

# [TSW] Recipy Documentation

**index.php:** Landing page of the application.

**/components:** Contains reusable UI components implemented in our prototype, such as the header, footer, carousel, recipe card versions, and search bar.

**/css:** CSS files that define the styling and layout of the project. `styles.css` contains the general style of the page and its components, while the rest of the layout is compartmentalized into different files.

**/dashboard:** Pages related to the user's dashboard, where they can manage their recipes. Both `dashboard.php` and `add-recipe-view.php` represent views, while `add-recipe.php` handles the connection with the database.

**/explore:** Pages used to explore available recipes, similar to a feed or gallery. `explore.php` represents the view of the explore page, `userKitchen.php` shows the recipes a user has posted, `exploringRecipe.php` fetches a specific recipe, and `search-recipe.php` is responsible for the logic of searching for a specific recipe.

**/header:** Contains files for the views and actions that can be performed from the header navigation.

- `contact.php` represents the contact page view.
- `logout.php` handles the logout logic.
- `edit-profile-vue.php` and `edit-profile-db.php` are two versions of the edit profile page view and logic, one using Vue.js and the other not.

**/img:** Stores all images used in the application, including icons, profile pictures, and recipe photos.

**/js** JavaScript files handling frontend logic and dynamic interactions.

- **add-recipe-jquery.js:** Manages the recipe addition process with jQuery, validates the recipe form, and submits it via AJAX. It allows dynamic step addition/removal and error handling.
- **explore.js:** Manages the recipe carousel, handles dynamic loading and displaying of recipes, enables the "like" functionality, and implements the search feature.
- **recipe.js:** Loads and displays recipe comments, handles new comment submissions, and checks if the user is authorized to comment.
- **rememberMe.js:** Manages login, validates credentials via AJAX, and stores login data if the "remember me" option is selected.
- **validateForm.js:** Validates the registration form, checks for errors, and submits data via AJAX to register the user.

**/login:** Contains **login\_form.php** with the view and **login.php** handling the logic.

**/recipe:** Pages or components that display individual recipe details, allowing users to interact with them (save, rate, comment).

- **addComment.php:** Handles comment submission for a selected recipe. It checks if the user is logged in and if a recipe is selected, then inserts the comment into the database.
- **check\_author\_of\_recipe.php:** Verifies if the logged-in user is the author of the selected recipe, returning a boolean result.
- **delete.php:** Deletes the selected recipe from the database if the user is logged in and authorized. After deletion, it redirects the user to the dashboard.

- `loadComments.php`: Loads and displays all comments for the selected recipe. It checks if the user is logged in and retrieves the comments from the database.
- `recipe.php`: Displays the details of a specific recipe, including its ingredients, instructions, and comments. It also manages the 'like' functionality and the option to delete the recipe for the author.
- `saveRecipe.php`: Handles saving and unsaving recipes by the logged-in user. It adds or removes the recipe from the saved list in the database based on the user's action.

**/registration**: Contains `register.php` with the view and `registration.php` handling the logic.