

THINGS NEEDED

1. User Login Functionality

The application should allow users to log in using a **valid email (username)** and password. Authentication is done by making a **GET** request to the following endpoint:

```
/users?username=example@gmail.com
```

If the username exists in the API data, the user is allowed to proceed to the dashboard. Otherwise, an error message is displayed.

2. Creating a New Expense

Users should be able to create a new expense by submitting a form. The data is sent using a **POST** request to the following endpoint:

```
/expenses
```

The request body should contain:

- **name**: the title or category of the expense
- **amount**: how much money was spent
- **description**: (optional) additional information about the expense

3. Viewing a Specific Expense

To view the details of an individual expense, the app should make a **GET** request to:

```
/expenses/{expenseId}
```

Here, `{expenseId}` is the unique ID of the expense selected by the user.

4. Listing All Expenses

The application should list all saved expenses by sending a `GET` request to:

`/expenses`

This will return a list of all expenses that have been created.

5. Deleting an Expense

Users should be able to delete an expense by sending a `DELETE` request to:

`/expenses/{expenseId}`

Where `{expenseId}` is the ID of the expense to delete.



PAGES / SCREENS TO BE DEVELOPED

Login Page:

This screen allows users to log in using their email address.

It sends a GET request to `/users?username=...` to verify the user.

If the email exists, the user is authenticated and redirected to the dashboard.

Home Page:

After login, the user sees this page.

It provides an overview of spending, summary charts, and buttons for:

- Creating a new expense

- Viewing all expenses

Create Expense Page:

This screen allows the user to input:

- Expense name
- Amount
- Description (optional)

It sends the data to `/expenses` using a POST request.

All Expenses Page:

This page shows a scrollable list of all the user's expenses.

Data is retrieved using a GET request to `/expenses`.

Each item in the list should be clickable.

Expense Detail Page:

When a user clicks an expense, this page opens.

It uses GET `/expenses/:id` to show all information about that expense.

It also includes a delete button, which sends a DELETE request to `/expenses/:id`.