Technologies Used in the Project

Techstack used for Frontend:

- HTML
- CSS
- Javascript
- jQuery
- AJAX

Techstack used for Backend:

- PHP
- Phalcon
- MongoDB

Additional technologies and services:

- Docker
- Composer
- Firebase JWT

Steps to run the Project

Prerequisite (technologies) required to run the Project:

- Docker
- Composer

Steps:

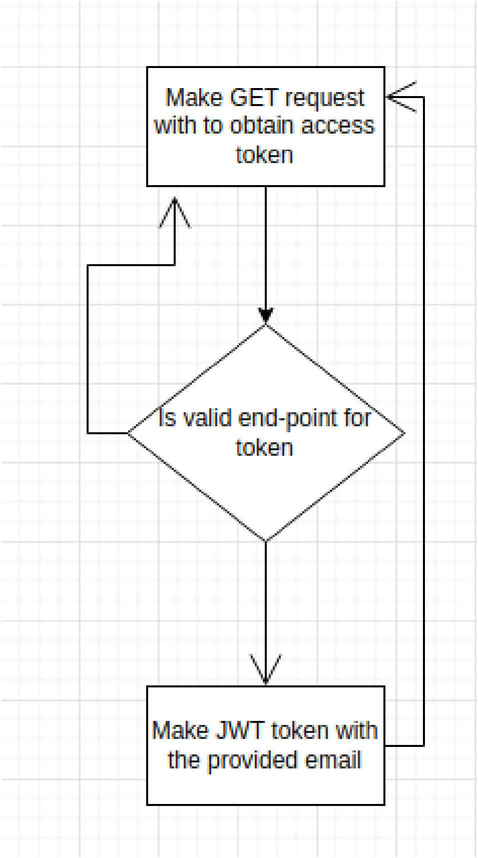
- Clone the repository inside a folder.
- Make sure all the required technologies all installed.
- Go inside the *backend* folder, then inside the *src* folder
- Make sure that you are at the correct place by verifying the presence of *composer.json* file
- Install the dependencies using the command – *composer install*, if the command does not run properly use *sudo* i.e. – *sudo composer install*
- Then compose the docker using the command – *sudo docker-compose up*
- Now the backend is running and one can make API calls using the POSTMAN
- To run frontend one can open it directly open in the browser or open the index.html using live server.

*** Note :- All commands are for LINUX system**

Project Overview

Process for APIs access :

- All the API endpoints are protected using JWT, except the *getToken* endpoint
- To access any API endpoint, one needs to pass *bearer* as a param in the URL e.g. http://0.0.0.0:8080/feedbacks?bearer={bearer_token}
- Bearer token is obtaned by hitting *getToken* endpoint and passing email in the URL as follows :- {base_url}/getToken/{email}
- After accessing the bearer_token, one can easily call any endpoint and passing the bearer along in the URL e.g. - http://0.0.0.0:8080/feedbacks?bearer={bearer_token}


Make GET request with to obtain access token
Is valid end-point for token
Make JWT token with the provided email

Flow of obtaining token

Process for Frontend:

- I needed email for getting the access token hence I have also created one field for that as well. Though in real life scenario, the user will be already logged in so it wouldn't be required in that case.
- After validating all the details, JS make an AJAX request on the *getToken* endpoint to obtain the access token
- Once the access token is obatined, the JS make another request to POST the feedback data on the /feedbacks endpoint

Response Codes used:

- 200 – OK, for successfully getting the from GET request
- 201 – Created, for successfully sending the data
- 400 – Missing data, when the POST request have some missing fields or data