LeSanShoes Document and File System Guide

**File system directory:**

LeSanShoes

--------admin  
 ------//contains all of the temporary admin html files and files for admin only

--------adminSuper

-------//contains all the php files exclusive for admin (inventory, managed, UserMng)

--------assets

-------css

--------//contains all custom css for various pages

-------images

--------//contains all images used in website and uploaded images

-------js

--------// contains all customs js files for front end

-------swal

-------//local swal (independent of cdn)

-------DBExportFiles

-------//contains sql file

-------includes

------layouts

--------//should be used for SPA

------logic

--------//contains all middleware and php to DB

userAuth

-------//contains login, signup and logout php files

Users

-------//contains user front end html and php files.

**Filename format:**

Admin: [name]SuperAdmin.php or [name]Admin.php

User: [name].php (main page should be index.php)

Send to DB php: [name]ToDB.php

Middleware: [name]Middleware.js

**General file dependencies (sending data):**

* Front end php => middleware.js => ToDB.php

**General file dependencies (receiving data):**

* Load middleware.js with ajax function => query to get[name].php => dynamically set received data to front end php

**Document for files’ dependencies:**

**User Management:**

**Admin**

* Files involved:
  + UserMngSuperAdmin.php (main file)
  + CustomAdminPage.css (minor css changes)
  + newAdminMiddleware.js (sending and receiving data)
  + userAuthToDB(sends data from middleware to DB)
  + getUsers.php(fetches registered users and sends to middleware)
* Functions:
  + Add New Super Admin
  + Remove a Super Admin
  + Edit Admin Details
  + Add New Admin
  + Delete Admin
  + Displays admin accounts and user accounts
* Access: Super Admin Only.
* Process:
  + Each button has its own functions and ajax to send queries to DB.
  + Submission via onclick.
  + ID of each input fields are queried to get the data.
  + Data validation is custom. Built-in data validation is not used.
  + Json file is stringified before sending.
  + User and admin tables are inside loadTables(). Function is called on every update.
  + Two PHP Backend files in use: userAuthToDB and getUsers
  + userAuthToDB holds queries for user and admin authentication
* Notes:
  + Inefficient code due to repeated use of ajax and failure to have a modified json echo from the php server.
  + Use control F and search for the desired function to quickly navigate through the middleware.

**Users**

* Files involved:
  + Login.php
  + logoutAdmin.php
  + logoutUser.php
  + signup.php
  + signupMiddleware.js
  + loginMiddleware.js
* Functions and process:
  + User-end front-end to sign up and login their accounts. Redirects them to index.php
  + Two separate Middleware for handling signup and login queries.
  + Queries sent to userAuthToDB.php
  + Functions are used to send data, IDs are queried to get input values.

**Manage Database:**

* Files Involved:
  + ManageDBSuperAdmin.php
  + manageDBMiddleware.js
  + manageDBToDB.php
  + getManageDB.php
  + config.php
* Functions:
  + Modifying independent tables.
* Access: Super Admin only
* Process:
  + IDs are queried to get the value
  + Form submitted via onclick
  + Functions for submitting are in order in middleware
  + Shares a common ajax file
  + Json file is stringified before sending.
  + Uses single line echo for feedback from php server
  + Action is used to determine which line of codes will be executed in php server
  + Tables are enclosed inside loadTables function
  + loadTables are called on every update.
* Notes:
  + Very long code, make sure to use ctrl-f for faster navigation.

**Inventory:**

**Manage Shoes**

* Files Involved:
  + InventorySuperAdmin.php
  + inventoryMiddleware.js
  + inventoryToDB.php
  + inventoryGetOptions.php
* Functions:
  + Modifying Shoe model
  + Choosing existing data from manage DB tables
  + Inline edit and delete features.
  + Deleting a shoe model will also delete colorways associated with the deleted row.
* Access: Super Admin only
* Process:
  + Submission via form by querying form
  + Getting data via formData.get()
  + Using browse’s built-in form validation
  + Shares single ajax function
  + Json file is stringified.
  + PHP server uses switch case to determine the process.
  + Options are generated in select input by running a function from inventoryGetOptions.
  + inventoryGetOptions must be included at the start of the file.
  + Tables are received via inventoryGetOptions.
* Notes:
  + Manage Shoe, colorways, and stocks are all in the same middleware. It may be confusing to navigate due to different ways of sending data.

**Manage Shoes**

* Files Involved:
  + InventorySuperAdmin.php
  + inventoryMiddleware.js
  + colorwayToDB.php
  + inventoryGetOptions.php
* Functions:
  + Modifying colorways
  + Colorways are displayed in card format
  + Choosing existing data from manage DB tables
  + Inline edit and delete features.
  + Uploading photos.
  + Data are NOT sent via json
* Access: Super Admin only
* Process:
  + Submission via form and POST
  + Utilizes different ajax function from manage shoes
  + Using browse’s built-in form validation
  + formData is sent directly via POST
  + Different way of transferring data to a php variable
    - $\_POST[] instead of $obj -> name
  + PHP server uses switch case to determine the process.
  + Options are generated in select input by running a function from inventoryGetOptions.
  + inventoryGetOptions must be included at the start of the file.
  + Tables are received via inventoryGetOptions.
  + Middleware processes the creation of cards with dynamic data
* Notes:
  + Image data are not allowed to be sent via JSON. Sending data via POST is utilized hence the creation of a new php file.