**TripsWiki Website - Analysis and Design**

**General Security Measures**

To prevent possible security breaches of the website every SQL statement used on the website will be prepared statements. When variables are retrieved and set in PHP I will use the htmlentities function to encode the variables preventing the risk of Cross Site Scripting.

**Task A – Adding a new place of interest**

**The user needs to enter details of a new place of interest.**

A HTML form is needed for this.

**The details should be validated before addition to the database.**

The user input is validated using JavaScript.

**The server needs to add the details to the database.**

AJAX is used for this so that the user doesn’t see the page change. PHP is also used to actually add the details to the database.

**Show confirmation to the user.**

No extra script is needed as it can be displayed using the response from the previously mentioned PHP script inserted into a div on the HTML page holding the HTML form.

**Task B – Search for a place of interest by type**

**The user chooses their desired type of interest to search for.**

A HTML form is needed for this.

**Server searches the database for places of interest for the given type.**

AJAX is used for this so the page displays the results without refreshing the page. A PHP script is also used as it requires the type of POI as a query string from the AJAX request.

**Server displays the results to the user.**

No extra script is needed as the results can be displayed using the response from the previously mentioned PHP script inserted into a div on the HTML page with the form.

**Task D – Viewing all reviews for a place of interest**

**User chooses to view all the reviews for a given place of interest.**

No extra script is required as the user will just be clicking a link.

**Server searches the database for the reviews for given place of interest.**

A PHP script is needed for this as it needs to read in the ID from a hidden field from the previous page.

**Server displays results to the user.**

No extra script is needed for this as the previously mentioned PHP script can display this information.

**Task F (extension of E) – Reviews have to be approved by admins**

**Admins should be able to see all pending reviews needed for approval.**

On load of every page, a JavaScript functions runs and returns a value based on the users status (not logged in, regular user and admin) and will change the visible links accordingly.

A PHP script is needed for this to search the database for all reviews which have not yet been approved by an admin. The results can also be displayed using the same PHP script. To ensure only admins can see this information when a user logs in, the PHP script handling the log in will start a new session, assigning the users admin status to a session variable.

**Admins should be able to approve a review.**

AJAX is needed for this so that the page doesn’t refresh when the admin has selected one to approve. A PHP script is also needed as the review ID which is being approved needs to be passed as a query string to update the chosen review in the database.

**Review should be hidden from the Admin.**

JavaScript will be needed to hide the review from the Admin once it has been approved.

**Task H – Searching for and reviewing places using AJAX**

**A user should be able to search for a location without page refreshing.**

A HTML form will be needed so that the user can type in their desired location name. AJAX is used so that the page does not refresh once a user has input a place name. A PHP script is also needed to search the database using the user input passes as a query string. The results will be displayed by using JavaScript to put the response from the PHP into a div on the HTML page with the form.

**A user should be able to review a place without page refreshing.**

The HTML form for this will have been loaded with the results in the previous task. JavaScript will be needed to validate the review ensuring that it is not empty. AJAX will be needed so that the page will not refresh once a review has been submitted. A new PHP script will be needed using the place ID from a hidden field on the form to add the review to the database.

**Task I (Extension of C) – Rating out of 10**

**A user should be able to give a place a rating out of 10.**

An image, housed by a div, displaying 10 stars will be needed as this is how the user will choose what rating to give a place, this however will not require extra script as it will be displayed with the search results for a place or type. The image will have transparent stars but opaque elsewhere. JavaScript will be needed to control the highlighting of the stars when the cursor is hovering over them, indicating to the user what rating they will give the place of interest. The highlighting will be done by changing the width of the div housing the image and therefore the colour of the div will be shown through the transparent parts of the image. AJAX will be used to submit the rating once a user has selected what rating they would like to give a place. A PHP script will be needed using the rating chosen and the ID for the place being rated as query strings.

**Task J – Graphically displaying average rating**

**The user should be able to immediately see the average rating for a place upon rating**

Following on from the previous task, once the PHP script has updated the database it will then search the database again for the new average rating and number of raters. The new average rating will be used to calculate what the width of the div behind the stars should be and these stars will be displayed to the user using JavaScript to input the response text into a div on the search results page.

**Pseudocode**

**Task A – Adding a new place of interest**

*In JavaScript:*

Read add details (name, type, country, region, description) from form.

Create a new http request.

Open the request to add place PHP script with the details as query strings.

Send the request.

*In PHP:*

Assign query string parameters to variables and encode using htmlentities function.

Connect to database.

Prepare a SQL INSERT statement to insert into the values into the places table.

Bind the parameter using bindParam function.

Execute the statement.

**Task B – Search for a place of interest by type**

*In JavaScript:*

Read the value (type) of the drop down.

Create a new http request.

Open the request to search by type PHP script with the type as query string.

Send the request.

*In PHP:*

Assign query string parameters to variables and encode using htmlentities function.

Connect to database.

Prepare a SQL SELECT statement to find all places with desired type.

Bind the parameter using bindParam function.

Execute the statement.

IF there are no results found

Tell user there are no results found.

ELSE

LOOP through each search results

Display div containing place details,

IF user is logged in

Show rating functionality

ELSE

Show message informing they have to be logged in to rate.

Display links to review and view reviews.

IF user is logged in

Load review form (initially hidden)

ELSE

Load you must be logged in to review message (initially hidden).

**Task D – Viewing all reviews for a place of interest**

Read value of placeID from hidden field loaded when place was searched for.

Open PHP script with placeId as a query string.

Assign query string to a variable.

Connect to the database.

Prepare a SQL SELECT statement to search database for all reviews which have been approved and have the placeID retrieved from query string.

Bind the parameter using bindParam function.

IF no results are returned.

Display no reviews found message to user.

ELSE

Display frame of a table.

Loop through results.

Add results as rows in the table.

**Task F (extension of E) – Reviews have to be approved by Admins & list of approvals**

*On every page:*

User logs in.

IF user is Admin

Make link to see approvals visible.

ELSE

Hide link.

*On approvals page:*

Connect to the database.

Send a SQL SELECT statement to get all reviews which have not yet been approved.

IF no results are returned

Display a message telling Admin there is no outstanding reviews.

ELSE

Display a table.

LOOP through results.

Add results as rows in table with options to approve or delete reviews.

User clicks the approve option.

*In JavaScript:*

Read value of review ID from hidden field loaded with the results previously.

Read in the row relating to review the user chose to approve.

Create a new http request.

Add Event Listener so that a callback function is called when loading happens.

Pass the row to the function.

Open the request to make approval PHP script with the placeID as query string.

Send the request.

*In PHP:*

Assign query string to a variable.

Connect to the database.

Prepare statement to UPDATE the review in the database to make approved true where the reviewID is the same as the variable.

Bind parameter using bindParam function.

Execute statement.

*In callback function:*

Remove the row from the table on screen.

**Task H – Searching for and reviewing places using AJAX**

***Searching:***

*In JavaScript:*

Read the value (location) of the drop down.

Create a new http request.

Open the request to search by location PHP script with the location as query string.

Send the request.

*In PHP:*

Assign query string parameters to variables and encode using htmlentities function.

Connect to database.

Prepare a SQL SELECT statement to find all places where the country or region is LIKE the location.

Bind the parameter using bindParam function.

Execute the statement.

IF there are no results found

Tell user there are no results found.

ELSE

LOOP through each search results

Display div containing place details,

IF user is logged in

Show rating functionality

ELSE

Show message informing they have to be logged in to rate.

Display links to review and view reviews.

IF user is logged in

Load review form (initially hidden)

ELSE

Load you must be logged in to review message (initially hidden).

***Reviewing:***

*In JavaScript:*

Get placeID from hidden field from form.

Get review from the form.

IF the review is empty.

Display error message.

ELSE

Create a new http request.

Open the request to review a place PHP script with the placeID and review as query strings.

Send the request.

*In PHP:*

Start a session.

Get username from session variable.

Assign place id and review to variables.

Connect to the database.

Prepare SQL INSERT statement to input the review into the reviews table.

Bind parameters using bindParam function.

Execute statement.

*In JavaScript:*

Hide element which trigged event in first place.

Display confirmation message.

**Task I (Extension of C) – Rating out of 10**

**Task J – Graphically displaying average rating**