Hemo-cardiac Application

© 2023 University of Central Florida. All rights reserved.

**Script: add-student.php**

* Establishes the connection to the database
* Using POST method saves Students' Data (First Name, SID, ClassSection)
* Creates a new record in the Students table. If the query was successful, echo “New record created successfully”, or “Error” otherwise.

**Script: addCaseAttempt.php**

* Establishes the connection with the database
* Using POST method saves Case attempt data (SID, CID, Grade, TimeSpent, Answer, Login)
* Creates a new record in the cardaic\_case\_attempts. If the query was successful, echo “New record created successfully”, or “Error” otherwise.

**Script: addFlashcardAttempt.php**

* Establishes the connection with the database
* Using POST method saves Case attempt data (SID, FID, Grade, TimeSpent, Confidence, Login)
* Creates a new record in the cardaic\_case\_attempts. If the query was successful, echo “New record created successfully”, or “Error” otherwise.

**Script: Cases.php**

* Establishes the connection with the database
* Using REQUEST method saves the Section number
* Returns a .json with Cases information for specified Section number. Information includes CID, Description, Rhythm, AnswerDescription, A, B, C

**Script: config.php**

* Sets the configuration for the database from the env. variables of the app

**Script: Flashcards.php**

* Establishes the connection with the database
* Using REQUEST method saves the Section number
* Returns a JSON with Cases information for specified Section number. Information includes FID, Promt, Answer

**Script: Hemo-create-answer.php**

* Establishes the connection with the database
* Using POST method saves Case attempt data (puzid, sid, answer)
* Creates a new record in the drhemo\_answers table. If the query was successful, echo “New record created successfully”, or “Error” otherwise.

**Script: Hemo-create-attempt.php**

* Establishes the connection with the database
* Using POST method saves Escape room attempt data (TeamName, TimeSpent, Completed, SID 1-5)
* Creates a new record in the drhemo\_answers table. If the query was successful, creates and returns a new code aid, or “Error” otherwise.

**Script: Hemo-create-puzzleStep.php**

* Establishes the connection with the database
* Using POST method saves Escape room attempt data (Game ID, PuzzleName, PuzzleStep, Hints taken, Time taken)
* Creates a new record in the drhemo\_puzzlesteps table. If the query was successful, returns a puzzleID, or “Error” otherwise.

**Script: Hemo-create-puzzleStep2.php**

* Establishes the connection with the database
* If the PuzzleStep doesn't exist
  + Using POST method saves Escape room attempt data (Game ID, PuzzleName, PuzzleStep, Hints taken, Time taken)
  + Creates a new record in the drhemo\_puzzlesteps table. If the query was successful, returns a puzzleID, or “Error” otherwise.
* Else, returns a PuzzleStep

**Script: Hemo-get-top-scores.php**

* Establishes the connection with the database
* Returns a JSON object with top 3 fastest competed attempts

**Script: Hemo-submit-attempt.php**

* Establishes the connection with the database
* Using POST method gathers Escape room attempt data (Game ID, Completed, TimeSpent)
* Creates a new record in the drhemo\_attempts table. If the query was successful, returns “Run submitted”, or “Error” otherwise.

**Script: Hemo-update-attempt.php**

* Establishes the connection with the database
* Using POST method gathers Escape room attempt data (Game ID, Completed, TimeSpent, SID, JoinNum)
* Updates a record in the drhemo\_attempts table. If the query was successful, returns “Attempt updated”, or “Error” otherwise.

**Script: Hemo-update-puzzleStep.php**

* Establishes the connection with the database
* Using POST method gathers Escape room attempt data (PUZSTEPid, HintsTaken, TimeTaken)
* Updates a record in the drhemo\_puzzlesteps table. If the query was successful, returns “Step updated”, or “Error” otherwise.

**Script: index.php**

* Default screen for the PHP app. Echos the Welcome text

**Script: login.php**

* Establishes the connection with the database
* Using REQUEST method saves the SID and Password
* If no student with SID prompted is found:
  + Returns “No student found in the table”
* If the password matches the password from the database
  + Updates the Loggen in counter in the database
  + Returns a JSON object with SID, Section number, Login Counter
* Else, returns “invalid password”

**Script: read-students.php**

* Establishes the connection with the database
* Returns JSON object with all student data (SID, ClassSection, LoggedIn)

**Script: students.php**

* Establishes the connection with the database
* Shows student data (SID, ClassSection, Name)