Project Title : Documentation for JOBIE

Project Start Date : September 5th, 2018

Project End Date : December 2018

Budget Information : There is no specific budget for this project as it was developed for non-profit

organization.

Project Manager : Maulik Patel ([mpatel59@una.edu](mailto:mpatel59@una.edu))

Project Objective : To develop a documentation for JOBIE website that can help the future updater to find all the information regarding the website as well its flow in systematic order.

Project Success Criteria: After the completion of documentation the updater must have all the information they are looking for as well as arranged and organized.

Approach:

* Getting down all the information regarding JOBIE (why, what, when) it was created.
* Meeting al the team members who are directly or in directly associated with the project and asking them about their requirements and the problems faced.
* Developing a business case and listing all the details of JOBIE.
* Sorting and analyzing the duplicate files(.php) and the unused or unwanted tables in database.
* Noting down all the requirements of code and the database relations associated in the project.

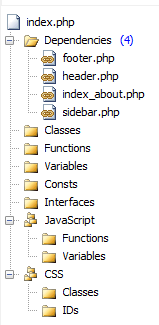
Roles and Responsibility

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Responsibility** | **Contact Information** |
| Dr. K.D. Malone | Project Sponsor | Editor | kdmalone@una.edu |
| Mr. M. Moon | Project Advisor | Checking code | mrmoon@una.edu |
| Dr. B. Cumbie | Program Manager | Managing the project | [bcumbie@una.edu](mailto:bcumbie@una.edu) |

Overview of the website

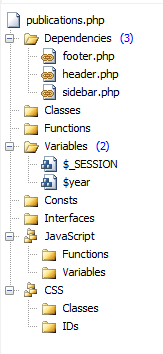
* The website has total of 32 pages.
* Website uses MYSQL database.
* JavaScript used:
  + Jobie.js
  + phpChecks.js
* Cascading Style Sheet used:
  + Mystyle.css
  + Themejobie.css
* There are mainly 25 different types of dependencies used in the website.
* [Dependency matrix](Dependency%20Matrix.xlsx).

1. Index.php



* Important Notes:
* session\_start();
  + auto session sign out funtion built in footer.php actived in login.php reset times by POST method
  + display publications by years function built in sidebar.php and publications.php
* require("header.php");
  + will control the auto log out at footer.php
  + This file is the home page of the website with links to different php files
  + structure: header.php sidebar.php/sidebarmenu.php content.php footer.php
* require("sidebar.php");
  + require sidebarmenu.php after logged in
* require("index\_about.php");
  + content for index.php
* require("footer.php");
  + will control the auto log out at footer.php

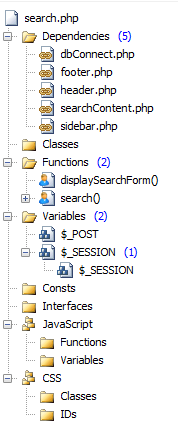
1. publications.php



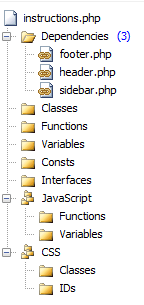
* Important Notes:
* session\_start();
  + display online publications on homepage.
* require("header.php");
* require("sidebar.php");
* switch ($year)
  + $year define in sidebar.php
  + display links by years.
* require("footer.php");

1. search.php

* Important Notes:
* $\_SESSION
  + set 20 mins before session expired
* $searchItem
  + Takes input from searchContent.php
* require("dbConnect.php")
  + connect to database
* $searchItem = mysqli\_real\_escape\_string($dbc, trim($searchItem))
  + fix the SQL Injection
* ($searchItem != "")
  + Checks title, author, and keywords for variable
* $dbc->close()
  + Close the mysql connection

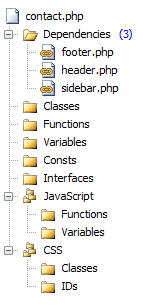


1. instructions.php



* Important Notes:
* This file contains the text for the Call for Papers page.
* Contains a link to submissionCreate.php for article submission.

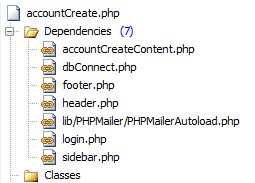
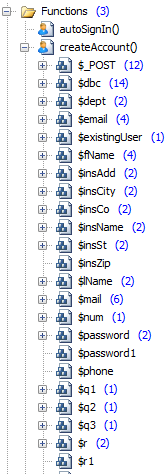
1. contact.php

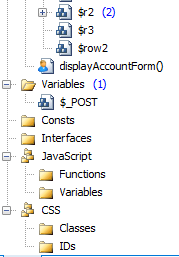


* Important Notes:
* This page holds the contact information for the JOBIE staff in the event a user needs to contact them.

1. accountCreate.php

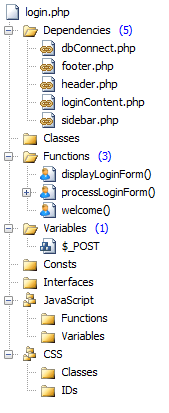
* Important Notes:
* require 'lib/PHPMailer/PHPMailerAutoload.php';
  + require\_once 'lib/swift\_required.php';
* isset($\_POST['\_submit\_check'])
  + Generates the form
* function createAccount()
  + function to create account and insert information into database
  + username stored in database will be an email address
  + info retrieved from accountContent.php
* Use mysqli\_real\_escape\_string() to to prevent SQL injection
* make sure that the email doesn't exist in database.
* insertUser query
* run the query
* addRole
* after successful account created, requires the user to login
* send email from FROM\_EMAIL to user email
* Create the mail transport configuration
* Create the message
* Send the email

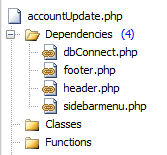


1. login.php

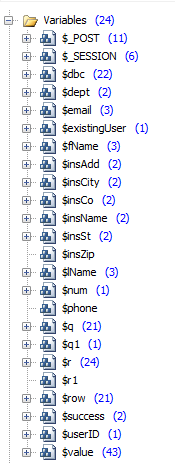
* Important Notes:
* This is the file containing the functions for a user to login
* isset($\_POST['\_submit\_check'])
  + Checks if the user has posted on the loginContent.php file
* function processLoginForm()
  + function to process the input of the user to make sure their ID and password match those stored in the database.
* $email = $\_POST['email']; || $password = $\_POST['password'];
  + Input from loginContent.php
* require("dbConnect.php")
  + connect to the database
* $q = "SELECT userID,fName,lName FROM users WHERE email='$e' AND password=SHA1('$p')"
  + Retrieve the user\_id and first\_name for that email/password combination
* function welcome()
  + take user to welcome page upon successful login
  + require("welcome.php")
  + otherwise, continue to display the form until user inputs valid information

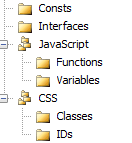


1. accountUpdate.php



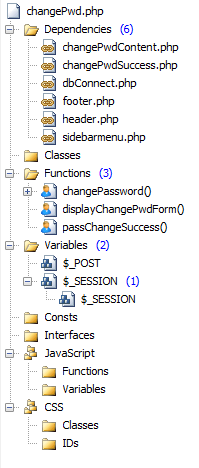
* Important Notes:
* $success = false;
  + check if \_submit\_check is successful
* if (isset($\_POST['\_submit\_check']) && isset($\_SESSION['userID']))
  + This file contains the code for a user to update an account.
* reset 20 mins before session expired
* $userID = $\_SESSION['userID']
  + update account contact info
* $email = mysqli\_real\_escape\_string($dbc, trim($email));
  + Use mysqli\_real\_escape\_string() to to prevent SQL injection
* make sure that the user does not use another email already have an account
* insertUser make the query
* run the query
* Verifiy that the user is logged in before submission.



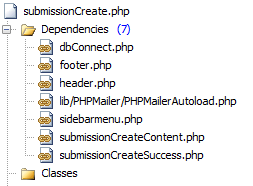


1. changePwd.php

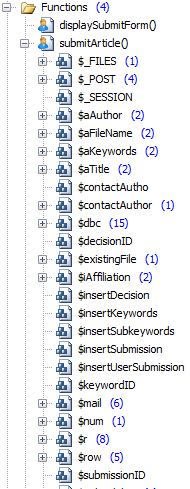
* Important Notes:
* session\_start();
  + This file contains functions to change the user's password in the database
* reset 20 mins before session expired
* require("dbConnect.php")
  + connect to database using dbConect function from dbConnect.php file
* check to see if username already exists in the database
* makes sure the email and password match what is in the database before update

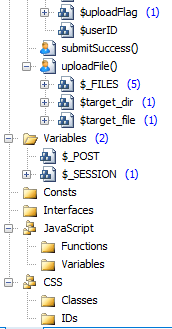


1. submissionCreate.php



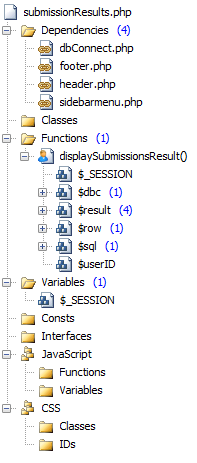
* Important Notes:
* require("header.php");
  + This file contains the code for a user to submit an article.
* require 'lib/PHPMailer/PHPMailerAutoload.php'
  + require\_once 'lib/swift\_required.php'
  + lib swift throwing error.
* Verifies that the user is logged in before submission.
* Verifies that the user is logged in before submission.
* function submitArticle()
  + function to submit article and insert information and file into database.
* $userID = $\_SESSION['userID']
  + session variables for link between tables in database.
* variables from submissionContent.php
* get last submisssionID.
* insert into usersubmission.
* insert into submissionkeyword.
* Create the mail transport configuration.
* Send the email.



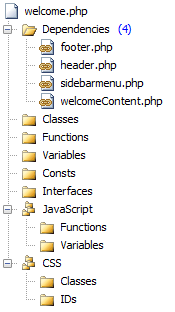


1. submissionResults.php

* Important Notes:
* Generate the form.
* Pull the information from the database and displays it in a table.
* $result = $dbc->query($sql)
  + Check if it was successful.
* Make sure there are some files in there.
* Print the top of a table.
* Print each file.
* Close table.
* Free the result.
* Close the mysql connection.

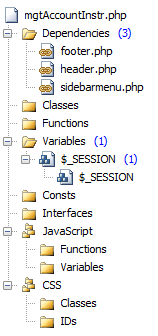


1. welcome.php



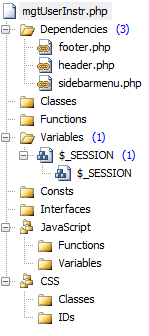
* Important Notes:
* This file contains links to files after login.

1. mgtAccountInstr.php



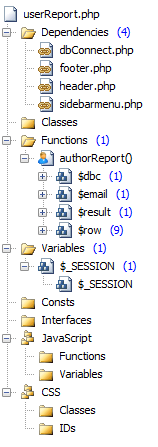
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");

1. mgtUserInstr.php



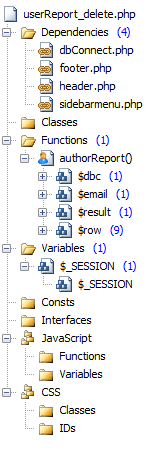
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");

1. userReport.php



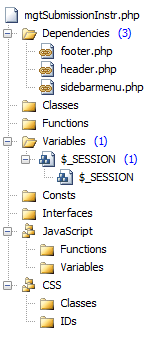
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* require User Privileges query.

1. userReport\_delete.php



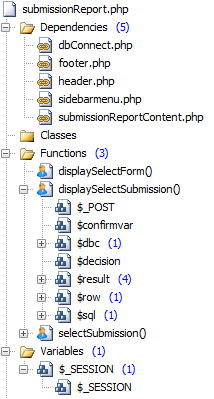
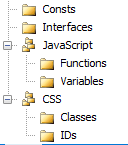
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* require User delete/drop query.

1. mgtSubmissionInstr.php



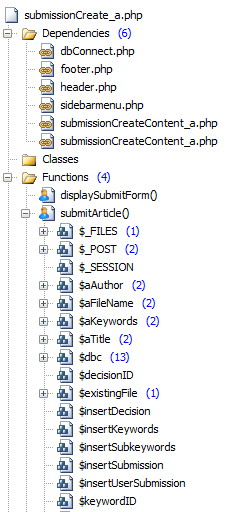
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");

1. submissionReport.php

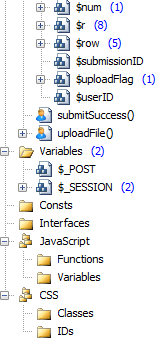


* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* Reset 20 mins before session expired.
* Fetch the information from the database and displays it in a table.
* Check if database connection was successful.
* Make sure there are some files in there and then print the top of a table.
* Print each file.
* Close the mysql connection.

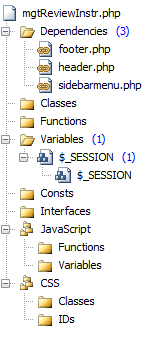
1. submissionCreate\_a.php



* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* This file contains the code for an admin to submit an article without type limits.
* Verify that the user is logged in before submission.
* Reset 20 mins before session expired.
* function submitArticle()
  + function to submit article and insert information and file into database.
* $userID = $\_SESSION['userID'];
  + session variables for link between tables in database
* Variables from submissionContent.php
* Get last submisssionID
* Insert into usersubmission
* Insert into submissionkeyword
* Make sure the file was sent without errors.

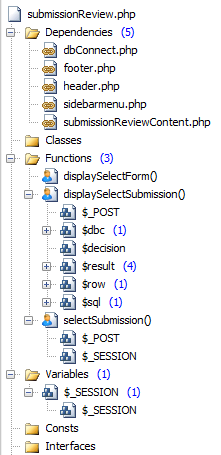


1. mgtReviewInstr.php

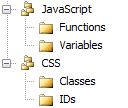


* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* Check if the user is already signed in before providing access.

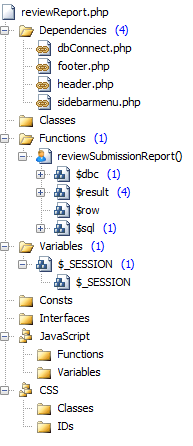
1. submissionReview.php



* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* Reset 20 mins before session expired.
* Pull the information from the database and displays it in a table.
* Check if it was successful.
* Make sure there are some files in there and then display the top of a table.
* Print each file.

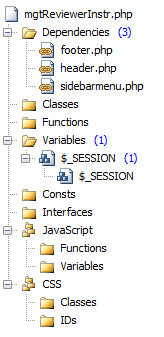


1. reviewReport.php



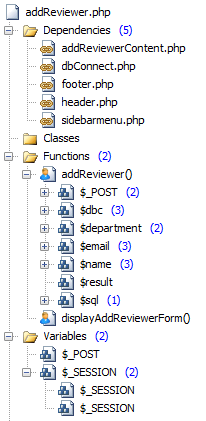
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* Pull the information from the database and displays it in a table.
* Check if it was successful.
* Make sure there are some files in there and then display the top of a table.
* Print each file.
* Close the mysql connection.

1. mgtReviewerInstr.php

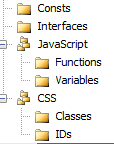


* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* Check if the user is already signed in before providing access.

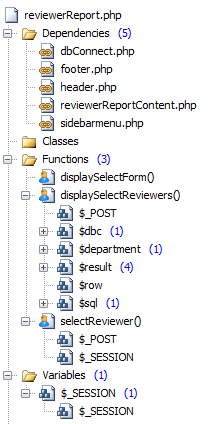
1. addReviewer.php



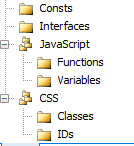
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require ("dbConnect.php");
* This file contains the code for admin to add a reviewer.
* Verify that the user is logged in before submission and verify admin role.
* Reset 20 mins before session expired.
* Use mysqli\_real\_escape\_string() to to prevent SQL injection.



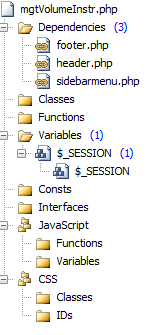
1. reviewerReport.php



* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* Pull the information from the database and displays it in a table.
* Check if it was successful.
* Make sure there are some files in there and then display the top of a table.
* Print each file.
* Close the mysql connection.

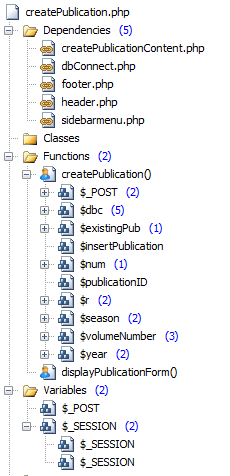


1. mgtVolumeInstr.php

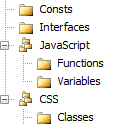


* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* Check if the user is already signed in before providing access.

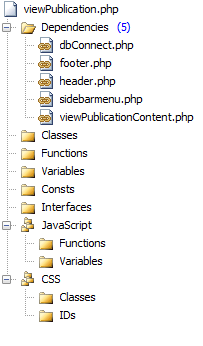
1. createPublication.php



* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* Reset 20 mins before session expired.
* Write a query to avoid duplication of name or id for publication.

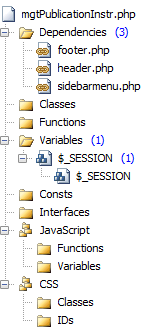


1. viewPublication.php



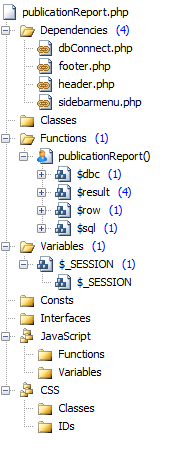
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* require("viewPublicationContent.php");

1. mgtPublicationInstr.php



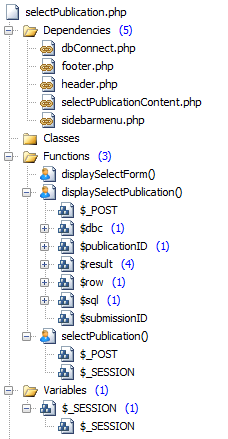
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* Check if the user is already signed in before providing access.

1. publicationReport.php

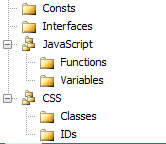


* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* Pull the information from the database and displays it in a table.
* Check if the connection was successful.
* Make sure there are some files in there and then print the top of the table with each file.
* Close the mysql connection.

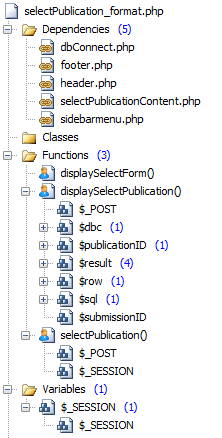
1. selectPublication.php



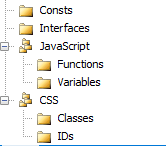
* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* Pull the information from the database and displays it in a table.
* Check if the connection was successful.
* Make sure there are some files in there and then print the top of the table with each file.
* Close the mysql connection.



1. selectPublication\_format.php



* Important Notes:
* require("header.php");
* require("sidebarmenu.php");
* require("footer.php");
* require("dbConnect.php");
* Pull the information from the database and displays it in a table.
* Check if the connection was successful.
* Make sure there are some files in there and then print the top of the table with each file.
* Close the mysql connection.



Query used in Jobie

* search.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission |
| Fields | s.title, s.author, k.keyword, s.fileName, s.publicationID |
| Condition | Inner Join |

* accountCreate.php

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Users |
| Fields | email, password, fName,lName, phone, dept, institutionName, address, city, state, country, zipCode |
| Values | $email', SHA1('$password'), '$fName', '$lName', '$phone', '$dept', '$insName', '$insAdd', '$insCity', '$insSt', '$insCo', '$insZip' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | userID |
| Condition | email = '$email |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Userrole |
| Fields | userID\_ur,roleID\_ur |
| Values | '$row2[0]',1001 |

* login.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | userID,fName,lName |
| Condition | email='$e' AND password=SHA1('$p') |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Userrole |
| Fields | roleID\_ur |
| Condition | userID\_ur= '$userID' |

* accountUpdate.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | Email |
| Condition | email = '$email', userID != '$userID' |

|  |  |
| --- | --- |
| Query Type | UPDATE |
| Table Name | Users |
| Fields | email ='$email',fName = '$fName',lName= '$lName',phone = '$phone', dept ='$dept', institutionName ='$insName', address = '$insAdd', city = '$insCity',state= '$insSt',country = '$insCo', zipCode = '$insZip' |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | email |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | fName |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | lName |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | Phone |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | Dept |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | institutionName |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | Address |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | City |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | State |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | Country |
| Condition | userID = '$userID' |

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | zipCode |
| Condition | userID = '$userID' |

* changePwd.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | userID |
| Condition | userID='$uid' AND password=SHA1('$p1') |

* submissionCreate.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission |
| Fields | filename |
| Condition | fileName = '$aFileName' |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Keyword |
| Values | '$aKeywords' |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Decision |
| Values | NULL,'Pending',CURRENT\_TIMESTAMP |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Submission |
| Values | '$aTitle', '$aAuthor', '$iAffiliation', '$contactAuthor', '$aFileName',CURRENT\_TIMESTAMP,'$decisionID' |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Usersubmission |
| Values | $userID','$submissionID |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Subkeywords |
| Values | $submissionID','$keywordID |

* submissionResult.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission, usersubmission |
| Fields | s.submissionID, s.title, s.fileName, s.dateSubmitted, s.payment, d.decision |
| Condition | INNER JOIN |

* userReport.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | userID |
| Condition | ORDER BY |

* userReport\_delete.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Users |
| Fields | userID |
| Condition | ORDER BY |

* submissionReport.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission |
| Fields | submission.submissionID, title, submission.author, submission.fileName, submission.dateSubmitted, submission.payment, decision.decision, submission.publicationID |
| Condition | INNER JOIN |

* submissionCreate\_a.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission |
| Fields | Filename |
| Condition | fileName = '$aFileName' |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Keyword |
| Values | $aKeywords |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Decision |
| Values | NULL,'Pending',CURRENT\_TIMESTAMP |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Submission |
| Values | $aTitle', '$aAuthor', '$aFileName',CURRENT\_TIMESTAMP,'$decisionID |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Usersubmission |
| Values | $userID','$submissionID |

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Subkeywords |
| Values | $submissionID','$keywordID |

* submissionReview.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission |
| Fields | s.submissionID, s.title, s.author, s.fileName, s.dateSubmitted, s.payment, d.decision, s.publicationID |
| Condition | INNER JOIN |

* reviewReport.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission |
| Fields | s.submissionID, s.title, s.dateSubmitted, d.decision |
| Condition | INNER JOIN |

* addReviewer.php

|  |  |
| --- | --- |
| Query Type | INSERT |
| Table Name | Reviewer |
| Values | NULL,'$name','$department','$email |

* reviewerReport.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Reviewer |
| Fields | reviewerID, name,email |
| Condition | department = '$department' |

* createPublication.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Publication |
| Fields | publicationID |
| Condition | publicationID = '$publicationID' |

* publicationReport.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Publication |
| Fields | s.submissionID, s.title, s.author, s.fileName, s.dateSubmitted, s.payment, d.decision, s.publicationID |
| Condition | INNER JOIN |

* selectPublication.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission |
| Fields | s.title,s.author,s.page, k.keyword,s.submissionID |
| Condition | INNER JOIN |

* selectPublication\_format.php

|  |  |
| --- | --- |
| Query Type | SELECT |
| Table Name | Submission |
| Fields | s.title,s.author,s.page, k.keyword,s.submissionID |
| Condition | INNER JOIN |