Pronapse Setup Guide (React + Vite)

# 1. Environment Setup (macOS M1)

Install Homebrew:

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile

eval "$(/opt/homebrew/bin/brew shellenv)"

Install Node.js (via Homebrew):

brew install node

Install Git:

brew install git

Install VS Code:

brew install --cask visual-studio-code

# 2. Create React + Vite Project

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

cd pronapse

npm install

# 3. Project Structure

pronapse/

├── public/

│ └── favicon.svg

├── src/

│ ├── assets/

│ │ └── logo.svg

│ ├── components/

│ │ ├── Tile.jsx

│ │ ├── TopBar.jsx

│ │ ├── ClassificationFilter.jsx

│ ├── App.jsx

│ ├── main.jsx

├── .gitignore

├── index.html

├── package.json

├── vite.config.js

├── README.md

└── Makefile

# 4. Component Design Overview

Tile.jsx - Displays data tile with classification dots.

TopBar.jsx - Houses navigation buttons: blue/green (data sources), purple (actions), and Rules button.

ClassificationFilter.jsx - Lets users filter visible tiles based on classification.

# 5. Vercel Deployment

1. Push your project to GitHub:

git init

git add .

git commit -m "Initial commit"

git remote add origin <your-repo-url>

git push -u origin main

2. Go to Vercel.com, import the GitHub repo.

3. Framework preset: Vite

4. Click Deploy.

# 6. Makefile

PROJECT = pronapse

setup:

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

echo 'eval "$$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile

eval "$$(/opt/homebrew/bin/brew shellenv)"

brew install node git

brew install --cask visual-studio-code

init:

npm create vite@latest $(PROJECT) -- --template react

cd $(PROJECT) && npm install

cd $(PROJECT) && git init

cd $(PROJECT) && git add .

cd $(PROJECT) && git commit -m "Initial commit"

build:

cd $(PROJECT) && npm run build

run:

cd $(PROJECT) && npm run dev

# 7. Next Steps

- Add styling with Tailwind or CSS Modules.

- Define routes for data source pages and rule management.

- Integrate Drools/BRMS backend logic later.

- Connect each classification to filters and allow tiles to update dynamically.

Use this setup as a baseline to build a structured, scalable front end for Pronapse.