ReadMe

Zendesk Coding Challenge

Project Structure and Languages Used: -

There are two major parts in this code base

1. Backend -> This is a Flask app which is a Python framework. The sole purpose of this is to make the API call to the Zendesk Tickets API and provide it to the frontend on demand. As we cannot make API call to Zendesk API using frontend JS because of CORS issue a separate backend application was created.
2. Frontend -> This is an angular app which is a JavaScript frontend framework developed using Nodejs. All the logic other than ticket fetching from Zendesk is implemented here.

The easiest way to run the app: -

1. Backend
    I. I assume you have python and pip package installer in your machine, and you are using windows machine to execute this program. However, you could easily find the Mac and Linux execution instructions online.
    II. In the downloaded Git repo, you will find a folder named "backend" open a command prompt inside this director and run "***pip install -r requirements.txt***" this will download all the required python packages. Then run "***python app.py***" in the same directory. This will spawn a local server on port 7999. If there is any other application running on that port either change the port of this app or kill the other program. That's it, major part is done now we have a running backend server.
2. Frontend
    I. After downloading the git repo follow this path to reach the prod build of Angular app. Zendesk -> zenticket -> dist -> zenticket. In this directory you can find index.html file on double clicking it your favorite browser will start executing it. See as I said before this is the easiest way to execute.

Running the Development server for Frontend (Not a mandatory): -

There is no change in spawning backend server. But for the frontend we can run development server by following these steps.

Firstly, install the nodejs from https://nodejs.org/en/ . Then open zenticket directory from the repo and run "***npm i***" this will install all the packages required for running the application. Then run "***npm start***" or "***ng serve***" this will spawn a development angular server in watch mode at http://localhost:4200/. *This way of running things only helps you if you want to make changes in the code and see how the app behaves.*

How to use the app: -

It is great you are now successfully running the app and just need some credentials for the ticket API authentication.

After running the front end, you will see three prompts one after the other asking email, password, and subdomain please carefully enter your credentials because they are key to fetch and populate data. However, in the process of entering you might enter wrong credentials unknowingly in such situations strong error handling techniques were in place to guide you to correct you mistake. ***In order to reauthenticate you please reload the browser again***.

Hurray! Now you are seeing your tickets in list view in the bottom of the app you will find a paginator which can be used to configure how many items you want to see in one page. Once you find the ticket you are searching click on the row to open a detailed view and get more information about the ticket. Then you can come back to list view by clicking "Back to All Tickets" button.

This is about the visual part of the application.

Understand the structure and developmental principles of the Application: -

TypeScript which is a superscript of JavaScript is used to code in Angular because this will allow us to write object-oriented style code with strict typing which saves tons of developmental issues in long term.

UI decisions and Enhancements: -

The ticket list view is given in a table so users can easily understand that there are many tickets, and each row represents one ticket. Hovering on a row changes the background color of that row. Although this is a simple feature this gives information like the rows are interactive and clickable and which row user is clicking on.

The table header and pagination bar are sticky in the UI this will give information about which column represents which feature instantly rather than scrolling all the way top. Paginator is like a control panel so fixed paginator will allow user to quickly control the UI as a result user experience will be enhanced.

Loaders play a major role in enhancing the user experience this application has loaders developed in almost all the situations where we need.

UI is responsive open it in mobile, tablet, PC this application gives better experience irrespective of the device because of its responsive design.

Dates in the application are formatted to give most information to the user rather than providing in some string format.

Error handling: -

Most of the components in this app requires human interaction or other application interaction which keeps us in a place where errors are common. So, handling those errors are important else users are not informed about the state of the UI.

This application handles ***invalid subdomain***, ***invalid credential***, and other ***network errors*** and informs user with nice UI and with appropriate action they need to take.

Code Modularization: -

The code in this application has four components header, ticket list, ticket detail, and main component. Writing in multiple components rather than in single one allows us to easily debug, understand, reuse code. For example, rather than writing the same header code in ticket list and details page we can write standalone header component and reuse it.

Unit Testing: -

Unit testing allows us to find the bugs during the developmental state itself and provides us clear information if we disturb the old features while adding new features. Although in small scale projects unit testing does not make significant difference it will show a stellar difference in huge multiple-team development projects.

In this project we have used Karma-Jasmine to write unit tests and achieved ***95%*** test coverage. In order to see the coverage report, go to Zendesk -> zenticket -> coverage -> zenticket and open the index.html file in this directory. This will provide a nice GUI view of the test coverage.

In order to perform the tests again go to zenticket directory and run "***npm test***" this will provide the response in cmd and updates the coverage report file directly (mentioned in above paragraph).

*****Refresh the page if you want to try with some other login*****