110" />
        <span className="sr-only">Budget Buddy</span>
      </a>
      <a
        href="/home"
        className="flex items-center gap-4 px-2.5 text-muted-
foreground hover:text-foreground"
      >
        <Home className="h-5 w-5" />
        Dashboard
      </a>
      <a
        href="/dashboard"
        className="flex items-center gap-4 px-2.5 text-muted-
foreground hover:text-foreground"
      >
        <BadgeIndianRupee className="h-5 w-5" />
        Expenses
      </a>
      <a
        href="/analytics"
        className="flex items-center gap-4 px-2.5 text-muted-
foreground hover:text-foreground"
      >
        <LineChart className="h-5 w-5" />
        Analytics
      </a>
    </nav>
  </SheetContent>
</Sheet>
<Breadcrumb className="hidden md:flex">
  <BreadcrumbList>
    <BreadcrumbItem>
      <BreadcrumbLink asChild>
        <a href="/home">Dashboard</a>
      </BreadcrumbLink>
    </BreadcrumbItem>
    <BreadcrumbSeparator />
    <BreadcrumbItem>
      <BreadcrumbLink asChild>
        <a href="/">Analytics</a>
      </BreadcrumbLink>
    </BreadcrumbItem>
  </BreadcrumbList>
</Breadcrumb>
<div className="relative ml-auto flex-1 md:grow-0">
  <Search className="absolute left-2.5 top-2.5 h-4 w-4 text-muted-
foreground" />
  <Input
    type="search"
    placeholder="Search..."
    className="w-full rounded-lg bg-background pl-8 md:w-[200px]
lg:w-[336px]"
  >

```

```

    />
  </div>
  <DropdownMenu>
    <DropdownMenuTrigger asChild>
      <Button
        variant="outline"
        size="icon"
        className="overflow-hidden rounded-full"
      >
        
      </Button>
    </DropdownMenuTrigger>
    <DropdownMenuContent align="end">
      <DropdownMenuLabel>My Account</DropdownMenuLabel>
      <DropdownMenuSeparator />
      <DropdownMenuItem>Settings</DropdownMenuItem>
      <DropdownMenuItem>Support</DropdownMenuItem>
      <DropdownMenuSeparator />
      <DropdownMenuItem>Logout</DropdownMenuItem>
    </DropdownMenuContent>
  </DropdownMenu>
</header>
<main>
  <div className="flex flex-wrap p-5">
    <Card className="mb-5 flex-1 mx-4">
      <CardHeader>
        <CardTitle>Today's Expenses</CardTitle>
      </CardHeader>
      <CardContent>
        {/* Logic to calculate and display today's total expenses */}
        <div>{todayTotal}</div>
      </CardContent>
    </Card>
    <Card className="mb-5 flex-1 mx-4">
      <CardHeader>
        <CardTitle>This Week's Expenses</CardTitle>
      </CardHeader>
      <CardContent>
        {/* Logic to calculate and display today's total expenses */}
        <div>{weekTotal}</div>
      </CardContent>
    </Card>
    <Card className="mb-5 flex-1 mx-4">

```

