

Technical Design

Pages/Filenames:**1) index.php**

- This page will be coded entirely in PHP and will include a switch statement dictating the functionality of the application, as I will be using the model-view-controller pattern for this project. It will also include all the get/post variable declarations involved in database interaction. In the switch statement, the default will be set to include("homepage.php").

2) db.php

- This will be the page that handles database connectivity, including the credentials required to access the database.

3) homePage.php

- This will be the homepage of the website. All pages will contain a button labeled *About Us* in the header div that will direct the user to this page. This page will include information about the app and why users should be motivated to use it.

4) signIn.php

- This file will contain code pertaining to the user login page. If valid login credentials are entered by the user, the \$_SESSION['username'] variable will be set to an array containing that user's account information and the user will be redirected to their homepage, *loggedInHome.php*.

5) signUp.php

- This file will contain code for the user sign up page. The user will be prompted to enter their desired username, a password for their account, their first and last name, as well as their date of birth. If the username they've entered is unique, their account will be added to the database and they will be redirected to their homepage, *loggedInHome.php*.

6) loggedInHome.php

- This page will be the home page for users after they have successfully logged into, or signed up for, an account. In the center of the screen there will be a large button labeled *Create Flashcards* that, when clicked, will redirect the user to *createFlashcards.php*. If

the user wants to view flashcards they have already created, they can click the button that will be in the header div labeled *My Flashcards*, which will redirect the user to *flashcards.php*.

7) **flashcards.php**

- This file will contain a drop-down menu displaying the title names for each set of flashcards the user has created, if the user has created any. There will also be a button labeled *Create Flashcards* which will allow the user to add a new set of flashcards to their collection. Below the drop-down list, if there are any categories to select from, there will be a button labeled *View*. If the user selects an existing flashcard set and clicks *View*, they will be redirected to *flashcards.php*. If the user clicks the *Create Flashcards* button, the user will be redirected to *createFlashcards.php*.

8) **createFlashcards.php**

- This page will allow users to create a set of flashcards. There will be three textboxes included on this page. One textbox will be labeled *Group Title*, another will be labeled *Question*, and the final one will be labeled *Answer*. Below the *Answer* textbox will be a button labeled *Submit*. When clicked, that button will first store the title of the group into the *Categories* table, then store the question and answer input into the *Flashcards* table using the same category id as that of its corresponding group title. The user will then be redirected to *flashcardSet.php* where they will be given the option to edit the group or add another card to group.

9) **flashcardSet.php**

- This page will display the flashcards stored within the flashcard title selected by the user from *loggedInHome.php*. Each flashcard will display a question provided by the user with a button below it labeled *Answer* that, when clicked, will display the answer assigned to that particular flashcard. There will also be buttons labeled *Edit Group* and *Add New Card to Group*. As implied, these buttons will allow the user to either edit the set of flashcards or add a new flashcard to the set.

10) **editCards.php**

- This page will display a table containing all information for all cards stored within the group the user chose to edit. There will be a column with a link reading *Delete Card* appearing in each row of the table that, when clicked, will delete the card displayed in that same row. There will also be a column with links labeled *Edit Card* in each row that,

when clicked will redirect the user to *createFlashcards.php*. However, instead of displaying a page with blank textboxes, the textboxes will contain the current information associated with the selected card. The user will be able to change the content within these textboxes. There will be a button labeled *Update* below the *Answer* textbox that, when clicked will update the information for the card and bring them back to *editCards.php*. At the bottom of this page there will be a button labeled *Delete Group*. If clicked, this button will delete the entire group of cards from the *Flashcards* table as well as that group name's information from the *Categories* table.

11) functions.php

- This file will contain all the functions used by the application. I will put the functions in commented sections for organizational purposes. For example, I will have a line of commented code reading “//CREDENTIALS AND USER ACCOUNTS -----” above all functions involving searching, adding, or editing user account information, with a line of commented out dashes below the final function in the section to indicate the end of that section. I will also write a comment above each function explaining its purpose.

Functions:

The names of the functions will be indicative of the function's action. For example, *addFlashcardCategory* will be the name of the function that, when called, will add a new category to the *Categories* table. Some other functions include:

- *AddUserAccount()* – function that adds a new user account to the *Users* table
- *UpdateUserAccount()* – function that edits/updates existing user account information
- *CheckCredentials()* – function that searches *Users* table for the username and password provided by the user at login
- *CheckForUniqueUname()* – function that selects all usernames from the *Users* table to determine whether a desired username is one that does not already exist