Macro Dash

this tile model is going to be like a buyable product. A tile will represent a tile on a website that contains model or feature of the website. e.g the tile could contain a chat window that allows you to also chat with other users that have the chat window tile. One could contain fred release calendar in a simple searchable sortable list. Another could have a live chart with multiple stock tickers on a watchlist. Can you give a plan, the most appropriate structure for this, a file structure overview an the scaflholing fo rit with the 3 things i mentioned, base on what you know about my app so far, he's some files in my current build so far and the file structure I currently have

can we just focus on the backend for now. I dont want to have endpoints for adding and deleting tiles yet. I just want the database record and the ability for users to have tiles in their own list of tiles to be able to view it. each user will have their own settings for the tiles like the configuration they use. when they go to the website they need to be able to get the data for each one based on what it does, this will be done in different ways, for a chat and life data we will need a socket connected to the backend so the user gets live data. for the others it can fetch the data when it loads so we need endpoints for that. the user needs to be able to shop and buy new tiles.

Thought for the payment structure (optional)
There will be a subscription service based on what you use each month. A tile could cost around 1 dollar or even 50 cents. You get a minimum charge of 5 dollars a month.

Some apps like chat will always be free so you can use them whenever and you wont get charge.

Each one will be resizable with a minus size before gets cover to make it bigger

What I have so far

- Login frontend, backend, with auth
- Signup
- Dashboard with mosaic library
- User records with hashed password
- Release calendar records with Fred data
- Cron job that can run hourly, daily, weekly, e.t.c
- Integration to fetch fred data, does calendar so far
-

What I need to add

- Chat tile frontend
- Chat tile backend with database records socket for communication
- Add user settings database record
- Add user owned tile list
- Add tile shop list with prices and information, title, e.t.c
- Plotly charts to frontend
- Websockets for showing live data to frontend
- Frontend page of buying tiles

File structure of my project

tree --prune -I 'node_modules|__pycache__|venv'

```
.
├── README.MD
├── backend
│   ├── __init__.py
│   ├── controllers
│   │   ├── applications_controller.py
│   │   ├── auth_controller.py
│   │   └── helper_controller.py
│   ├── cron
│   │   ├── jobs.py
│   │   └── scheduler.py
│   ├── helpers
│   │   ├── api_exception.py
│   │   ├── api_helpers.py
│   │   └── database.py
│   ├── integrations
│   │   └── fred_integration.py
│   ├── models
│   │   ├── applications.py
│   │   ├── fred_release_calendar.py
│   │   ├── tile.py
│   │   └── users.py
│   ├── requirements.txt
│   └── run.py
└── frontend
```

```
├── package-lock.json
├── package.json
├── public
│   ├── MuscleMotionLogo.jpg
│   ├── index.html
│   ├── manifest.json
│   └── robots.txt
├── src
│   ├── app.tsx
│   ├── components
│   │   ├── display
│   │   │   └── Loading.tsx
│   │   ├── header
│   │   │   └── HeaderContainer.tsx
│   │   └── input
│   │       ├── Button.tsx
│   │       ├── Error.tsx
│   │       ├── Text.tsx
│   │       └── TextArea.tsx
│   ├── hooks
│   │   ├── useFetchApi.ts
│   │   └── usePostApi.ts
│   ├── index.tsx
│   ├── pages
│   │   ├── dashboard
│   │   │   └── DashboardContainer.tsx
│   │   ├── login
│   │   │   ├── LoginContainer.tsx
│   │   │   └── LoginForm.tsx
│   │   ├── signup
│   │   │   ├── SignupContainer.tsx
│   │   │   └── SignupForm.tsx
│   │   └── tile_store
│   │       └── TileStoreContainer.tsx
│   ├── react-app-env.d.ts
│   ├── recoil
│   │   └── user.ts
│   ├── types
│   │   ├── ApiTypes.ts
│   │   ├── ObjectTypes.ts
│   │   └── index.d.ts
│   └── utils
│       ├── GlobalStyles.tsx
│       ├── PageNotFound.tsx
│       └── ProtectedRoute.tsx
└── tsconfig.json
```