File: **index.php**

Description: This PHP file serves as the main page for a restaurant menu website. It includes a navigation bar, a menu section with various categories, and a dynamic display of dishes based on user-selected filters. The code utilizes PHP sessions for user authentication and a MySQL database for storing and retrieving dish information.

Contents:

1. Error Reporting and Session Initialization:

- PHP error reporting is set to suppress errors.

- Session is started, and the session garbage collection probability is adjusted.

2. HTML Head Section:

- Sets up the HTML document with metadata, title, and links to external stylesheets and favicon.

3. Navigation Bar:

- Displays navigation links based on the user's login status.

- Options include Registration, Sign In, Order, Log Out, and Change Profile.

4. Article Section:

- Categories: Displays a form with buttons to filter dishes by categories such as "All," "Top Sale," and "By Cost."

- Retrieves and displays dishes from the database based on the user's selection.

- Utilizes a MySQL database connection to fetch and present dish details.

5. Footer Section:

- Displays the author's name.

File: **entry.php**

Description: This PHP file serves as the entry form for user authentication on a restaurant menu website. It allows users to sign in using their email, phone number, and password. The code checks the entered credentials against the database and grants access to the main page or dashboard based on the user's role (Manager or non-Manager).

Contents:

1. Session Initialization and Error Reporting:

- Session is started, and error reporting is set to suppress errors.

- The session save path is configured to 'sessions', and the session garbage collection probability is adjusted.

2. Database Connection and Variable Initialization:

- Establishes a connection to the database using the include/db.php file.

- Initializes variables for user information.

3. Form Processing:

- Defines a function (getPosts) to retrieve values from the login form.

- Handles form submission and performs a search query based on entered email or phone number.

- Uses prepared statements for enhanced security against SQL injection.

- Validates the entered password and redirects users to the appropriate page (menu.php for Managers, index.php for others).

4. HTML Document Structure:

- Sets up the HTML document with a title, viewport meta tag, and links to external stylesheets and favicon.

5. Sign-in Form:

- Displays a form for users to enter their email, phone number, and password.

- Submits the form to validate and process the login request.

6. Sign-up Link:

- Provides a link for users to navigate to the registration page if they don't have an account.

File: **registration.php**

Description: This PHP file provides a user registration form for the restaurant menu website. It allows users to register by providing their name, email, phone number, password, and password confirmation. The code validates the entered information, checks for existing records in the database, and inserts a new user record if the input is valid.

Contents:

1. Namespace and Error Reporting:

- Uses the 'example' namespace.

- Sets up error reporting to suppress errors.

2. Database Connection and Variable Initialization:

- Establishes a connection to the database using the include/db.php file.

- Initializes variables for user information and error messages.

3. Validation Functions:

- Defines functions for validating phone numbers, emails, password confirmation, and passwords.

- Checks for existing records in the database to avoid duplicate entries.

4. Form Processing:

- Handles form submission and calls validation functions to ensure data integrity.

- If the input is valid, inserts a new user record into the 'Employees' table.

- Utilizes password hashing for improved security.

5. HTML Document Structure:

- Sets up the HTML document with a title, links to external stylesheets and favicon.

6. Registration Form:

- Displays a form for users to enter their registration details.

- Submits the form to validate and process the registration request.

7. Sign-in Link:

- Provides a link for users to navigate to the sign-in page if they already have an account.

File: **order.php**

Description: This PHP file represents the order page of a restaurant menu website. It allows authenticated users to view the menu, add items to their order, and place the final order. The code checks if a user is logged in using session information and redirects them to the main page if not logged in.

Contents:

1. Session and Error Reporting:

- Session is started to check user authentication.

- Error reporting is set to suppress errors.

2. HTML Document Structure:

- Sets up the HTML document with metadata, title, links to external stylesheets, and favicon.

3. Main Content:

- Header: Displays the "Order" header.

- Menu Section: Utilizes a JavaScript file (order\_script.js) to dynamically load and display the menu.

- Order Section: Allows users to add items to their order, view the order, and see the total sum.

- Place Order Button: Allows users to finalize and place their order.

4. Footer:

- Displays the author's name.

Note: The code assumes that user authentication is handled elsewhere (such as during login), and it redirects users to the main page if they are not logged in. Ensure that proper security measures are in place, and consider additional validation for user authentication in a production environment. Additionally, the functionality of the order\_script.js file is not provided in this code snippet.

File: **menu.php**

Description: This PHP file serves as the backend API for fetching dishes from the database in JSON format. It is intended to be accessed via AJAX requests to dynamically update the menu on the client side.

Contents:

1. Database Connection:

- Establishes a connection to the database using the include/db.php file.

2. Fetch Dishes from Database:

- Executes a SQL query to select all dishes from the 'Dishes' table.

3. Check Results:

- Verifies if there are results from the query.

4. Process Results:

- If there are results, fetches the dishes and stores them in an array.

5. Return JSON Response:

- Sets the content type to JSON and encodes the dishes array as JSON.

- Echoes the JSON response to be consumed by the client-side script.

6. Error Handling:

- If no dishes are found, returns a JSON response with an error message.

7. Close Connection:

- Closes the database connection.

Note: This file is designed to be accessed by AJAX requests from the client side (possibly using JavaScript). It returns a JSON-encoded array of dishes retrieved from the database. Proper security measures, such as input validation and secure database handling, should be implemented in a production environment.

File: **ord\_script.js**

Description: This JavaScript file is responsible for dynamically fetching and displaying the menu on the client side. It allows users to add items to their order, discard items, and place the final order. It communicates with the server-side API (order\_connection.php) to retrieve menu information.

Contents:

1. Fetch Menu:

- Uses the Fetch API to make an asynchronous request to 'order\_connection.php' to retrieve the menu.

- Handles the JSON response and passes the menu data to the displayMenu function.

2. Display Menu:

- Receives the menu data and dynamically creates HTML elements to display each dish.

- Appends menu items with 'Add to Order' buttons to the 'menu' container on the page.

3. Add to Order:

- Adds selected menu items to the user's order dynamically.

- Updates the total sum based on the added items.

- Calls the updateTotalSum function.

4. Discard from Order:

- Removes selected items from the user's order dynamically.

- Updates the total sum based on the discarded items.

- Calls the updateTotalSum function.

- Displays an alert notifying the user of the discarded item.

5. Update Total Sum:

- Updates the displayed total sum on the page.

6. Place Order:

- Displays an alert notifying the user that the order has been placed successfully.

- Reloads the page to reset the order.

Note: This JavaScript file assumes a connection to the order\_connection.php API for menu data. It interacts with the DOM to dynamically update the menu and order sections. Ensure proper error handling and security measures are implemented for a production environment.

File: **manager\_dashboard.php**

Description: This PHP file represents the manager's dashboard for a restaurant menu website. It provides access to various management functionalities, including managing providers, employees, dishes, and products. The page is accessible only to users with the 'Manager' role, and unauthorized users are redirected to the login page.

Contents:

1. Session and Error Reporting:

- Session is started to check user authentication and role.

- Error reporting is set to suppress errors.

2. HTML Document Structure:

- Sets up the HTML document with metadata, title, links to external stylesheets, and favicon.

3. Main Content:

- Header: Displays a logo and the text 'Manager'.

- Navigation Bar: Provides links to different management functionalities (Provider, Employees, Dishes, Products, Log out).

- Greetings: Personalized greeting for the logged-in manager, acknowledging their role and responsibilities.

4. Authorization Check:

- Checks if the user is logged in and has the 'Manager' role. Redirects unauthorized users to the login page.

5. Footer:

- Displays the author's name.

Note: This file assumes that user authentication and role management are implemented elsewhere (e.g., during login). Ensure proper security measures, such as input validation, secure session handling, and database security, are implemented in a production environment.

File: dishes.php

Description: This PHP file serves as a management interface for dishes in a restaurant menu. It allows a manager to perform CRUD (Create, Read, Update, Delete) operations on dishes. The page is accessible only to users with the 'Manager' role, and unauthorized users are redirected to the login page.

Contents:

1. Session and Error Reporting:

- Session is started to check user authentication and role.

- Error reporting is set to suppress errors.

2. HTML Document Structure:

- Sets up the HTML document with metadata, title, links to external stylesheets, and favicon.

3. Form for Data Manipulation:

- Allows the manager to input and submit data for adding, updating, deleting, and searching dishes.

- Input fields for dishes\_id\_serial, emploeer\_id\_serial, name, category, weight, cost, and recipe.

- Submit buttons for Add, Update, Delete, and Find operations.

4. PHP Logic for Data Manipulation:

- Processes form submissions to execute CRUD operations on the 'Dishes' table.

- Uses prepared statements for secure database interactions.

- Displays feedback messages and the number of rows in the result set.

5. Table Displaying Dishes Data:

- Retrieves and displays data from the 'Dishes' table in a tabular format.

- Shows columns for dishes\_id\_serial, name, emploeer\_id\_serial, category, weight, cost, and recipe.

6. Back Button:

- Provides a back button that redirects to the 'menu.php' page.

Note: This file assumes that user authentication and role management are implemented elsewhere (e.g., during login). Ensure proper security measures, such as input validation, secure session handling, and database security, are implemented in a production environment.