|  |  |  |  |
| --- | --- | --- | --- |
| **CSY2028: Web Programming** | | | |
| Due for Issue  (week commencing): | **Thursday 19th November 2017** | **Last Date for**  **Submission:** | **Sunday 14th January 2018**  **23:59:59** |
| Agreed Date for late submission: |  | Module Tutor: | Thomas Butler |
| Student ID: |  | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aspect (& weighting) | Excellent  A | Good  B | Satisfactory  C | Needs some more work  D | Needs much more work  F |
| Functionality (35%) |  |  |  |  |  |
| Testing and Error Handling (15%) |  |  |  |  |  |
| Technical Documentation (20%) |  |  |  |  |  |
| Evaluation (15%) |  |  |  |  |  |
| Code Quality and Efficiency (10%) |  |  |  |  |  |
| Demonstration (5%) |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Specific aspects of the assignment that the marker likes: | | Specific aspects of the assignment that need more work: | | | |
| Tutor’s Signature: |  | Date: |  | Grade: |  |

The University of Northampton Policy on Plagiarism & Mitigating Circumstances will be strictly implemented. By submitting this assignment you are asserting that this submission is entirely your own/individual work.

Contents

[1) Testing: 4](#_Toc503731692)

[1.1) Test Tables: 4](#_Toc503731693)

[1.1.1) Header Tests: 4](#_Toc503731694)

[1.1.2) Left Side Tool Bar Tests When Logged Out: 5](#_Toc503731695)

[1.1.3) Log In Page: 5](#_Toc503731696)

[1.1.4) Sign Up Page: 5](#_Toc503731697)

[1.1.5) Search Function: 6](#_Toc503731698)

[1.1.6) Left Tool Bar Functions: 7](#_Toc503731699)

[1.1.7) Articles: 9](#_Toc503731700)

[1.2) Errors: 10](#_Toc503731701)

[2) Design & Technical documentation 11](#_Toc503731702)

[2.1) Directive: 11](#_Toc503731703)

[2.2) Information for another Developer: 12](#_Toc503731704)

[2.2.1) SQL code: 13](#_Toc503731705)

[2.2.2) Security: 13](#_Toc503731706)

[2.2.2.1) Validation: 13](#_Toc503731707)

[2.2.2.2) Passwords: 14](#_Toc503731708)

[2.2.3) Account to log in with: 14](#_Toc503731709)

[2.2.4) Simple processes to run: 14](#_Toc503731710)

[2.2.4.1) Adding a new page: 14](#_Toc503731711)

[2.2.4.2) Adding a new field to the news article page: 14](#_Toc503731712)

[2.2.4.3) Adding a new field to the news article comments: 15](#_Toc503731713)

[2.2.4.4) Adding a edit comments section: 15](#_Toc503731714)

[2.2.5) Naming convention and design decisions: 15](#_Toc503731715)

[3) Evaluation: 16](#_Toc503731716)

[3.1) Well Implemented Sections: 16](#_Toc503731717)

[3.2) Improvements: 16](#_Toc503731718)

[3.3) Code Usability: 17](#_Toc503731719)

[4) References: 17](#_Toc503731720)

# Testing:

I have created a table which explains what I have texted, what I expect the outcome to be, what the actual outcome is and whether it has been fixed or works as expected.

To best lay this out I will split the testing up by pages. Each page will have its own testing table with all the features included.

After the tests have been carried out I will then list all the errors and bugs. These will be explained and any fix for these will be documented here.

# Test Tables:

# Header Tests:

These tests are to see that everything In the header works as expected and to find any bugs within this area of the website

The following tests took place with the user being logged out.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: | What is the test: | Input entered: | Expected result: | What happened: | Webpages involved: |
| 01 | Home button will navigate to the welcome page. | Clicked home button | Show the welcome message and all navigation bars | Worked as expected. | header.php |
| 02 | Articles button will show articles created most recent first. | Clicked Article button | To show a list of the articles created most recent first. | Worked as expected. | header.php  latestArticle.php |
| 03 | Hover over the categories button in the header navigation. | Hovered over categories button | To show a list of all the news categories . | Worked as expected | header.php |
| 04 | Click the categories button. | Click categories button. | List of all the categories in the main content section. | Worked as expected | header.php  articleCategory.php |
| 05 | Click the contact us button | Click Contact us button | Changes main content to show contact us content | Worked as expected | header.php  Contact.php |

# Left Side Tool Bar Tests When Logged Out:

I will be testing the column on the left hand side as a logged out user. This shows some extra features of the website for users that don’t have an account.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: | What is the test: | Input: | Expected result: | What happened:  ()error no. | Webpages involved: |
| 06 | search function takes you to a search function. | Click the search function text | Will display the search function. | Worked as expected | leftColumn.php  search.php |
| 07 | Sign up link takes you to the sign up page. | Click the sign up text | Display a sign up page. | Worked as expected | leftColumn.php  signUp.php |
| 08 | Log in link takes you to a log in screen. | Click the log in text | Display in the main content log in information. | Worked as expected | leftColumn.php  logIn.php |

# Log In Page:

I will be testing all the functions on the log In screen. When looking at the site this will be what is stored in the main content area when Log In link has been selected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: | What is the test: | Input: | Expected result: | What happened:  () error no. | Webpages involved: |
| 09 | To log in successfully. Left column updates. | “test”, “test” | Be able to log in successfully. | Works as expected. | LogIn.php  logInCompleted.php |
| 10 | Enter just an email address | “test” | To have a error message display asking for a password. | (1)Doesn’t show error message but doesn’t log you in. | logIn.php  logInCompleted.php |
| 11 | Rerun test 09 & 10 | “test” , “test” & “test” | Same as before | Both work as expected | logIn.php  logInCompleted.php |
| 12 | Enter just a password | “test” | Error saying credentials aren’t correct | Worked as expected | logIn.php  logInCompleted.php |

# Sign Up Page:

I will be testing all the functions on the Sign up screen. This is when a user clicks on the sign up link above log in on the home screen.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: | What is the test: | Input: | Expected result: | What happened:  () error no. | Webpages involved: |
| 13 | Can add a user successfully | “test”, “test”, “03-04-1993”, ”test”, “test” | To be able to add a user and have a confirmation screen. | Adds user to database and requests them to log in. | signUp.php  signUpCompleted.php |
| 14 | Fill out just the first name of the page. | “test1” | To have a invalid entry error page show. | Worked as expected. | signUp.php  signUpCompleted.php |
| 15 | Fill out just the surname. On the page | “test1” | To have a invalid entry error page show. | Worked as expected. | signUp.php  signUpCompleted.php |
| 16 | Enter a invalid entry into dob. | “test” | Shouldn’t be able to. | Worked as expected un able to enter text into date. | signUp.php |
| 17 | Fill out just the email field. | “test1” | To have a invalid entry error page show. | Worked as expected. | signUp.php  signUpCompleted.php |
| 18 | Fill out just the password field | “test1” | To have a invalid entry error page show. | Worked as expected. | signUp.php  signUpCompleted.php |

# Search Function:

I will be testing all the search function in the left hand column when a user isn’t logged in.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: | What is the test: | Input: | Expected result: | What happened:  () error no. | Webpages involved: |
| 19 | Select a valid radio option just one and to enter a keyword to test the search. | Clicked “get article by title”, “test” | To execute a search and bring back the correct data. | Worked as expected. | Search.php searchResult.php |
| 20 | Select all the radio buttons | Clicked all radio buttons and entered “test” | To throw out an error or to run all searches. | (2)Ran all searches and showed the error message for categories. | Search.php searchResult.php |
| 21 | Select incorrect radio button and try and unselect it. | Clicked “get article by title” then clicked it again. | Should let me unselect the radio button | (3) couldn’t unselect the radio button unless I refreshed the page. | Search.php searchResult.php |
| 22 | Leave the search box blank |  | Show a error message | Worked as expected. | Search.php searchResult.php |
| 60 | Only enter a keyword | “test2” | Show error message | (12) No error message was shown. | Search.php searchResult.php |

# Left Tool Bar Functions:

I will be testing that all the function on the left hand column work as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: | What is the test: | Input: | Expected result: | What happened:  () error no. | Webpages involved: |
| 23 | Unauthorize a comment by selecting the unauthorized link. | Clicked the link unauthorize a link | To delete the comment | Worked as expected | Comment.php |
| 24 | Authorise a comment. By clicking on the authorize link on a comment. | Click on the authorize link | The comment should be displayed on the article. One authorized. | Worked as expected | Comment.php  Article.php |
| 25 | Add a new article. Use correct information to add an article. | Click add a new article. Enter “test1”, “test1”, 2017-04-02, “Music”, ”test” | The article should be added and viewable through categories and latest Article | Worked as expected | Article.php  addArticle.php |
| 26 | Edit the categories function should add new category | Add category text box enter “test” | The category should be added. | (4) added but no confirmation text. | category.php |
| 27 | Retest the above | Add category text box enter “test1” | The category should be added | (5) added and successful message appears but if you don’t use the other features then their unsuccessful messages appear. | Category.php |
| 28 | Delete a category | Delete a valid category “test2” | The category should be deleted | (6) deleted successfully but other functions that weren’t used displays messages. | Category.php |
| 29 | Edit category name | Edit category “test” to “test3” | The category should be changed. | (7)changed category but displayed successful messages from other fields. | Category.php |
| 30 | Fill out all fields for the edit category page | “test3”  “test1”  “test”  “test2” | All fields should execute changes as expected. | Worked as expected | Category.php |
| 31 | Try adding a category that is already there | “test” | Came back with error message | (8)error messages came back from all fields. Correct error message is showing for this field | Category.php |
| 32 | Try deleting a category that isn’t there. | “test1” | Error message should be returned | (9) error message came back for all fields. The correct one is showing for this field. | Category.php |
| 33 | Try editing a category that doesn’t exist. | “test1” | Error message should return | (10)error message showed as expected but other field messages came through as well. | Category.php |
| 34 | Edit content with correct article and text | “test1”, “test4” | Should edit article content correctly and display a message of confirmation. Executes update as expected | Worked as expected | editArticle.php |
| 35 | Edit content of an article that isn’t there | “test4”,”test” | Should show an error message saying article couldn’t be found | Worked as expected | editArticle.php |
| 36 | Edit article name successfully | “test1”, “test” | Should show a successful message of the change. Executes update as expected | Worked as expected | editArticle.php |
| 37 | Edit article name with invalid name | “test4”, ”test” | Should show an error message | Worked as expected | editArticle.php |
| 38 | Edit article author successfully | “test”, “test4” | Should show a completed message. Executes update as expected | Worked as expected | editArticle.php |
| 39 | Edit article author with an invalid article | “test4”,”test4” | Should display an error message | Worked as expected | editArticle.php |
| 40 | Edit an articles category | “test”, “Music” | Should show a success message | Worked as expected | editArticle.php |
| 41 | Edit article category with invalid article | “test4”,”Music” | Should show error message | Worked as expected | editArticle.php |
| 42 | Edit article category with invalid category | “test”, “test4” | Should display error message | Worked as expected | editArticle.php |
| 43 | Delete article | “test” | Should delete article and display confirmation message | Worked as expected | EditArticle.php |
| 44 | Delete article entering invalid article | “test4” | Should return an error message | Worked as expected | editArticle.php |
| 45 | Edit a users first name | “test”, “test1” | Should successfully execute and display a confirmation message | Worked as expected | Users.php |
| 46 | Edit a users surname | “test”, “test1” | Should change the surname and display a confirmation message. | Worked as expected | Users.php |
| 47 | Change users DOB | “test”, “2000-03-07” | Makes the change to user DOB and shows a confirmation message | Worked as expected | Users.php |
| 48 | Change users email | “test”, “test1” | Changes users email and shows confirmation message | Worked as expected | Users.php |
| 49 | Change users password | “test1”, “test” | Changes password and shows confirmation of change | Worked as expected | Users.php |
| 50 | Delete a user | “test1”,“test1” | Should delete user and show confirmation message | Worked as expected | Users.php |
| 51 | Make changes all at once (apart from delete a user) | [test@gmail.com](mailto:test@gmail.com), “test”, “test”, 10-04-2001, “test”,”test” | Should make changes to the user and display confirmation messages for everything that was actioned. | Worked as expected | Users.php |
| 52 | Log out successfully | Click log out link then button | Should log a user out only if the button is clicked | Worked as expected | logOutPage.php |
| 53 | Log out button but don’t press log out again | Don’t click the log out button | Should return to previous page as normally | (11)This logs a user out. | logOutPage.php |
| 61 | Edit category name update articles | “Music” , “test” | Should change the category music to test | (13) Changes category name but articles still have the category assigned to them. | Categories.php |

# Articles:

I will be testing all the functions on an article page and routes too an article so going through the categories and the latest article options.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test No.: | What is the test: | Input: | Expected result: | What happened:  () error no. | Webpages involved: |
| 54 | Articles take you to a list of articles in earliest first | Click articles | Show a list of articles in correct date order | Worked as expected | latestArticle.php |
| 55 | Articles take you to their article page when clicked | Click the article title | Shows the articles page | Worked as expected | latestArticle.php  Article.php |
| 56 | Select a category | Click “music” | bring up a list of the articles that are in that category | Worked as expected | ArticleCategory.php |
| 57 | Select article title when music is selected as category | Click “test2” | Bring up test 2 article page | Worked as expected | ArticleCategory.php  Article.php |
| 58 | Check article look when logged in and out. | Click log out | Adding comments should be visible when logged in but not when logged out. | Worked as expected | Article.php  LogOutPage.php |
| 59 | Check comment text box lets you add a comment | “test” and click submit button | Should accept the comment pass a confirmation message and hold comment in the authorize comments section. | Worked as expected | Article.php |

# Errors:

Fixed errors and bugs are in green.

Not hard fixed but have a workaround.

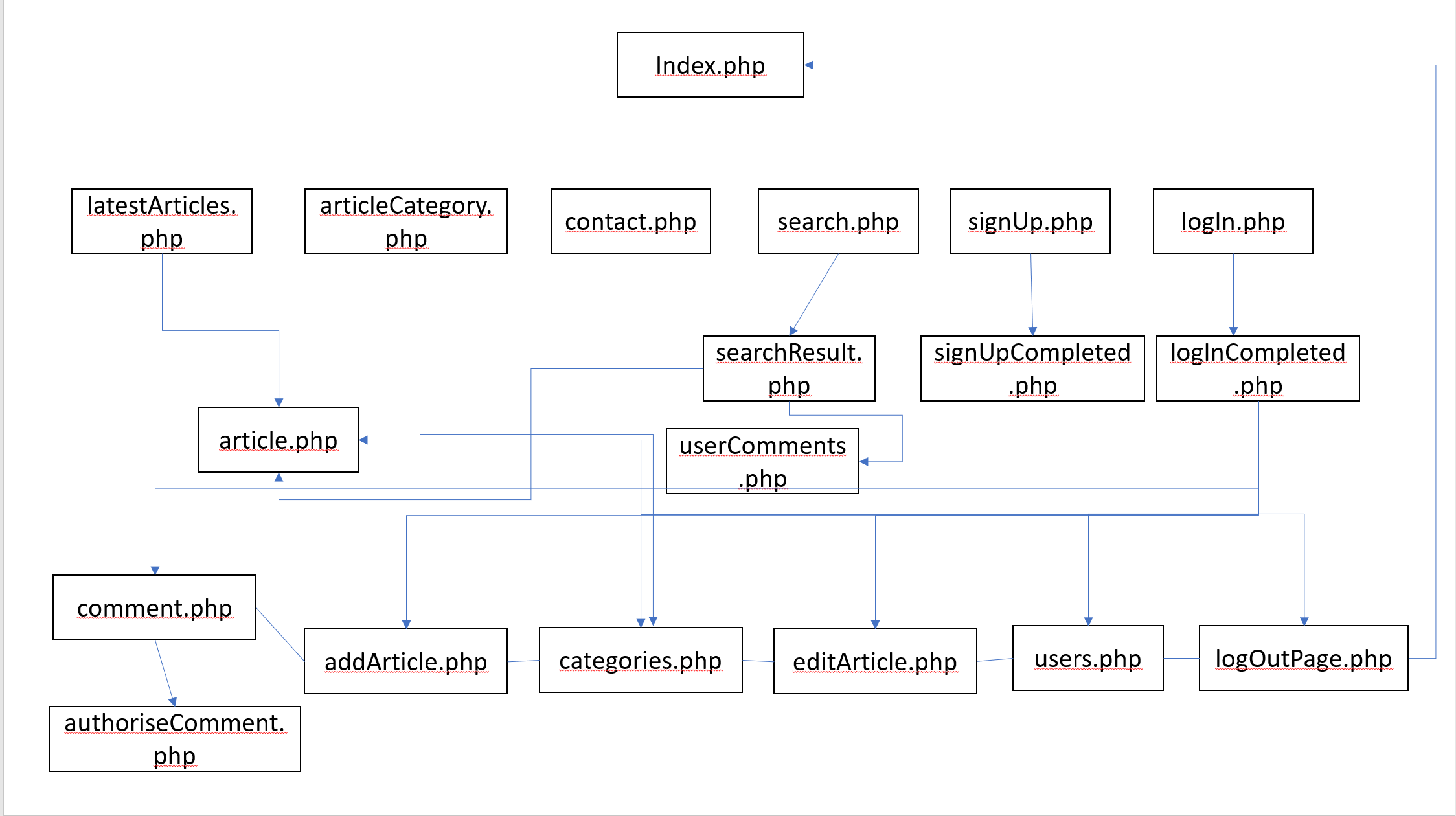
Unfixed errors are in red.

1. Test 10) The widow displays that the user has successfully logged in when they haven’t successfully logged in. Solution: lines of text wasn’t in the correct place I have replaced the text and will run both test 10 and test 09 again.
2. Test 20) Either the search function should pull through just search results or show an error message. It shouldn’t do both. Solution: I have written some text to try and stop people from selecting multiple radio buttons.
3. Test 21) Radio buttons wouldn’t let me unselect them unless I refreshed the page. Solution: I could make all options of the radio button have the same variable this would mean that only one radio button could be selected but I would have to re do the search feature and come up with a way of using one variable.
4. Test 26) Added category doesn’t let the user know if the category was added or not. Solution: I have entered some text to confirm that the adding category function was successful.
5. Test 27) If you use only one edit category feature the result will pull through error messages saying the other features haven’t been successful. Solution: need to add another if statement to the categories.php page that will read if this is empty or not set then read this has not been used or leave blank.
6. Test 28) If you use only the delete category feature the result will pull through messages saying the other features have been successful when they weren’t used. Solution: need to add another if statement to the categories.php page that will read if this is empty or not set then read this has not been used or leave blank. You could also put them as different options on different pages but this will add a lot more pages to your directory.
7. Test 29) This is the same issue as above but just on the edit function. Solution: would be the same as the previous error.
8. Test 31) This is the same issue as above where other fields error messages are being pulled through.
9. Test 32) correct error message for field used is shown but the other fields messages are being pulled through.
10. Test 33) Same error as above. Solution: could use get to pull through the correct field that was being changed.
11. Test 53) The log out function runs whether or not the log out button is clicked. Solution: this will be because the $session variable is being destroyed so will have to move it to the if statement that runs if the button is pressed.
12. Test 60) When entered just a keyword no error message is pulled through. Solution: The file searchResults.php will need a else statement with an error message inside when a keyword has been set but no radio button option.
13. Test 61) The articles don’t get updated with category change of name. Solution: can implement a query that retrieves all articles that have the category and then update each one with the new category name. Temp solution for now: I have but a note in saying to only change names of empty categories.

# Design & Technical documentation

# Directive:

This is a flow chart showing how each page reacts with each other. This will explain the navigation of the site and what pages link up to what. There are some pages that won’t be included I will give a brief explanation of why and what they do.



Pages that aren’t included:

Database.sql – This is the database file that has saved the database.

dbAccessObject.php – This is a class for connecting to the database and has all the sql functions stored in it. If you want to write a SQL then it should be in this file.

Footer.php – This is The footer of the website and is required on all the pages.

Header.php- This is the header on every page and is linked by a require.

leftColumn.php- This is the left column that runs on every page. This also sets where the column is in logged out mode or logged in and shows all the features.

Styles.php- This is the css style for the website.

Randombanner.php- This runs a random banner on the site.

# Information for another Developer:

All of the pages have a standard structure this is to add three components of a page together:

* Header: this connect the $SESSION variable to the page, also the database variables are set up here. The header also sets the top navigation bar and the categories drop down menu.
* leftColumn: This is where the left navigation bar is created and displayed it runs on an if statement from the $session variables so if you’re not logged in you get one navigation bar and if you’re logged in you will have access to the more detailed navigation bar. If you wish to create more function and tools then this will be the place to add them. I use this column as the “toolbox” for the admins.
* Footer: This just displays what’s at the bottom of the page after the main content and the column ends.

# 2.2.1) SQL code:

The website has one page that stores all the SQL written that will talk to the database. You will need to call this class and the function on the page you want to query. Best thing to do is look at the current method the comments will explain what the function does if none meet your needs then you can write another method at the end to query the database. An example of running a piece of SQL code is shown below:

$emailVerificationResult = $dbAccessObject->getUsersByEmail($\_POST['email']);

# 2.2.2) Security:

# 2.2.2.1) Validation:

Validation is a method at the bottom of some pages that take $POST variables the method will be passed the $POST array. This method will then run some if statements on the variable to make sure all the data has been entered correctly. The method returns a variable of whether the data is valid or not. An example is shown below:

$validationResults = validateFields($\_POST);

function validateFields($fields){

// array for validating the fields if they are false or blank then the field isn't valid

$valid = [

'isValid' => true,

'invalidField' => ''

];

if(!isset($fields['comment']) || empty($fields['comment'])){

$valid['isValid'] = false;

$valid['invalidField'] = 'comment';

}

This should be the main validation method used or to run an if statement to check whether a variable has been set or not.

# 2.2.2.2) Passwords:

Security is mainly to talk about the password. The way the passwords are stored is firstly any table requesting a password should use the input type of password. When storing passwords into the data base I add the password to the email and then use a function to hide it in a scramble of characters which is then visible in the database. This is so if someone view the database you won’t be able to see the passwords.

Example: 'password' => sha1($\_POST['email'] . $\_POST['password'])

# Account to log in with:

You can use the Sign Up section to create your own account if not you can use an account already made.

Website: http://assignment.v.je/

Email: test5

Password: test5

# Simple processes to run:

# Adding a new page:

To Add a new page just follow these step by step instructions:

* Add the file to the directory with all the rest.
* Name it with what its aiming to do and add .php to the end.
* You will then need to add the website structure which is to require then header.php, leftColumn.php and the footer.php. The best way to do this is to copy and past the contactUs page and delete everything in the main content which isn’t a require.

This is all you need to do to add a new page with the same layout as all the other pages.

# Adding a new field to the news article page:

To accomplish this you will need to follow the steps below:

* Firstly add the field you want to the database table articles this will then allow you to pull the data through.
* Next add the field in the first foreach loop on the article.php page. This will allow the information to be pulled through from the loop.
* Make sure the database function is pulling through the right columns it should be everything being pulled through the function is called retrieveArticle which is just above the foreach loop.

# Adding a new field to the news article comments:

To add a field to the comments section then follow the below steps:

* Add the field to the comments table in the database.
* Open the article.php file. You can then edit the foreach loop below the article foreach loop to display the data can use template of the other information.
* Check that the information is being pulled through from the database. This function is the retrieveArticleComments() function make sure it is selecting all information then you can display any new rows. This is located just above the foreach loop.
* Just below the foreach loop there is an if statement for inserting comments into the database. This section you can add the field rating and pass it the variable the result is stored in. The function addComment function will need updating you will have to add the new column to is.
* After you have done all this just below the if statement is the table where a comment table is displayed you can add a new line for the ratings.

# Adding a edit comments section:

You can do this by two ways. Firstly you can use the authorize page and extends it or create a new link to a page and add the feature to the leftColumn page so all the tools are in the same place.

* You can do this with multiple ways you can either copy the edit article way which is create separate links in the left column. I wasn’t a fan of this. The best one is the edit category way where all the changes can be located on one page. This uses POST and GET to decide what edit you want to run and from there use a similar style of if statement as the other edit features.
* I would use the validate function to check all the data you are setting for the page and then run if statements depending on what has been selected.
* By using a table layout it will allow the changes to be clear and easy for a user to use.
* You should write all the database scripts in the dbAccessObject file and link the pages together this will keep all the data together.

# Naming convention and design decisions:

* File names: The main idea behind the page names is to describe what they are doing and to give a clue that the page is linked to ass well. Other pages are named by what they do for example the Header is created by the header.php file. This makes it easy to pick out what pages you need for your task.
* Variable names: The variable names are long and as descriptive as possible it makes it easy for someone else to pick up and understand what they do and what data they pull through. This means they can be easily checked and recreated if needed.
* dbAccessObject file: This file was created so you can store all the database code in one place. All you need to do is reference the class run the method and add that method to the end of the page. This keeps it all ordered and comments help explain what the method does.
* Requires: I decided to use requires to split all the webpages up into sections. The webpage code should be all about the function and point of that webpage if the page has the same parts as other pages then a require would save on repeated code and make the page easier to read and edit.

# Evaluation:

# Well Implemented Sections:

A few things I think has been implemented well are:

* Webpage structure: using require and keeping the main function of the webpage as the only code on it allows the page to be easily read and easily edited instead of having a load more extra lines of code that is all the same on all the pages.
* Toolbox idea: I like the toolbox idea where all the features and functions that a user would want to carry out are in one position this is so the user knows exactly where to go. This also helps when future developers add features to the site all they are all in one place already.
* dbAccessObject page: This is my favorite idea as it keeps all the database code together and easily searchable and usable for future functions. The idea here is to write a class that links to the database and to the page. So if you want to write a query to the database then you will have to call the class then call the correct function. This means there is less code on the pages which makes them easier to read and if you have any issues with results being pulled through there is a paper trail that tells you what each function is doing which makes the error easier to find.

# Improvements:

* There are a few bugs which are down to else sections in the code not isolating an issue. For example if I log out the session destroy function shouldn’t run until a button is clicked but the position it is in within the if statement means it runs anyway. I would like to clean that up and make it more sturdy so you know exactly what error message to get when something happens. In some instances you get three messages at once.
* I would like to use less database functions. This is because I added functions to the bottom and didn’t look for ways to reuse other functions by pulling through extra information than needed. The broader the query then the more reusable the function would be.
* I would like to re do the users part of the site. This is due to a issue with who I thought the user is. If the user is in the company and they are suppose to add a comment then this is fine. But it means normal users who won’t want to sign up won’t be able to add comments. I would basically add another column which would make users admins or something else which can be used as a criteria for the features. So if they didn’t have the admin as a “Y” (just like authorising the comments) then they could add comments and articles but make no changes and not be able to access some of the features. If the user had a “Y” then they would be viewed as a admin and can then do all the additional features. The way the site is set up all users that sign up are considered an admin. This wouldn’t be a hard change you would just need to add an extra piece of validation to each feature and an extra column to the Users table. If it’s a “Y” I could change the left navigation column to show what it does now. If its an “N” then the column will have a few less features.

# Code Usability:

I believe the code is really easy to read and to make changes too. All the pages are only about the main focus of the page. Example: header is setting the header lay out. LeftColumn sets the layout of the toolbox on the left hand side. Contact page just as the information that is displayed on the contact page.

Pages do link up which makes it even easier to read. There is comments that give you an over view of what each section of code is doing. I also think that the naming conventions of the variables help understand what information is stored and what the reason for the variable is.

I like how the validation of fields is done as a function and at the bottom of the page.

Downsides of the code are the fact that the validation isn’t used through out. there is two different validations either the if statement starts with isset criterias which should be in a validation function because it makes the code easier to read. The other way is that the code runs from a is valid variable set to true which is a much easier if statement to read and if there is any errors with the data the output tells you what the field is that is causeing the issue.

# References:

1. $commentContent=preg\_replace("/[\n]/","<br>",$\_POST['comment']); - Page used on: article.php, addArticle.php, editArticle.php This piece helps store spaces I got it from the following site: <https://stackoverflow.com/questions/12498337/php-preg-replace-replacing-line-break>
2. <label>Change date of birth:</label> <input type="date" name="dob" /> -Page used on: users.php, signUp.php, addArticle.php, editArticle.php. This piece allowed me to set the date for forms so that text couldn’t be entered into it. Website: <https://www.w3schools.com/html/html_form_input_types.asp>
3. <label>get article by title:</label> <input type="radio" name="article" /> - This helped me set up and set variables to radio buttons on the search function. Page used on: search.php, website: <https://www.w3schools.com/html/html_form_input_types.asp>