

Application Manual

1 ARCHITECTURE

As the tools and architectures used are mentioned in the Design Document the information will not be repeated here in general, but will talk about the specific implementation in our website.

Using NodeJS and Angular as a framework, the following helpers, services, components, and assets have been created which allows the application to function.

Setup instructions are provided in the Special Instructions document.

2 Helpers

2.1 AuthGuard

Checks that the user is authenticated via both the Smart App Launcher and our provider's custom authentication before allowing access to any authentication-only pages.

3 Services

3.1 FhirService

Checks that the user is authenticated via the Smart App Launcher and makes live FHIR calls for production deployment on GitHub pages, while returning mock content for local deployment.

3.2 PatientService

Checks that the user is authenticated via our provider's custom authentication and returns mock content for our provider's existing APIs.

4 Components

4.1 AskAQuestionComponent

The "Ask a Question" page of the website. Allows the user to submit a question to our provider about treatment, financing, etc. Uses our provider's existing API to submit.

4.2 CareTeamComponent

The "Care Team" page of the website. Calls our provider's existing API to get information about the individuals on the team who dealt with the user's acute visit and displays their information.

4.3 ContactInformationComponent

The “Contact Information” page of the website. Displays static information about the contact information of our provider.

4.4 FooterComponent

The static footer of the website which simply states it is copyright content. Sticks to the bottom of the webpage.

4.5 HeaderComponent

The header of the website which dynamically changes based on authentication status. Includes a navbar for in-app navigation.

4.6 HomeComponent

The “Home” page of the website. This is the only component that makes live FHIR calls and displays the information to the user.

4.7 LabResultsComponent

The “Lab Results” page of the website. Calls our provider’s existing API to get information about the lab results that were run during their acute visit, a description of each file, the date the files were uploaded, and allows them to download them.

4.8 LaunchComponent

Created for the implementation of authentication using the Smart App Launcher. After this page authenticates, routes to the LoginComponent to finish authentication.

4.9 LoginComponent

Created for the implementation of authentication with our provider’s authentication APIs. Runs after the LaunchComponent and verifies the user is registered in our company database.

4.10 NotRegisteredComponent

The “Not Registered” page of the website. Displays static information to the user that they are not fully authenticated / registered to use our application. Currently tells the user how to authenticate both in production and local environments for the sake of testing the application.

4.11 PageNotFoundComponent

The “Page Not Found” page of the website. Displays static content to the user informing them that that URL they tried to access is not valid. Should not happen if using nav-bar navigation.

4.12 PatientEducationComponent

The “Patient Education” page of the website. Calls our provider’s existing API to get information about the possible treatment suggestions and any other important information about the condition they were diagnosed with during their acute visit.

5 Assets

Holds all of the mock data for our application, including static pictures and API responses. If any of this information needed to be secured additional steps would need to be taken, however that is currently not the case. Would be a good future enhancement regardless though.