Bilancio

├─ backup

│ ├─ BilancioDataBase1.db

│ ├─ BilancioDataBase20.04.2025.db

│ ├─ BilancioDataBase20.04.2025a.db

│ └─ tsconfig.app.json..txtbackup

├─ desktopIcon.png

├─ electron-builder.json

├─ eslint.config.js

├─ index.html

├─ package-lock.json

├─ package.json

├─ playwright.config.ts

├─ README.md

├─ src

│ ├─ assets

│ │ ├─ trayIcon.png

│ │ └─ trayIconTemplate.png

│ ├─ electron

│ │ ├─ dataBase

│ │ │ ├─ dbClass.ts

│ │ │ ├─ dbFunction.ts

│ │ │ ├─ dbQuerySqlString.ts

│ │ │ └─ enum.ts

│ │ ├─ dataBase.ts

│ │ ├─ main.ts

│ │ ├─ menu.ts

│ │ ├─ pathResolver.ts

│ │ ├─ preload.cts

│ │ ├─ resourceManager.ts

│ │ ├─ sharedTypes

│ │ │ ├─ status.backup..js

│ │ │ └─ status.ts

│ │ ├─ tray.ts

│ │ ├─ tsconfig.json

│ │ └─ util.ts

│ └─ ui

│ ├─ App

│ │ ├─ App.css

│ │ └─ App.tsx

│ ├─ assets

│ │ └─ react.svg

│ ├─ BaseChart.tsx

│ ├─ Chart.tsx

│ ├─ components

│ │ ├─ ButtonCancel

│ │ │ ├─ ButtonCancel.module.scss

│ │ │ └─ ButtonCancel.tsx

│ │ ├─ CheckboxSlider

│ │ │ ├─ CheckboxSlider.module.scss

│ │ │ └─ CheckboxSlider.tsx

│ │ ├─ ComboBox

│ │ │ └─ ComboBox.tsx

│ │ ├─ Context

│ │ │ ├─ ElectronProvider.tsx

│ │ │ └─ useOptionsImage.ts

│ │ ├─ DateTimePicker

│ │ │ ├─ DateTimePicer.module.scss

│ │ │ └─ DateTimePicker.tsx

│ │ ├─ FormAddInvoice

│ │ │ ├─ FormAddInvoice.module.scss

│ │ │ └─ FormAddInvoice.tsx

│ │ ├─ FormAddInvoiceDocuments

│ │ │ ├─ FormAddInvoiceDocuments.module.scss

│ │ │ └─ FormAddInvoiceDocuments.tsx

│ │ ├─ FormHomeDate

│ │ │ ├─ FormHomeDate.module.scss

│ │ │ └─ FormHomeDate.tsx

│ │ ├─ GlobalFunctions

│ │ │ └─ GlobalFunctions.ts

│ │ ├─ MainTable

│ │ │ ├─ MainTable.module.scss

│ │ │ └─ MainTable.tsx

│ │ ├─ ModalAddInvoice

│ │ │ ├─ ModalAddInvoice.module.scss

│ │ │ └─ ModalAddInvoice.tsx

│ │ ├─ Navigation

│ │ │ ├─ Navigation.module.scss

│ │ │ └─ Navigation.tsx

│ │ ├─ SingleInput

│ │ │ ├─ SingleInput.module.scss

│ │ │ └─ SingleInput.tsx

│ │ └─ TextInput

│ │ ├─ TextInput.module.scss

│ │ └─ TextInput.tsx

│ ├─ hooks

│ │ ├─ useAllDocumentName.ts

│ │ ├─ useAllInvoices.ts

│ │ ├─ useConnectedTableDictionary.ts

│ │ ├─ useLocalStorage.ts

│ │ ├─ useTableDictionaryDocuments.ts

│ │ └─ useToggle.ts

│ ├─ index.css

│ ├─ main.tsx

│ ├─ pages

│ │ ├─ HomePage

│ │ │ ├─ HomePage.module.scss

│ │ │ └─ HomePage.tsx

│ │ ├─ LayoutPage

│ │ │ ├─ LayoutPage.module.scss

│ │ │ └─ LayoutPage.tsx

│ │ └─ ReportDataPage

│ │ └─ ReportDataPage.tsx

│ ├─ TempStart

│ │ ├─ TempStart.css

│ │ └─ TempStart.tsx

│ ├─ useStatistics.ts

│ └─ vite-env.d.ts

├─ structure.txt

├─ tsconfig.json

├─ tsconfig.node.json

├─ types.d.ts

└─ vite.config.ts

to jest moja struktura plików, chciałbym do mojego programu pisanego w Electron dodać funkcjonalość dodawania faktury w formacie pdf. Użytkownik będzie przeciągał lub wybierał z otwartego okna fakturę w pdf. Będzie ona później dodawana do bazy danych (jeśli będziesz potrzebował strukturę bazy danych to mi napisz). Na tym etapie będzie w pamięci. Faktura będzie dodawana poprzez pole (aktualnie jeszcze go nie ma) znajdujące się w komponencie FormHomeDate.tsx obok przycisku "Dodaj fakturę". Po dodaniu faktury powinna być ona odczytana (chciałbym żeby to było zrobione przez AI offline, ewentualnie online jeśli nie ma takiej możliwości, ponieważ chciałbym żeby uczyła się prawidłowo odczytywać dane z faktury, ewentualnie może być biblioteka w VisualStudio, js, ts jeśli takie są). Z faktury będą odczytywane takie dane jak: nazwa faktury, data faktury, nazwa produktu za który jest wystawiana faktura, ilość produktu, jego cena. Po odczytaniu danych będzie otwierało się okno modala w tym komponencie ModalAddInvoice.tsx, który zawiera komponent FormAddInvoice.tsx. W komponencie w TextInput inputName="invoiceName" powinna się wyświetlić nazwa faktury. Komponent FormAddInvoice.tsx zawiera komponent FormAddInvoiceDocuments.tsx w nim znajdują się pola: <Select<ComboBoxOption> //Dokument Combobox, <Select<ComboBoxOption> //MainType Combobox, <Select<ComboBoxOption> //Type Combobox, <Select<ComboBoxOption> //Subtype Combobox, <TextInput inputName="quantity", <TextInput inputName="price". Pola select będą wyświetlały nazwy dokumentu lub dokumentów jeżeli jest ich więcej niż 1. Pola tekstowe będą wyświetlały odpowiednio ilość produktów oraz cenę jednostkową. Zwróć uwagę że produkt o nazwie "dowód rejestracyjny" będzie zawarty tylko w Select Dokument ponieważ nie ma on podtypów. A produkt o nazwie "tablica rejestracyjna tymczasowa samochodowa jednorzędowa żółta" będzie wyświetlany w czterech Select. W Select Dokument będzie "tablica rejestracyjna", w Select MainType będzie "tymczasowa", w Select Type będzie " samochodowa jednorzędowa", w Select Subtype będzie "żółta". Napisz czy potrzebujesz żebym wysłał Ci kod któregoś z komponentów. Zwróć uwagę że wykorzystywane AI lub biblioteki powinny być darmowe