**Advanced Higher Computing Project – leBaie**

**2023/2024**

AH Project Analysis

I will develop an online storefront called "leBaie".  The website will allow registered users to list their own items/products to the site and be able to view and edit their details (profile). Furthermore "leBaie" should allow all users to browse the website and view products listed by others (Each item listed will have a picture price and name associated with it, which is then stored in the database). Registered Users and Guests should also be able to add available products to their basket. Consumers should be able to easily access their basket (stored in a MySQL database), search for products either by search or browsing in the appropriate categories and see the chosen products of the month. When users first enter the website, they will be on the landing page which presents them with options to sign in, register or seamlessly browse the website as a guest and add or remove items from their basket as they see fit. Registered Users will be able to list their own products on the site as well, products listed must have an image, product name and the category associated with the item being listed (this will be done by storing the file pathway in the SQL database).

Due to my project meeting the Advanced Higher Computing Criteria, it is a suitable project which consists of easily managed inputs from users- making it simple to validate inputs entered by users. My project will authenticate all login details entered by users by comparing the entered details to the ones stored in the database, update users’ profiles inside database if they choose to change their details, insert items added to basket into the database, I will also utilise session variables across the site and allow users to delete items from their basket (which then deletes it from the database) this will be done via integrating the SQL databases using PHP.

Constraints

 The technical, economic, legal, and time constraints that apply to this project include:

* MySQL database that is used to store products listed on website, customer queries, user’s baskets and user login details.
* To access the database using MySQL, I will use Windows 11 Operating System along with Mac OS.
* To run the database integrated with the webserver I will use xampp control panel.
* There will be no costs involved in the development of the project as I am using software that is fully licensed and available in school for free.
* Notepad ++ 8.5.7 will be used to develop the site as this is the program, I feel the most confident developing with
* The project must be completed by the deadline of the end of February.

Project Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stage | Task | Duration (Days) | Possible Start Date | End Date |
| ANALYSIS  **6 DAYS** | Description of Project | 3 | 26/09/23 | 29/09/23 |
| UML Diagram | 1 | 01/10/23 | 02/10/23 |
| Requirement Spec | 1 | 02/10/23 | 03/10/23 |
| Project Plan | 1 | 04/10/23 | 05/10/23 |
| DESIGN  **12 DAYS** | AH Concepts | 1 | 06/10/23 | 07/10/23 |
| Design Data Structure | 3 | 09/10/23 | 12/10/23 |
| Integrate Project | 2 | 13/10/23 | 15/10/23 |
| Ensure program is fit for purpose | 1 | 14/10/23 | 15/10/23 |
| User Interface | 5 | 16/10/23 | 21/10/23 |
| IMPLEMENTATION  **35 DAYS** | Ensure Program Matches Design | 8 | 22/10/23 | 30/10/23 |
| Integration | 10 | 01/11/23 | 11/11/23 |
| User Interface | 9 | 12/11/23 | 21/11/23 |
| New Skills | 5 | 21/11/23 | 26/11/23 |
| Ongoing Testing | 3 | 26/11/23 | 29/11/23 |
| TESTING  **9 DAYS** | Test Plan | 4 | 01/12/23 | 05/12/23 |
| Requirements Testing | 3 | 07/12/23 | 10/12/23 |
| Test Cases | 2 | 11/12/23 | 13/12/23 |
| EVALUATION  **5 DAYS** | Fitness For Purpose | 2 | 10/01/24 | 12/01/24 |
| Maintainability | 2 | 14/01/24 | 16/01/24 |
| Robustness | 1 | 17/01/24 | 18/01/24 |

**UML diagram**

**A diagram of a diagram

Description automatically generated**

Requirement Spec

For my project to be able to fulfil its requirements, it must meet several criteria:

My project will comply with GDPR as it does publicly share user’s information and it’s securely stored in a SQL database. The form submissions containing sensitive data is shared using the POST method as it does not display users’ info in URL unlike the GET method.

Registering an Account Requirements

Users of my system will be able to create an account if they enter a unique username, first name, surname, their date of birth and a password for their account. The registration functionality will include:

1. The registration page will allow the user to enter

a) A unique username

b) a valid forename and surname

c) a valid date of birth

c) A password that must be re-entered

2. The user will then press the “registration” button, which will check that the username, forename, surname and password they have entered is valid by:

a) Checking that no field is empty

b) Checking that the password is within 20 characters and the username is within 10, date of birth is before 31st December 2013 and that forename and surname are less than 20 characters.

c) ensure password and confirm password fields match via JavaScript function

d) Importing stored usernames from the SQL database table “registration”

e) Checking that the username does not already exist in the database table

3. If the user passes this validation:

a) The User will be redirected to the home landing page for logged in users and their details are inserted into the database table “registration”

Login System Requirements

Login System Requirements Users of my system will be able to create a unique username-password combination to access the full functionalities of the site. The login functionality will include:

1. The Login page will allow the user to enter

a) A valid username

b) A valid password

2. The user will then press the “Login” button, which will check that the username and password they have entered is valid by:

a) Checking that neither field is blank

b) Checking that the password is within 20 characters and the username is within 10

c) Importing stored usernames and passwords from the SQL database table “registration”

d) Checking that the username and password combination exists in the table

3. If the user passes this validation:

a) The User will be redirected to the home landing page for logged in users with a popup stating they have successfully signed in/registered. Session variable will also be set.

Listing a product requirement

Signed in Users of my system will be able to list a product on the website when a Product Name, Product Category, Price and Image are entered into the form. The listing functionality will include:

1. The listing page will allow the user to enter

a) A valid product names

b) a product category from a drop-down list

c) a valid price

c) An image is attached

2. The user will then press the “create listing” button, which will check that the product name, product category, price and image they have entered/selected is valid by:

a) Checking that no field is empty

b) Checking that the product name is within 20 characters, a category is selected, price within 0 and 100,000 is entered.

3. If the user passes this validation:

a) The User will be redirected to the home landing page for logged in users that displays a popup confirming listing was successful and their listing is inserted into the database table “products”. Session variable will also be set.

Editing profile requirements

Signed in Users of my system will be able edit their profile on the website when the edit button on the profile page is pressed. forename, surname, date of birth and password are the values of the form elements which can then be edited by the user. The listing functionality will include:

1. The registration page will allow the user to enter

a) a valid new forename and surname

b) a valid new Date of Birth

c) a valid new date of birth

c) A valid new password

2. The user will then press the “save changes” button, which will check that the forename, surname and password they have entered is valid by:

a) Checking that no field is empty

b) Checking that the password is within 20, date of birth is before 31st December 2013 and that forename and surname are less than 20 characters.

3. If the user passes this validation:

a) The User will be redirected to profile page and their details are updated in the database table “registration” and the updated details will be displayed

Contact Us requirements

Users of my system will be able submit queries/issues on the website form on the contact us page when the submit button on the contact us page is pressed. username, email, subject and descriptions are the form elements which can then be edited by the user. The contact us functionality will include:

1. The contact us page will allow the user to enter

a) a valid username (not required)

b) an email (required)

c) subject (required)

c) a description (required)

2. The user will then press the “save changes” button, which will check that the email, subject and description they have entered is valid by:

a) Checking that no fields mentioned above is empty

b) Checking that the username is within 10 characters (if entered), email within 30 characters and that subject is within 250 characters (validated in SQL table).

3. If the user passes this validation:

a) The User will be redirected to the logged in-home page with a popup stating that the query is submitted if the username entered in the form matches one in the registration table, if the username field was left empty it will redirect the user to the homepage for guests which will display a popup stating that query was successfully submitted. The form data will also be inserted into the database table “contactUs” so that the query can be viewed at a later date so user can get assistance.

Basket requirements (registered users and guests)

Users of my system will be able to view items they added to their basket in previous pages, which is then displayed in the basket page. The contact us functionality will include:

1. The basket page will allow users to

a) remove items from basket

2) display user’s basket

3) checkout

1a. if the user presses the “remove from basket” button, it will proceed to remove that item from the user’s basket in either the “basket” table or “guestBasket” table:

a) Select item from “basket” or “guestBasket” tables that corresponds to the product. Then the database will be queried, and then the item will be removed from the database table. The user will then be redirected back to the basket page and it will display the rest of the basket or state that its empty

2. The user’s basket will be displayed using php and retrieving products added to basket from the SQL table “guestBasket” for unregistered users and “basket” for signed in users:

a) SELECT items in “basket” table that correspond with have the same username as the session variable started after the user signed in/registered (FOR SIGNED IN USERS ONLY)

b) QUERY database (SIGNED IN USER ONLY)

C) Display results in table that were queried (SIGNED IN USER ONLY)

d) SELECT all products in the table “guestBasket” (GUEST ONLY)

e) QUERY database (GUEST ONLY)

f) Display results in table queried from database table (GUEST ONLY)

3. CHECKOUT

Due to budget limitations and time constraints I was unable to implement this feature.

Clothing, Electronics, Vehicles and Game page requirements (registered users and guests)

Users of my system will be able to browse different categories for products that have been listed (Can access hyperlinks which will take them to pages displaying these via using the navigation bar). The pages listed above functionality will include:

1. The pages for clothing, electronics, vehicles and game will allow users to:
2. Access navigation bar – contains links to home, clothing, electronics, vehicles and game pages

2. display products from appropriate categories

3. allow users to add items to basket

1a) Clothing, Electronics, Vehicles and Game pages will feature a navigation bar which will contain hyperlinks to the Home, Clothing, Electronics, Vehicles and Game pages clothing

2) The products for the appropriate category for the page will be displayed using php and retrieving products from the “products” table, which will then products that align with the page of the category being viewed.

b) QUERY database

c) Display results from “products” where appropriate conditions are met

3) the users will be able to add items that are displayed into their basket when “add to basket” button is clicked

a) assign value of the product name to the product name of the row being displayed

b) insert the value of the product name into the database table “guestBasket” for users who haven’t signed in and “basket” table for users that are logged in/registered

Home page requirements (Registered Users and Guests)

Users of my system will be able to browse products of the months that have been listed. They can access hyperlinks which will take them to different product category pages displaying these via using the navigation bar, login or register or proceed to add products to basket and browse website as guest. The home functionality will include:

1. The pages for the home pages will allow users to:
2. Access navigation bar – contains links to home, clothing, electronics, vehicles and game pages

2. display products of the month

3. allow users to add items to basket

1a) Home pages will feature a navigation bar which will contain hyperlinks to the Home, Clothing, Electronics, Vehicles and Game pages clothing

2) The products of the month will be displayed using php and retrieving products via a select statement which looks for names of the products that were chosen as the products of the month from the “products” table.

b) QUERY database

c) Display results from “products” where appropriate conditions are met

3) the users will be able to add items that are displayed into their basket when “add to basket” button is clicked

a) assign value of the product name to the product name of the row being displayed

b) insert the value of the product name into the database table “guestBasket” for users who haven’t signed in and “basket” table for users that are logged in/registered

DESIGN

USER INTERFACES

**A screen shot of a login page

Description automatically generated**Login Page

A screen shot of a computer screen

Description automatically generatedRegistration Page

About us page

A screen shot of a computer

Description automatically generated

Clothing Page

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generatedElectronics Page

Games Page

A screenshot of a computer

Description automatically generated

Listing Page

A screenshot of a computer

Description automatically generated

Failed Search

A graph with text and arrows

Description automatically generated

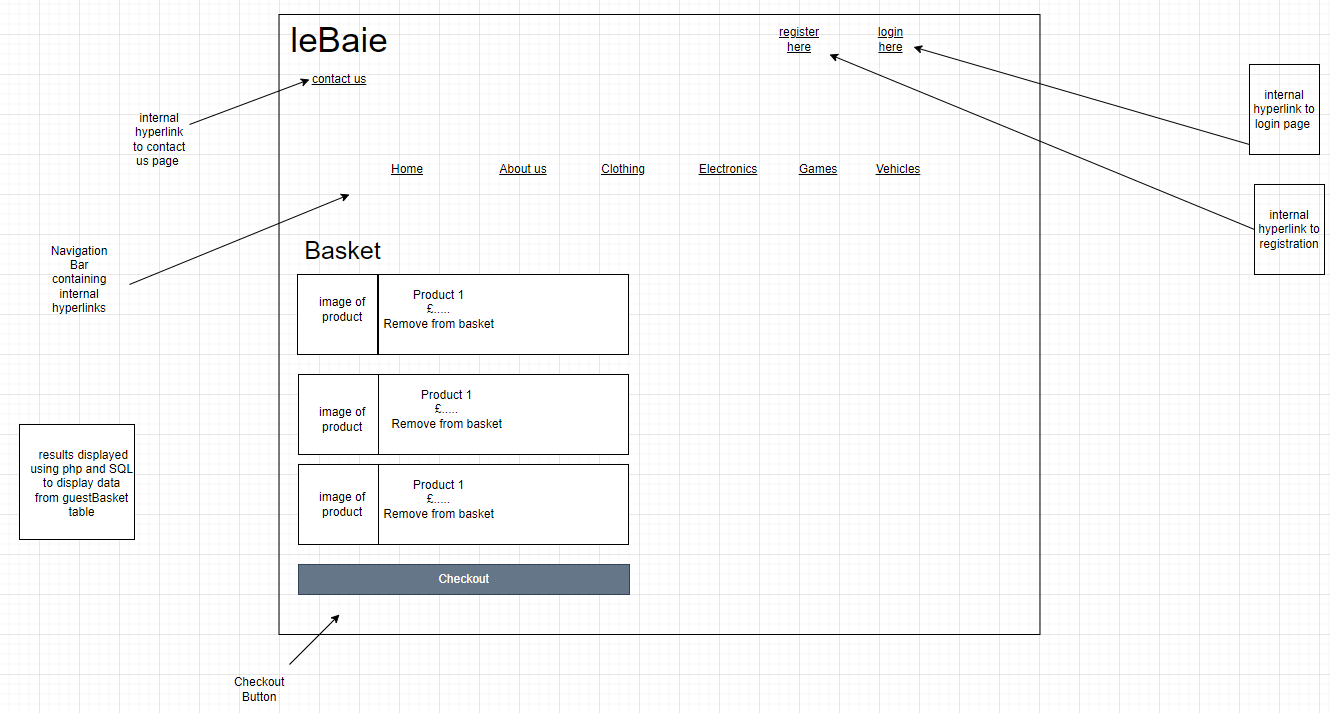
Vehicles Page

A screenshot of a computer

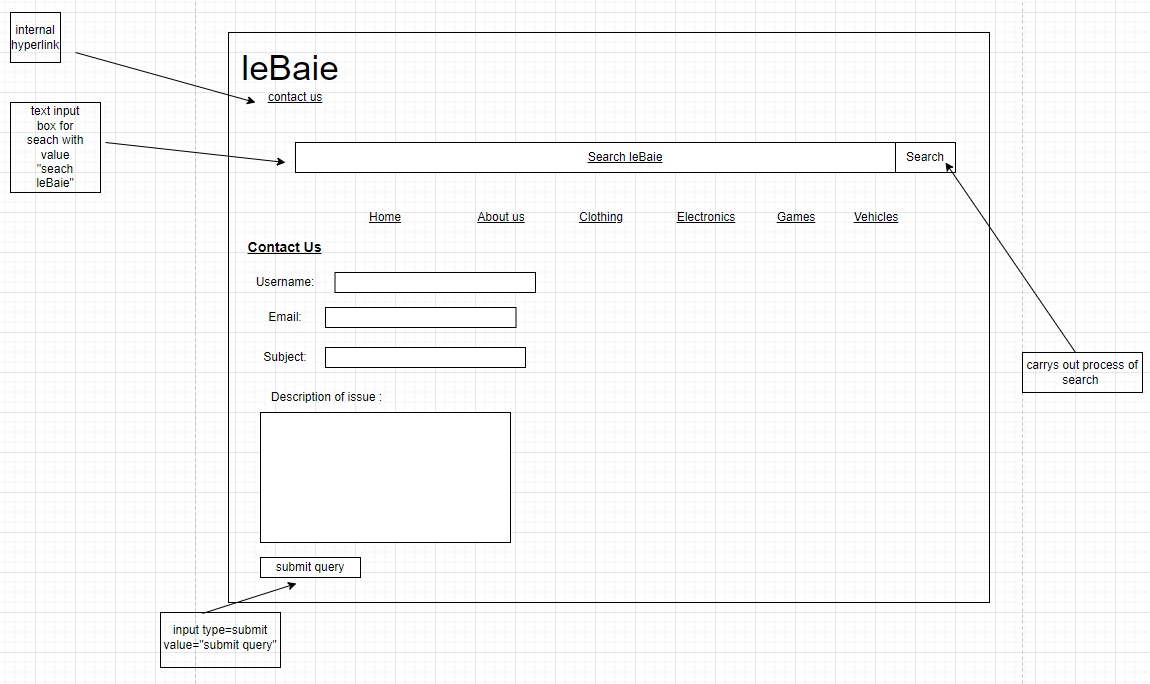
Description automatically generated

Unregistered User Interfaces

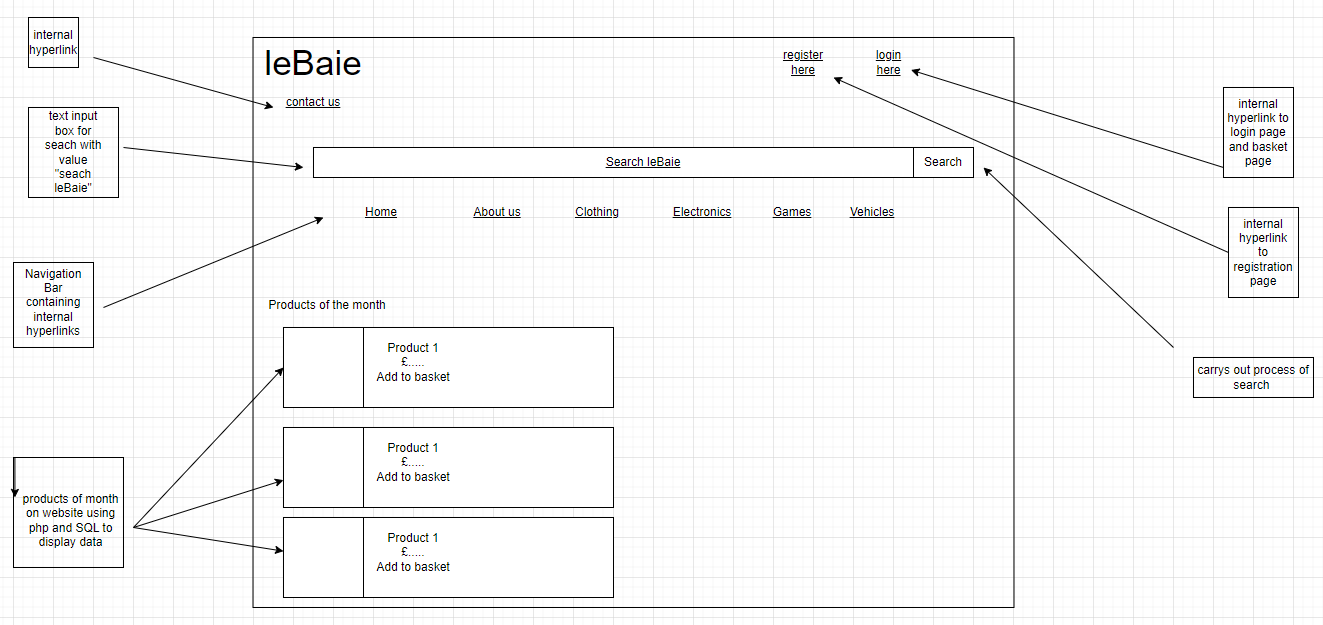
Basket Page (Guest)



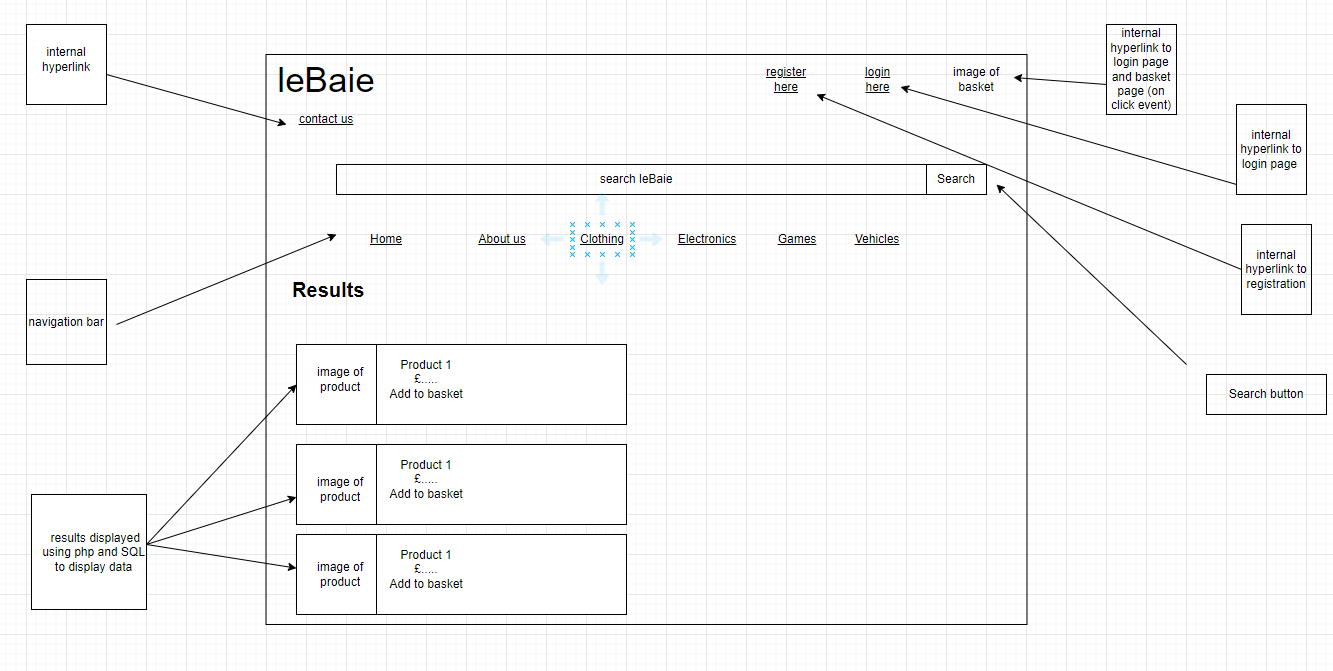
Contact Us Page (Guest)



Home Page (Guest)



Results page (guest)



Basket page when empty (Guest)

A screen shot of a graph

Description automatically generated

Basket when empty (registered user)

A white grid with black text

Description automatically generated

Basket page (registered user)

A screen shot of a computer

Description automatically generated

Contact us page (registered user)

A screenshot of a computer

Description automatically generated

Profile page (registered user)

A screen shot of a computer screen

Description automatically generated

Edited profile form page (registered user)

A diagram with text and arrows

Description automatically generated with medium confidence

Home page (for registered users that have logged in/signed up)

A screenshot of a computer

Description automatically generated

Results page (for registered users)

A screen shot of a computer

Description automatically generated

Pseudocode

Registration Page

1. RECEIVE HTML FORM VALUES USING POST AND ASSIGN TO PHP VARIABLES
2. SETUP SESSION VARIABLES FOR USERNAME, FIRSTNAME, SURNAME, DATE OF BIRTH AND USERS’ PASSWORD ASSIGNED VIA PHP VARIABLES
3. ASSIGN CONNECTION DETAILS TO PHP VARIABLES
4. CONNECT TO DATABASE
5. IF ERROR CONNECTING
6. DISPLAY ERROR MESSAGE
7. END IF
8. QUERY DB TO SELECT USERNAMES FROM REGISTRATION TABLE THAT MATCH PHP VARIABLE THAT WAS ASSIGNED TO USERNAME
9. IF NUMBER OF RESULTS>0
10. DISPLAY ERROR SAYING USERNAME ALREADY EXISTS AND REDIRECT USER TO FAILED REGISTRATION PAGE
11. ELSE
12. INSERT USERS’ DETAILS INTO REGISTRATION TABLE
13. IF QUERY CHECKING INSERT AND CONNECTION IS SUCCESSFUL
14. DISPLAY SUCCESS MESSAGE
15. END IF
16. REDIRECT TO REGISTERED HOME PAGE

Login Process

1. START SESSION
2. RECEIVE HTML FORM VALUES VIA POST AND ASSIGN TO PHP VARIABLES
3. SETUP SERVER DETAILS USING PHP VARIABLES
4. SET VALID TO FALSE
5. CONNECT TO SERVER
6. IF CONNECTION FAILED DISPLAY ERROR MESSAGE
7. END IF
8. SELECT USERNAME, FIRSTNAME, SURNAME, DOB AND PASSWORD FROM DATABASE TABLE REGISTRATION WHERE THE USERNAME IN TABLE MATCHES PHP VARIABLE ASSIGNED FROM FORM
9. QUERY DATABASE AND ASSIGN RESULTS TO PHP VARIABLE
10. IF NUMBER OF RESULTS>0
11. IF PASSWORD STORED IN DATABASE TABLE MATCHES THE PHP VARIABLE USING FORM VALUE
12. SET VALID TO TRUE
13. SETUP SESSION VARIABLES FROM THE DATABASE ROW RESULT
14. END IF
15. END IF
16. CLOSE CONNECTION TO DATABASE
17. IF VALID IS TRUE
18. REDIRECT USER TO REGISTERED HOME PAGE
19. ELSE
20. REDIRECT USER TO FAILED LOGIN PAGE
21. END IF

Listing Process

1. START SESSION
2. RECEIVE HTML FORM VALUES VIA GET AND ASSIGN TO PHP VARIABLES
3. SETUP SERVER DETAILS USING PHP VARIABLES
4. CONNECT TO SERVER
5. IF CONNECTION FAILED
6. DISPLAY ERROR MESSAGE
7. END IF
8. INSERT HTML FORM VALUES THAT WERE ASSIGNED TO PHP VARIABLES INTO PRODUCTS TABLE
9. IF QUERY IS SUCCESSFUL
10. DISPLAY MESSAGE STATING THAT PRODUCT IS LISTED
11. REDIRECT USER BACK TO REGISTERED HOME PAGE
12. ELSE
13. DISPLAY ERROR MESSAGE
14. END IF

CONTACT US PROCESS

1. START NEW SESSION
2. RECEIVE HTML FORM VALUES VIA GET
3. ASSIGN CONNECTION DETAILS TO PHP VARIABLES
4. IF CONNECTION FAILED
5. DISPLAY ERROR MESSAGE
6. END IF
7. QUERY DATABASE TO SELECT USERNAME FROM DATABASE THAT MATCHES THE PHP VARIABLE HOLDING HTML VALUE
8. ASSIGN RESULTS FOUND TO PHP VARIABLE
9. INSERT VALUES INSIDE PHP VARIABLES FROM HTML VALUES INTO CONTACTUS TABLE
10. IF QUERY WAS SUCCESSFUL
11. REDIRECT USER TO REGISTERED HOME PAGE THAT WILL DISPLAY SUCCESS MESSAGE
12. ELSE
13. REDIRECT USER TO GUEST USERS HOME PAGE THAT WILL DISPLAY SUCCESS MESSAGE
14. END IF

DISPLAYING PROFILE

1. START SESSION
2. ASSIGN SESSION VARIABLES TO PHP VARIABLES
3. ASSIGN CONNECTION DETAILS TO PHP VARIABLES
4. CONNECT TO SERVER
5. IF SESSION VARIABLES ARE SET
6. DISPLAY PROFILE DETAILS USING HTML AND PHP VARIABLES
7. END IF

EDIT PROFILE PROCESS

1. START SESSION
2. ASSIGN SESSION VARIABLES TO PHP VARIABLES
3. ASSIGN CONNECTION DETAILS TO PHP VARIABLES
4. CONNECT TO SERVER
5. IF CONNECTION FAILED
6. DISPLAY ERROR MESSAGE
7. ASSIGN VALUES OF THE FORM ELEMENTS TO PHP VARIABLES

UPDATING PROFILE PROCESS

1. START SESSION
2. RECEIVE HTML FORM VALUES VIA POST
3. ASSIGN CONNECTION DETAILS TO PHP VARIABLES
4. CONNECT TO SERVER
5. IF CONNECTION FAILED
6. DISPLAY ERROR MESSAGE
7. END IF
8. UPDATE THE REGISTRATION TABLE TO THE VALUES CONTAINED INSIDE THE PHP VARIABLES STORING THE FORM VALUES
9. IF QUERY IS SUCCESSFUL
10. DISPLAY ACCOUNT UPDATED MESSAGE
11. END IF
12. ASSIGN SESSION VARIABLES TO THE VALUES OF PHP VARIABLES
13. REDIRECT USER BACK TO PROFILE PAGE
14. CLOSE CONNECTION

LOG OUT PROCESS

* 1. DESTROY SESSION
  2. REDIRECT USER TO HOME PAGE FOR UNREGISTERED USERS

REGISTERED USER DISPLAYING BASKET PROCESS

1. START SESSION
2. ASSIGN PHP VARIABLE TO SESSION VARIABLE
3. ASSIGN CONNECTION DETAILS TO PHP VARIABLES
4. CONNECT TO SERVER
5. IF CONNECTION FAILED
6. DISPLAY ERROR MESSAGE
7. SELECT ROWS FROM BASKET TABLE WHERE THE USERNAME IN TABLE MATCHES THE PHP VARIABLE
8. QUERY DATABASE AND ASSIGN RESULTS TO PHP VARIABLE
9. IF NUMBER OF RESULTS >0
10. REPEAT WHILE DISPLAYING DATA OF CORRESPONDING ROWS USING PHP VARIABLES INSIDE HTML FORM ELEMENT VALUES
11. END REPEAT WHEN ALL SELECTED ROWS HAVE BEEN DISPLAYED
12. ELSE
13. DISPLAY MESSAGE STATING BASKET IS EMPTY
14. END IF
15. CLOSE CONNECTION