

EventHub Documentation

Ethan Streets, Nevon Xayavongsa, Yash Patel, Grace McPherson

October 22, 2024

1 Database Documentation

There are three different files required to create and communicate with the database. These are config.php, TestEntries.sql, and BuildDB.sql.

- The config.php file is used within the php files to establish the connection to the database for queries.
- The BuildDB.sql script is used to build the database's tables and relationships.
- The TestEntries.sql script is used to enter the required statuses into the EVENT_STATUS table and insert dummy records into the database.
- More documentation and instructions on how to run the scripts can be found in the **How to Run the DB Script** file.

2 Website Documentation

2.1 Account Login, Creation, and Forgotten Password

2.1.1 Login page: SE.php

This page collects data from the user required to log in. If they do not have an account or if the password is incorrect, it will not log them in. If they are successfully able to log in, it will redirect them to the Customer Mode Home Page (ES.php). From this page they can also click a button to redirect to the Create Account page (Create.php) or a button to the Forgot Password page (FP.php).

2.1.2 Account Creation page: Create.php

This page collects data from the user required to make an account. Once they've successfully made an account, it redirects them to the Customer Mode Home Page (ES.php). If they leave any required fields blank, it will not allow them to create an account.

2.1.3 Forgot Password page: FP.php

This page collects data from the user required to reset their password. Once they've successfully reset their password, it redirects them to the Customer Mode Home Page (ES.php).

2.2 Customer Mode Pages

2.2.1 Home page: ES.php

This is where users will initially be redirected to after logging in or creating an account. The users (currently customers, as they would be in customer mode on this page) will see all the events laid out in a card format. They can also RSVP to any of the events if they choose to. From this page, customers can click on the navbar to filter by date (FilterDate.php), location (FilterLocation.php), or category (FilterCategory.php). They can also hover on the top right to visit their profile page (userSettings.php), see the events they've RSVP'd for (ESMyEvents.php), change from Customer Mode to Creator Mode (ESManager.php), and log out.

2.2.2 Filter by Date page: FilterDate.php

This is where customers can filter what events they wish to see between two different dates. They input the start and end date in the format YYYY-MM-DD and it returns a table with the events occurring between those dates.

2.2.3 Filter by Location page: FilterLocation.php

On this page, a customer can filter what events they wish to see within a mile-radius of their current address. This page collects all events from the database, calculates their distance from the current address, and if it is within the radius, displays it to the page.

2.2.4 Filter by Category page: FilterCategory.php

This page is where customers can filter the events by category. They will choose the category from a dropdown box, and the events are returned in a table format.

2.3 Creator Mode Pages

2.3.1 Home page: ESManager.php

When the customer goes into Creator Mode, it redirects them to this page. It is like the other home page as it lays out the events in a card format and allows the creator to RSVP to an event. From the navbar, the creator can filter by date (FilterDateManage.php), location (FilterLocationManage.php), or category (FilterCategoryManage.php). They can also create an event (ESCreate.php) and manage their events (ESManage.php). Like in Customer Mode,

they can visit their profile page, see their RSVP'd events, can switch to Customer Mode, and can log out.

2.3.2 Event Creation page: ESCreate.php

On this page, the creator can input data to create their event, such as its name, description, address, etc. Once it has been successfully added, it will then appear on the creator's manage page (ESManage.php).

2.3.3 Event Management page: ESManage.php

On this page, the creator can edit and delete the events they have created. If they hit the edit button, it redirects to a different page (edit_record.php) to update the event's information. If they hit the delete button, it redirects to a confirmation page (delete_record.php) to ensure the creator wishes to delete the event.

2.3.4 Event Editing page: edit_record.php

This page is used by a creator to edit their event information.

2.3.5 Event Deletion page: delete_record.php

This page is used to confirm that the creator wishes to delete their selected event.

2.3.6 Filter by Date page: FilterDateManage.php

This is where creators can filter what events they wish to see between two different dates. They input the start and end date in the format YYYY-MM-DD and it returns a table with the events occurring between those dates.

2.3.7 Filter by Location page: FilterLocationManage.php

On this page, a creator can filter what events they wish to see within a mile-radius of their current address. This page collects all events from the database, calculates their distance from the current address, and if it is within the radius, displays it to the page.

2.3.8 Filter by Category page: FilterCategoryManage.php

This page is where creators can filter the events by category. They will choose the category from a dropdown box, and the events are returned in a table format.

2.4 Customer and Creator Pages

2.4.1 RSVP'd Events page: ESMYEvents.php

On this page, a user can view what events they have RSVP'd for and change their RSVP status for the event.

2.4.2 Saving Event Status page: save_RSVP.php

This file is used to save the status a user has with an event. It will be one of the following: Not Going, Interested, or Going. If a user is going to the event, it will appear on their My Events page (ESMYEvents.php).

2.4.3 User Profile page: userSettings.php

On this page, a user can change their profile settings as needed.

2.4.4 Confirm Logout page: Confirm_Logout.php

This page confirms that the user wants to log out. Once they've confirmed, their session is ended.