**Architecture**

**Web**

**Tests**: Found under umbuy/project/umbuy/web/src/tests

Inside the tests folder, you have a *unit* folder for all unit tests and an *integration* folder for all integration tests.

**Presentation layer (Front-end)**: Found under umbuy/project/umbuy/web/src/app/presentation

Inside the presentation folder, you have a *html* folder for all html templates and a *css* folder for all css or scss files.

**Business logic layer**: Found under umbuy/project/umbuy/web/src/app/business

Inside the *business* folder, you have all the business logic of the application. You will notice that files have different names. The *app-routing.module.ts* deals with all the different possible routes in our angular app. The *app.module.ts* contains all the components we are using, the modules we are using and the providers or services we are using. Finally, the files that have component in their name, is business logic and has a html and css file for it under the *presentation* folder.

**Persistence (Database)**: Found under umbuy/project/umbuy/web/src/app/persistence

Inside the *persistence* folder, you have a *localhost* folder which contains the *localhost.js* file used to start the local database and the *database.mwb* which contains all the tables and their schemas. Also, in the *persistence* folder there are files used to query the database using http requests.

How it works is you have the business logic layer do a call to a method in a file in the *persistence* folder such as *advertisement.service.ts*. Then that method will do an http request to the node.js file named *localhost.js* which is found in the *localhost* folder in the *persistence* folder. In localhost.js, we are using Express.js and it will match the part of the url after *localhost:3000* for localhost. It will match the url using Express.js and then will perform the method that will execute the sql query on our mysql database.

**Shared**: Found under umbuy/project/umbuy/web/src/app/shared

Inside the *shared* folder, you all files that didn’t quite fit under the presentation, business or persistence layer. As well the files in the *shared* folder are files that are being shared in more than one file and made the most sense to be in this folder to avoid confusion.

**Architecture**

Our app is a 3-tier architecture. It consists of a presentation layer that consists of all the html and css files, a business logic layer which consists of all the logic in the app in form of .ts files and persistence layer that communicates with the database.

A typical page will consist of a html, css and business logic layer file with the word *component* in it. The business logic layer file knows which html and css file belongs to it. That way we can have 2-way data binding. In other words, we can take data from the business logic file and show it in the html file using {{ }} also called interpolation or we can send data to the business logic layer using [ (ngModel) ] which can also be used to update the variable in the business logic layer and also show the changes in the presentation layer. As well, the business logic layer can do calls to the persistence layer which will do http requests to the mysql database using node.js with express.js.