

# Work Process of Expense Tracker App

## Introduction

No one likes to spend, but wants to note them down when there is an expense. I am someone who loves to keep my spending in check and so I decided to build an Expense Tracker in my journey of becoming a web developer.

### Step 1: Thought process

At first I jumped into my imagination of how I want my app to look like and what information I want to record for my expenses. Started making some wireframes in Figma and thought of adding fields to insert **date, name of the expense, amount and details** of each entry. All the inserted data will be shown in a list just below the form. I decided to keep an option to Edit and Delete any entries if needed.

The rough user interface I first designed in Figma is shown below:

EXPENSE TRACKER	
Expense Name	<input type="text"/>
Amount	<input type="text"/>
Details	<input type="text"/>
<input type="button" value="SUBMIT"/> <input type="button" value="CANCEL"/>	
28 Aug - Grocery - Rice, Meat, Noodles - \$70 <input type="button" value="Edit"/> <input type="button" value="Del"/>	
28 Aug - Fuel - Trip to Mont-Tremblant - \$120 <input type="button" value="Edit"/> <input type="button" value="Del"/>	
Total Expense : \$190	

Didn't add the option to insert the date of the entry at first but I added it later which you will be able to see in my final interface.

### Step 2: Creating database table.

To store the data I needed a database and for what I had planned I needed only one table. Tried to keep it simple at first. So I created a table called 'expenses', added four columns as required.

### Step 3: Little bit of front-end and connecting to DB

Created the simple form with required fields and labeled them using just HTML. Then I wrote the PHP code to connect the form with my database to test the INSERT and SELECT queries.

When I was able to successfully insert and see the data in the database, it was time to work to display the data list to the user.

[illegible]

```
[
- {
    id: 53,
    expense_name: "Coffee",
    amount: 10,
    details: "Tim Horton",
    added_on: "2023-09-08"
},
- {
    id: 54,
    expense_name: "Movie",
    amount: 25,
    details: "Mission Possible",
    added_on: "2023-09-20"
},
- {
    id: 63,
    expense_name: "Utility",
    amount: 150,
    details: "Electricity",
    added_on: "2023-09-15"
},
- {
    id: 64,
    expense_name: "AC",
    amount: 350,
    details: "Portable AC",
    added_on: "2023-09-22"
}
]
```

### Step 4: Javascript & CSS

Created a table using DOM to show the list of data inserted. Used a little bit of bootstrap to get a primary look with responsiveness and styling. First function to implement with Javascript was the insertion of data by fetching the PHP file. I used the 'POST' HTTP request for fetching and inserting data with Javascript.

### Step 5: Edit & Delete

There can always be human errors while entering data or a need to change data. So, I decided to include EDIT and DELETE options to be able to modify the data entered which helped me to learn how to manipulate the database.

The final look of my Expense Tracker App so far is shown below:

The screenshot shows the 'Expense Tracker' app interface. At the top, there's a teal header with the title 'Expense Tracker'. Below it, there's a form with the following fields: 'Select a Date:' with a date picker showing 'yyyy-mm-dd', 'Expense Name' with a text input 'Ex: Gas', 'Amount' with a text input '\$', and 'Details' with a text input 'Optional'. Below the form are two buttons: 'Add' (green) and 'cancel' (red). Underneath is a table with columns: 'Date', 'Name', 'Amount', 'Details', 'Edit', and 'Delete'. The table contains three rows of expense data. At the bottom, there's a red bar showing 'Total Expense= \$250'.

Date	Name	Amount	Details	Edit	Delete
2023-09-08	Coffee	\$50	Tim Horton	Edit	Delete
2023-09-20	Movie	\$50	Mission Possible	Edit	Delete
2023-09-15	Utility	\$150	Electricity	Edit	Delete

Total Expense= \$250

### Step 6 : Improving user interface

At first I was re-populating the form with the data for which the edit button was pressed which seemed not that user friendly to me and would require an effort for the user to understand what is happening. And for the delete option it was deleting the entries with just one press on the delete button without giving any confirmation alert.

So, later I decided to add popup options for both the functions which made it more user friendly and reduced the chance of deleting an entry by mistake. The popup interfaces are as shown below:

This screenshot shows the app with a modal popup for editing an expense. The popup has the following fields: 'Select a Date:' with a date picker showing '2023-09-08', 'Expense Name' with a text input 'Coffee', 'Amount' with a text input '10', and 'Details' with a text input 'Tim Horton'. At the bottom of the popup are two buttons: 'Update' (blue) and 'Cancel' (red). The background shows the same table as the previous screenshot, but it's dimmed.

This screenshot shows the app with a modal popup for deleting an expense. The popup has a single text input: 'Are you sure you want to delete this expense?'. At the bottom of the popup are two buttons: 'Yes' (red) and 'No' (green). The background shows the same table as the previous screenshot, but it's dimmed.

## ***CONCLUSION***

So far I was able to achieve what I had planned for my simple Expense Tracker app. I was able to learn about INSERT and SELECT query, learned to use FETCH function and did a lot of DOM that cleared many basic concepts. There are a lot of improvements that can be made to this app and I wish to do it for my final project.

Improvements that I am thinking of are - using relational tables, being able to add receipt images, filtering entries according to dates and showing graphical presentation. Really excited to add these new features to my app.