**Folder: Admin**

|  |  |  |
| --- | --- | --- |
| ***Class*** | ***Requires*** | ***Function*** |
| **Questiondetails.php** | Dbutils.php | Get details of questions from the database and display them. This includes displaying possible answers, both correct and incorrect |
| **Questionlist.php** | Dbultils.php | Displays list of questions. It calls questiondetails.php to display data |
| **Remove\_image.php** | Dbutils.php | Deletes an image from the questions\_images database |

**Folder: Classes**

|  |  |  |
| --- | --- | --- |
| ***Class*** | ***Requires…*** | ***Function*** |
| **Dbutils.php** | None | Class for setting up databases   |  |  | | --- | --- | | *\_\_construct* | Takes in database username, password, and name arguments. Class constructor to create a new database connection object | | *closeConnection* | Close the mysql connection | | *escapeMySqlString* | Takes special characters arguments. Escapes the special characters used in the MySQL statement | | *getDatasetWithParams* | Takes sql select query, data types, and parameters arguments. Creates a dataset based on an SQL Select query and a set of parameters for the WHERE clause. This method only works with SELECT queries | | *getDataset* | Takes a SQL SELECT query argument. Creates a dataset on an SQL SLECT query where the WHERE clause is concatenated | | *executeQuery* | Takes SQL SELECT query, types, and parameters arguments. Executes an INSERT, UPDATE, DELETE query with a set of parameters for the WHERE clause | | *getRecordsetAsJson* | Takes SQL Select query, types, and parameters arguments. Creates a JSON object form data returned by an SQL Select query and a set of parameters for the WHERE clause. This only works with SELECT queries | |
| **HtmlUtilities** | None | Cannot find use in current code base. |
| **Image.php** | None | Class for image object   |  |  | | --- | --- | | *\_\_construct* | Constructs image object | |
| **Levelsecurity.php** | getaccess.php | Checks to make sure the user isn’t jumping levels |
| **Question.php** | Image.php | Class for question object   |  |  | | --- | --- | | *\_\_construct* | Constructor function for question object. Splits depending whether or not user is in practice mode. | | *logUsedQuestion* | Inserts a question that has been used into used\_question\_log database | | *loadDistractors* | Load distractor images for the questions | | *toJSON* | Converts image List to Json | |  |  | |
| **Quiz.php (actually for Quiz Question)** | None | Class for Quiz Question object   |  |  | | --- | --- | | *\_\_construct* | Takes database username, password, and name arguments. Constructor to create a new database connection object | | *closeConnection()* | Close SQL database connection | | *escapeMYSqlString* | Takes special character arguments. Creates an escape string for the sql database | | *getDatasetWithParams* | Takes SQL Select query, types, and parameters arguments. Creates a dataset based on a select query and a set of parameters for the WHERE clause. This method only works with SELECT queries | | *getDataset* | Take SQL SELECT query parameters. Creates a database on an SQL SELECT query where the WHERE clause is concatenated | | *executeQuery* | Takes SQL SELECT query, types, and parameters arguments. Executes an INSERT< UPDATE< DELETE query with a set of parameters for the WHERE clause | | *getRecordsetAsJson* | Takes SQL SELECT query, types, and parameters arguments. Creates a JSON object form data returned by an SQL SELECT query and a set of parameters for the WHERE clause. This method only works with SELECT queries | |
| **Quizquestion.php** | None | Pretty sure this was abandoned |
| **security.php** | None | Checks if user id in the session is blank or “invalid”. Redirects to authfailed.php |
| **stringutils.php** | None | Replaces certain characters in strings   |  |  | | --- | --- | | *cleanMySqlInsert* | Takes sql String argument, replaces characters that present potential danger for SQL Injections or can break SQL syntax | |

**Folder: Rest**

|  |  |  |
| --- | --- | --- |
| ***Class*** | ***Requires…*** | ***Function*** |
| **getaccess.php** | Dbutils.php | Checks if user ID is set in the session. Set accordingly. Files an SQL SELECT query for everything from page access database where the ID matches the session ID. |
| **getDistractors.php** | Dbutils.php | Gets distractor images for questions.  Selects question ID, diagnosis name, hint, image ID, image folder, and image name  from the distractor and questions databases  where IDs match. Randomizes them, encodes in json, and displays the data. |
| **getPracticeQuestions.php** | Dbutils.php | Retrieves practices questions.  Selects question ID, image ID, category name, diagnosis name, hint, number of images, image folder, and image name.  from categories and questions databases  where IDs match. Loops number of times as the “limit” in the session |
| **getscore.php** | Dbutils.php | Sets the IDs from the session data.  IF the user ID is “TOPFIVE”, retrieves the top five score, encodes them into JSON and displays them.  Otherwise, select the question ID, image ID, if there is a bonus, level ID, attempt ID, score received, date and time started, and date and time ended.  Gets data from score database where IDs match.  Then selects question ID, score date, score time, level ID, and attempt ID.  Gets data from questioncomplete database where IDs match.  Displays score and question data. |
| **getuserlist.php** | Dbutils.php | Gets ID from session.  Select name  from drupal\_users  where ID matches session ID.  Sets database name, username, and password. Creates a new DbUtilities object from previous data, turns it into an array and encodes to Json |
| **Leaderboard\_ws.php** | Dbutils.php | Retrieve the Leader board scores  Selects user ID, level ID, and max level score  From the dentalgame score log  Where level is complete  Group by user and level id  Order by max level score  Limit to 3 queries  Encode to JSON  Select user ID, level ID, and sum of Max Overall score  From dental game score log  Where the level is complete  Group by ID  Order by the max overall score  Limit to 5 results  Encode to Json |
| **levelsummary.php** | Dbutils.php | Displays the level summary, including score and answers  Get the game attempt, level, and user IDs from the session data  Select Max level attempt number  From the score log  Where the user and attempt ID matches  Select the level attempt number, if the level is complete, final level score, number of questions answered, number of questions correct, and number of questions incorrect  From Scorelog  Where attempt, user, and level IDs match  Save as score and encode in JSON  Select question ID, image ID, diagnosis name, image Folder, image Name, and whether it is correct  From answer\_log, questions\_images, questions, and images databases  Where image, question, attempt, user, and level IDs match  Save as answers  Display scores and answers |
| **Logquizanswer.php** | Dbutils.php | Set user ID, question ID, choice ID, and if item is correct from session data  Insert data above into quizanswerlog database |
| **Question\_ws.php** | Dbutils.php | Gets question and distractor info from databases and display them  If the level ID is set from the session  Select question ID, diagnosis name, hint, image ID, image folder, and image name  From questions and questions\_levels databases  Where IDs match  Randomize  Encode to JSon and save as questions  Select question ID, diagnosis name, forQuestion ID, image ID, image folder, and image name  From distractors and questions databases  Where IDs match  Randomize  Encode to Json and save as distractors  Display questions and distractors   |  |  | | --- | --- | | *Utf8ize* | IF argument is an array, for each item, use recursion to call itself for each string  If string, encode it into utf8 | |
| **Question\_ws\_obj.php** | Dbutils.php  Question.php | Create new question object and encode in JSON. Question object depends on user’s current level  Get game attempt and level IDs from session  Retrieve practice categories if any  Display questions differently depending on level |
| **Quizquestion\_ws.php** | Dbutils.php | Creates a QuizQuestion and QuizQuestionChoice object  Selects question ID question stem, choice Id, choice text, and if item is correct  From quizquestions and quizquestionchoices databases where IDs match and items are active  Order by question ID  Save as collection list  Iterate through collection list and put questions in collection list into new array list called question list  Also construct new questions based on data from sql query |
| **Quizscorecard\_ws.php** | Dbutils.php | Displays user’s scorecard  Select date answered, if correct, and count  From quizanswerlog  Where IDs match  Group by Date answered and if correct  Save as collectionlist, encode in json and display |
| **Resultstojson.php** | Dbutils.php | Convert entire sql database to JSON? (Actually not sure if this works) |
| **Scorecard\_ws.php** | Dbutils.php | Display user’s scorecard (not sure how this differentiates from quizscorecard\_ws.php)  Select level ID, log time and date, level attempt, final level score, number of questions answered, number correct, and number incorrect  From the scorelog  Where user IDs match and the level is complete  Order by log date time descending and level ID ascending  Convert to json and display |
| **Session.php** | None | Set name and value session data and display  (not sure if this is used) |
| **Updateaccess.php** | None | Insert new access data in page access database  Depends if session data is already set |
| **Updateanswerlog.php** | None | Update the answer log database  Retrieve session data and display  Then insert into answer log database with SQL |
| **Updatescore.php** | Dbutils.php | Update the score database  Retrieve session data and display  Then insert into score database with SQL |
| **Updatescorelog.php** | Dbutils.php | Update the scorelog database  Retrieve session data  Insert data into score log database with SQL |

**Folder: Main**

|  |  |  |
| --- | --- | --- |
| ***Class*** | ***Requires*** | ***Functions*** |
| **Auth.php** | Dbutils.php | Sets default database name, username, and password. Authenticate user  Retrieve list of users  from drupal\_users  Where ID matches  Drupal\_users has one entry already, set it to session  Otherwise, authentication failed |
| **Authfailed.php** | Header.php  Footer.php | Page for failed authentication |
| **Caught\_cheating.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Page to tell user they were cheating |
| **Completedlevelsummary.php** | Security.php  Header.php  levelSummary.php | Display completed level summary  Once document has been loaded into browser  Set all session info  Iterate through all answers and say whether it was answered correctly or not  If level is completed, redirect to next level  Otherwise, offer to repeat level   |  |  | | --- | --- | | *setupTimerBar* | Set up timer for the level | | *updateTimerBar* | Change timer appearance upon completion of level (unsure if actually used) | |
| **Footer.php** | None | Footer for page |
| **Gameattempt.php** | Security.php  Dbutils.php  Updateaccess.php | If level ID is already set, redirect to incorrect\_level.php  Otherwise, send user to level 1 tutorial |
| **Game-level-example.php** | Dbutils.php  Question.php | Show an example of the game level  When document is loaded, set button to each image and say whether each answer is right or wrong   |  |  | | --- | --- | | *updateScoreData* | Takes user ID, selected image id, and if image is correct arguments  Set score data variables from arguments, JSON and send score to updatescore.php | |
| **Header.php** | None | Header to be displayed on top of most pages |
| **Homepage.php** | Header.php | Display homepage for game. Includes links to each mode |
| **Leaderboard.php** | Header.php  Footer.php  Security.php  Dbutils.php  Question.php | Display the leaderboard for the game |
| **Level1.php** | Security.php  Header.php  Footer.php  Question\_ws\_obj.php | Page for level 1 of game   |  |  | | --- | --- | | *initQuestion* | Get question from question\_ws\_obj.php and initialize | | *selectDxImage* | Take button object argument. Check if user’s answer is correct or not.  If user has reached 5 correct, level is passed  If user missed 2, game over | |
| **Level1a.php** | Security.php  Dbutils.php  Levelsecurity.php  Question.php  Header.php  Footer.php | Determines if the selected image is correct or not. If it is correct, send info to update score as true and bring up new question. IF false, send info to updatescore data and upload new question   |  |  | | --- | --- | | *getScore* | Take amount argument. Calculate score | | *hintPointDeduction* | Formula for deducting points for hints | |
| **Level1tutorial.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Display the tutorial for level 1 |
| **Level2.php** | Header.php  Footer.php | Page for level 2 of the game   |  |  | | --- | --- | | *initQuestion* | Initialize question | | *selectDxImage* | Determine if user’s answer was right | | *SubmitLevel2answers* | Submit answers for level 2 | | *processAnswersChoices* | Increase score if answer was right. If user wins or loses the level, send them to the correct page | | *getNumberAvailableCorrect* | Function to get number available correct | |
| **Level2a.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Determines if the selected image is correct or not. If it is correct, send info to update score as true and bring up new question. IF false, send info to updatescore data and upload new question   |  |  | | --- | --- | | *getScore* | Takes amount variable, calculates score | |
| **Level2tutorial** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Display tutorial for level 2 |
| **Level3.php** | Security.php  Header.php  Footer.php | Page for level 3 of the game   |  |  | | --- | --- | | *initQuestion* | Initialize question | | *loadLevel3Question* | Take data argument, load level 3 questions | | *getImageDataFromSelection* | Iterate through images and get data | | *randomizeSelectableTerms* | Randomize choices for question | | *submitLevel3Answers* | Submit answers for level 3, redirect depending on amount right or wrong | |
| **Level3a.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Determines if the selected image is correct or not. If it is correct, send info to update score as true and bring up new question. IF false, send info to updatescore data and upload new question   |  |  | | --- | --- | | *saveSelectedPairs* | Save image matches to be scored | | *selectDiagnosisMatch* | Function to store the diagnosis selection | | *selectImageMatch* | Function to store the image selection | | *calculateScore* | Calculate score | | *getScore* | Function to calculate score | |
| **Level3tutorial.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Tutorial for level 3 |
| **Logout.php** | None | Reset session data |
| **Practice.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Load up practice page for game |
| **Practicemenu.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Display the practice menu |
| **practiceQuestion.php** | Security.php  Dbutils.php  Levelsecurity.php  Question.php  Header.php  Footer.php | Display a practice question |
| **Quiz.php** | Security.php  Header.php  Footer.php  logquizanswer | Page for quiz   |  |  | | --- | --- | | *loadQuestion()* | Load quiz question | | *submitQuestion* | Submit answer | | *getCorrectQuestionChoice* | Function to get correct question choice | |
| **Quizscorecard.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php  Quizscorecard.php | Page to display quiz score card |
| **Scorecard.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Page for the game score card |
| **Tutorial.php** | Security.php  Dbutils.php  Question.php  Header.php  Footer.php | Page to display quiz tutorial |
| **Win.php** | Security.php  Dbutils.php  Levelsecurity.php  Question.php  Header.php  Footer.php | Win page for game |

Database Accesses

|  |  |
| --- | --- |
| ***Class*** | ***Access Which Databases*** |
| **Admin Folder** |  |
| *Questiondetails.php* | Questions  Categories  Questions\_images  Distractors |
| *Questionlist.php* | Questions |
| *Remove\_image.php* | Questions\_images |
| *Question.php* | Questions  Questions\_levels  Questions\_images  Images  Used\_question\_log  Distractors |
| **Rest Folder** |  |
| *Getaccess.php* | Pageaccess |
| *getDistractors* | Distractors  Questions |
| *getPracticeQuestions* | Categories  Questions |
| *Getscore.php* | Score  Questioncomplete |
| *Getuserlist.php* | Drupal101.drupal\_users |
| *Leaderboard\_ws.php* | Dentalgame.scorelog |
| *Levelsummary.php* | Scorelog  Answerlog |
| *Logquizanswer.php* | Quizanswerlog |
| *Question\_ws.php* | Questions  Questions\_images  Images  Distractors  Questions\_levels |
| *Quizquestion\_ws.php* | Quizquestions  Quizquestionchoices |
| *Quizscorecard\_ws.php* | Quizanswerlog |
| *Scorecard\_ws.php* | Scorelog |
| *Updateaccess.php* | Pageaccess |
| *Updateanswerlog.php* | Answerlog |
| *Updatescore.php* | Score |
| *Updatescorelog.php* | Scorelog |
| **Main Folder** |  |
| *Auth.php* | Drupal101.drupal\_users |
|  |  |

Databases (Primary Key is Bolded) – Double Check with db\_07\_30\_2017\_2.sql

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Key | Constraint | References |
| **answerlog** |  |  |  |
| **ScoreID** |  |  |  |
| userID |  |  |  |
| Fk\_questionID | Fk\_answerlog\_questions\_idx | Fk\_answerlog\_questions | questions  (question ID) |
| Fk\_imageID | Fk\_answerlog\_images\_idx | Fk\_answerlog\_images | Images  (imageID) |
| isCorrect |  |  |  |
| attemptID |  |  |  |
| levelID |  |  |  |
| levelAttemptNumber |  |  |  |
| dateTimeSelected |  |  |  |
| **Categories** |  |  |  |
| **categoryID** |  |  |  |
| categoryName |  |  |  |
| **Distractors** |  |  |  |
| **distractorID** |  |  |  |
| Fk\_forQuestionID | Fk\_distractors\_questions\_idx | Fk\_distractors\_questions\_forQuestionID | Questions  (questionID) |
| Fk\_distructorQuestionID | Fk\_distractor\_questions\_distractorQuestionID\_idx | Fk\_distractor\_questions\_distractorQuestionID | Questions  (questionID) |
| **Images** |  |  |  |
| **imageID** |  |  |  |
| imageFolder |  |  |  |
| imageName |  |  |  |
| **Pageaccess** |  |  |  |
| **pageaccessID** |  |  |  |
| userID |  |  |  |
| AttemptID |  |  |  |
| levelAccess |  |  |  |
| accessDateTime |  |  |  |
| **Questions** |  |  |  |
| **questionID** |  |  |  |
| diagnosisName | Idx\_diagnosisName |  |  |
| Fk\_categoryID | Fk\_questions\_categories\_idx | Fk\_questions\_categories | Categories  (categoryID) |
| Hint |  |  |  |
| numberofImages |  |  |  |
| **Questions\_images** |  |  |  |
| **Fk\_questionID** |  | Fk\_questions\_images\_questions | Questions  (questionID) |
| **Fk\_imageID** | Fk\_questions\_images\_images\_idx | Fk\_questions\_images\_images | Images  (imageID) |
| **Questions\_levels** |  |  |  |
| **Fk\_questionID** |  | Fk\_questions\_levels\_questions | Questions  (questionID) |
| **levelID** |  |  |  |
| **Quizanswerlog** |  |  |  |
| **logID** |  |  |  |
| userID |  |  |  |
| Fk\_quizQuestionID | Fk\_quizanswerlog\_questions\_questionID\_idx | Fk\_quizanswerlog\_questions\_questionID | Quizquestions  (questionID) |
| Fk\_quizChoiceID | Fk\_quizanswerlog\_questionchoices\_questionID\_idx | Fk\_quizanswerlog\_questionchoices\_questionID | Quizquestionchoices  (choiceID) |
| isCorrect |  |  |  |
| dateTimeAnswered |  |  |  |
| **Quizquestionchoices** |  |  |  |
| **choiceID** |  |  |  |
| Fk\_questionID |  |  |  |
| choiceText |  |  |  |
| isCorrect |  |  |  |
| **Quizquestions** |  |  |  |
| **questionID** |  |  |  |
| questionStem |  |  |  |
| isActive |  |  |  |
| **Scorelog** |  |  |  |
| **autoID** |  |  |  |
| attemptID |  |  |  |
| levelID |  |  |  |
| Fk\_userID |  |  |  |
| logDateTime |  |  |  |
| levelAttemptNumber |  |  |  |
| isLevelComplete |  |  |  |
| finalLevelScore |  |  |  |
| numberQuestionsAnswered |  |  |  |
| numberCorrect |  |  |  |
| numberIncorrect |  |  |  |
| **Subclassifications** |  |  |  |
| **Fk\_questionID** |  |  |  |
| **Fk\_categoryID** |  |  |  |
| **Used\_question\_log** |  |  |  |
| **attemptID** |  |  |  |
| **questionID** | Fk\_used\_question\_log | Fk\_used\_question\_log\_questions | Questions  (questionID) |
| **levelID** |  |  |  |
|  |  |  |  |