Smart Budget Tracker Application

R1: Application High-level Description

Our final project is a personal budgeting application designed to help users effectively manage their finances. Key features of the application include a budget logger, expense management tools, spending pattern visualizations, and smart spending optimization suggestions. Additionally, the application supports a group enrollment feature, allowing users to track shared expenses conveniently. The primary users of this app are individuals and groups, such as families, students, and roommates, who seek a robust platform for managing personal or collective financial activities. To demonstrate the functionality of the app, we will create a production dataset as the principal administrators and developers.

R2: System Support Description

The application is hosted locally and accessed through a web interface, to provide a desktop-based user-friendly experience rather than mobile use. Our technology stack consists of Flask and React for dynamic web development, with HTML, CSS, and JavaScript enhancing the front end. For database management, we utilize SQLite3, ensuring scalable user data handling.

R3/4: Database with Sample & Production Dataset

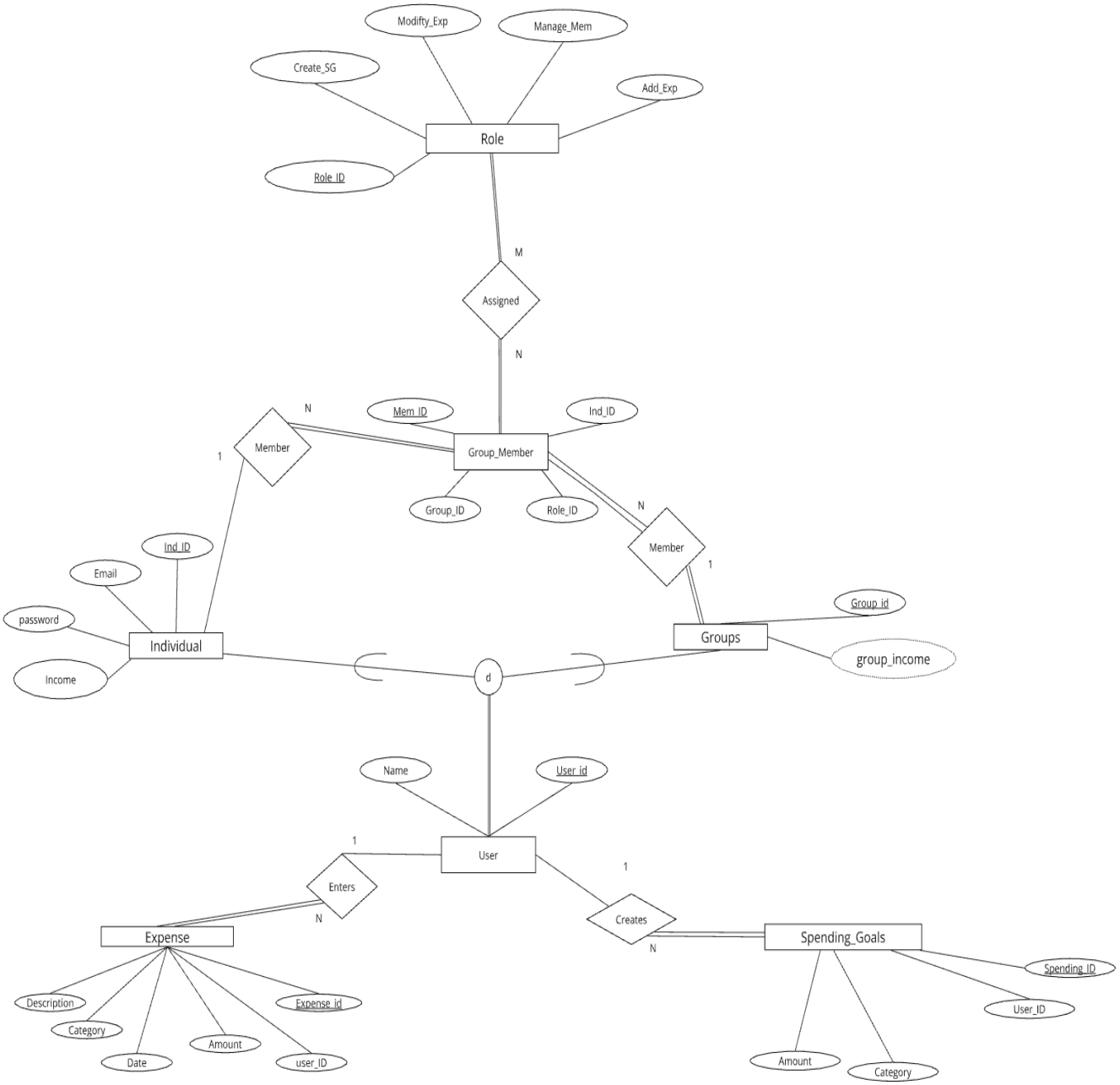
To simulate realistic user interaction, we manually generated fictional entries that represent typical user transactions. For a comprehensive look at how our application handles data in a live environment, our production database is available on our GitHub repository.

R5: Database Schema Design

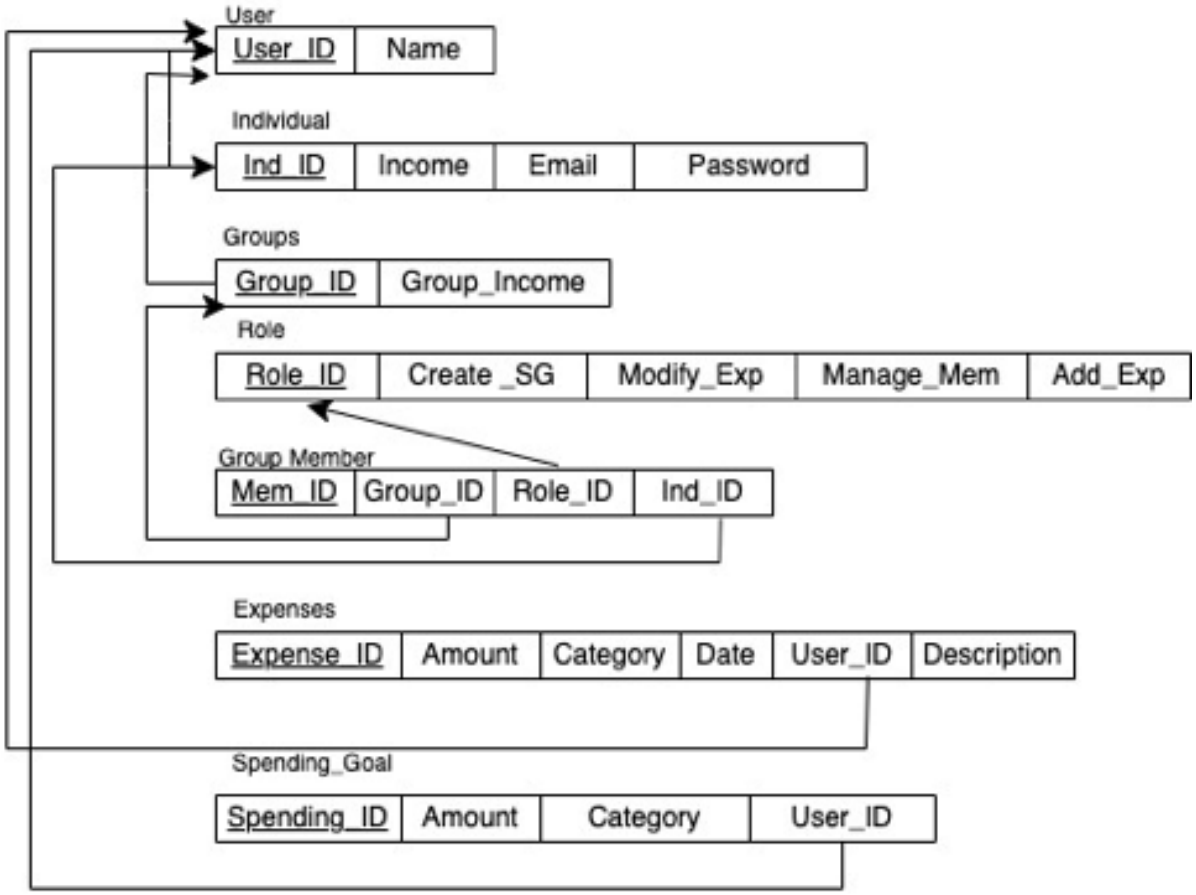
R5a: Assumptions

- All user inputs, such as expenses and budget goals, are recorded in a single currency (Canadian Dollars).
- Users can set budget goals for various categories and must log expenses under predefined categories only.
- Group creators can set spending goals, adjust expenses, and modify both group members and their permissions.
- Each email address is unique to an individual user but is not used as the primary key in the 'Individual' table.
- A Groups's income will be determined by the income of all Individuals who are members
- The Individual who creates a Group will have all Role that provides access to all functionalities

R5b: E/R Diagram for Database Design

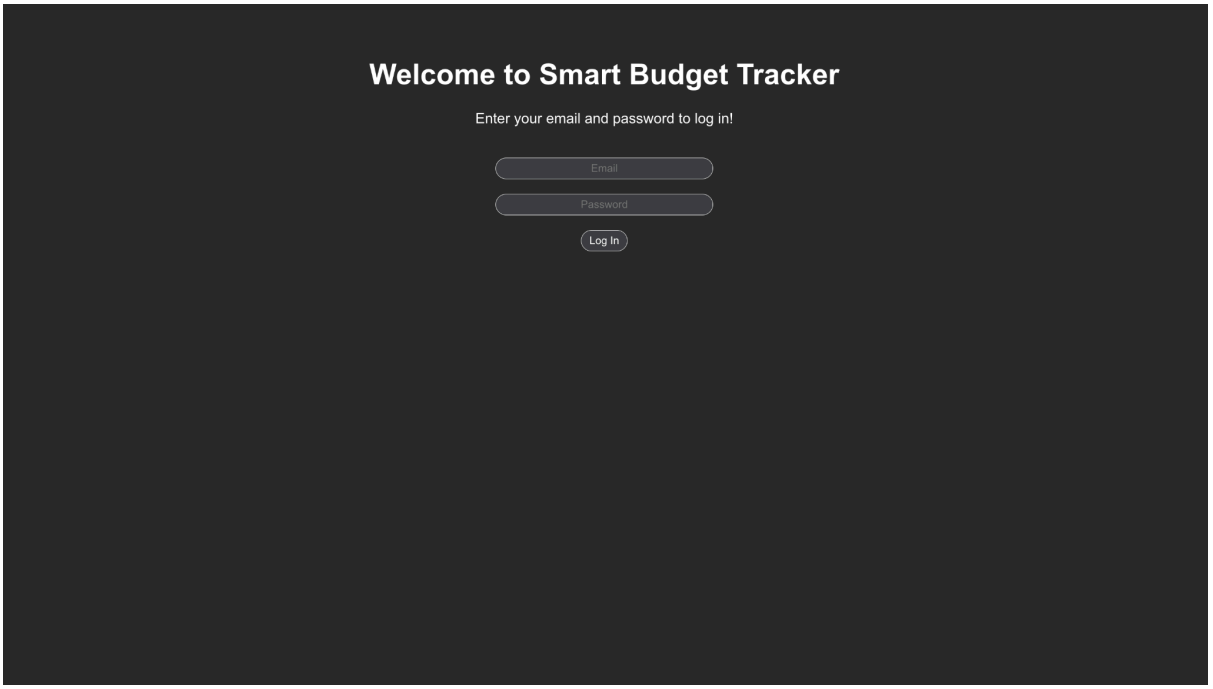


R5c: Relational Data Model

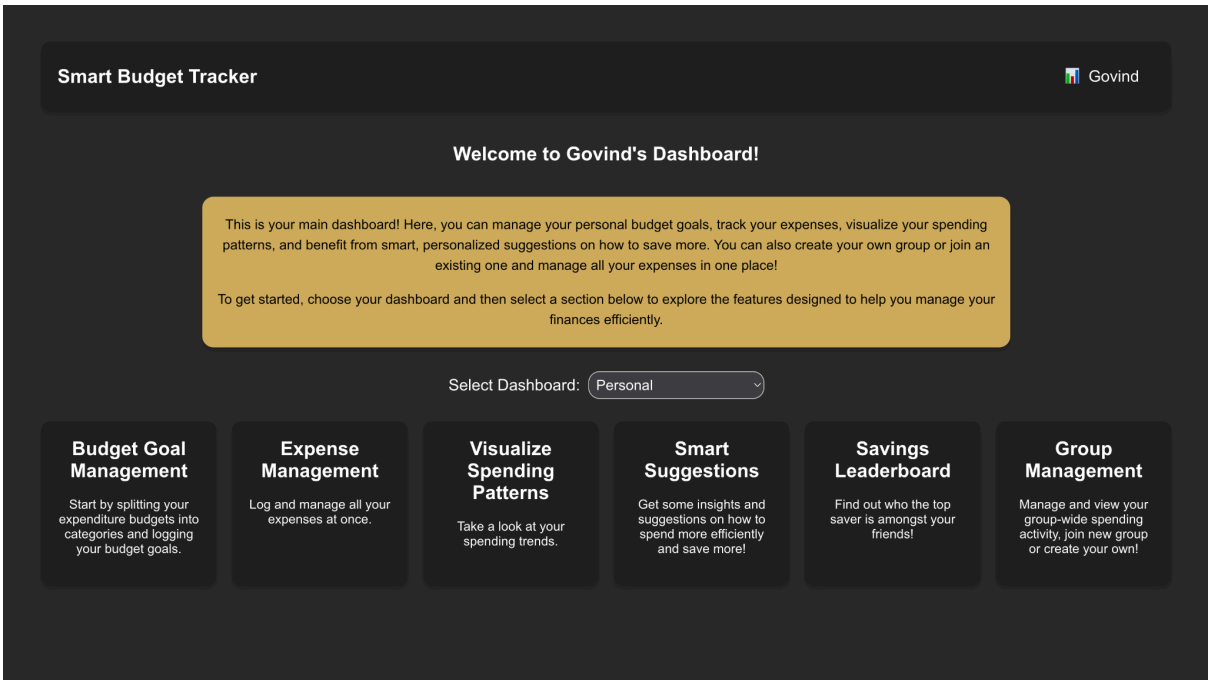


Overview of Application

Login Page



Dashboard



On the dashboard page of our app, users are welcomed by a series of interactive cards that serve as gateways to various specialized features and functionalities. Each card is distinctly titled to reflect its purpose: “Budget Goal Management,” “Expense Management,” “Visualize Spending Patterns,” “Smart Suggestions,” “Savings Leaderboard,” and “Group Management.” Clicking on any of these cards will navigate users to separate pages where they can engage with the specific tools and services designed to enhance their financial management experience.

R6: Fetch & Display User's Budget Goals and Expenses (Feature 1)

R6a: Feature Interface design

- **Budget Goal Management**: Clicking on this card displays a table with headers "Categories," "Budget ($)," and "Actions." Here, users can define their main spending categories, set a monthly budget for each category, and modify or delete their budget goals using the "Edit" and "Delete" buttons. This feature simplifies managing and tracking financial goals per major spending category.
- **Expense Management**: Clicking on this card reveals a comprehensive table with headers "Description," "Amount ($)," "Category," "Date," and "Actions." Users can see detailed entries for each expense, including description, cost, category, and date. Similar to budget management, expenses can be edited or deleted using the corresponding buttons in the "Actions" column.

R6b: SQL Query (Testing with Sample Data)

- **Budget Goal Management**:

```
> SELECT * FROM spending_goal WHERE user_id = I55555;
SG4|Food|200.0
SG5|Rent|900.0
SG6|Clothes|150.0
SG7|Miscellaneous|200.0
SG_1721051068|Groceries|250.0
```

- **Expense Management**:

```
> SELECT * FROM expenses WHERE user_id = I55555;
expense_1721050957|157.0|Clothes|2024-05-26|I55555|Lululemon
expense_1718992742|780.0|Rent|2024-06-01|I55555|Rent
expense_1718992762|35.0|Miscellaneous|2024-06-04|I55555|Gifts
expense_1718992483|4.78|Food|2024-06-05|I55555|Iced capp
...
```

R6c: SQL Query (Testing with Production Data)

- **Budget Goal Management**:

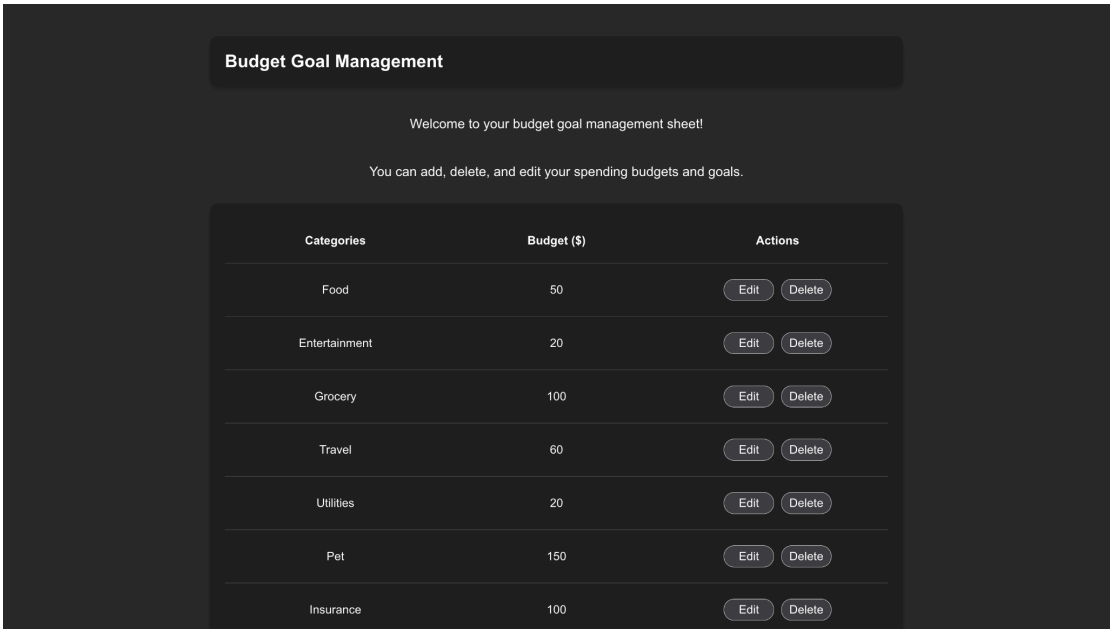
```
1|Food|50.0
2|Entertainment|20.0
3|Grocery|100.0
4|Travel|60.0
5|Utilities|20.0
...
```

- **Expense Management**:

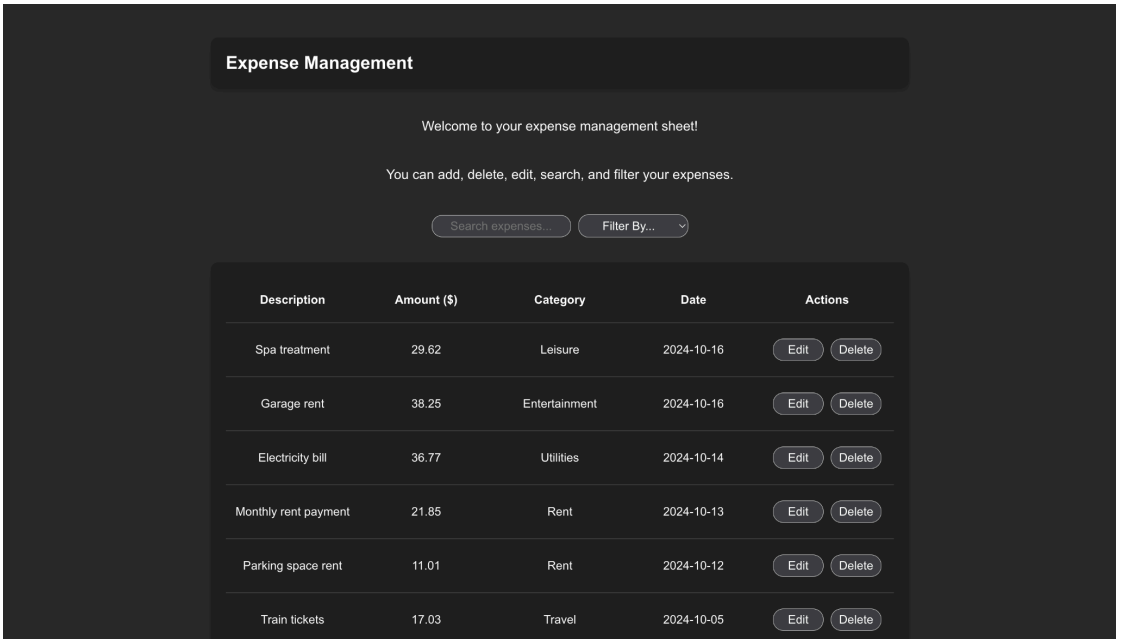
```
1|36.61|Food|2024-07-21|12|Dinner at a restaurant
3|16.72|Food|2024-07-28|12|Lunch at a fast-food restaurant
8|14.28|Entertainment|2024-07-28|12|Concert tickets
2|36.86|Food|2024-07-29|12|Breakfast at a cafe
24|21.41|Travel|2024-08-02|12|Car rental fee
...
```

R6d: Snapshot

- **Budget Goal Management**:



- **Expense Management**:



R7: Add Budget Goal and Expense (Feature 2)

R7a: Feature Interface design

Users can add new budget goals and expenses via intuitive "+" buttons located on the last row of their respective tables. By clicking the "+" button, users can add a new row with new information on new budgets and expenses.

- **Adding a Budget Goal**: In the new row, users input a new category name and budget goal amount into the "Categories" and "Budget ($)" fields respectively. They can then confirm their entries by clicking the "Log Budget" button to add this new budget goal to the `spending_goal` table associated with the user's ID.

- **Adding an Expense**: In the new row, users input a new description of their expense into the “Description” field, a new expense amount into the “Amount ($)” field, and select from a dropdown list of categories in the “Category” field, taken from their previously entered categories in their Budget Management sheet. They must also enter a date when the expense was made before clicking the “Add Expense” button to add this new expense to the `expenses` table associated with the user's ID.

R7b/c: SQL Query (Testing with Sample & Production Data)

- **Adding a Budget Goal**:

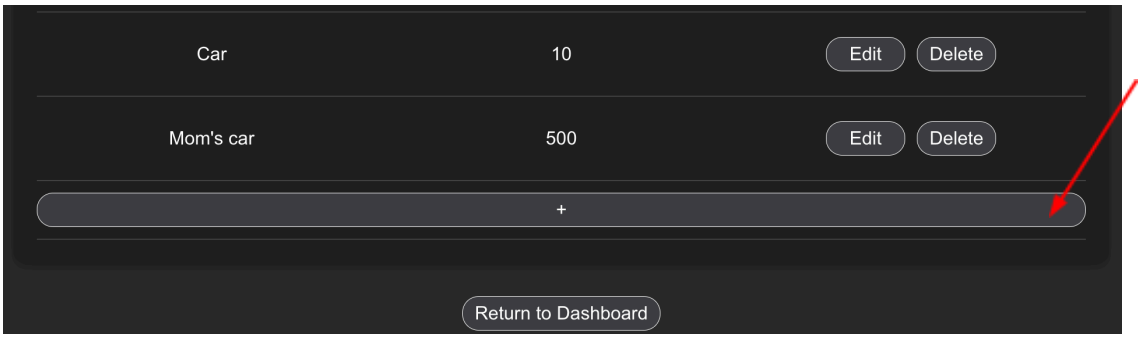
```
> INSERT INTO spending_goal VALUES (12987312, '1.23', 'New Category',
'I5555')
```

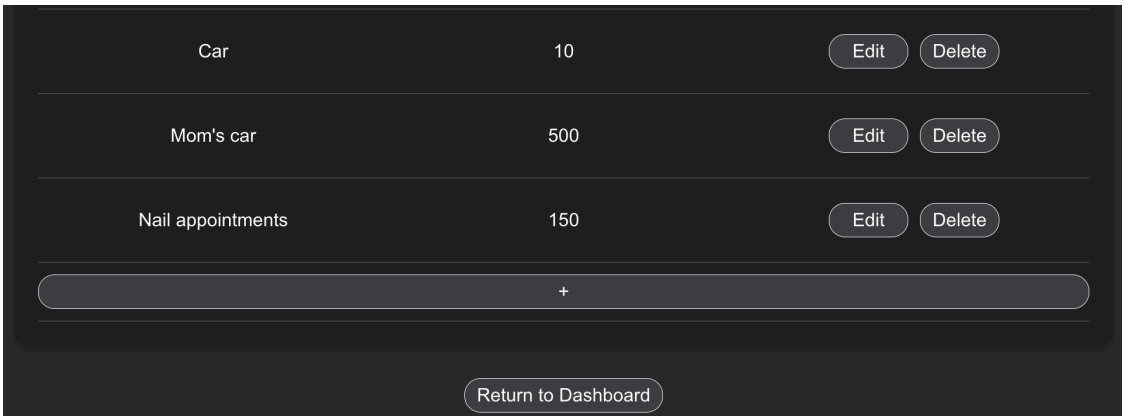
- **Adding an Expense**:

```
> INSERT INTO expense (spending_goal_id, amount, category,
date, user_id, description) VALUES (18276312, ‘90’, Clothes’,
‘2024-06-07’, ‘I55555’,  Aritzia’);
```

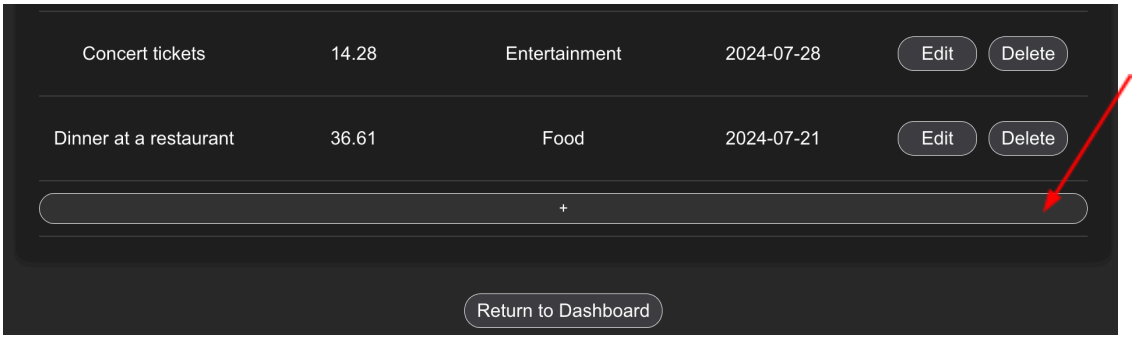
R7d: Snapshot

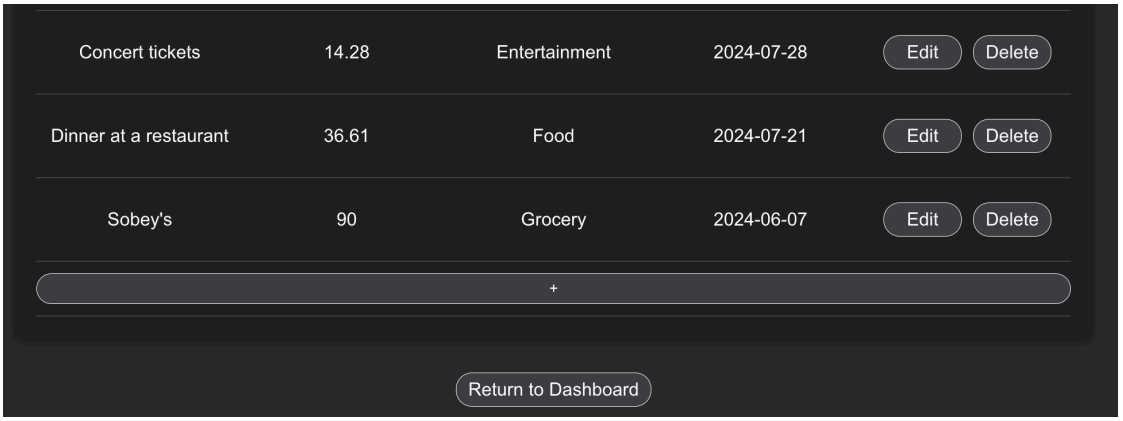
- **Adding a Budget Goal**:





- **Adding an Expense**:





R8: Edit Budget Goal and Expense (Feature 3)

R8a: Feature Interface design

Users can modify existing budget goals and expenses using the "Edit" button located in the “Actions” column of the respective tables. Clicking this button, allows users to update the values in any field in that row before clicking “Save Edits” to update the contents of the `spending_goal` and `expenses` tables in the backend database.

R8b/c: SQL Query (Testing with Sample & Production Data)

- **Edit a Budget Goal**:

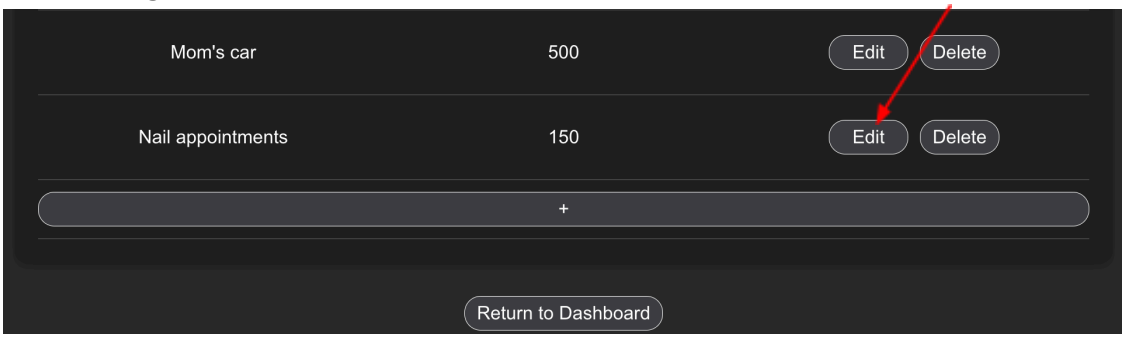
```
> UPDATE spending_goal SET category = 'New New Category', amount = '1.24' WHERE spending_id = 12987312
```

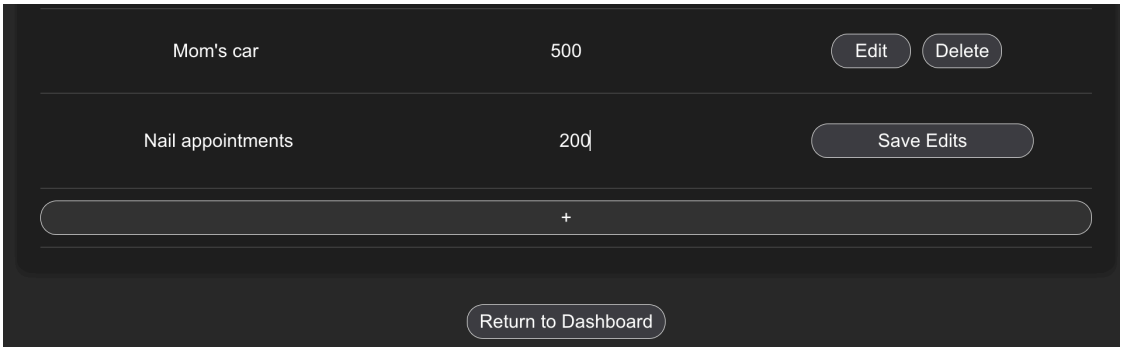
- **Edit an Expense**:

```
> UPDATE expenses SET description = ‘new description’, amount = 1.23, category = ‘new category’, date = ‘2024-07-17’ WHERE expense_id = 18276312;
```

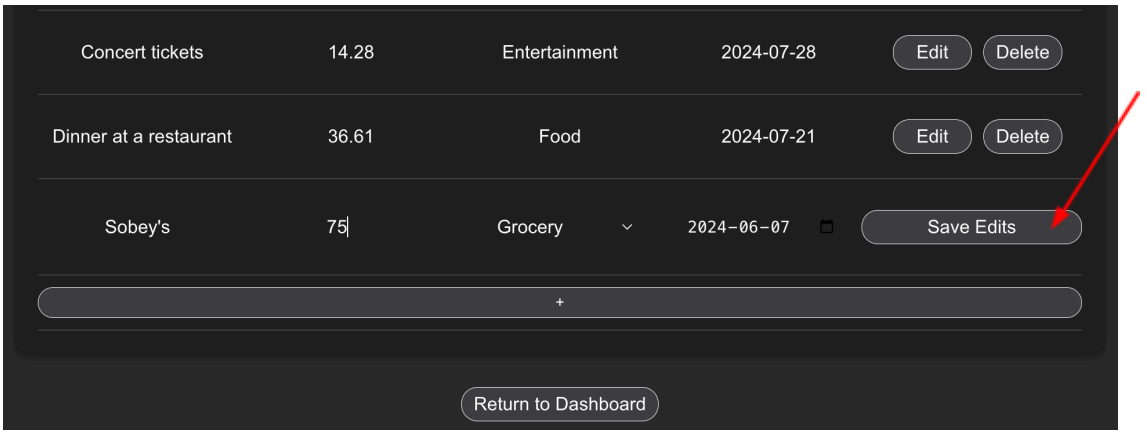
R8d: Snapshot

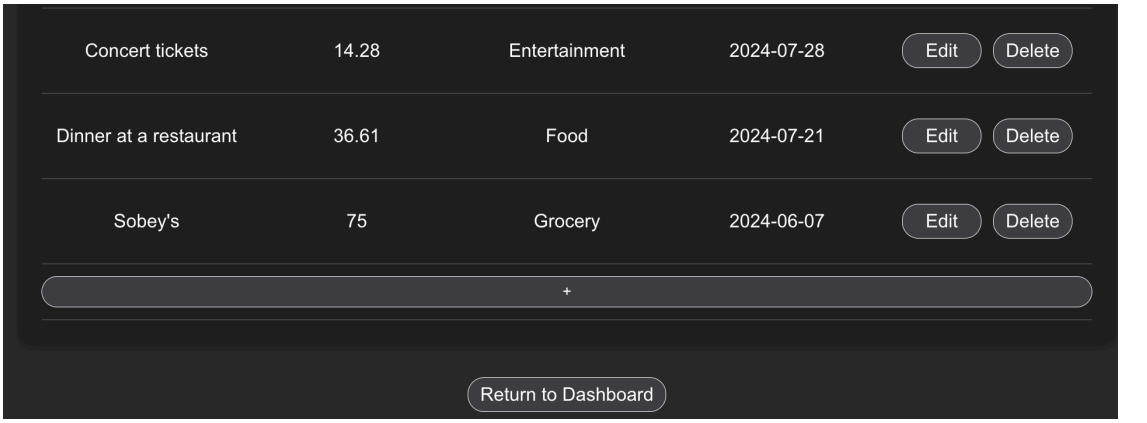
- **Edit a Budget Goal**:





- **Edit an Expense**:





R9: Delete Budget Goal and Expense (Feature 4)

R9a: Feature Interface design

Users can delete existing budget goals and expenses using the "Delete" button located in the "Actions" column of the respective tables. Clicking this button deletes that row from the table displayed as well as from the `spending_goal` and `expenses` tables in the database.

R9b/c: SQL Query (Testing with Sample & Production Data)

- **Delete a Budget Goal**:

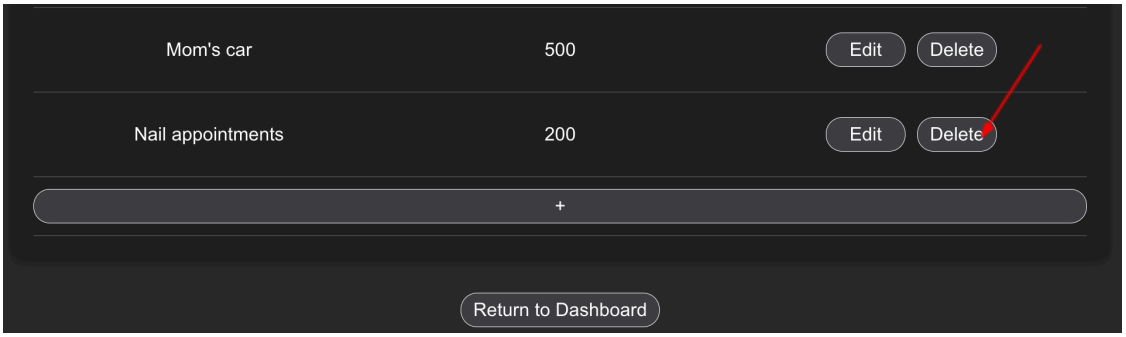
```
> DELETE FROM spending_goal WHERE expense_id = 18276312;
```

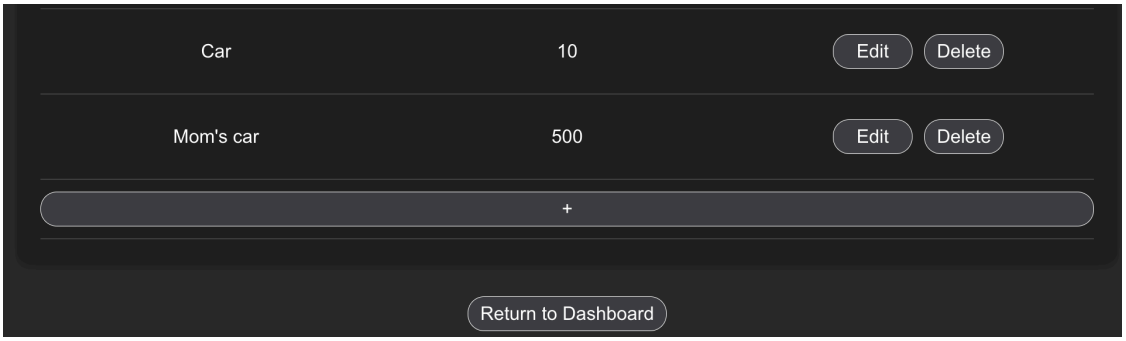
- **Delete an Expense**:

```
> DELETE FROM expenses WHERE expense_id = 18276312;
```

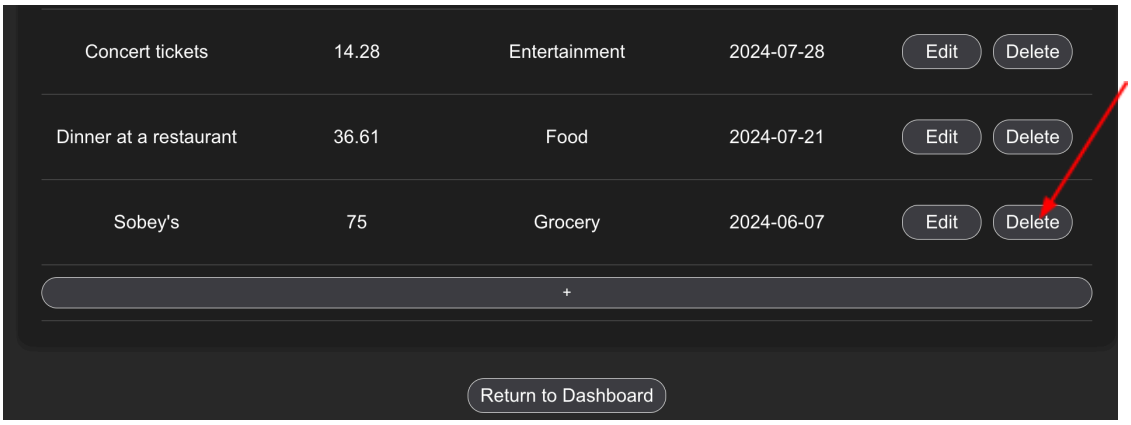
R9d: Snapshot

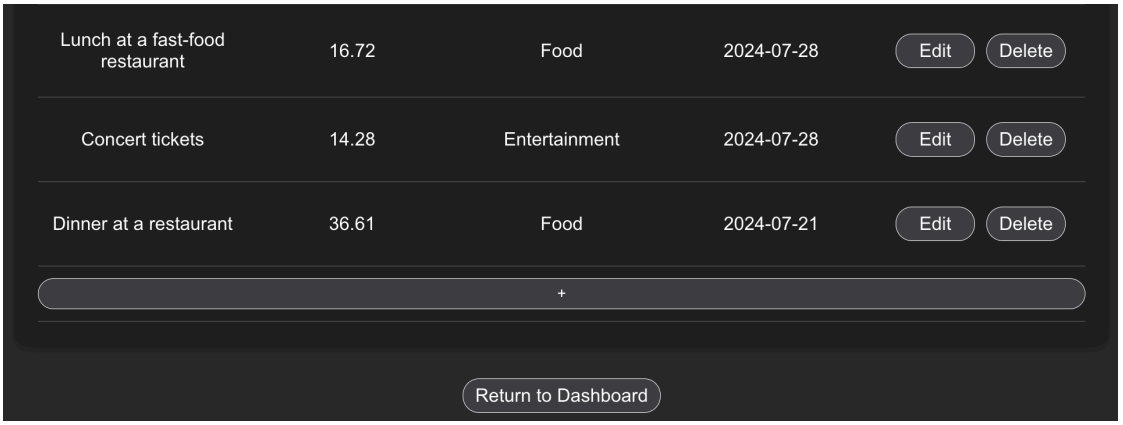
- **Delete a Budget Goal**:





- **Delete an Expense**:





R9: Savings Leaderboard (Feature 5)

R9a: Feature Interface design

This card directs users to a leaderboard page comparing their spending efficiency with that of their peers within the same groups. It ranks each member based on how they manage their finances to motivate users to optimize their spending habits by providing a clear and comparative insight into how they fare against their peers.

R10b: SQL Query (Testing with Sample Data)

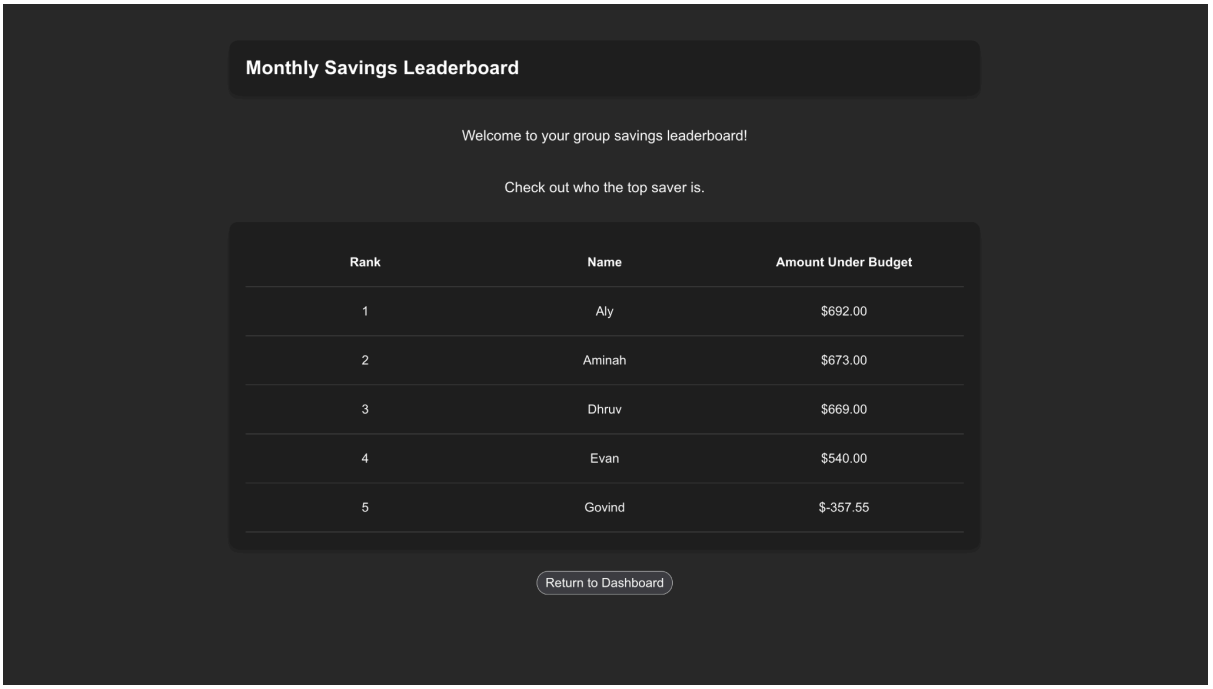
```
> WITH recent_expenses AS (SELECT ex.user_id, SUM(ex.amount)
                           AS total_expenses FROM expenses ex
                          WHERE ex.date >= datetime('now', '-1 month')
                          GROUP BY ex.user_id),
       total_spending_goals AS (SELECT sg.user_id, SUM(sg.amount)
                                AS total_goals FROM spending_goal sg
                                GROUP BY sg.user_id),
```

```
        relevant_individuals AS (SELECT DISTINCT gm.ind_id FROM
                                 group_member gm
                                 WHERE gm.group_id IN (SELECT gm.group_id
                                                       FROM group_member gm
                                                       LEFT JOIN user
                                                       ON gm.group_id =
                                                   user.user_id
                                                       WHERE gm.ind_id = ?))
        SELECT u.name,
        COALESCE(tg.total_goals, 0) - COALESCE(re.total_expenses, 0)
        AS net_savings FROM relevant_individuals ri
        LEFT JOIN user u ON ri.ind_id = u.user_id
        LEFT JOIN recent_expenses re ON u.user_id = re.user_id
        LEFT JOIN total_spending_goals tg ON u.user_id = tg.user_id
        GROUP BY u.name ORDER BY net_savings DESC;
Evan|89730433124.0
Aly|1195.0
Aminah|500.0
Govind|300.0
Dhruv|0
```

R10c: SQL Query (Testing with Production Data)

```
Aly|692.0
Aminah|673.0
Dhruv|669.0
Evan|540.0
Govind|-784.38
```

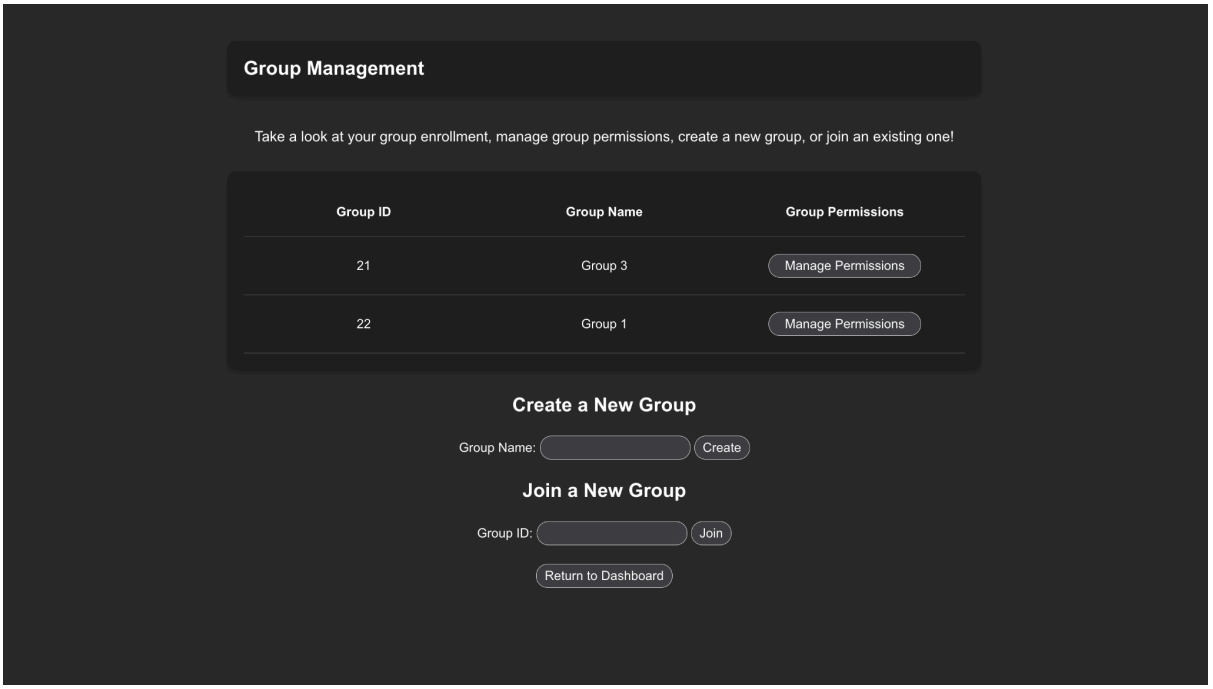
R10d: Snapshot



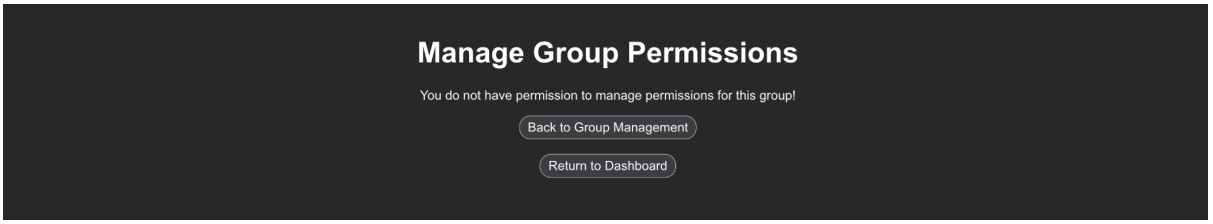
R11: Group Permissions Management (Feature 6)

R11a: Feature Interface design

This card allows users to oversee and adjust which group members have the authority to modify permissions, update group spending goals, and add or modify group expenses. Additionally, users can create a new group by entering a new group name and clicking “Create” or join an existing group using a specific group ID and clicking “Join”. This feature enhances collaborative financial management, making it easier for groups to coordinate and track their shared financial activities efficiently.



The “Manage Permissions” buttons take the user to the group permissions management page for the corresponding group the user is enrolled in. If the logged-in user does not have permission to modify the permissions of the selected group, they will see this message:



If they have the authority to manage group permissions, they will see a checkbox table that they can modify by altering the states of the checkboxes and clicking “Save Permissions” to update the database contents.

R11b: SQL Query (Testing with Sample Data)

Check if the user is an Admin

```
> SELECT role.manage_mem FROM role
   LEFT JOIN group_member gm on gm.role_id = role.role_id
   WHERE gm.ind_id = 1 AND gm.group_id = 1
|1|
```

Fetch Group Permissions

```
> SELECT user.name, role.create_sg, role.modify_exp, role.manage_mem,
role.add_exp, role.role_id FROM groups
   LEFT JOIN group_member gm on groups.group_id = gm.group_id
```

```
   LEFT JOIN role on gm.role_id = role.role_id
   LEFT JOIN user on user.user_id = gm.ind_id
   WHERE groups.group_id = 1;
Aminah|1|1|1|1|role1
Govind|1|1|0|1|role2
Evan|1|0|1|1|role3
Dhruv|0|0|0|1|role4
Aly|0|0|0|0|role5
```

Save Group Permissions

```
> UPDATE role
  SET
      create_sg = 1,
      modify_exp = 1,
      manage_mem = 1,
      add_exp = 1
  WHERE
      role_id = 1
```

R11c: SQL Query (Testing with Production Data)

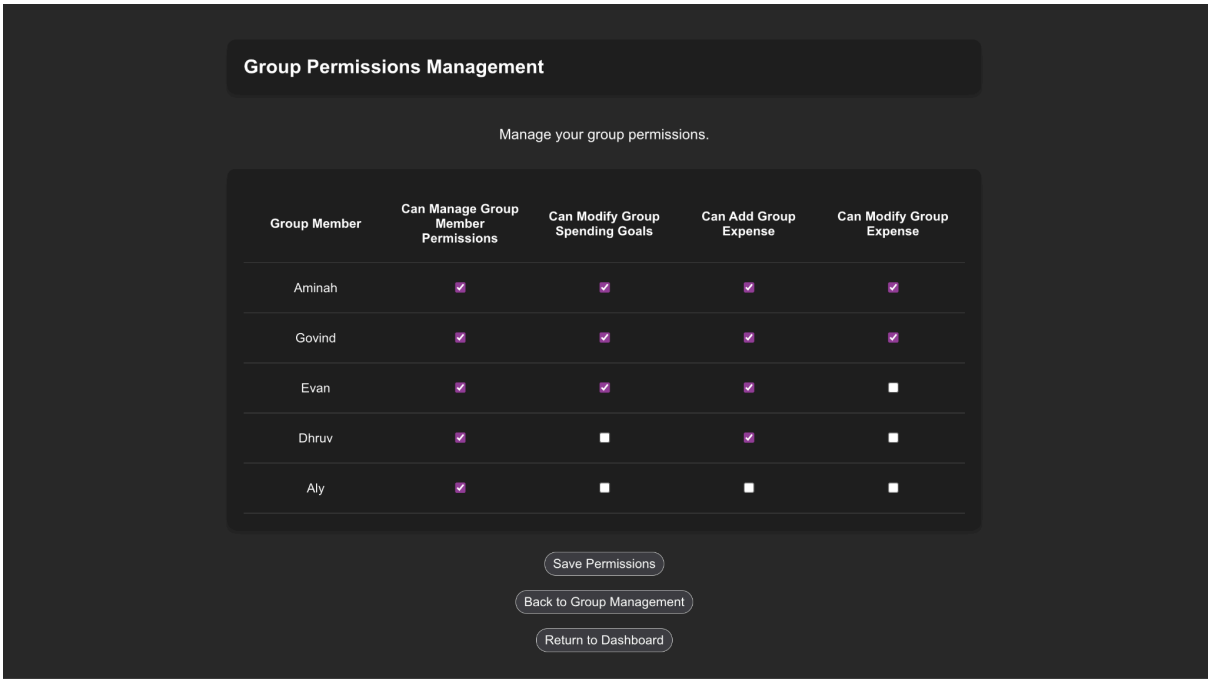
Check if the user is an Admin

```
|1|
```

Fetch Group Permissions

```
Aminah|1|1|1|1|1
Govind|1|1|1|1|2
Evan|1|0|1|1|3
Dhruv|0|0|1|1|4
Aly|0|0|1|0|5
```

R11d: Snapshot



R12: Expenditure vs. Allotted Budget (Fancy Feature 1)

R12a: Feature Interface design

This feature is accessed through the "Visualize Spending Patterns" card on the dashboard. It provides users with a visual representation of their financial habits by displaying a line chart plotting their total expenditures as well as their set budget goals for each category. This clear visual format allows users to easily assess how closely they are adhering to their financial plans. A detailed rundown by category is also provided, indicating how much users have spent either under or over their budget, helping them pinpoint areas for financial adjustment.

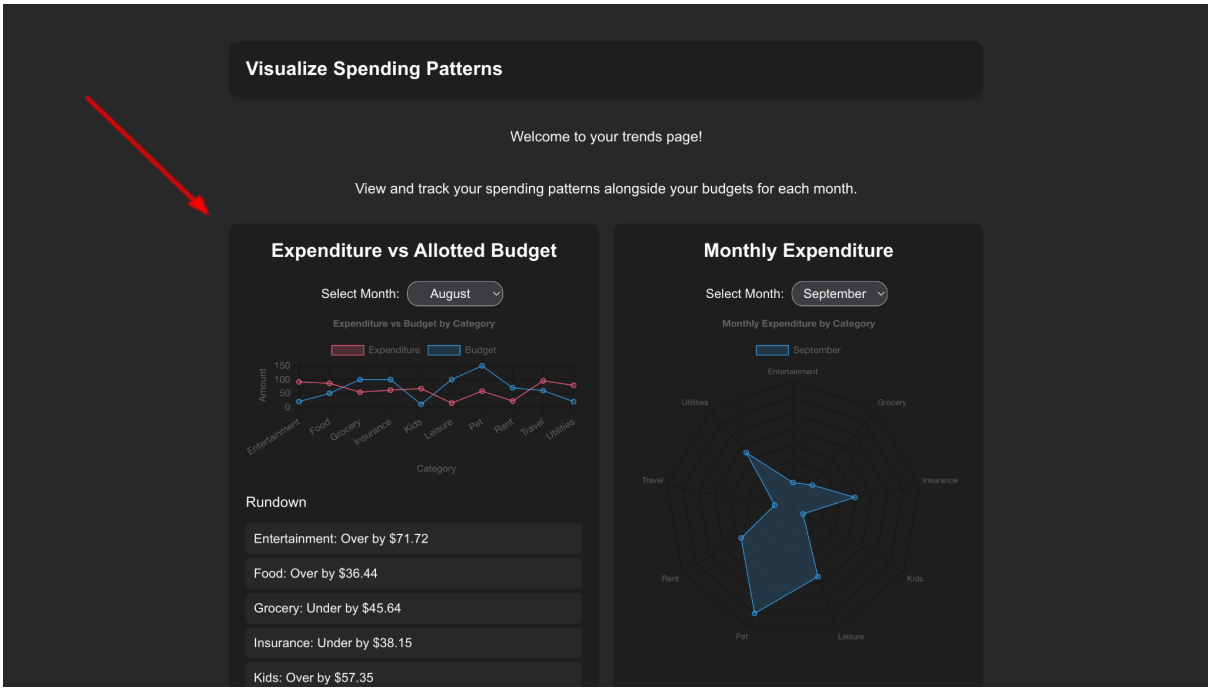
R12b: SQL Query (Testing with Sample Data)

```
> SELECT e.category, SUM(e.amount), sg.amount FROM expenses e
   LEFT JOIN spending_goal sg ON e.category = sg.category AND e.user_id =
sg.user_id WHERE e.user_id = 'I55555' GROUP BY e.category, sg.amount
Clothes|165.0|150.0
Food|4.78|150.0
Groceries|100.25|
Miscellaneous|35.0|200.0
Rent|780.0|900.0
```

R12c: SQL Query (Testing with Production Data)

```
Entertainment|176.33|20.0
Food|214.4|50.0
Grocery|69.0|100.0
Insurance|140.34|100.0
Kids|116.88|10.0
Leisure|135.84|100.0
Pet|198.83|150.0
...
```

R12d: Snapshot



R13: Monthly Expenditure (Fancy Feature 2)

R13a: Feature Interface design

This feature, also accessible through the "Visualize Spending Patterns" dashboard card, presents users with a radial chart that graphically depicts spending across various categories for a selected month. This visual tool helps users quickly identify which categories they spend the most on, allowing for an intuitive and immediate understanding of their spending habits and helping them to manage their monthly budget better.

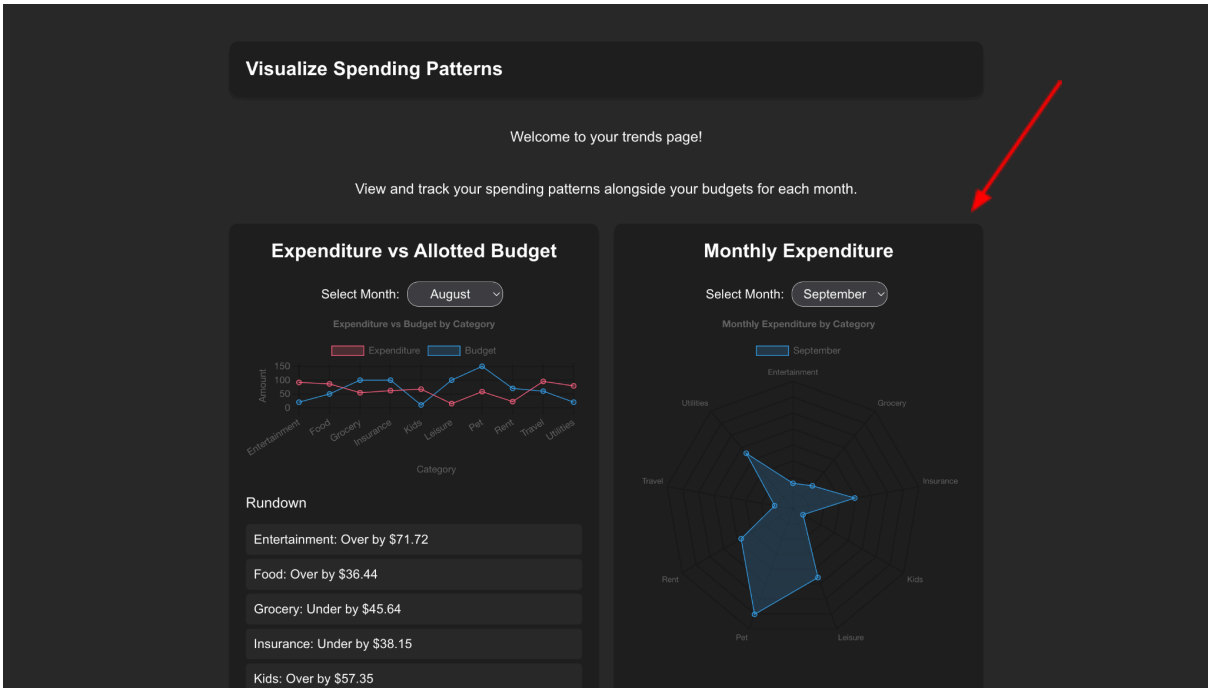
R13b: SQL Query (Testing with Sample Data)

```
> SELECT substr(date, 4, 2) AS month, category, SUM(amount) AS total_amount
  FROM expenses WHERE user_id = 'I55555'
  GROUP BY month, category ORDER BY month
4-|Clothes|322.0
4-|Food|299.78
4-|Groceries|400.25
4-|Miscellaneous|35.0
4-|Rent|780.0
```

R13c: SQL Query (Testing with Production Data)

```
4-|Entertainment|176.33
4-|Food|214.4
4-|Grocery|69.0
4-|Insurance|140.34
4-|Kids|116.88
4-|Leisure|135.84
...
```

R13d: Snapshot



R14: Smart Suggestions (Fancy Feature 3)

R14a: Feature Interface design

Clicking on this card leads users to a page offering personalized smart suggestions aimed at optimizing spending and saving habits. These recommendations are tailored to help users meet their specific budget goals more effectively.

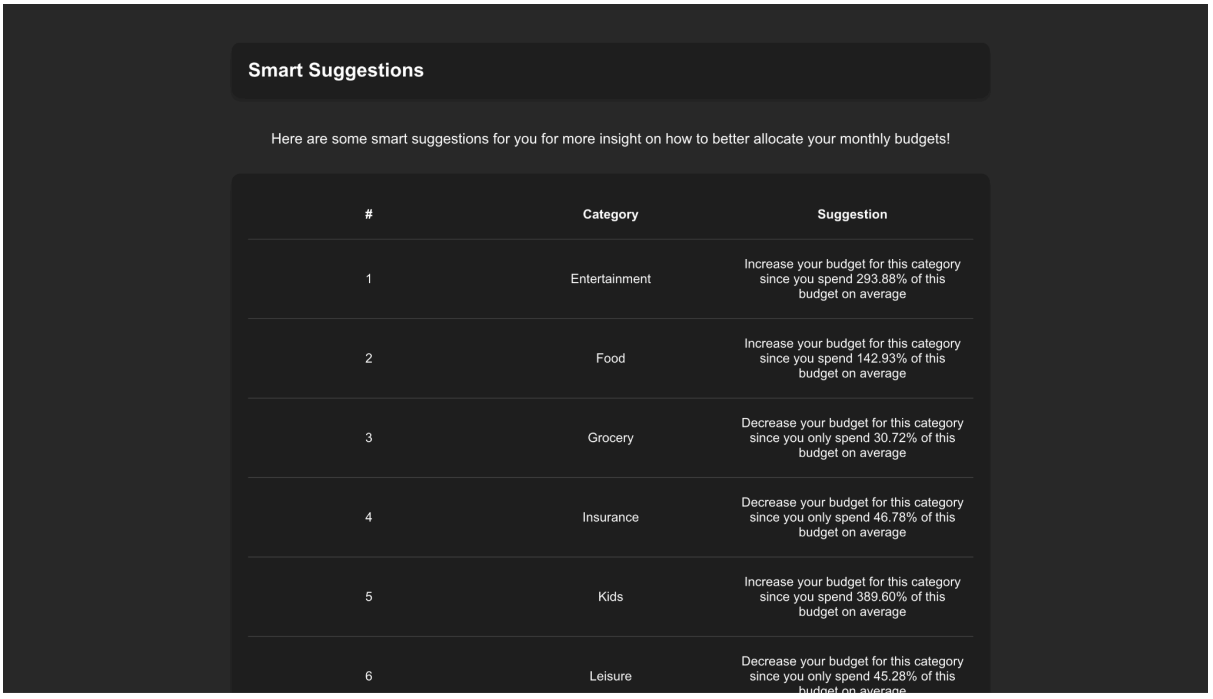
R14b: SQL Query (Testing with Sample Data)

```
> SELECT sg.category as category, sg.amount,
   AVG(ex.amount) as avg_spending, sg_avg.amount as avg_budget
   FROM spending_goal sg LEFT JOIN expenses ex ON ex.user_id = sg.user_id
   AND sg.category = ex.category AND ex.date >= DATE('now', '-3 months')
   LEFT JOIN (SELECT category, AVG(amount) AS amount FROM spending_goal
              GROUP BY category) sg_avg ON sg.category = sg_avg.category
   WHERE sg.user_id = ? GROUP BY sg.category
Entertainment|10000.0||5150.0
Food|80.0|10.0|201.25
Groceries|100.0|123.0|175.0
Personal Care|89730423074.0||29910141025.6667
```

R14c: SQL Query (Testing with Production Data)

```
Entertainment|20.0|29.3883333333333|206.666666666667
Food|50.0|35.7333333333333|350.0
Grocery|100.0|17.25|300.0
Insurance|100.0|23.39|200.0
Kids|10.0|29.22|270.0
Leisure|100.0|27.168|100.0
...
```

R14d: Snapshot

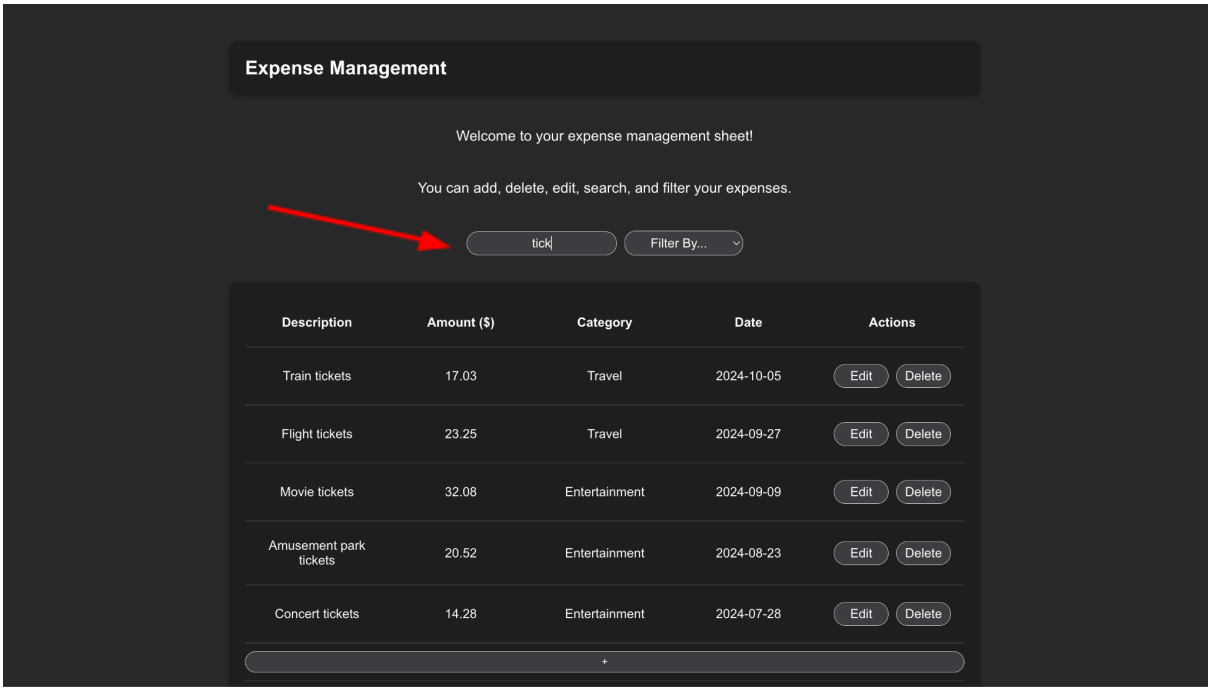


R15: Search Expenses (Fancy Feature 4)

R15a: Feature Interface design

On the Expense Management page, users can quickly locate specific expenses in real-time using a search bar. This feature is designed to simply enhance the frontend user experience by enabling immediate and straightforward access to expense records without the need for database queries.

R15d: Snapshot

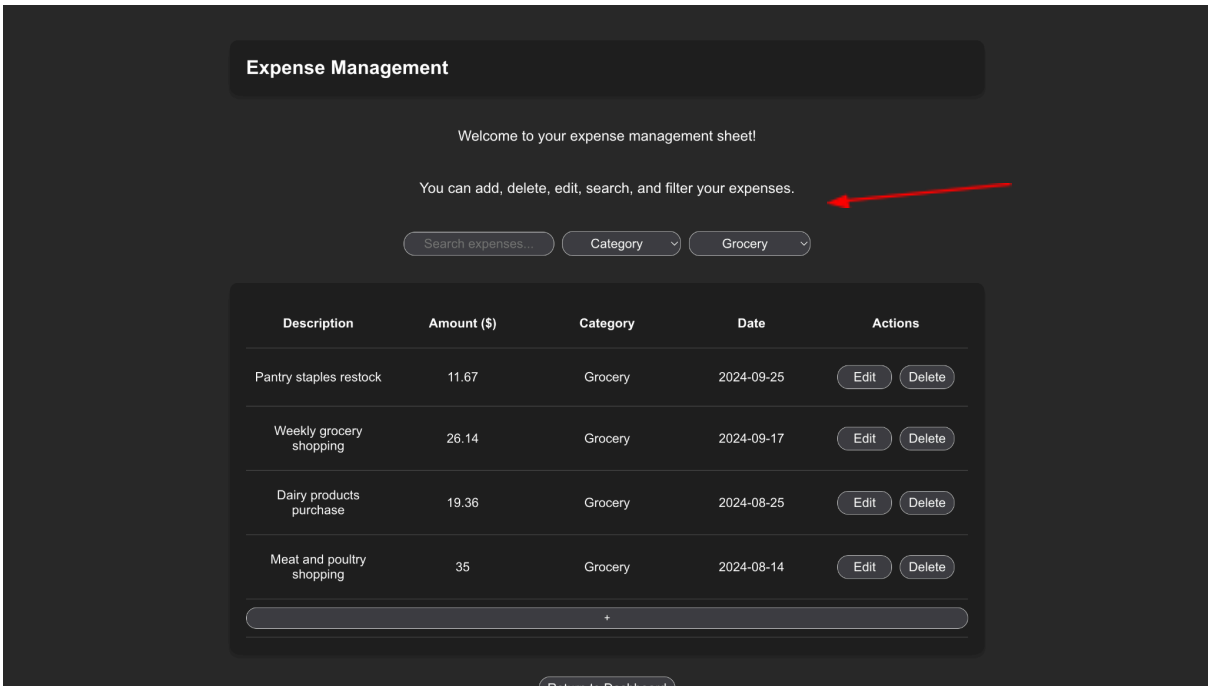


R16: Filter Expenses (Fancy Feature 5)

R16a: Feature Interface design

Users can also refine their view of expenses on the Expense Management page by setting various filters in the "Filter by…" dropdown menu. Options for filtering include setting lower or upper bounds on the amount, choosing specific categories, and selecting particular months or years for which they have previously recorded expenses. This tool was designed to enable effortless navigation and organization of each user's financial records according to their specific needs, without direct database queries.

R16d: Snapshot



R17: Members and Contribution

- Evan Warner: Feature Brainstorming, App Brainstorming, Description of Platform, Developed Expense Management Prototype, Developed Smart Recommendations, Developed Savings Leaderboard, Developed Group Management
- Govind Babu: Feature Brainstorming, App Brainstorming, Description of Application, E/R diagram, Relational Database model, Sample & production database
- Aminah Kirefu: Feature Brainstorming, App Brainstorming, Description of Application, E/R diagram, Relational Database model, Sample database
- Dhruv Mathur: Feature Brainstorming, App Brainstorming, Creation of GitHub Repo, Developed Feature to Log Budget Goals, Developed Trends
- Alicia Mathew: Feature Brainstorming, App Brainstorming, Description of Platform, Developed Database connection to frontend, Developed and edited Expense Management, Budget Management, Finalized and enforced frontend consistency on all app pages

Source Code Preparation

Smart Budget Tracker (GitHub Repo)