

## AI Assistance Usage Notes (ChatGPT)

I used ChatGPT sparingly to fix runtime bugs, data inconsistencies, and complicated TypeScript problems that hindered productivity while creating the Credit Risk Analytics Dashboard (Node.js + Express backend, React + TypeScript frontend). The following are the main areas where AI support was beneficial:

### 1. Type Mismatch in Chart Components

**Issue:** TypeScript threw an error when the customers array was passed to the IncomeExpenseChart since it was expecting ChartDataItem[] rather than CustomerType[].

```
<IncomeExpenseChart data={customers} />
```

Error Message:

```
Type 'CustomerType[]' is not assignable to type 'ChartDataItem[]'.  
Property 'month' is missing in type 'CustomerType'.
```

**Resolution:** ChatGPT assisted in determining that before passing the customer data, it needed to be converted into the proper chart format. This is the updated method:

```
const chartData: ChartDataItem[] = Array.from({ length: 6 }, (_, i) => {  
  const month = `Month ${i + 1}`;  
  const income = customers.reduce((sum, c) => sum + c.monthlyIncome, 0);  
  const expenses = customers.reduce((sum, c) => sum + c.monthlyExpenses, 0);  
  return { month, income, expenses };  
});  
  
<IncomeExpenseChart data={chartData} />
```

### 2. JSX Expression Must Have One Parent

**Issue:** I made a JSX syntax error when rendering multiple components in Dashboard.tsx because I neglected to wrap them in a parent element.

```
<DashboardCards ... />
```

```
<Card ... />
```

**Resolution:** I was reminded by ChatGPT to enclose several sibling JSX elements in a parent `<div>` or fragment:

```
<>
  <DashboardCards ... />
  <Card ... />
</>
```

### 3. Unresolved Variables in Dashboard.tsx

**Issue:** Before defining or importing variables like `totalIncome`, `setSearchText`, `computeRiskScore`, and `riskData`, I made reference to them. This resulted in:

```
Cannot find name 'setSearchText'.ts(2304)
Cannot find name 'riskData'.ts(2304)
```

**Resolution:** I was able to confirm with ChatGPT which variables should be defined locally as opposed to being passed as props or derived from the customer data.

Example fix:

```
const [searchText, setSearchText] = useState<string>("");
const riskData = computeRiskDistribution(customers);
```

### 4. Interface Missing in TypeScript File

**Issue:** Using a custom type like `CustomerType` without importing or defining it threw errors:

```
Cannot find name 'CustomerType'.ts(2304)
```

**Resolution:** I followed ChatGPT's instructions to extract shared interfaces, such as CustomerType, ChartDataItem, etc., into a different types.ts file and import them as required:

```
export interface CustomerType {  
  customerId: string;  
  name: string;  
  monthlyIncome: number;  
  monthlyExpenses: number;  
  creditScore: number;  
  status: "Approved" | "Rejected" | "Review";  
}
```

```
import { CustomerType } from "../types";
```

## Node.js + Express Backend

### 5. POST Body Parsing Not Working in Express

**Issue:** The request body was not being correctly received by my POST /update-customer-status endpoint.

```
app.post("/update-customer-status", (req, res) => {  
  const { customerId, newStatus } = req.body; // ❌ undefined  
});
```

**Resolution:** ChatGPT helped identify the missing middleware:

```
app.use(express.json());
```

## 6. In-Memory Data Persistence with Side Effects

**Issue:** Changes made to the customer's status were not reflected in GET calls that followed. I discovered that I was altering a copied object rather than the original in-memory array thanks to ChatGPT.

```
const customer = customers.find(c => c.id === id);  
customer.status = newStatus;
```

**Resolution:** Ensure mutation is on the original array object:

```
const index = customers.findIndex(c => c.id === id);  
if (index !== -1) customers[index].status = newStatus;
```

These were the main technical obstacles that I overcame with the help of ChatGPT and other AI tools, not for complete implementation but rather to effectively fix particular TypeScript and runtime problems. The styling, and core app logic were all written by hand.

**LOOM VIDEO LINK:-**

<https://drive.google.com/file/d/1oumk9vcz0p6qFsW2dOrmHZfA550M65aM/view?usp=sharing>