

Analytics Dashboard (Angular)

Objective:

Create a single-page analytics application using Angular that can render various metrics through Key Performance Indicators (KPIs) and charts. The application should allow users to configure the metrics' properties, such as their position on the page, type of chart, height, and width. Users should also be able to search for specific metrics/values throughout the page. Additionally, use a state management technique (e.g., NgRx) for managing the application state. Lastly, include test cases to ensure the application functions correctly.

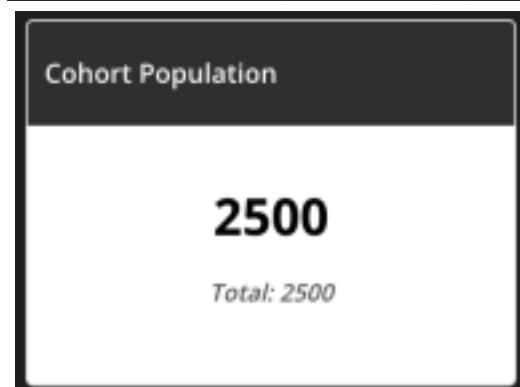
See attachments for charts/metrics that should be initially present on the screen.

Instructions:

1. Set up a new Angular project using the Angular CLI. Initialize a git repository here and add first commit. Make sure to add incremental commits with sensible commit messages.
2. Create the necessary components, services, and modules for the application.
3. Design the application layout and styles to present the metrics and charts appropriately. You can utilize the provided template (<https://akveo.github.io/ngx-admin/>) as a starting point or create your own layout.
4. Implement a configuration file (e.g., JSON) that stores the metrics' properties, such as their type, position, and size.
5. Load the configuration file dynamically using Angular's HttpClient and use it to position and render the metrics on the page.
6. Implement a search feature that allows users to search for specific metrics/values throughout the page.
7. Use a state management technique (e.g., NgRx) to manage the application state. Set up the necessary actions, reducers, and selectors to handle the metrics and search functionality.
8. Integrate a charting library (e.g., Highcharts) to render the metrics based on the configuration. Follow the library's documentation for integration and usage.
9. Ensure the application is responsive and compatible with different devices and screen sizes. Implement appropriate responsive design techniques and test the application on various devices.
10. Write test cases to validate the functionality of the application. Include unit tests for components, services, and reducers. You can use testing frameworks such as Jasmine and Karma for writing tests.
11. Include any necessary instructions for setting up and running the application. Provide clear steps for installing dependencies, starting the development server, and running the tests.
12. Share the final git repository by making it public.

Note: You are **not** required to implement any backend functionality or connect to a live data source. Focus solely on the frontend part of the application using Angular.

Attachments:



Total ER

April-2023 (Till Today)

3969 ▼ -11.0%

March-2023: 4459

Total Admission

April-2023 (Till Today)

4179 ▼ -7.4%

March-2023: 4511

Average Cost

April-2023 (Till Today)

\$11.31 ▼ -1.7%

March-2023: \$12

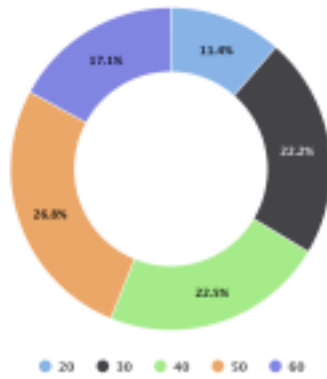
Total Specialist Visit

April-2023 (Till Today)

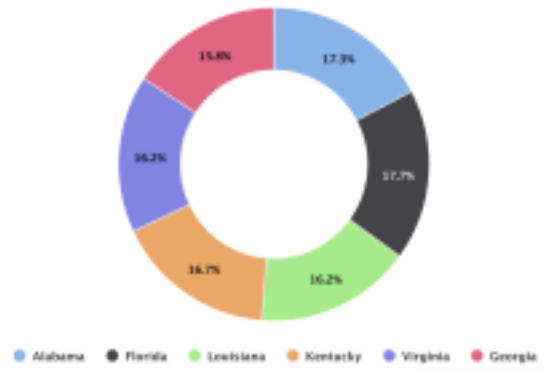
5244 ▼ -9.3%

March-2023: 5779

Population By Age



Population By Location



Year On Year Analysis

AVG ADMISSION AVG ER AVG SPECIALIST AVG COST

