**CDC Data Visualization Application Documentation**

**Project Overview**

This project aims to provide Mars leadership with visual insights into vaccination data from the CDC. The application fetches data from the CDC website, processes it, stores it in a MongoDB database, and provides an API for frontend visualization. The frontend allows users to filter data by various criteria such as national or jurisdictional, vaccination status, intent, and demographics.

## Technologies Used

* **Frontend**: React
* **Backend**: Node.js, Express.js
* **Database**: MongoDB
* **Data Fetching**: Axios
* **Scheduling**: Node-Cron

## Backend

### 1. **Server Setup**

The server is set up using Node.js and Express.js. It connects to MongoDB and serves the API endpoints. The server.js file initializes the server, connects to MongoDB, and sets up a cron job to fetch data from the CDC website daily.

### 2. **Data Ingestion**

Data ingestion is handled by the fetchData.js script. This script fetches data from the CDC API, transforms it to match our database schema, and stores it in MongoDB. The script also schedules daily data fetching using the node-cron package.

### 3. **API Endpoints**

API endpoints are defined in api.js. These endpoints allow fetching vaccination data with various filters such as jurisdiction, vaccination status, intent, demographics, and year. The data is retrieved from MongoDB and sent to the frontend.

### 4. **Data Model**

The data model is defined in VaccinationData.js. This file sets up the schema for the vaccination data stored in MongoDB. The schema includes fields like jurisdiction, population group, measure type, measure value, date range, year, and more.

### 5. **Data Validation and Transformation**

The data fetched from the CDC is considered dirty. It undergoes validation and transformation to ensure that only clean and relevant data is stored in the database. This includes converting string values to numbers where applicable and filtering out incomplete or invalid entries.

## Frontend

### 1. **Service for API Requests**

The api.js file in the services directory handles API requests to the backend. It uses Axios to fetch data from the backend API and returns it to the frontend.

### 2. **VaccinationData Component**

The VaccinationData.js component fetches and displays vaccination data. It uses the API service to request data from the backend and displays it in a table format. The component also supports filtering data based on various criteria, including national or jurisdictional, vaccination status, intent, and demographics.

## Filtering Functionality

### 1. **National or Jurisdictional Filtering**

Users can filter the data to view either national estimates or jurisdictional (state-level) estimates.

### 2. **Vaccination Status**

Users can filter the data based on vaccination status such as vaccinated (>=1 dose) and other relevant categories.

### 3. **Intent and Demographics**

Users can filter the data based on intent (e.g., intention to get vaccinated) and demographic categories.

## How It Works

1. **Data Fetching**: The fetchData.js script fetches vaccination data from the CDC website daily. The data is transformed and stored in MongoDB.
2. **Data Validation and Transformation**: The fetched data is validated and transformed to ensure cleanliness and relevance before storing it in MongoDB.
3. **API Endpoints**: The Express server provides API endpoints to fetch data from MongoDB. These endpoints allow filtering data by national or jurisdictional, vaccination status, intent, demographics, and year.
4. **Frontend**: The React frontend fetches data from the backend API and displays it in a user-friendly format. Users can filter the data based on the specified criteria.