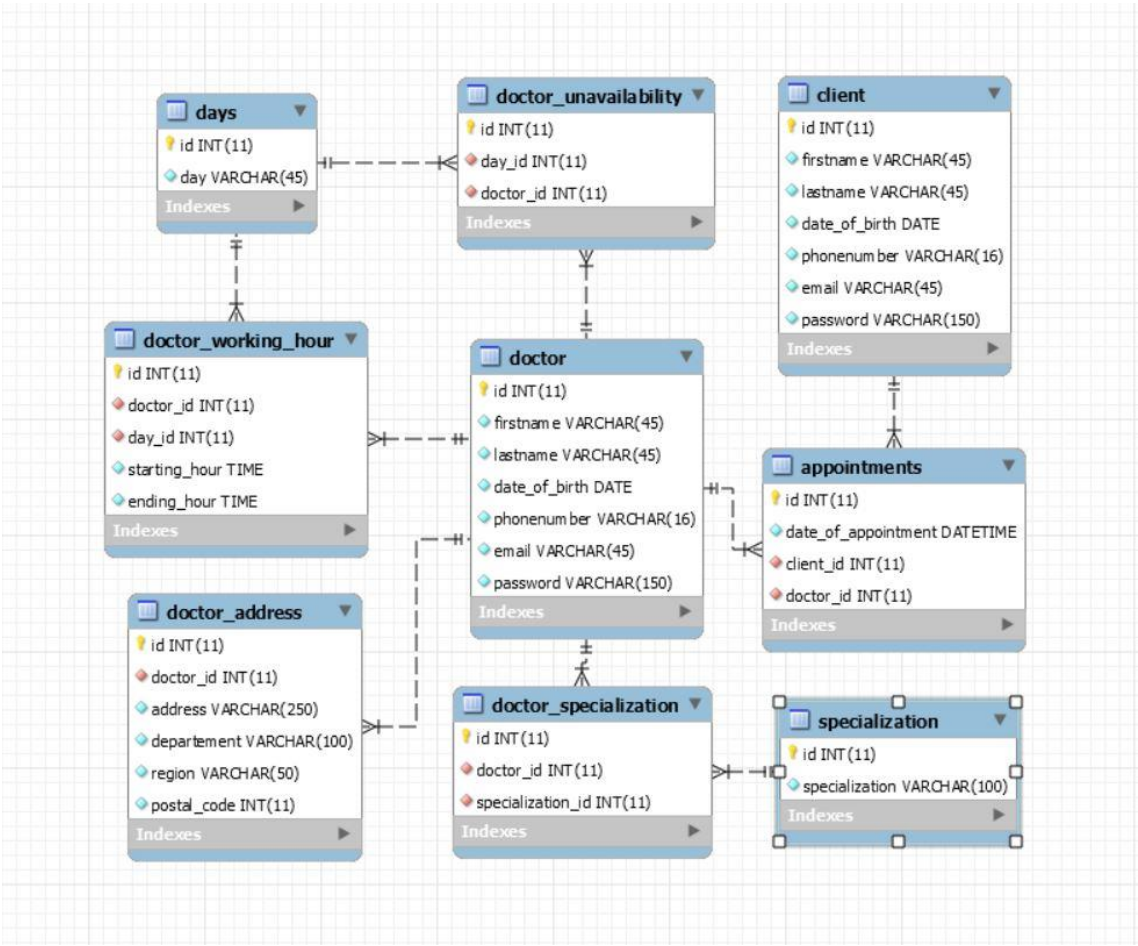
2PROJ Backend-Documentation :

Database Structure:



The Database structure of the Medical Platform was created using MYSQL Workbench and is separated into 9 main tables:

- **Client:**

The client table includes all the information that pertains to the client user.

- **Specialization:**

The Specialization table is a table that includes the types of medical practices that the doctor can register as on the site.

Though the access to the table itself is currently restricted and cannot be accessed by website directly nor the backend as to prevent inappropriate practices to be added onto the interface.

Currently, only those who have access to the database directly can add more specializations.

- **Doctor:**

The Doctor tables contains only the personal information of the doctor themselves and not their work information as to keep an orderly structure of information. Because a doctor may have multiple practices, addresses and different work schedule.

- **Doctor Specialization:**

The Doctor Specialization table keeps a record of relationship between the various specialization that a doctor may have by associating the doctor's unique id to the id of the specialization they choose.

Thus, allowing the database to keep track of the various doctor specialization.

- **Doctor Address:**

The Doctor Address table keeps a list of the various addresses a doctor may have during their practices.

- **Days:**

The Days table contains a list of days within a week and is used to help establish a doctor's schedule alongside the Doctor_Working_Hour and the Doctor_Unavailability.

- **Doctor Working Hour:**

The Doctor Working Hour has information on the exact time that a doctor will be working in a day from the start of their day to the end.

This information is then later used to establish a time that the client can book an appointment with that doctor.

- **Doctor Unavailability:**

The Doctor Unavailability table is created as a precaution for the Doctor Working Hour to override the hours they would usually work, in case the doctor is going to be busy that day and cannot work without changing their schedule entirely.

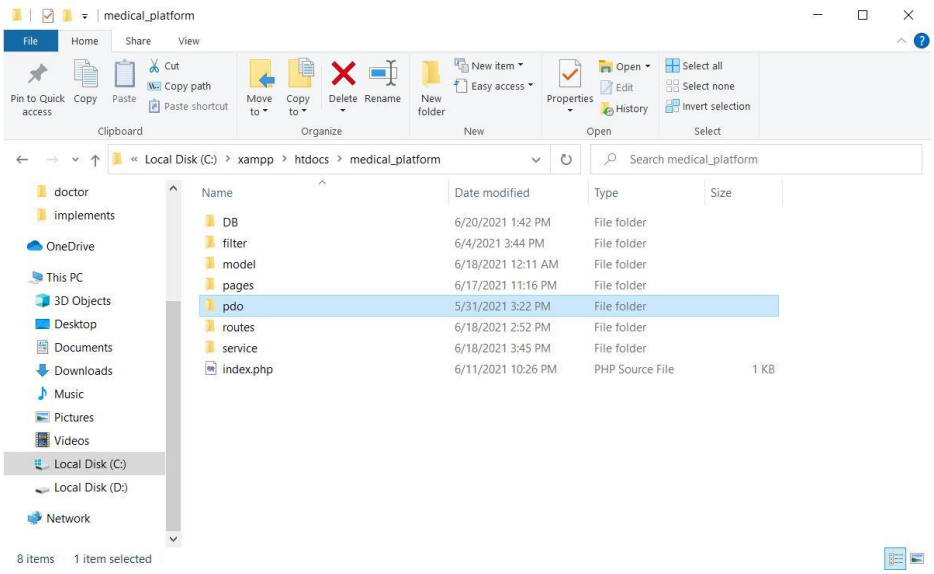
- **Appointments:**

The Appointments table is used to set up a datetime of the rendez-vous between a client and a doctor.

The language used in the Backend of this project is **PHP** as an **Object-Oriented Programing** Language.

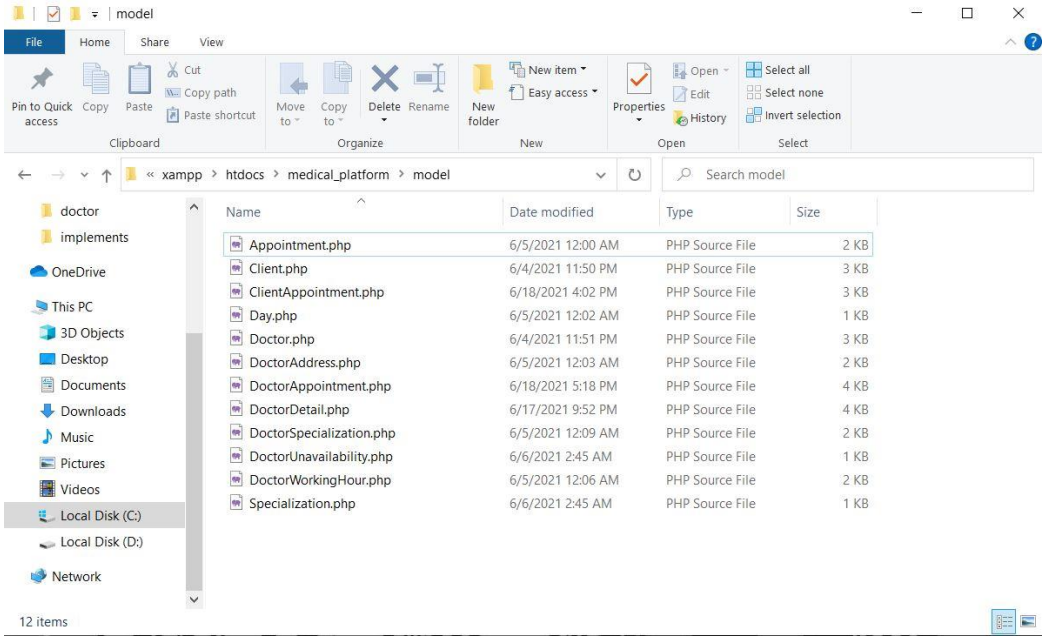
The Backend portion of this project is dived into **5 main folders** that make up the entire structure.

Each folders has their own purposes and rely on each other in order to smoothly run the entire process of the site.



Model:

The Model folder is the folder that contains classes that is the representation of the data that has all its attributes and the getter and setter methods.



PDO:

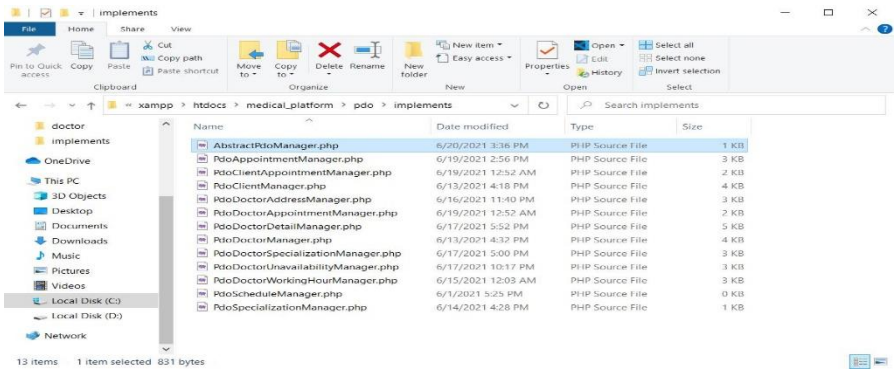
The PDO represent the connection of the backend to the database.

This folder contains 3 important subcategories:

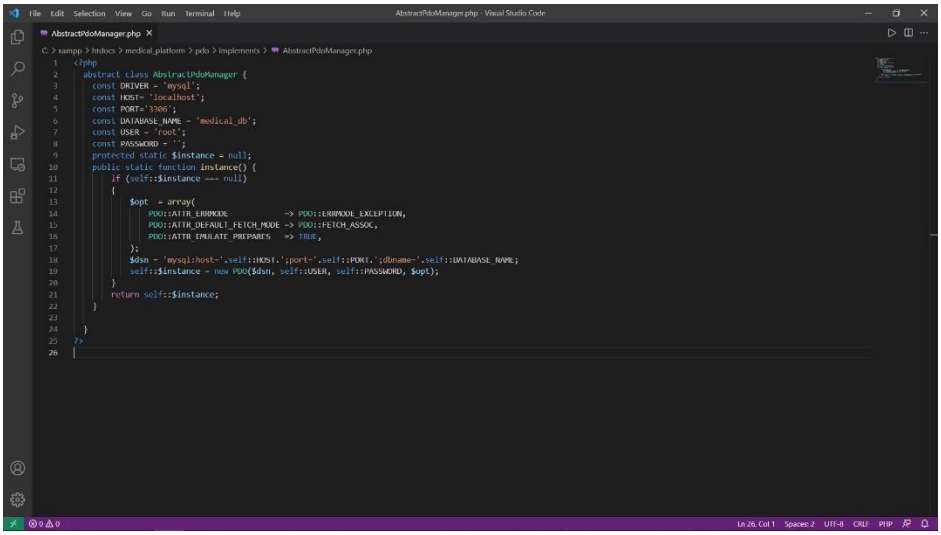
- ***Implements*:**

  The Implements folder contains the functions that allows you to directly create, update, retrieve and delete the information within the database.*

  Each Pdo class inherits the AbstractPdoManager in order to consume the PHP MySQL Connection and implements the corresponding interface.

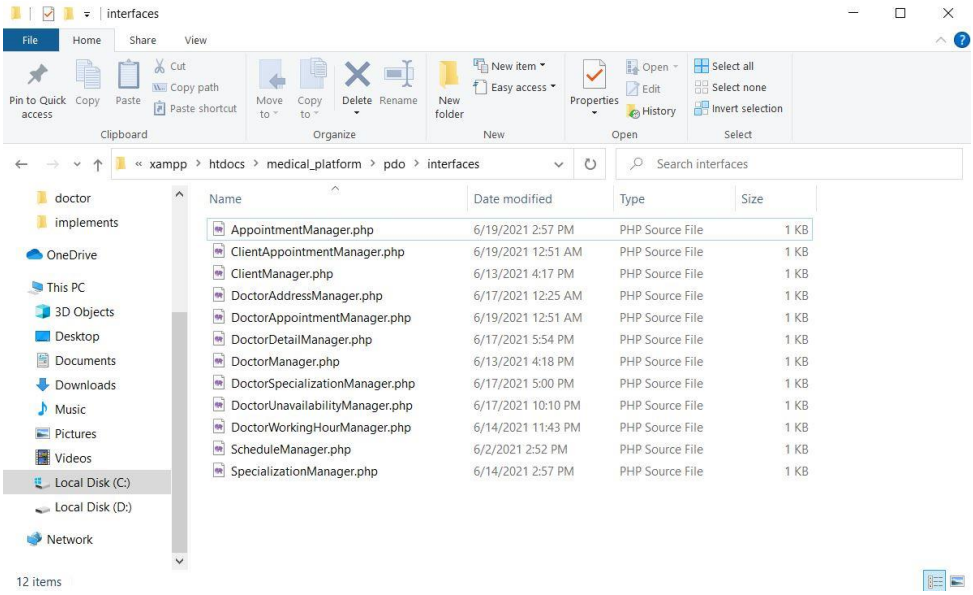


*AbstractPdoManager.php is the file that establishes the connection of the backend to the medical database that we previously created.



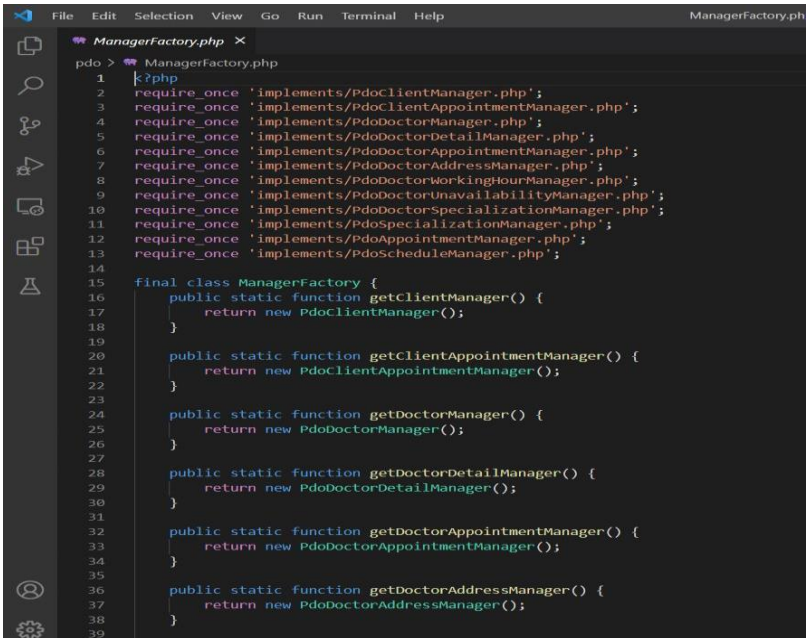
- *Interfaces:*

The Interface folder contains the files that sets the rules of what function can be implemented in a class for each of its respective PdoManager file from the implements folder.



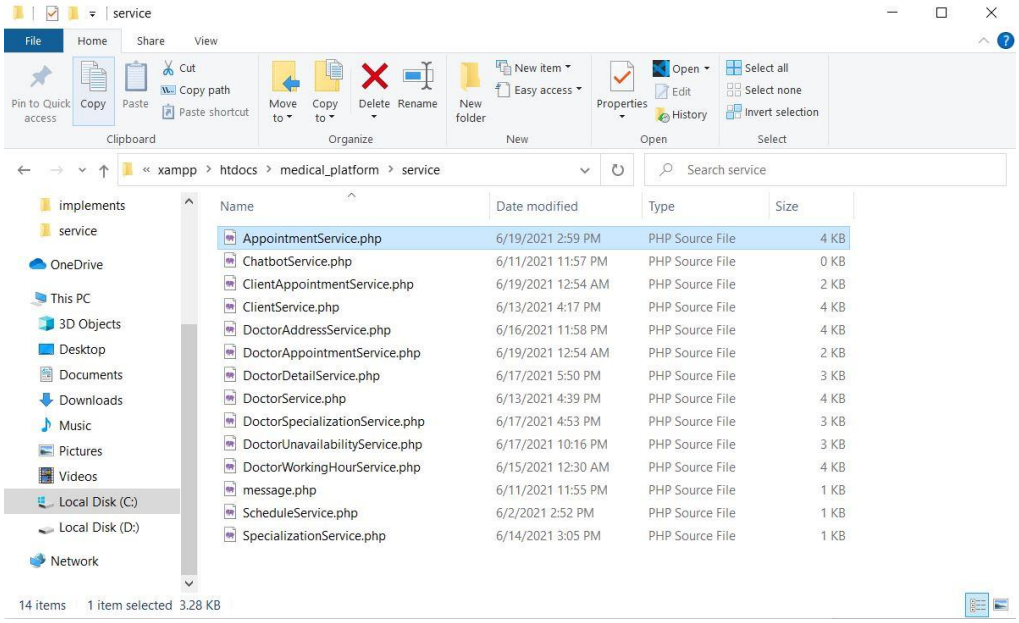
- *Manager Factory:*

The Manager Factory exports all the classes that we have implemented from the implements folder.



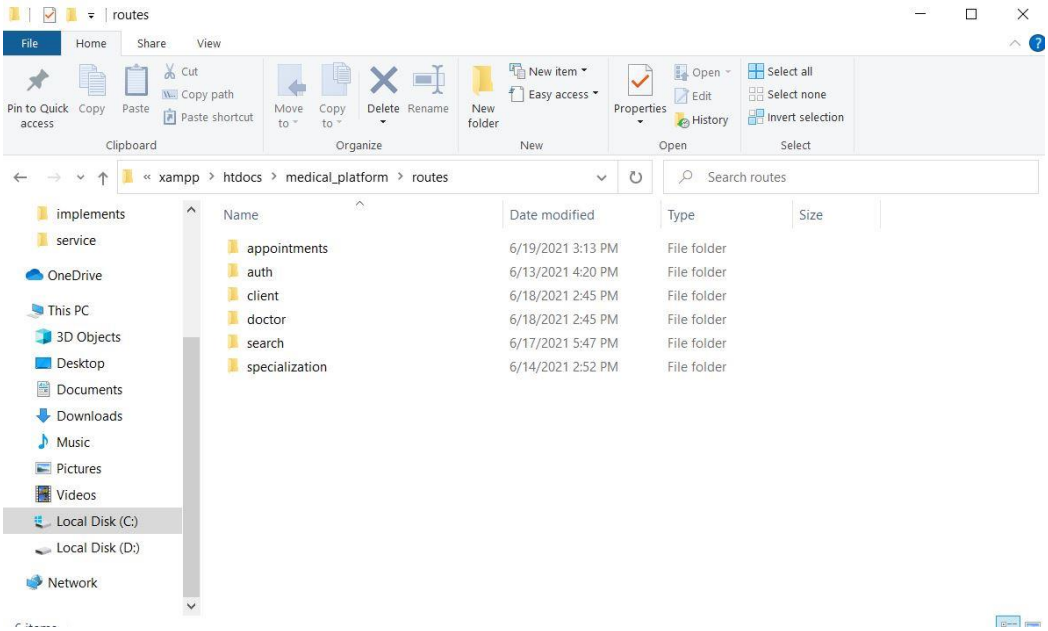
Services:

The Service folder contains files that houses the functions that defines the business logic of the application rather than having it in the databases directly and slowing the procedure down.



Routes:

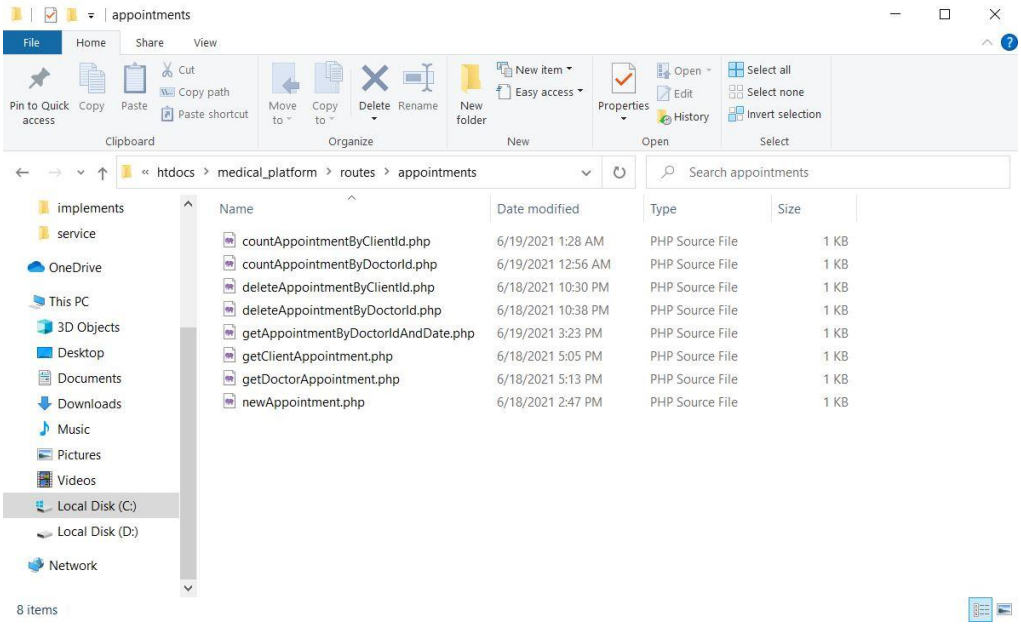
The Routes folder contains the php files that is used for receiving the requests and responds by launching the services function in accordance with that particular request.



*The route is set to respond to everything in json.

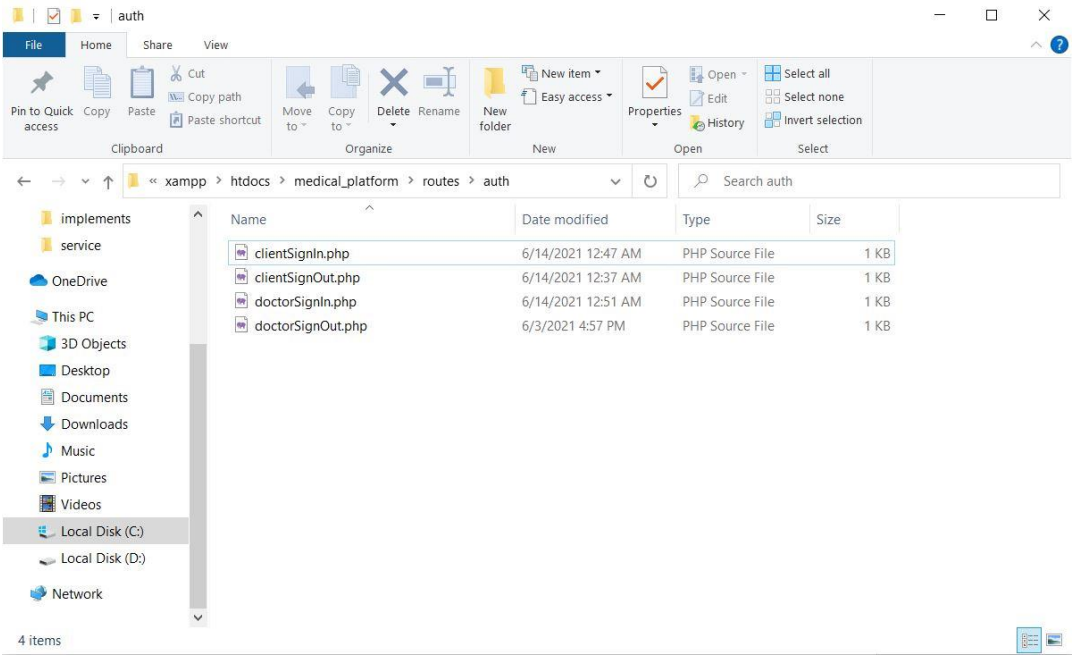
- ***Appointments**:*

  The appointments folder contains all of the files that relates to all of the management of the appointments request within the site.



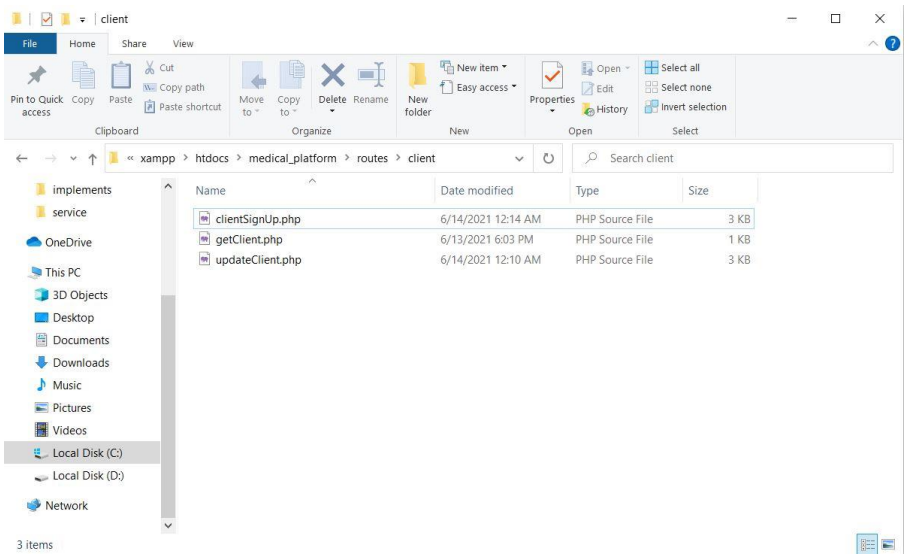
- *Auth:*

The auth folder is where the user session is managed.



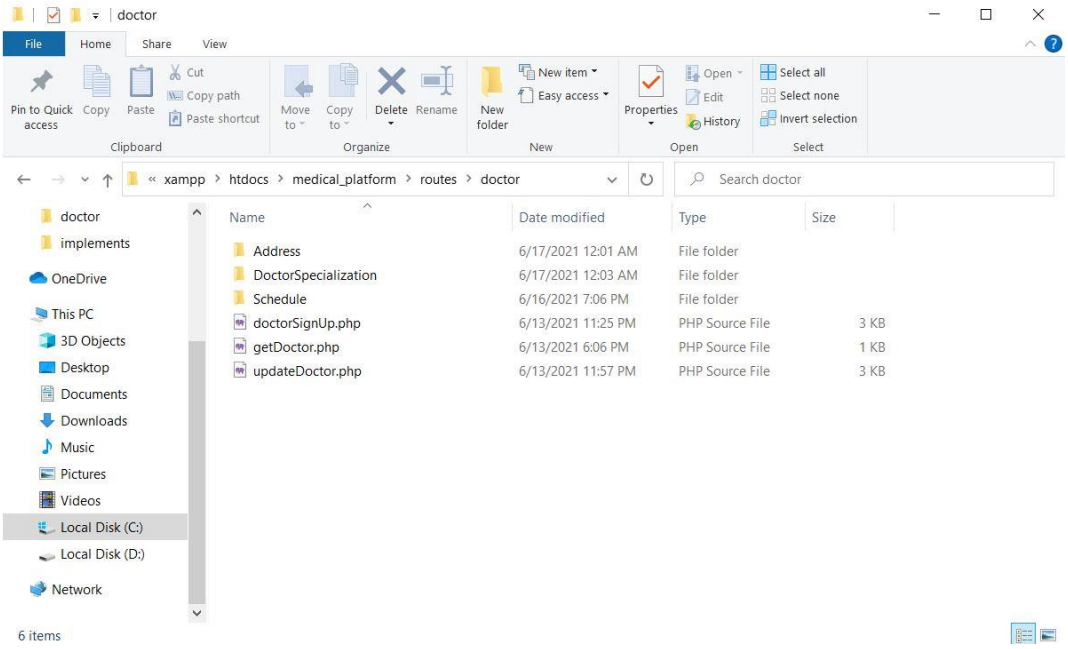
- *Client:*

The client folder is where all of client information is managed.

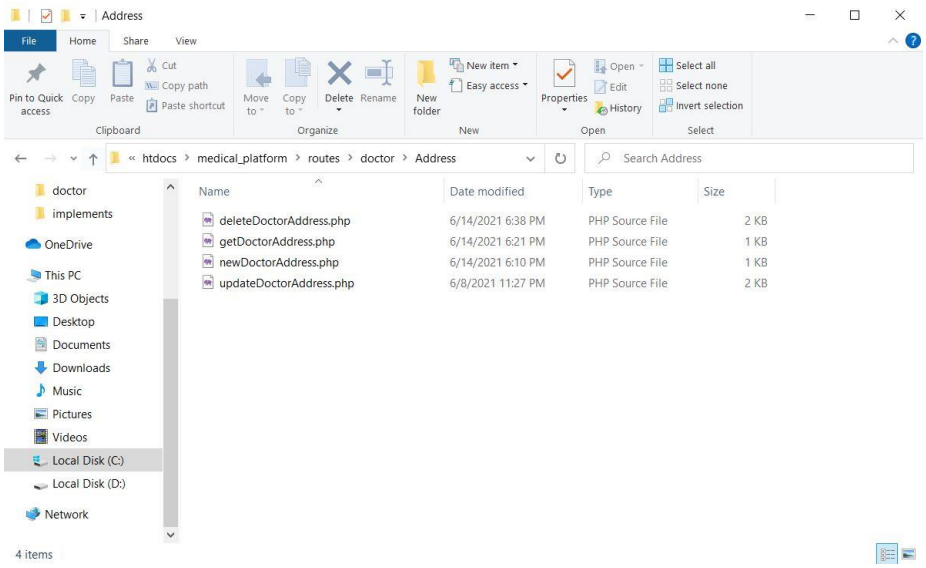


- *<u>Doctor</u>:*

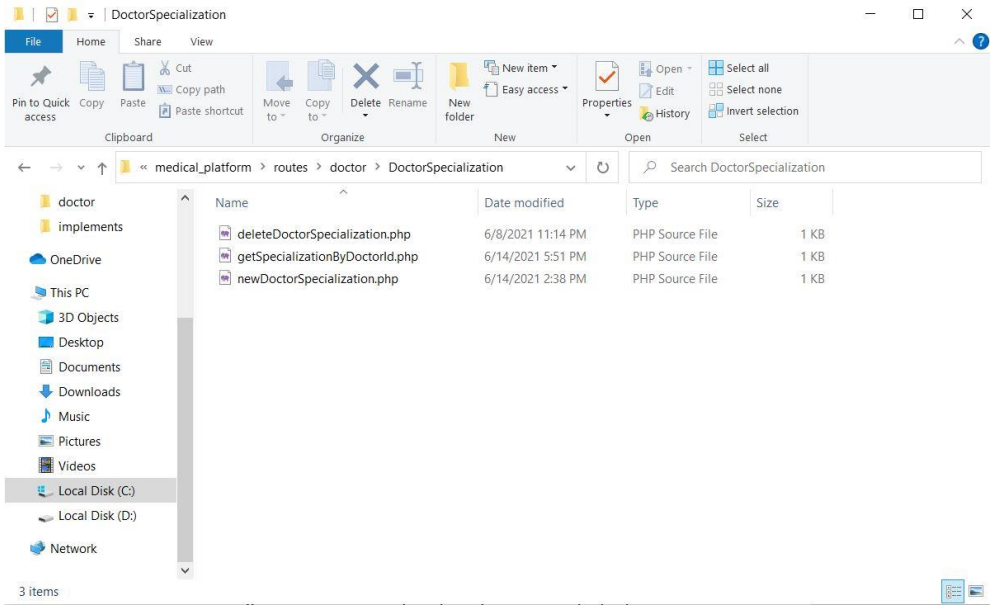
The doctor folder contains all of the files that manages everything that is related to the doctor from their personal information to their specializations.



<u>Address</u>:

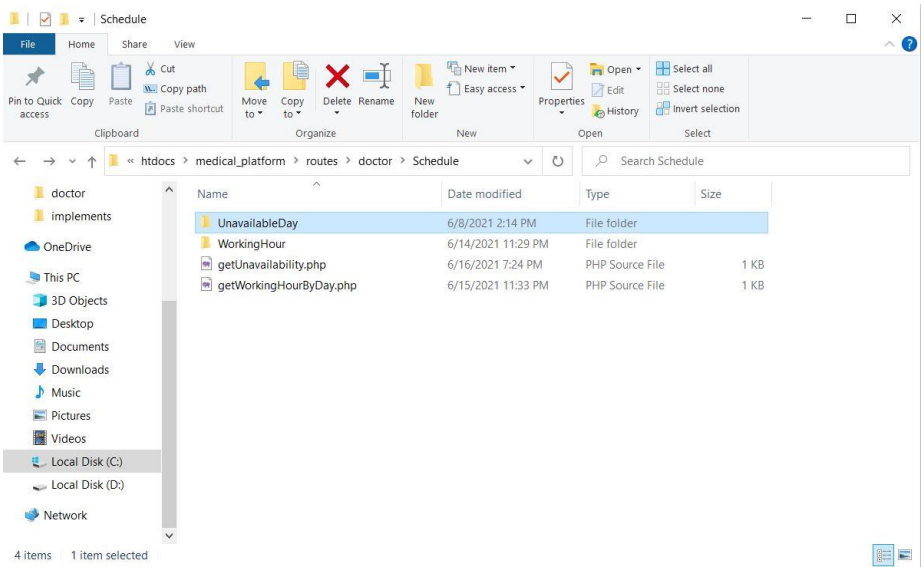


DoctorSpecialization:

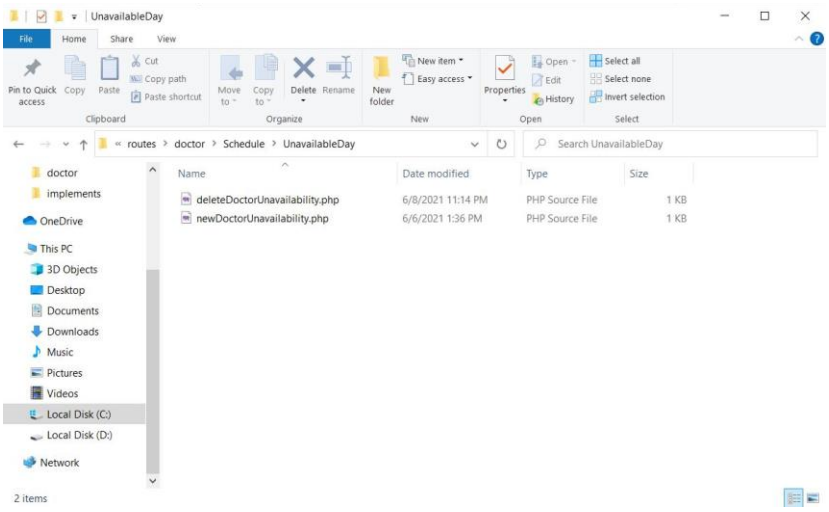


Schedule:

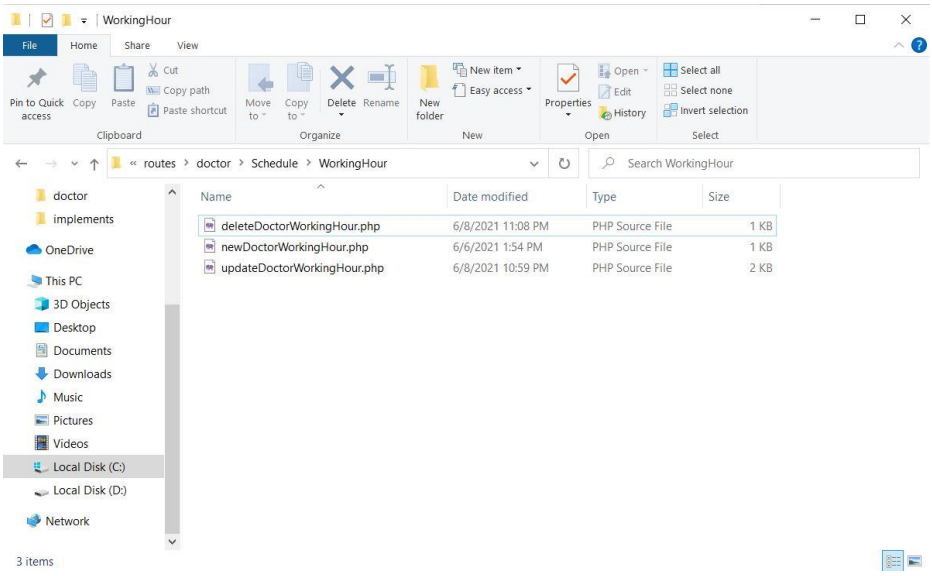
The schedule folders manage all of the requests relating to the doctors' time during the week.



UnavailableDay:

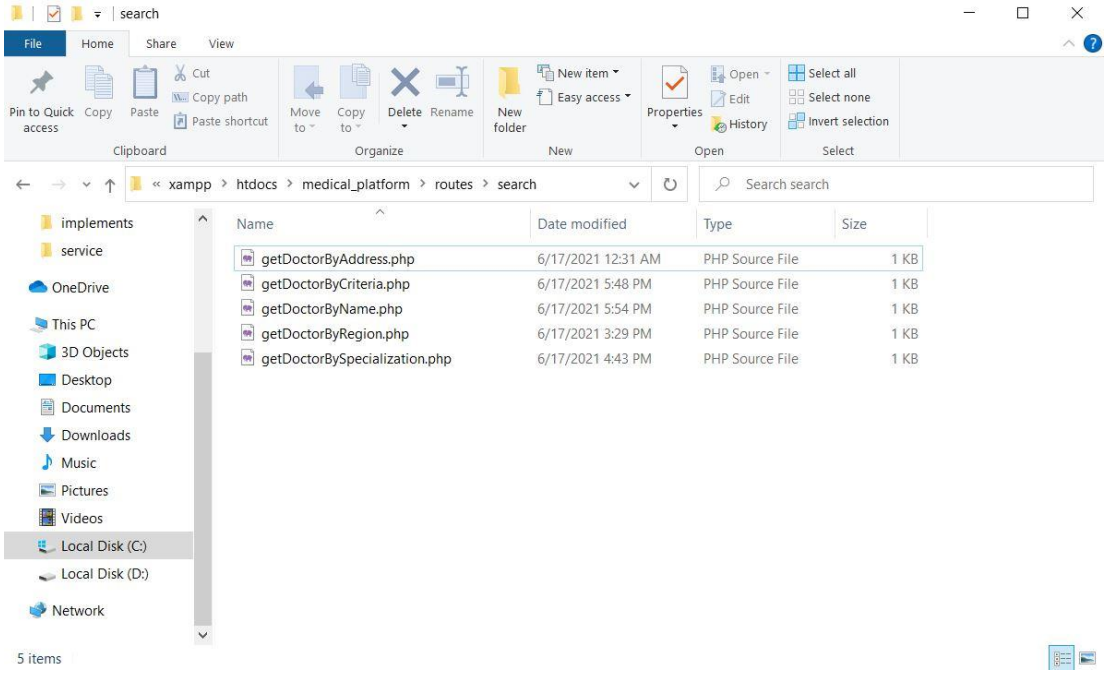


WorkingHour:



- *<u>Search</u>:*

The search folder manages all of the search request (as requested of the project) within the site that relates to the finding and booking of a doctor.



- *<u>Specialization</u>:*

The specialization folders contain the *getSpecialization.php* file that grabs a list of specialization from the database for the doctor to choose from as their specialization.

Filter:

The filter folder acts as a gatekeeper that prevents unauthenticated users from accessing certain pages.

In the case of this project, we have created 2 types of filters that requires the proper authentication to access the **Client Based** Pages and the **Doctor Based** Pages.