**Functionality Guide**

**1. all\_recipes.php**

**Purpose**: Display a list of all recipes with search and filter capabilities. Users can also mark recipes as favorites.

* **Session Management**:
  + Includes a session validation via assert\_logged\_in.inc.php.
  + Retrieves the current logged-in user's information.
* **Database Connection**:
  + Uses dbconnection.php to connect to the database.
  + Queries the recipes and favorites tables to fetch all recipes, including whether each recipe is marked as a favorite by the logged-in user.
* **Search Functionality**:
  + Users can search for recipes based on the name, cuisine, or description using the search parameter.
  + The search modifies the SQL query to include LIKE conditions.
* **Filter Functionality**:
  + Users can filter recipes by cuisine using checkboxes. The filters are applied using an OR clause in SQL.
  + A helper function (get\_html\_checkbox) generates checkboxes for available cuisines.
* **Output**:
  + Displays a list of recipes based on the search and filter criteria.
  + If no search or filter is applied, it displays all available recipes.

**2. create.php**

**Purpose**: Allow users to add a new recipe to the system.

* **Session Management**:
  + Includes a session validation to ensure the user is logged in.
* **Database Connection**:
  + Uses dbconnection.php for database access.
  + Handles POST requests to collect data from the recipe creation form.
* **Form Processing**:
  + Collects recipe details: name, description, cuisine, image URL, ingredients, and instructions.
  + Ingredients and instructions are handled as newline-separated strings, converted to JSON format for storage.
  + Data is sanitized using basic PHP functions like json\_encode.
* **Database Interaction**:
  + Inserts the new recipe data into the recipes table using an INSERT INTO SQL statement.
  + After insertion, the user is redirected to the all\_recipes.php page.

**3. delete.php**

**Purpose**: Allow users to delete an existing recipe.

* **Session Management**:
  + Includes a session validation to ensure the user is logged in.
* **Database Connection**:
  + Uses dbconnection.php for database access.
  + Handles both GET and POST requests for deleting recipes.
* **Recipe Deletion**:
  + If a recipe\_id is provided, fetches the recipe details to confirm deletion.
  + On form submission (POST request), deletes the recipe from the recipes table using a DELETE SQL query.
  + Redirects to the all\_recipes.php page after successful deletion.

**4. edit.php**

**Purpose**: Allow users to edit an existing recipe's details.

* **Session Management**:
  + Includes a session validation to ensure the user is logged in.
* **Database Connection**:
  + Uses dbconnection.php for database access.
  + Handles POST requests to update recipe details.
* **Form Handling**:
  + Displays the current details of a recipe for editing.
  + Collects updated data and sanitizes it before storing it in the database.
  + Updates the recipes table with the new information via an UPDATE SQL query.
  + On success, redirects to a detailed view page for the edited recipe.

**5. favorites.php**

**Purpose**: Display a list of recipes marked as favorites by the logged-in user.

* **Session Management**:
  + Includes a session validation to ensure the user is logged in.
* **Database Connection**:
  + Uses dbconnection.php for database access.
  + Queries the recipes and favorites tables to fetch recipes favorited by the user.
* **Output**:
  + Displays the user's favorite recipes or a message if no favorites are found.
  + Uses an included file (recipe\_list.inc.php) to render the list of recipes.

**6. functions.php**

**Purpose**: Contains helper functions to support the main application.

* **Function: get\_html\_checkbox**
  + **Input**: An array of all available cuisines ($arr) and an array of selected cuisines ($arr\_selected).
  + **Output**: Generates HTML checkboxes for each cuisine, marking checkboxes as checked if they are selected.

**Database Tables Overview**

* **recipes**:
  + Fields: recipe\_id, name, description, cuisine, image\_url, ingredients\_json, instructions\_json.
  + Stores all recipe details, including ingredients and instructions in JSON format.
* **favorites**:
  + Fields: username, recipe\_id.
  + Tracks the recipes marked as favorites by users.

**Additional Considerations**

* **Security**:
  + User inputs are directly included in SQL queries without proper escaping, making the application vulnerable to SQL injection.
  + Consider using prepared statements with parameterized queries for improved security.
  + Inputs are sanitized using basic PHP functions but should include stronger validation mechanisms, especially for the POST request data in create.php and edit.php.
* **User Experience**:
  + The application supports essential CRUD (Create, Read, Update, Delete) operations for recipes.
  + A simple and intuitive interface with search and filter options is provided for better user interaction.
  + Error handling for database interactions (e.g., failed queries) is limited and should be enhanced for better feedback to the user.