**WeCasa File Architecture**

**HAGSJP.WeCasa VSCode Project**

-> **/Frontend**

-> **/ClientApp**: *Files necessary for all frontend elements*

-> **/src**

-> **/assets**: *Photos*

-> **/components**: *Javascript files for user interface components/views for every group of work (ReactJS)*

*- Generic ReactJS component file structure always includes*:

- export const function for the component itself

- return statement containing all HTML for view

- export default component

- *Hooks within ReactJS components*:

- allows the data to be passed between different views/components

- useState

- AuthContext.js: global variables within frontend views

*->* **/scripts**: *Javascript files related to functional scripts within the frontend*

*->* **/styles**: CSS files for styling (these CSS files do not override AntD/Bootstrap/third-party component styling)

-> **index.css**: This file overrides AntD/Bootstrap/third-party component styling

-> **/Controllers**: *Files for all Web API controllers (accesses Services in the backend)*

-> **/WebAPIModels**: *Files for all Data Transfer Objects / Models (sending server/controller data to and from ClientApp for API calls)*

Other file directories outside of /Frontend are in the backend

-> **/DataAccessLayer**: *Classes necessary for database access*

*->* **/Services**: *Class implementations for the services that will be called in Frontend Web API Controllers*

*-* Services and managers are the main ways to make calls to the /DataAccessLayer

-> **/Models**: *Backend & Frontend models*

*-> Shared representation of data for front-end and back-end models*

*-> Models should mimic the database*

*->* **/Security**

-> **/AppIdentity.cs**

-> **/AppPrincipal.cs**

-> **/AuthorizationCommon.cs**

-> **/Claim.cs**

-> **/Claims.cs**

-> **/OTP.cs**

->