

Documentation of Interviewer Task.

Name: Kazi Ruhol Kabir Moahhed

Index.html

In this html file I have made unsorted list of the given task. First there is Authentication Part. In this part I have made Facebook and Google authentication. Facebook redirect to facebook_button.php and Google redirect to google_login.php. Then Product part has 5 items of Add Product, Update Product, Read Product, Delete Product, Read Single Product. Add Product redirect to add_product.php, Update Product redirect to update_product.php, Read Product redirect to read_product.php and it continues till Delete login history.

Authentication of Facebook

In facebook_button.php it requires facebook_config.php. In facebook_config.php \$fb is a variable where app_id, app_secret key, and graph version has provided. All this has been taken from facebook developers website. In \$login_url I have assigned the localhost url through getLoginUrl() function. Then \$access_token is getting a JSON web token(JWT) through getAccessToken() function. If access_token gets the token then it will redirect to facebook_button.php where authentication of facebook will be completed. If any exception occurs then \$e will represent the exception as string via getTraceAsString() function. This authentication will take the login only once.

Authentication of Google

First I have declared my google client Id. I got it from google developer's website. Then I have synced a script file of google which is apis.google.com/js/platform.js. Then I have created a sign in button. Once the user sign in then a function called onSignIn will provide the googleUser information of Basic Profile. It will provide profile ID, Name, Family name, Image Url, Email and an id token. Then a div class content will show the name of the google user after login.

Products

Add Products: add_product.php has a form of html where user will provide the information of products. Product_name, product_price, product_image, product_color, product_size, product_description will be provided by the user. Then after clicking the submit button this page will redirect to add_product_connect.php which will store the product information to the database. First I have assigned the variable accordingly to the name of each form by post method. Like this-

```
$product_name = $_POST['product_name']
```

Then I connected the database with my php file. Then checking that the database is connected or not. If not connected yet then it will show an error. If connected then it will proceed to the next code. It will run a query to insert the product information into products_table. Then I have bind the parameters to the VALUES accordingly. Just pressing the submit button it will add the form values to the database.

Update Product: At first User will search for a product that exists or not. If the data found in database then it will show the product details where product_name matches in database. If product found in database then user will provide the updated information via a form that has been provided below to the php file. After clicking the submit button it will redirect the update_product_connect.php where updated product information will be stored in the database. Here the MySQL query will be for the update not insert.

For database Connection

```
$conn = mysqli_connect('localhost','root','');
```

```
$db = mysqli_select_db($conn,'products');
```

This two line of code is executed. In \$conn the three parameters represent localhost, then root, the localhost password. In \$db products parameter represent the database name.

For Query Run:

```
$query_run = mysqli_query($conn,$query);
```

This code has two parameter. One is for \$conn which is connection between the database and php file and another is \$query which execute the MySQL query that provided into it.

Update Product: At first User will search for a product that exists or not. If the data found in database then it will show the product details where product_name matches in database.

Read Product: Here just a simple query will be executed to fetch the data from the database of a particular table.

```
$query = "SELECT * FROM products_table";
```

This query will fetch the whole table information to present in the webpage.

To show the table data this

```
$row = mysqli_fetch_array($query_run)
```

code executed in a while loop because it will executed till the last row of table.

Delete Product: In delete_product.php a simple delete query has been executed to delete the product information. This query will delete the row where the product_name matches with the product_name column value. This php file will delete the information as product name matches with the database.

Read Single Product: In this php file a query will search by the given product name via user in the database. Once it matches with the database then it will show all the exact product information fetched from the database.

CART CRUD

In Cart (CREATE, READ, UPDATE, DELETE) I have used the same process as the products process. I have used `cart_table` and `cart_quantity` table to store the information of cart.

Add Cart: In this part I have made a form to collect product name and product quantity then a submit button to proceed next. It will redirect to `add_cart_connect.php`. At first I have connected to the database then checking connection. If connected, then inserting the form values to the database. HERE a different mechanism is working. I am fetching the data of `products_table` into `cart_table`. Where the `product_name` is matched with the `product_table's product_name`. Here is another query that is for quantity. I am storing the quantity value in `cart_quantity` table.

Read Cart: Read cart is presenting the data inserted into `cart_table` and `cart_quantity` data table.

Update Cart: Update product is updating the product values to the database.

Delete Cart: Delete product is deleting all the information stored in the database according to product name.

I have built this three process same as before. I am collecting the `product_name` and executing the MySQL query accordingly.

Login History

Add Login History: In this part I have built a login button. When a user gives a username and password and press login button then this process will work. I just need the click on login button and the program will automatically collect the information of the user device, location and time. At first I have set the time zone of Dhaka. Then there is a `date()` function which is providing me the time and date of login and `$_SERVER['HTTP_USER_AGENT']` is providing me the logged in device information. But the location information has a bit lengthy process. I have used a JSON API link to collect the location of the user. This link is providing me all the location information but I am just collecting City, Country and ZIP code. At last I am storing the `$dateAndtime`, `$device`, `$location` in the `location_history` table of the database.

Read Login: Read login is representing all the data inserted in the database table of `login_history`

Delete Login: In this, a login query is working which will delete the data where device matches with `login_device` column of the `login_history` data table.

Details about All variables and Parameters:

FB Authentication:

`$_SESSION['access_token']`: This is for the access token getting from facebook

`$user->getField('name')`: IT is actually getting the User name of the logged in person

`require "vendor/autoload.php"`: This file was loaded after installing COMPOSER in my cmd

`$fb=new Facebook\Facebook` : Here I have provided the app id, app secret, and default graph version which I got from the developers.facebook.com

`$helper=$fb->getRedirectLoginHelper()` : It is a helper which will proceed further after login

`$_SESSION['access_token'] = (string)$accessToken` : Here I am changing the variable type as string then assigning it with `$_SESSION['access_token']` this parameter

`echo $e->getTraceAsString()` : This will print the exception as string

`$result = $fb->get('/me?locale=en_US&fields=name,email')` : This is providing the user's name and email

Google Authentication:

`function onSignIn(googleUser)` : This function is assigning the person who logged in

`var profile = googleUser.getBasicProfile()` : In profile variable I have assigned the Basic profile details of the logged in person

`var id_token = googleUser.getAuthResponse().id_token` : This is for the token which is provided by the google

`document.querySelector('#content').innerText = googleUser.getBasicProfile().getName()` : This is to represent the name of the user

`$product_name = $_POST['product_name']` : It is assigning the value which is taken from the html form as a POST method

Products:

`$conn = new mysqli('localhost','root','','products')` : This is the connection of database to the .php file. Localhost, root, password, database name are given as parameter

`$stmt->close()` : Closing the statement

`$conn->close()`: Closing the database connection

`$query = "SELECT * FROM products_table where product_name='$product_name' "` : This is a query variable assigning a MySQL query to it

`$query_run = mysqli_query($conn,$query)` : Executing the mySQL query

`while($row = mysqli_fetch_array($query_run))` : Fetching the all the data table information till the last row

`<input type="text" name="product_name" value="<?php echo $row['product_name'] ?>"/>` : This is a form format where I am visualizing the database's particular value

Cart CRUD: Same as products but in a customer view

Login History:

`<input type="submit" name="add_time" value="LOG IN">` : Just a login button

`$timezone = date_default_timezone_set("Asia/Dhaka")` : Setting the timezone of Dhaka

`$dateAndTime = date("Y-m-d H:i:s")` : Getting the time as Year, month, date, space, Hour, minute, second through the date function

`$device = $_SERVER['HTTP_USER_AGENT']` : Getting the device information of the user through this parameter

`$location = $city.' '.$country.' '.$zip` : Joining three values city, country, zip in one variable to store in a perfect combination of location to the database