**Prompt:**  
I am creating a website to track my monthly expenses. I need your help to design the website. Overview of website: First there will be a simple landing page (there will be option to login). After login user will be shown a dashboard. In the left side there will be a side menu bar. There will be options, like Transaction

Analytics

Settings

etc..

In Transaction, I want to add money, spend the money on different thins, show current balance, recent transactions.

In analytics, I want to show in which month how much money added, current balance, expense analysis, option to download report etc,

in settings user can change the username password, option to send report by email , logout , delete account etc.

I want you to read and understand the website and its requirements.

Great! Since you're using the **MERN stack** and will build the **frontend with Vite + React**, here’s a clean and scalable folder structure tailored for your **Expense Tracker Web App**.

**✅ Folder Structure for Frontend (Vite + React)**

expense-tracker-frontend/

├── public/

│ └── favicon.ico

│ └── index.html

│

├── src/

│ ├── assets/ # Images, icons, logos, etc.

│ ├── components/ # Reusable UI components (buttons, charts, etc.)

│ ├── features/ # Feature-specific folders (Transaction, Analytics, etc.)

│ │ ├── dashboard/ # Dashboard landing page

│ │ ├── transactions/ # Transaction UI & logic

│ │ ├── analytics/ # Analytics graphs and reports

│ │ ├── settings/ # Settings page

│ ├── layout/ # Layout components (Sidebar, Header, Wrapper)

│ ├── pages/ # Route-based pages (Login, Landing, Dashboard)

│ ├── services/ # API service calls (Axios, fetch, etc.)

│ ├── hooks/ # Custom React hooks

│ ├── context/ # Context providers (Auth, App state, etc.)

│ ├── utils/ # Helper functions (formatting, validation)

│ ├── constants/ # Static config (routes, chart configs, etc.)

│ ├── App.jsx # Main app component

│ ├── main.jsx # Entry point

│ └── index.css # Global styles (Tailwind imported here)

│

├── .env # Environment variables

├── .gitignore

├── package.json

├── tailwind.config.js # Tailwind setup

├── postcss.config.js

└── vite.config.js

**✅ Highlights of the Structure**

| **Folder / File** | **Purpose** |
| --- | --- |
| **/components** | Common reusable UI elements like Button, Input, ChartCard, Modal, etc. |
| **/features** | Feature-based structure for encapsulation (good for scaling) |
| **/layout** | Common layout parts like Sidebar, Topbar, Layout wrapper |
| **/pages** | Top-level route components, like LandingPage, LoginPage, DashboardPage |
| **/services** | Centralized API calls, using axios or fetch, possibly with interceptors |
| **/context** | Contexts like AuthContext, ThemeContext, etc. |
| **/hooks** | Custom hooks, like useAuth, useTransactions, useToggle |
| **/utils** | Helpers: currency formatting, date formatting, etc. |
| **/constants** | App-wide constants like route paths, API endpoints, chart settings |

**✅ Tailwind Setup (Assuming Tailwind CSS)**

In index.css:

@tailwind base;

@tailwind components;

@tailwind utilities;

**✅ Example Pages Breakdown**

**pages/LoginPage.jsx**

import LoginForm from '../features/auth/LoginForm'

export default function LoginPage() {

return (

<div className="min-h-screen flex items-center justify-center bg-gray-100">

<LoginForm />

</div>

)

}

**features/transactions/AddTransactionForm.jsx**

export default function AddTransactionForm() {

// Form logic here

}

Would you like me to now generate the **backend folder structure**, or would you prefer I scaffold the initial code for something like the layout with the sidebar and routing using React Router?

**Configuring Folder structure:**

npm create vite@latest frontend -- --template react

cd frontend

npm install

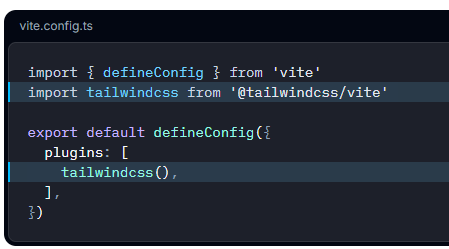
npm run dev

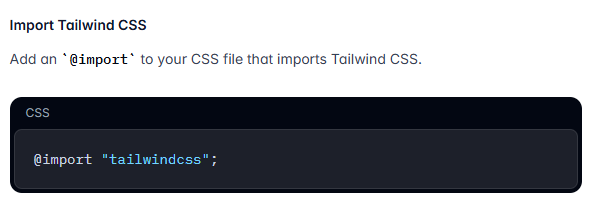
**TailwindCSS:**

npm create vite@latest my-project

cd my-project

npm install tailwindcss @tailwindcss/vite





**✅ OPTIONAL: Test Tailwind**

Edit src/App.jsx to:

jsx

CopyEdit

function App() {

return (

<div className="min-h-screen bg-gray-100 flex items-center justify-center">

<h1 className="text-4xl font-bold text-blue-600">Tailwind is Working!</h1>

</div>

);

}

export default App;

If you see styled text, **Tailwind is working perfectly**.