**FMA Assignment**

The project is divided into 3 parts:

1. Frontend
2. Backend
3. Intranet files

**Frontend:**

It includes the files:

1. Index.php (Homepage):

The homepage contains a paragraph about the DCS department. The page is accessible to both logged in, and to logged out users.

1. Login.php (Login page):

The login page has a form which takes the input *username* and *password*. Both are mandatory to fill as *required* attribute of HTML is used in the input tag. Username input is of type *text*, and Password input is of type *password* (so that the input is censored by stars).

When someone attempts to log in, the request is redirected to the backend part of the code *functions.php*, which logs in user and redirects to homepage. If there’s an error, then it redirects back to the *login.php* page and sends an error using the query string ‘*?error=’nouser’*’ which is then captured by the *$\_GET* superglobal variable, and an error is displayed that ‘*No user found with this record*’.

The backend part of the code figures out if staff or admin is logged in, and logs them in (code written in *functions.php*)

*Pseudocode: $\_GET['error'] is used to figure out errors from query string*

1. Intranet.php (Intranet page):

If no user is logged in, then *login.php* file included instead of the intranet pages content.

If a user is logged in, then the page displays 3 links to display the 3 intranet files when clicked.

When the link is clicked, it redirects to the same page adding the query string:

1. "*?php=true*"
2. "*?dt=true*"
3. "*?pfp=true*"

I use *$\_GET* superglobal variable with if-else conditions to understand which link was clicked and display that file.

The logic of display file is handled by *displayBody()* function which passes the filename as a parameter, and is written in *includes/functions.php* file. Hence *includes/functions.php* file is included in the *intranet.php* file.

*Pseudocode: isset $\_GET['php'] to find if php page is clicked on via query string*

1. Signup.php (Admin’s page to register staff users):

I check the status of logged in user using the *$\_SESSION* superglobal variable.

If a staff is logged in, then the page displays the error “*Login as Administrator to access Admin area.”.* Otherwise the admin gets a form to sign up a new staff user.

1. Includes/header.php (Shared header file across all the frontend pages):

This file is shared across all the pages. It displays a navbar containing links to *index.php*, *intranet.php*, *signup.php*, and *logout.php* (If a person is logged in, which is checked by the *$\_SESSION* superglobal variable).

1. Img/logo.png: Logo image which is present in the header
2. Css/style.css: CSS to style the whole frontend

**Backend:**

It includes the files:

1. Includes/functions.php:

The functions included are:

1. Display intranet pages logic:

It includes the displayBody function, which is called by the intranet.php file to display the intranet pages.

The function accepts the file location as a parameter, and reads the file line by line using *feof* (till end of file is reached) and *fgets* (read line by line) methods, and prints it using *echo*.

1. Login logic:

When someone tries to login, this page is called. I first verify that the *data/users.txt* file is present and is readable using *fopen* method with ‘*r*’ access.

Then I read all the users data using file using *feof* method (read till the end of file) and *fgetcsv* (read each line as a csv) and store the data in the *formData* array.

Then I loop on the *formData* array to find if the login credentials match, if it matches, then I start the session and set the *$\_SESSION[‘username’]* to the username who just logged in, and redirect to the homepage.

If no user is found, then I redirect to the login page, and send an error ‘*nouser’* using query string.

1. Signup logic:

When the admin attempts to register a new staff user, this page is called. I first sanitize the inputs using *trim* and *htmlentities*. Then I check for possible errors, which if present, I redirect to the *register.php* file and send the error using query string.

Then I hash the password using the *password*\_hash method (to store hashes only in the *data/users.txt* file instead of the storing password in plain text).

Then I create a string of all the inputs, separated by commas, and use file handling to append the string into the *data/users.txt* file, completing the signup process.

1. Some reusable functions to validate alphanumeric words, alphabetical words, and save user data.
2. Logout.php: It unsets and destroys the session, so that no user info (who was logged in) remains, and redirects to the homepage.

*Pseudocode: session\_destroy() is used to destroy sessions after resetting session cookie to expired time (yesterdays)*

1. Data/users.txt: It contains data of each registered user, including the admin, on each line.

**Intranet files:**

It includes the following files:

1. DTresults.php
2. P1results.php
3. PfPresults.php

A session check is present on all the 3 files, so that only logged in members can access the intranet files.

*Pesudocode: if !isset $\_SESSION['username']) is used to check if someone is logged in*