Swinburne University of Technology

Faculty of Engineering, Computing and Science

Assignment 2

COS30020 Web Application Development

Semester 2, 2024

DUE ON: Week 12

Assignment Worth: 50% of total marks

Background Description

In this assignment, you are to continue working on Assignment 1 to improve it and to add additional functions to produce a complete system. This includes adding the Admin user type with different functionalities. This assignment will make use of MySQL table creation, MySQL database access from PHP, and PHP sessions.

You are to make a copy of assignment 1 and use it to continue with assignment 2. You are advised to place your functions in another PHP page and include the page when needed. This will make it easier for you to debug your program.

Part 1: Database Table Creation for the Plant Biodiversity project

In this task, you are also expected to:

- Create a database called `PlantBiodiversity` programmatically if it does not exist. Embed the codes in `main.php`. Create **three** tables `under` `PlantBiodiversity` database. Only create tables if they do not currently exist in the database using the command "`CREATE TABLE IF NOT EXISTS`".

1. Create the table named `user_table` to store the user's information. It should have the following fields:
   - `email` –Set this field's properties as '`varchar`', '`size=50`', and '`not null`' and '`primary key`'
   - `first_name` – the first name of the user. Set this field's properties as '`varchar`', '`size=50`', and '`not null`'.
   - `last_name` – the last name of the user. Set this field's properties as '`varchar`', '`size=50`', and '`not null`'.
   - `dob` – the date of birth of the user. Set this field's properties as '`Date' and 'null'`.
   - `gender` – the gender of the user. Set this field's properties as '`varchar`', '`size=6`', and '`not null`'.
   - `contact_number` – Set this field's properties as '`varchar`', '`size=15`', and '`null`'.
   - `hometown` - Set this field's properties as '`varchar`', '`size=50`', and '`not null`'
   - `profile_image` – this field will store the path of the profile image stored in the server. Set this field's properties as '`varchar`', '`size=100`', and '`null`'.

2. Create a table called `account_table` to store the login credential of the users It should have the following fields:
    - `email` – The login name for the account. Set this field’s properties as ‘`varchar`’, ‘`size=50`’, and ‘`not null`’ and ‘`foreign key`’ with reference to the `email` field in the `user_table`. Define constraint for DELETE and UPDATE operations such that any changes to the `email` in the `user_table` will be reflected in this table as well.
        - `password` – the password for the account. Set this field’s properties as ’`varchar`’, ‘`size=255`’, and ‘`not null`’.
        - Type - `type` – the type of the account. It should be either “admin” or “user”. Set this field’s properties as ’`varchar`’, ‘`size=5`’, and ‘`not null`’.

3. Create a table called `plant_table` to store the list of news or events organized by the alumni association (contribution page). It should have the following fields:
    - `id` – a unique 4-digit numeric identifier, e.g. 1023. Set this field’s properties as ‘`integer`, ‘`not null`’, ‘`auto_increment`’ and ‘`primary key`’.
    - `Scientific_Name –` Scientific name of the plant. Set this field’s properties as ‘`varchar`’, ‘`size=50`’, and ‘`not null`’.
    - `Common_Name –` Common name of the plant. Set this field’s properties as ‘`varchar`’, ‘`size=50`’, and ‘`not null`’.
    - `family` – Set this field’s properties as ‘`varchar`’, ‘`size=100`’, and ‘`not null`’.
    - `genus` - Set this field’s properties as ‘`varchar`’, ‘`size=100`’, and ‘`not null`’.
    - `species` - Set this field’s properties as ‘`varchar`’, ‘`size=100`’, and ‘`not null`’.
    - `plants_image` – this field will store the path of the plants image stored in the server. Set this field’s properties as ’`varchar`’, ‘`size=100`’, and ‘`null`’.
    - `description` - this field will store the path of the description file (in pdf) stored in the server. Set this field’s properties as ’`varchar`’, ‘`size=100`’, and ‘`null`’.

Note:

- o Write the codes to populate the `user_table, account_table` and `plant_table` with at least 4 sets of dummy data.
- o Include a record for admin in the `user_table` and `account_table`.
- o Set the admin login credential to:
    - Login: admin@swin.edu.my
    - Password: admin
- o You may populate the table with more sample data. This will be useful in the following tasks.

Part 2: Register and Login Functions

In this part, you are only required to amend the existing pages from assignment 1

Task 2.1 Registration page [`registration.php`]

Before saving the information entered into the database:

- Check for the existence of the record in the database
- Hash the password to defend against password from being compromised
- Save all the information captured on the registration form into the `user_table` and `account_table`

Task 2.2: Log in page [`login.php`]

On form submit,

- validate the password against the hashed password.
- If login credentials are correct, set up the session variable(s) to store the `type` and the `email` from the `account_table` before redirecting it to the main menu page.

Part 3: User Interaction Pages of the project

Task 3.1: Main Menu page [`main_menu.php`]

Amend the main menu from assignment 1 to meet the following requirements:

- User must be logged in and the user type must be "user" in order to access the page, otherwise redirects the user to the login page. All this information can be obtained from the Session variables stored in the login page.

This part allows navigation to all respective PHP pages which involves Add, Edit, and Delete records to and from the tables

Task 3.2 Amend codes to retrieve from MySQL database

Amend the codes in the existing pages from assignment 1 to retrieve the data from the database. Write the codes to check that the user must be logged in in order to access this page. This includes the following pages:

- **View Plant detail Page** (`plant_detail.php`)
- **View profile page** (`profile.php`)

Task 3.3 Update profile page (`update_profile.php`)

Add functions to the existing page from assignment 1 to allow the user to change the profile photo and to upload the resume. Refer to the screenshot below.

In this page:

- User must be logged in as user in order to access the page
- Extract the user's information from the database and display it on the form when the page is loaded
- Perform input validation and display appropriate error messages upon failed validation
- Save the updated information on the form into the `user_table`. ** Remember to ensure that the `account_table` will be updated if there is a change in the email address
- Should have the section for the user to
  - Upload the profile photo from the local drive onto the server. Write the codes to only allow files with jpg, jpeg and png extensions. Limit the file size to 5MB. Store the profile photo in a folder called 'profile_images'.
  - Upload resume from the local drive onto the server. Write the codes to only allow files with pdf extension. Limit the file size to 7MB. Store the resume in a folder called 'resume'. **Remember to store the path to the files in the `user_table`.

Task 3.4: Contribution page [`contribute.php`]

In this page, user will upload photos and relevant information to store into `plant_table` .

Requirements:

- User must be logged in and the user type must be "user" in order to access the page, otherwise display a page which leads to the login page.
- The information will store into `plant_table.`
  - Upload description file from the local drive onto the server. Write the codes to only allow files with pdf extension. Limit the file size to 7MB. Store description file in a folder called 'plants_description'. Remember to store the path to the files in `plant_table.`
- When the uploading is successful, a simple message should be displayed, to notify the user that the uploading is done/successful.

Part 4: Admin Interaction Pages of the the project

Task 4.1: Admin Main Menu [**`main_menu_admin.php`**]

Create a main menu page with your own design. It will contain the functions for the admin account. Load this page when the user logins with the admin credentials. The main menu for the admin will have similar design as the normal user but with darker shade of blue for the navigation bar.

Requirements:

- User must be logged in and the user type must be "admin" in order to access the page, otherwise display a page which leads to the login page.
- Main menu page should have a darker navigation bar, a home button, a logout button and 2 options for the admin to choose from.
- Links/buttons are working as expected. Link the buttons to the respective pages

Task 4.2: Manage Users' Account Page [`manage_accounts.php]`

This page allows the admin to manage user's account such as, add, edit, and delete.

Once the user is deleted, remove the user's records in the `user_table`. Ensure that the record is removed from the `account_table` as well.

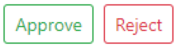
Task 4.3: Manage Plants Page `[manage_plants.php]`

This page allows the admin to manage the plants record such as delete, edit and add. Another feature will be added where it allows the admin to approve or reject the plants data.

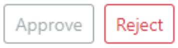
Requirements:

- User must be logged in and the user type must be "admin" in order to access the page, otherwise display a page which leads to the login page
- Display all the plants data on a table with the option to accept/reject the account and the option to delete the record.
- For the approval feature:
  - The plant record status will be pending until the admin has approved of the plant record. The admin may choose to reject the plant record as well.

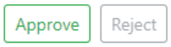
- If the status is pending, display the button for the admin to approve or reject.



- If the status is approved, disable the approve button and enable the reject button.



- If the status is reject, disable the reject button and enable the approve button



Task 4.4: About This Assignment Page `[about.php]`
Create a report webpage for presenting what you have done based on the question provided.

Requirements:
Page contains:

- Listing of answers to the following questions in bullet point.
  - What tasks have you not attempted or not completed?
  - Which parts did you have trouble with?
  - What would you like to do better next time?
  - What extension features/extra challenges have you done, or attempted, when creating the site?
  - Link to the video presentation
- Listing of links to the following pages
  - Home page - `index.php`

Part 5: Extension tasks

Task 5.1: Identify page [`identify.php`**]**

You are to create this page for user to identify the plant information based on the photo uploaded.

Output:

- The output will display the scientific plant name, common name and the photos of herbarium specimens.
- Description in pdf for user to download.

Requirements:

- The information will retrieve from `plant_table.`
- When the identify process is done, list out the plant information if the process is done successfully, otherwise display the message “Plant data not available”.

Submission & Assessment Process:

Submission & Assessment Process:

Create your site structure as described below:

assing2/	you must have this folder – case sensitive!
*.php	contains all the PHP files
profile_images/	folder for any profile
images/	folder for any images used for page content
css/	folder for CSS style sheet rules for your stylesheet (option)
plants_description/	folder to store Plants description file

Notes:

- All links to your files, including data files, should be relative.
    - Links must **not** be absolute, as these links will be broken when files are transferred for marking.
    - If links are broken, or an assignment does not run when tested, the result will be a fail for this assignment.
- Additional files are allowed depending on the adopted solution.

Assignment Submission:

The assignment should be submitted as an individual work to Canvas.

Please ZIP all your **source code files** and upload it through Canvas. Upload the video to YouTube (as unlisted) showcasing all functionalities of the website. Video should not be longer than 7 minutes. You can submit more than once (as long as it is before the due date) to Canvas as the new submission would overwrite the previous one.

** Make sure to test your website before submission.*
** Full marks for each task will not be awarded if there are errors or issues in the way the code is implemented and/or if usability is not considered properly.*
** If your assignment cannot run, your result will be 0 marks for this assignment.*