**Analysis and Design Report**

This report will describe the general role of the script and discuss what the script needs to do in terms of database queries, gathering input from the user. For example, form a form or form a query strings, this will link to other scripts or performing calculations. This report will also include details of techniques to maintain state such as sessions, query strings and cookies, and of which security measures are to be implemented and why they must be included.

**Task 1: (GRADE D)**

**Register page / Home page: (HTML page)**

The users have to enter their details to register

We need an HTML form for this; a form will include ‘username’ and ‘password’.

The details then will be added to the database

We need a PHP script for this; PHP Query strings and POST method.

**PSEUDO CODE: (Register page)**

Read details (username and password) from form

Set POST variables to add new details to the database

Connect to the database

Send SQL Post Query to the database

If details created successfully and add on to the database

Display message saying, ‘New User created successfully’

It will then be redirected to home page to login/register

**End Connection to database**

**----------------------------------------------------------------------------------------------------------------**

**Login page / Home page: (HTML page)**

The users will have to enter their details

We need an HTML form for this; this form will include ‘username’ and ‘ password’.

The server will be sent to the database to search for the records

A PHP script will be used for this to search the database.

If no records found, the users will see error message

A PHP script will be used for this too to display the message.

**PSEUDO CODE: (Login page)**

Session Start

Read the user’s details (username and password)

Set POST variables to verify the details from the database

Connect to the database

Send SQL SELECT query from users where username

If $row == false

It will echo out error message ‘Invalid Username or Password’

End Connection to databse

Re-directs to index.php

Else

The users will be logged in

Then will re-direct to home page

End connection to database

**----------------------------------------------------------------------------------------------------------------**

**Loggingout.php**

The whole session will be destroyed after logging out

A Session destroy PHP script and HTML logout button is required.

The session is destroyed for security

**PSEUDO CODE: (Loggingout.php)**

Session Start

Connect to database

The users are logged in

Session Destroy

The users will get logged out

Destroys the current session

Re-directs to home page

**----------------------------------------------------------------------------------------------------------------**

**Lockingout.php**

This page blocks the users from accessing the website unless they are logged in

Gatekeeper PHP script is required; it blocks the users.

When the users clicks menu bars without logging in, a message with a hyperlink will say ‘Please LOGIN to access’ will display on the screen.

It requires a ‘isset Session Gatekeeper’ PHP script

This script blocks the users from accessing the pages

**PSEUDO CODE: (Lockingout page)**

No Pseudo Code is required for this page

**----------------------------------------------------------------------------------------------------------------**

**Addplaces.php (HTML page)**

The users add new place details on this page using a search bar

HTML form is required including ‘name’, ‘type’, ‘description’, ‘country’, ‘region’, ‘recommend’ and ‘submit’ button.

Needs PHP script to make the submit button work

The details will add to the database after verifying

A PHP script such as $conn, query, $results and rowCount is required.

An error message will pop up if the details are incorrect

A rowCount script will be used to to add the details to the correct columns.

New detail will be added to the database if they are correct and will show the users’s usernames on the database, and a message will show on the screen saying, ‘A new place has been added successfully’

PHP script such as gatekeeper, username and a session\_start() is required to show the username of the user.

**PSEUDO CODE: (Addnewplaces.php)**

Session Start

Read the user’s details from the database

Set POST variables to add new place details to database

Set $\_SESSION["gatekeeper"];

Connect to database

Send SQL INSERT query into places and values of the new place to add in to the correct database in the correct rows and columns

If $results->rowCount() == 0

It will echo an error message, ‘Insert failed’

Else

It will echo a successful message, ‘Successfully added a new place’

End connection to database

Re-directs to adding new places page

**----------------------------------------------------------------------------------------------------------------**

**Searchandreviews.php (HTML page)**

This page will allow the users to search places by ‘type’

A PHP script such as $type and $\_GET, and $conn and query will be used when the users search for places by type.

An HTML search bar with a search button is required for it

If a place is not found on the database, then an error message will pop up on the page with a hyperlink for the users to go back and search again

$row, $results and fetch PHP script will be used to display the message.

If a place is found, then the results will load on the page

A PHP script would be GET method and an echo will echo out the matching results

**Search.php: (PSEUDO CODE)**

Session Start

Read the users’ details

Set GET variable using $type = $\_GET[“type”]

Connect to database

Send SQL SELECT query from places where type = ‘$type’

If $row == false,

Display an error message using echo if no results found

Else

While ($row)

$results and fetch to display the results on the screen

Display results of the place using echo

End connection to database

**----------------------------------------------------------------------------------------------------------------**

**Task 2: (GRADE C)**

**Reviews.php (HTML page)**

The users will write their reviews into the form

A HTML form including review textarea, date and submit button is required.

Hidden field is used on the review box, username, placeID and ID

A PHP script is required to create a Hidden Field and ID.

Need a submit button on both PHP script and HTML page

The review is then sent to the database and added

A PHP script is used to send the review to the correct location with the correct ID.

**Write Reviews.php: (PSEUDO CODE)**

Add a query string in HTML page in input value = ?php echo $placeID ?>

This will allow the users to add review to a specific place

This echo out the ID of a place

Session Start

Read the users’ details

Set POST variables to write a review of a chosen place using $pl = $\_GET[“placeID”]

Set $\_SESSION["gatekeeper"];

Connect to database

Send SQL INSERT query into reviews (placeID, review, username) and values to the database

Echo hidden fields for ID, PlaceID and Username

Echo a message saying, ‘You have added a review’

End connection to database

Re-directs to Search page.

**----------------------------------------------------------------------------------------------------------------**

**Show Reviews.php: (PSEUDO CODE)**

Session Start

Read the users’ details

Set GET variable to find reviews of places using $pl = $\_GET[“place”]

Connect to the database

Send SQL SELECT query from reviews where placeID = $’pl’

$row = $results->fetch() get the reviews of a chosen place

If $row == false

If results not found, then

t will echo a message saying, ‘sorry, no reviews found’

Else

While ($row)

All reviews of a place will be displayed using echo

End connection to database

**----------------------------------------------------------------------------------------------------------------**

**Recommend.php (HTML page)**

No HTML page required for this page as it only requires PHP script

**Recommend.php: (PSEUDO CODE)**

Add a query string in HTML page in input value - ?php echo $placeID ?>

This will allow the users to add review to a specific place

This echo out the ID of a place

Session Start

Read the users’ details

Set GET variable using $pl = $\_GET[“placeID”]

Set $\_SESSION["gatekeeper"];

Connect to the database

Send SQL Update query to update places SET recommended=recommended+1 where ID = ‘$pl‘’

When the user click recommend link, the recommend column in the database will change from 0 to 1

It will then echo back to where the users want to

End connection to database

Re-directs to search page

**----------------------------------------------------------------------------------------------------------------**

**Task 3: (Grade B)**

**Admin.php: (PSEUDO CODE)**

Session Start

Read the users’ details

Connect to the database

Send SQL SELECT query from reviews where approved = 0

$row = $results->fetch() will gather all pending reviews to be approved

If ($row == false)

If no results found, then

It will echo a message, ‘No Reviews to approve!’

Else

While($row)

Echo results with the approve links underneath

Adds 1 to the database when clicked

A review and a link will disappear

End connection to database

**Approve\_reviews.php: (PSEUDO CODE)**

Set GET query using $id = $\_GET[“id”]

Connect to the database

Send SQL UPDATE query to update reviews SET approved=1 where id=”$id”

It will add 1 to approve column in the database

End connection to database

It then re-directs to admin.php page

**----------------------------------------------------------------------------------------------------------------**

**SQL Injection and Cross – Site Scripting**

Add htmlentities on every PHP page on POST, GET and SESSION after Session start();. This is for security and it secures the PHP scripts as well as no one will be able to access the original code if they want to copy or harm the code.