Real Estate Chatbot Dashboard

A full-stack web application for querying real estate data using natural language.

Features

- Natural language queries for real estate trends, comparisons, and averages
- Interactive charts (line/bar) using Chart.js
- Data table display with Bootstrap styling
- Backend API built with Django REST Framework and Pandas

Tech Stack

- **Backend:** Django, Django REST Framework, Pandas, OpenPyXL
- **Frontend:** React, Bootstrap, Chart.js, Axios

Setup Instructions

Prerequisites

- Python 3.8+
- Node.js 14+
- Virtual environment (recommended)

Backend Setup

1. Navigate to the project root directory.
2. Create and activate a virtual environment:

```
python -m venv venv
venv\Scripts\activate  # On Windows
```

3. Install Python dependencies:

```
pip install django djangorestframework pandas openpyxl django-cors-
headers
```

4. Run migrations:

```
python manage.py migrate
```

5. Start the Django server:

```
python manage.py runserver
```

The backend will be running on `http://localhost:8000`.

Frontend Setup

1. Navigate to the frontend directory:

```
cd frontend
```

2. Install Node.js dependencies:

```
npm install
```

3. Start the React development server:

```
npm start
```

The frontend will be running on `http://localhost:3000`.

Usage

1. Open your browser and go to `http://localhost:3000`.
2. Enter a query in the search box, e.g.:
   - "Show trends for Wakad"
   - "Compare rent in Bandra vs Wakad"
   - "Show average price for Andheri over last 3 years"
3. Click "Submit" to get results including summary, chart, and table.

API Endpoint

- **POST** `/query/`: Accepts JSON with `query` field, returns summary, chart_data, table_data.

Data

The application uses `Sample_data.xlsx` with columns: Location, Year, Price, Demand, Supply.

Notes

- CORS is configured to allow requests from `http://localhost:3000`.

- The backend parses queries using regex for locations and years.
- Charts are rendered based on query type (trend: line, compare: bar, average: bar).