**Project Overview**

* **Title**: Mobile Telecom Services Catalog
* **Goals**: Create a web application to browse and manage a telecom services catalog, with distinct backend and frontend parts.

**1. Framework**

* **Backend**: Python with Flask for handling data and API requests.
* **Frontend**: [**Vue.js**](https://vue.js/?form=MG0AV3)for building a responsive and interactive user interface.

**2. Backend (Flask)**

* **Directory Structure**:

Copy

backend/

│

├── server.py

├── requirements.txt

* **Functionality**:
  + Serve catalog data via API endpoints.
  + Manage different customer types (Residential, Business, Partner) with specific pricing.
* **Endpoints**:
  + /catalog: Returns full catalog in JSON format.
  + /catalog/<customer\_type>: Returns catalog based on customer type.
  + /routes: Lists all available routes in the application.
* **How to Run**:
  + Create and activate a virtual environment:

sh

Copy

python -m venv venv

source venv/bin/activate # On Windows use `venv\Scripts\activate`

* + Install dependencies:

sh

Copy

pip install -r requirements.txt

* + Run the Flask application:

sh

Copy

python server.py

**3. Frontend (Vue.js)**

* **Directory Structure**:

Copy

frontend/

│

├── public/

│ ├── index.html

│ └── server.js

├── src/

│ ├── components/

│ │ ├── CatalogList.vue

│ │ ├── CatalogItem.vue

│ │ └── SearchBar.vue

│ ├── views/

│ │ ├── HomeView.vue

│ │ ├── CatalogView.vue

│ ├── store/

│ │ └── index.js

│ ├── router/

│ │ └── index.js

│ ├── assets/

│ ├── App.vue

│ ├── main.js

├── package.json

├── babel.config.js

├── vue.config.js

* **Functionality**:
  + Display catalog data fetched from Flask backend.
  + Provide interactive features like sorting, filtering, and searching.
* **How to Run**:
  + Navigate to frontend directory.
  + Install dependencies:

sh

Copy

npm install

* + Start the development server:

sh

Copy

npm run serve

**Integration**

* **Communication**: Frontend will use Axios or Fetch API to request data from the Flask backend.
* **Proxy Configuration**: Set up proxy in vue.config.js to forward API calls to the Flask server:

js

Copy

module.exports = {

devServer: {

proxy: 'http://localhost:5000'

}

}

This should get you back on track. Good to go?

**You**

could we store all what we did in github?

**Copilot**

Absolutely. You can create a GitHub repository to store and manage all your project files. Here’s a simple guide:

**Steps to Store Your Project on GitHub**

1. **Create a GitHub Account**: If you don't have one already, sign up at [**GitHub**](https://github.com/?form=MG0AV3).
2. **Create a New Repository**:
   * Go to your GitHub account.
   * Click on the New button to create a new repository.
   * Name your repository (e.g., MobileTelecomCatalog).
   * Add a description (optional).
   * Set the repository to public or private.
   * Click on Create repository.
3. **Initialize Git in Your Project**:
   * Open your project directory in the terminal.
   * Run the following commands to initialize a new Git repository and commit your files:

sh

Copy

git init

git add .

git commit -m "Initial commit"

1. **Add Remote and Push to GitHub**:
   * Add the GitHub repository as a remote:

sh

Copy

git remote add origin https://github.com/yourusername/MobileTelecomCatalog.git

* + Push your local commits to the GitHub repository:

sh

Copy

git push -u origin master