RZA Project Style Guide

File and Directory Structure

Structure:

- `db` - This is where a record of all SQL queries used to construct a database will go. To obtain a `.sql` file for importing, you typically export in phpmyadmin, which will create the SQL dump.
- `documents` - This is where anything supplementary to the project goes, such as important documentation and designs, as well as the project brief and information to be communicated by the client.
- `src` - This is where the source code of the application goes, such as `.php`, `.html`, `.css`, `.js`, and `.ini` files go. Anything that is used by the implementation of the website must go in here as well.
  - `src/.` - This is the root of the `src` directory. Unique pages that the user will visit go here.
  - `src/images` - Images and assets used by the website
  - `src/include` - Anything extra included that is utilised by the website as source code must go here, such as CSS styles, JavaScript, reusable HTML and PHP code, and settings.
    - `src/include/utils.php` - PHP functions shared across multiple pages/script files must go here in order to be used globally.

Important Files

There are three different files you will have to include for each page that you create:

- `utils.php`
- `base.php` (includes `header.php`)
- `footer.php`

Page Naming

You need to add a constant for the `<title>` element as follows:

```
const PAGE_TITLE = "Example Template";
```

If you include `base.php` without doing this, it will lead to errors about undefined constants.

Format

Your `.php` files (for when we start writing back-end code) should look like this:

```php
<?php

// This connects to the database and starts a session.
// The database connection is accessible with the global variable $pdo.
// Database settings are changed in include/db_settings.ini
// It also provides several utility functions that wrap common functionality
// so it can be used across different pages.
require_once "include/utils.php";

// Here is where you will put PHP code such as helper functions, SQL queries,
request/form handling,
// session handling, file handling, validation etc.
//
// For example:
//
// function helper_function() {
//     do_pdo_thing();
//
//     return fetch();
// }
// if (isset($_POST["submit"])) {
//     data = helper_function();
// }

// This will decide the name of the <title> element in `base.php` (name assigned
to the tab).
// WARNING - if it is not set, this will cause an error of undefined constant.
This is *intentional behaviour*
// as the page should not load if the title is set incorrectly so the programmer
does not ignore the problem.
const PAGE_TITLE = "Example Template";
include_once "include/base.php";
// base.php already includes all of the HTML boilerplate, such as <head>, <body>,
<title>, and DOCTYPE
// Any extra <link> elements the programmer wishes to add should go in base.php
(see comments there)

?>

←!— Place your HTML content here —→
<h1><?=PAGE_TITLE?></h1>
```

```
<p>This is an example template for what PHP pages/scripts should be formatted
like.</p>
<p>
Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatibus nobis ipsam
fugit a repellat natus id ea,
aliquam tempora voluptatem temporibus esse?
Aut id ipsam nemo quisquam quidem deserunt eligendi.
</p>
<button type="button">Lorem ipsum dolor sit amet consectetur adipisicing elit.
</button>

<!— Don't forget to include the footer at the end so the closing tags can work
with the remaining base content —>
<!— Footer already includes any external JavaScript or JS CDNs (e.g. Bootstrap,
Popper) —>
<?php include_once "include/footer.php";
```

See `src/example_template.php` for a template.

You need to write your processing logic before the `PAGE_TITLE` and `base.php` are declared, but after the `utils.php` script. This will ensure that the application functions correctly.

Code Style

Use the `redirect($url)` function from the `include/utils.php` module. **DO NOT** use `header()`. This is to mitigate a problem where relative paths cause problems with the web server redirecting the user.

Please use the `<?=$var?>` syntax rather than using `<?php echo $var ?>` when interpolating PHP variables into HTML content. Make sure that the variable actually exist before doing this as well.

Follow all standard SQL, PHP, HTML conventions when writing code.

SQL `SELECT` queries must **not** use the wildcard (`*`) operator to grab all columns. Reference only the columns you need as this is more conventional, safe, and readable. Use PDO's bound parameters instead of string interpolation where possible. Try to use helper functions for PDO and associated validation tasks.

Database Settings

Database settings, such as the address it is hosted at (through `$dsn`), database driver, username, and password, can all be changed at `src/include/db_settings.ini`.

Session Handling

You do not need to start a session in PHP as this is already started in `utils.php`. However, the code will not error if you try to start one anyway, but this would be redundant code.

The user's `username` is typically stored in `$_SESSION["user"]` and their role is stored in `$_SESSION["role"]`. `role` is either equal to `"customer"` or `"admin"`, so make sure to handle this in your PHP code.

Adding CSS and JS

CSS can be added either inline with the `<style>` tag or with `<link>` elements and separate CSS as seen below. Anything included must go in `base.php` as seen below.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title><?=PAGE_TITLE?></title>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.7.2/css/all.min.css" integrity="sha512-
Evv84Mr4kqVGRNSgIGL/F/aIDqQb7xQ2vcrdIwxfjThSH8CSR7PBEakCr51Ck+w+/U6swU2Im1vVX0SVk
9ABhg==" crossorigin="anonymous" referrerpolicy="no-referrer" />
        <!-- Add extra <link> elements here, such as CSS -->
        <!-- For example: -->

        <!-- <link rel="stylesheet" href="style.css"> -->

        <style>
            /* Add inline CSS here */
            .carousel-index-image {
                min-height:25vh;
                width: 100%;
            }
        </style>
    </head>
    <body>
```

```
        <?php include "header.php" ?>
        <main>
```

Note that HTML boilerplate is already included in `base.php`, as well as `<link>` elements for Bootstrap and FontAwesome. This means that you do not need to include these when writing front-end code. This also applies to `<script>` elements for Bootstrap and Popper JS as these are included in the `footer.php`. For example:

```
        </main>

        <footer>
            A footer
        </footer>

        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js
" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
    </body>
</html>
```