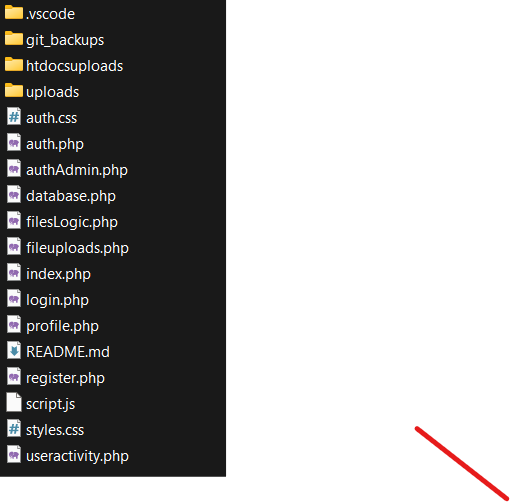
Document Tracker System Technical Documentation  
GROUP 2

App Structure

We didn’t follow any particular web framework infrastructure. But stuck with a simple and conventional Software Structure. Which is HTML, CSS and JavaScript for the front-end and PHP with MySQL for our backend.



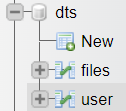


Figure 2. Project Database

Figure 1. Project Folder

DTS\_GROUP2 is the main folder containing all of the important files like project settings and configurations to run the software.

Our main landing page and app in contained in the “index.php” file. It is where the user interface, document tracker system is contained and all the other functionalities are connected to.

We created separate files for each of the major functionalities for modularity and ease of access of the developers. Also giving a controlled environment for adding new functions, testing and modifying functions.

In the project folder these files contain:

PHP/HTML

* auth.php contains the Login form
* register.php contains the Register form
* index.php contains the landing page and the app
* filesLogic.php manages the CRUD functions

CSS

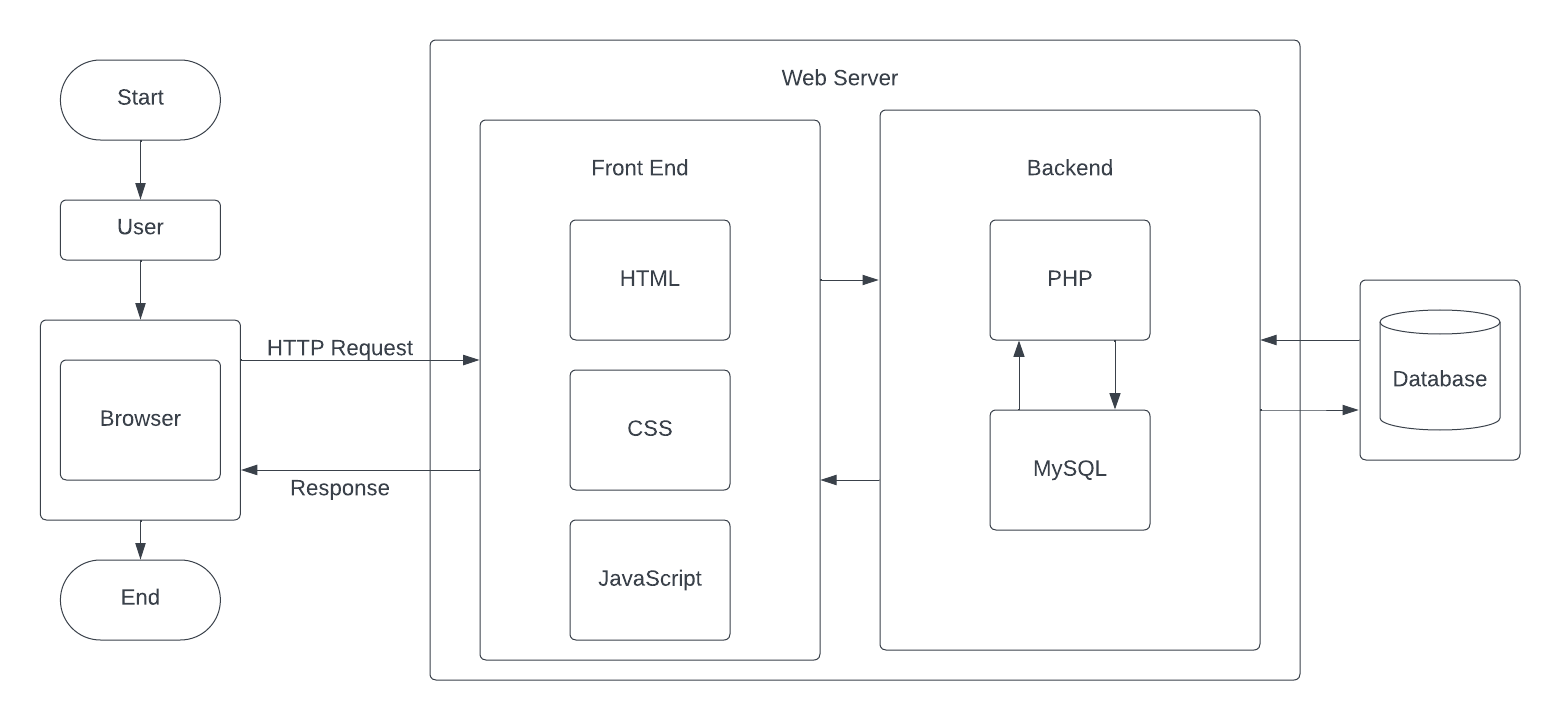
* auth.css is the styling of the Login and Register page
* styles.css is the styling of the landing page

JavaScript

* script.js manages the functionalities of the HTML

Folders

* uploads stores all our uploaded documents from the database



Software Architecture

The Architecture of the Document Tracker System follows the View Model Controller (VMC) software architecture. When the user is redirected to the landing page of the software, the user will able to see the functionalities of the system separated by widgets such as: import/export file controls, search and delete file controls, and the overview of the files table.

Behavior

When the user interacts with the buttons of the UI, the inner model then sends a request to the controller (database) according to its back end functionalities (i.e. import button activates the import PHP command to call the MySQL database file table to insert and store data). After successfully storing the files to the database table, the database will then send the updated table data to the inner functions of the model which is then displayed on the main landing page which the user can receive and view the updated table data, confirming that the files is successfully imported.