PHP Documentation

Jack Laborde, Sagar Patel

1. Auth.php
   1. This file simply stores the local user’s session, taking in the header value from the login.php file.
   2. This allows the user to maintain the session throughout separate php pages.
   3. Session\_start() starts the session, and if the session isn’t set with the username, use the header data from login.php, then exit.
2. Dashboard.php, db.php, index.php are UNUSED
3. Index\_2 and Index\_2 Patient.php
   1. These two index files are the same file, but in the demo only the doctor’s form is used.
   2. First, the auth.php file is included to maintain the user session.
   3. This is followed up by the HTML code that actually generates the site.
      1. Head:
         1. The head file simply generates the Page Title, stylesheets, and icons
      2. Body:
         1. The body shows the EHR AWS management dashboard. The section “Congratulations” just gives text.
         2. The “instructions” class allows the doctor to view all their patients (which takes them to view.php below).
         3. They are also allowed to logout at the end of the list of instructions.
4. Insert.php
   1. This PHP file connects to the database and inserts monitor values. This is done by the mobile application, which establishes a connection to this page.
      1. First, a connection to the database is established, and then the GET function retrieves all the passed values.
      2. Once a connection is made, there is a call to insert the data into the Monitors table, and an OKAY echo sends out the confirmation.
5. Login.php
   1. This page allows the patients or doctor to login.
   2. First, the PHP checks if the post call is made. If it is, it requests the username and password from the user.
      1. A SQL call is then followed, selecting the matching user data from the users table.
      2. The Result is then stored and the number of rows are retrieved.
      3. If the rows are equal to 1, then establish a session with that username. If not, echo back a mistake.
   3. If the post call is not made, display the form that allows the user to enter into their information.
6. Logout.php
   1. This simply destroys all existing sessions and redirecting to the Login page.
7. Registration.php
   1. The registration file acts similarly to the Login.php code as descrbied above. It makes the same connection to the same database, making sure it exists, then takes in a REQUEST call by a username.
      1. The code then requests the patient/doctors name, date of birth, and other personal data.
      2. Once this is done, it checks between if the “Is doctor” checkbox is checked, then it enters the new information into the database in either the Patient or Doctor table.
   2. If a call is not made, the form is displayed to the user for registration.
8. Retrieve.php and Scheduled.php are UNUSED
9. V\_edit.php
   1. This PHP file allows a doctor to edit their patient data, and is called from the view.php file.
   2. The database connection is established along with an auth call.
   3. The functions that follow simply call up the correct patient information, then allows the record to be edited by the doctor using a provided HTML form.
10. V\_Insert.php
    1. This allows for a new record to be inserted. This is the same as the registration page.
11. View.php
    1. This view file simply displays all the patients into a table that are under a certain doctor.
    2. The database connection is the same as before
    3. Then it calls up every patient and lists their data.
12. ViewSpecific.php
    1. This PHP file acts similarly to View.php file, instead just viewing the Monitor data for a specific patient.