# App Exercise – THE HACKETT GROUP - FULLSTACK

## Objective:

The objective of this exercise is to create a full stack application with an Angular frontend and a Django backend, which connects to the API provided by datausa.io and visualizes two charts, using a GraphQL API with two resolvers. Additionally, the Django backend should serve the Angular application’s distribution files at a specific endpoint. The application should follow the provided graphic template and meet the following requirements.

## Requirements:

1. **Frontend in Angular:**

* Create an Angular application with two components: one for each chart.
* Use the backend’s GraphQL API to fetch data for visualization.
* Use a chart library (preferably Chart.js) to create the charts.
* Implement the charts closely following the provided graphic template.
* Ensure responsiveness and compatibility across different devices and screen sizes.

1. **Backend in Django With GraphQL:**

* Create a Django backend that acts as an intermediary between the frontend and the DataUSA API.
* Implement a GraphQL API in Django using the graphene-django library.
* Create two resolvers in GraphQL to handle data requests for each chart:
* Resolver 1: Population data for Alabama, Florida, and California from 2013 to 2021.
* Resolver 2: Vehicle ownership data in US households for the year 2021.
* Serve the Angular application’s distribution files at a specific endpoint.
* Dockerize the Django backend to facilitate deployment and management.

## API Integration:

* **DataUSA API:** Provides access to various datasets related to the United States. The data must be properly handled, including error handling and data parsing.

## Chart Visualization:

1. **Multi-Axis Line Chart:** Shows the populations of Alabama, Florida, and California from 2013 to 2021.

* **API Endpoint Query**

https://datausa.io/api/data?drilldowns=State&measures=Population

1. **Pie Chart:** Shows the distribution of US households by vehicle ownership in 2021.

* **API Endpoint Query:** https://zircon.datausa.io/api/data?measure=Commute%20Means%20by%20GenderCommute%20Means%20by%20Gender%20Moe&geo=01000US01000US&drilldowns=Vehicles%20Available

## Graphic Template:

Please refer to the provided graphic template for guidance on the design and layout of the app page. Ensure that the colors, font (Google Font: Open Sans), and styling match the template as closely as possible. If needed, use the SVGs and images attached in the assets folder.

## Chart Libraries:

You are free to choose any chart library, but we prefer Chart.js ([Chart.js](https://www.chartjs.org)). Make sure to install and integrate the library into your Angular application properly.

## Deliverables:

* Source code of the Angular application.
* Source code of the Django backend.
* Dockerfile and Docker configuration for the backend.
* Documentation explaining the approach, implementation details, and any additional considerations.
* Instructions for running the application locally (if needed).
* Any additional notes or comments regarding the exercise.

## Submission:

Please submit your completed exercise, including the source code and documentation, within the specified timeframe. You can either send the files via email or provide a link to a GitHub repository.

## Additional Notes:

* Feel free to reach out if you have any questions or need clarification on the requirements.
* Focus on writing clean, maintainable code with proper comments and documentation.
* Thoroughly test the application to ensure functionality and responsiveness.
* Dockerize the backend to facilitate deployment and management.
* It is not necessary to implement or use a database, so the default Django configuration is sufficient.
* Good luck, and we look forward to reviewing your submission!