

Data Visualization Dashboard

A full-stack data visualization dashboard using **Flask**, **MongoDB**, and **React** to display insights from the provided JSON dataset.

Project Overview

This project extracts data from a JSON file, stores it in a **MongoDB** database, provides a **Flask API** to serve the data, and presents it in an interactive **React dashboard** with filtering and visualization features.

Technologies Used

Backend (Flask + MongoDB)

- **Flask** → Python web framework for API development.
- **MongoDB** → NoSQL database to store JSON data.
- **Flask-PyMongo** → Enables Flask to communicate with MongoDB.
- **Flask-CORS** → Allows frontend to make requests to the backend.
- **Pymongo** → Python driver for MongoDB.

Frontend (React)

- **React.js** → Frontend framework for building the dashboard.
- **Bootstrap** → Styling for a responsive layout.
- **Chart.js / react-chartjs-2** → Used for data visualization.

Installation & Setup Guide

Backend Setup (Flask + MongoDB)

Prerequisites

- **Python 3.8+**
- **MongoDB (Local or MongoDB Atlas)**

Steps

1. Navigate to the **backend folder**:

```
bash(in terminal)
```

```
cd backend
```

2. Create a virtual environment (optional but recommended):

```
bash
```

```
python -m venv venv
```

```
source venv/bin/activate    # On macOS/Linux
```

```
venv\Scripts\activate      # On Windows
```

3. Install dependencies:

```
bash
```

```
pip install -r requirements.txt
```

4. Ensure MongoDB is **running** (for local MongoDB, start it with **mongod**).

5. Start the Flask server:

```
bash
```

```
python app.py
```

6. **Verify the API is running:**

Open a browser and go to:

- <http://127.0.0.1:5000/data> → Should return JSON data.

- <http://127.0.0.1:5000/stats> → Should return aggregated statistics.

2 Frontend Setup (React)

Prerequisites

- **Node.js 14+**
- **NPM (Node Package Manager)**

Steps

1. Navigate to the **frontend folder**:

```
bash
```

```
cd frontend
```

2. Install dependencies:

```
bash
```

```
npm install
```

3.Start the React app:

```
bash  
npm start
```

4.Open the browser at <http://localhost:3000>.

Features

- ✓ Load data from JSON into MongoDB
 - ✓ Filter data dynamically by country, sector, topic, etc.
 - ✓ Interactive charts (Bar Chart showing average intensity per sector)
 - ✓ Responsive, clean dashboard UI with Bootstrap styling
-

API Endpoints

Endpoint	Method	Description
/data	GET	Fetch all records
/data?country=USA	GET	Filter data by country
/data?sector=Energy	GET	Filter data by sector
/data/oil	GET	Fetch records for topic "oil"
/stats	GET	Get average intensity per sector

Next Steps

- ✓ Deploy backend to **Heroku or Render**
- ✓ Deploy frontend to **Vercel or Netlify**
- ✓ Optimize frontend for performance