Manual

Project Name

SMART-on-FHIR app to pull PDMP

Team Name

Code Blue

Team Members

- Laura Pike
- Adam Sligar
- Imran Yousuf
- Alison Jing Huang
- Tony Leung

GitHub Link

https://github.gatech.edu/gt-cs6440-hit-spring2019/SMART-on-FHIR-App-to-pull-PDMP

Setting up the local development environment

To setup the local development environment, first install Docker and Docker Compose by following the installation instructions at https://docs.docker.com/install/ and https://docs.docker.com/compose/install/.

Next, open the terminal and start the local development server by running `docker-compose up` in the project root directory. The necessary dependencies will be automatically installed at this step. The app should be viewable at http://localhost:4200.

Developing the application

The application-specific source code resides in the `src` folder. Any changes made to the `src` folder will be automatically reloaded by the development server and viewable on the browser. There are three main folders in the src folder: - `app` - the main components of the app reside here. - `assets` - static assets and libraries are stored here. The FHIR JavaScript client is located here. - `environments` - contains environment-specific configurations. For example, this could be use to toggle different endpoints for a development or production FHIR server.

The SMART-on-FHIR app uses the Angular framework and follows the component-based architecture. Each component controls a portion of the app and could contain a set of HTML, JavaScript, and CSS files. The SMART-on_FHIR app contains the following components:

- `Find` - the component responsible for the Patient ID search box.
- `Report` - the component for generating the PDMP report for a specific patient.
- `FhirService` - contains an API wrapper for the FHIR client library located in the assets folder.
- `App` - the container component that holds the other components and controls the application's routing logic.

Additional components can be added in the same manner as the other components in order to extend the functionality of the application.

Testing the application

The Angular framework supports unit tests and end-to-end tests. End-to-end tests can be added to the e2e folder located in the project root directory and executed by running `ng e2e` using the Angular command line tool (https://angular.io/cli). Unit tests can be added to each component by specifying a *.spec.ts file (i.e. app.component.spec.ts) and running `ng tests`. Upon running the tests, the test results will be automatically displayed in a browser.

Running the application

Please refer to the Special Instructions PDF for instructions on how to run the application.