COMPSCI 326 Web Programming

Team 18

CoinSync

3/28/25

Project Overview

CoinSync

Problem: Many people struggle with personal finance tracking and budgeting.

Solution: A simple user friendly web-based budget tracker that helps users categorize expenses and monitor savings goals. This will help the user get closer to financial freedom.

Features:

Add and categorize profits/expenses

Budget planning per month or by paycheck

Visual charts for spending analysis in months or years (which categories the most money is spent in)

Export reports as CSV (Used to output monthly data clearly and store easily)

Why This Project?
Managing personal finances is a common challenge for many people. From tracking daily expenses to planning monthly budgets, individuals often struggle to maintain financial discipline. CoinSync provides a simple yet effective solution to help users take control of their finances and hit their goals.

Team Members

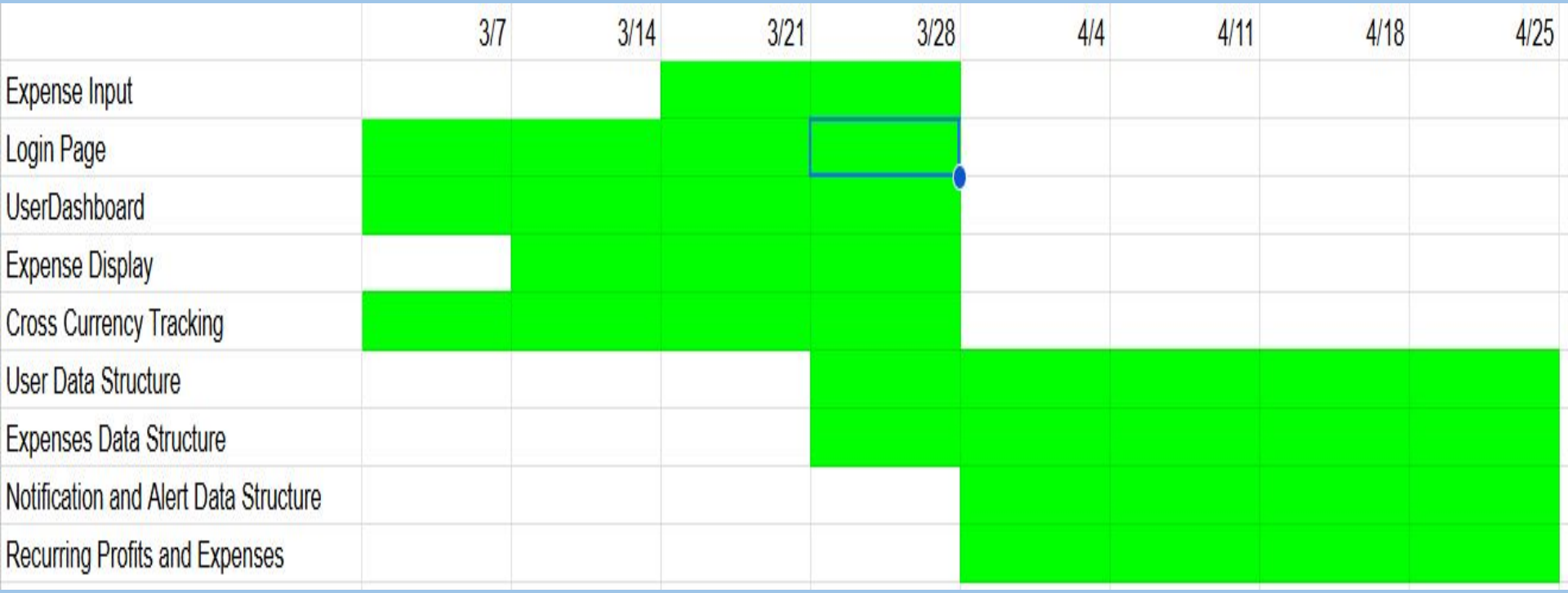
Kendrick Ong - Project Manager
Issues/Contributions: User data structure, Login Screen

Brian Tan - Timekeeper
Issues/Contributions: Expenses Data Structure, UserDashboard, Expenses Input, Git Folder Management

Nicholas Brojek - Notetaker
Issues/Contributions: Cross Currency Tracking, Recurring Profits/Expenses

Varun Pininty - Meeting Creator
Issues/Contributions: Notification & Alerts Data Structures, Expenses Display

Timeline


3/7
3/14
3/21
3/28
4/4
4/11
4/18
4/25
Expense Input
Login Page
UserDashboard
Expense Display
Cross Currency Tracking
User Data Structure
Expenses Data Structure
Notification and Alert Data Structure
Recurring Profits and Expenses

Assigned Work Summary: Kendrick Ong

Issues/Commits/PR: Project Manager, Login Page Development/PR, User Data Structure, Milestone #4 Git Organization

Description of Tasks Completed: Created the Login page HTML and styling with a header for an about us and home page that the user will land on at first. Organized our group's tasks under Milestone #4

PRs Closed: https://github.com/brtan8/CS326/pull/22, https://github.com/brtan8/CS326/pull/25

Unfinished Issues: https://github.com/brtan8/CS326/issues/5

Assigned Work Summary: Brian Tan

Issues/Commits/PR: Expenses Input, Timekeeper, Folder Organization Commits, UserDashboard PR, Inputs PR

Description of Tasks Completed: Created the organization of the folders in the GitHub. Also made the inputs for the user to add their expenses and category, along with the PR for that. Started to make the base of the UserDashboard that will clearly show the financial data of the user by showing graphs or grids in the future.

PRs Closed: https://github.com/brtan8/CS326/pull/24

Unfinished Issues: https://github.com/brtan8/CS326/issues/7

Assigned Work Summary: Nicholas Brojek

Issues/Commits/PR: Notetaker, Add CrossCurrencyTable PR, Basic RecurringPayments PR

Description of Tasks Completed: Added basic UI for a profits/expenses table where the user can input currency type & net profit/loss, added basic potential recurring payment UI where user can add/delete/edit weekly, biweekly, monthly, yearly recurring payments

PRs Closed:https://github.com/brtan8/CS326/pull/30
https://github.com/brtan8/CS326/pull/31

Unfinished Issues: https://github.com/brtan8/CS326/issues/16

Assigned Work Summary: Varun Pininty

Issues/Commits/PR: Meeting Organizer, Notification & Alerts Data Structure, Expenses Display Completed

Description of Tasks Completed: Created the initial UI for the Expenses Display that contains a pie graph with the key representing different categories for budgeting as well as table documenting purchases. Created Slide Deck necessary for Milestone #4

PRs Closed: https://github.com/brtan8/CS326/pull/35

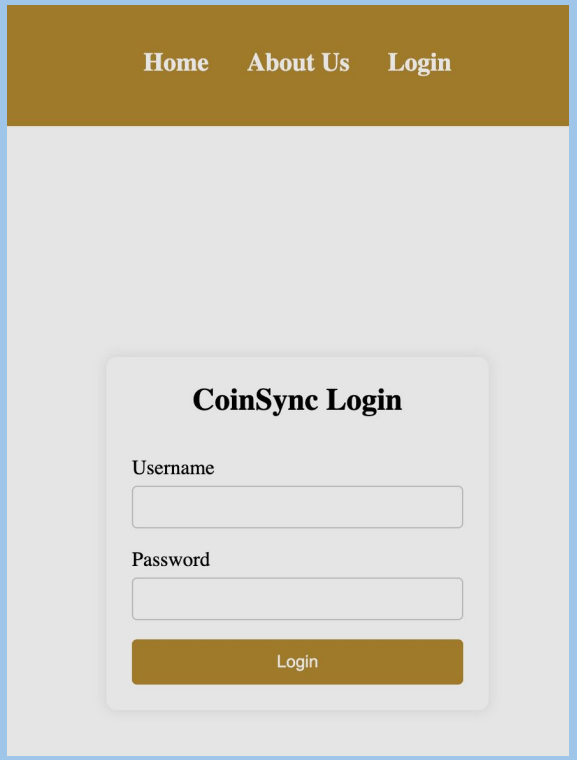
Unfinished Issues: https://github.com/brtan8/CS326/issues/13

Screenshots & Demonstration: Kendrick Ong

The header of the webpage is used for user navigation.

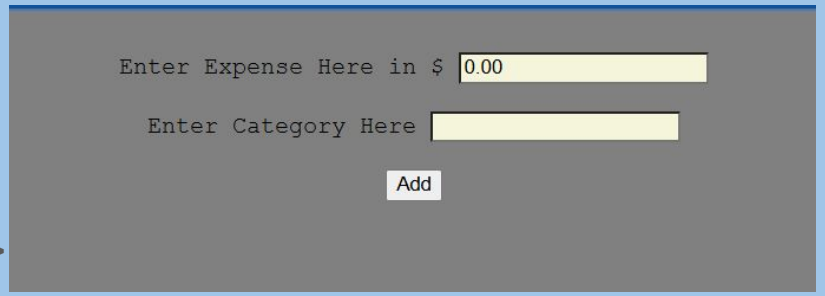
I added textboxs where a user can input their login information.

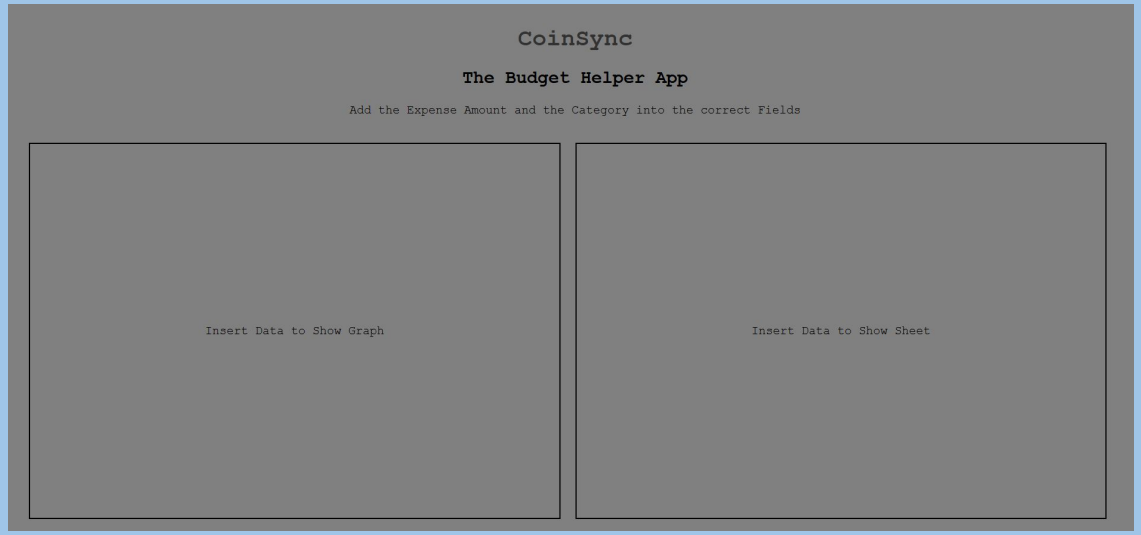
When the login button is Hovered, the button turns green.



Screenshots & Demonstration: Brian Tan

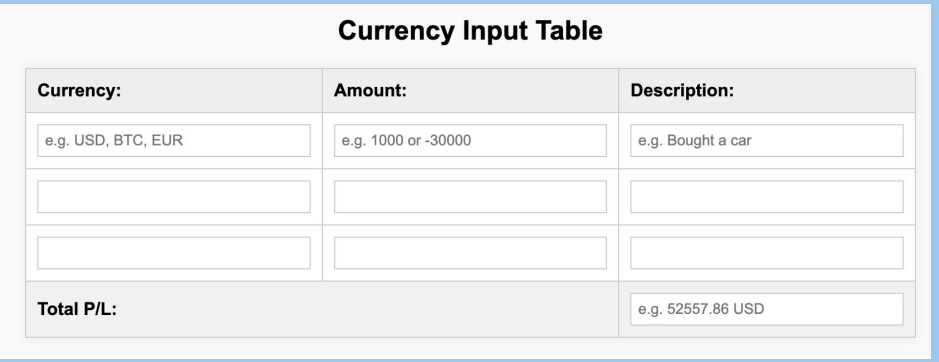
These are the ui for the inputs for our project. The expense input can only be numerical and the boxes/button changes color if the user is using it ->





<- This is the basis for the UserDashboard for our project. The bottom left will create a graph of the data and the right will be a list of the data.

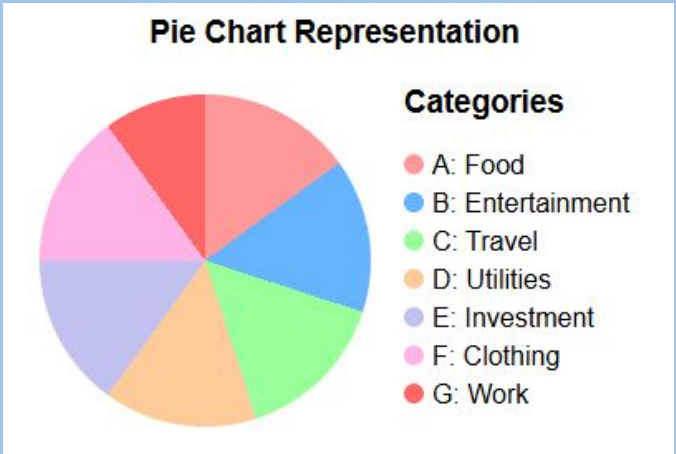
Screenshots & Demonstration: Nicholas Brojek



<- Basic currency input table where the user can add currency type, amount, description, and P/L output at end

<- Basic potential UI we can use for the user to add profits & expenses with amount, description, if it's recurring, frequency, etc

Screenshots & Demonstration: Varun Pininty



<- This is the UI for the Pie chart and the legend for the pie chart. It visually represents the proportion of each category/type of spending in proportion to the total amount of spending

A detailed expenses table that contains the name, price, date, and type of purchase for all purchases made ->

Detailed Expenses Table

Expense	Price	Date	Category
Purchase 1	$10.00	3/9/25	A
Purchase 2	$9.45	3/12/25	B
Purchase 3	$6.30	3/16/25	F
Purchase 4	$7.49	3/17/25	G
Purchase 5	$36.90	3/20/25	E
Purchase 6	$21.40	3/23/25	D
Purchase 7	$9.87	3/24/25	C

Code & UI Explanation: Kendrick Ong

The HTML of the header has anchor tags we can attach links to later in the project to have the user navigate through different pages.

The username and password input boxes were put into 1 div so that they would be styled together.

The CSS for the center container users display:flex to make the elements adjustable.

```
<header>
    <nav>
        <ul>
            <li><a href="#home">Home</a></li>
            <li><a href="#about-us">About Us</a></li>
            <li><a href="#login">Login</a></li>
        </ul>
    </nav>
</header>
```

```
.center-container {
    width: 100%;
    height: calc(100vh - 100px);
    display: flex;
    justify-content: center;
    align-items: center;
    padding-top: 20px;
}
```

```
<div class="center-container">
    <div class="login-container">
        <h2>CoinSync Login</h2>
        <form>
            <div class="input-group">
                <label for="username">Username</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="input-group">
                <label for="password">Password</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button type="submit" class="login-btn">Login</button>
        </form>
    </div>
</div>
```

Code & UI Explanation: Brian Tan

```
<div>
  <label for="expenseInput">Enter Expense Here in $</label>
  <input type="number" id="expenseInput" class="input" value="0.00">
  <br><br>
  <label for="categoryInput">Enter Category Here</label>
  <input type="text" id="categoryInput" class="input">
  <br><br>
  <button id="addButton" class="button">Add</button>
```

```
.button:hover {
  background-color: greenyellow;
}

.input {
  background-color: beige;
}

.input:focus {
  background-color: greenyellow;
}
```

These pieces of code create the inputs and the buttons that are necessary for our budget app to add data from the user. For the expense input I ensured that only numerical data could be entered and put the default value to 0.00. All inputs also have a label to go with it. The button and the inputs turn greenyellow when the user is hovering/inputting data.

This code will help the UserDashboard be more complete because the user can add their financial data.

During the implementation, one challenge was making the UI feel less plain so for the solution I added a changing background color.

Code & UI Explanation: Nicholas Brojek

```
<label>Amount</label>
<input type="number" placeholder="e.g. 15.99" />

<label>Recurring?</label>
<select>
  <option value="no">No</option>
  <option value="yes">Yes</option>
</select>

<label>Frequency</label>
<select>
  <option value="monthly">Monthly</option>
  <option value="weekly">Weekly</option>
  <option value="biweekly">Biweekly</option>
  <option value="yearly">Yearly</option>
</select>
```

```
      <div class="flex-row">
        <button class="btn btn-edit">Edit</button>
        <button class="btn btn-delete">Delete</button>
      </div>
    </div>

    <button class="btn btn-add">+ Add New Entry</button>
```

<- These pieces of code lets the user adjust their payments/profits to be recurring or not with the amount, frequency, etc and there are 3 buttons added for the user to either edit an existing P/L, add a new one or delete one. One issue I completely forgot was letting the user know to do +/- before their number to indicate whether it is a profit or expense, so I just committed a small note on that directly in main

```
<tbody>
  <tr>
    <td><input type="text" placeholder="e.g. USD, BTC, EUR" /></td>
    <td><input type="number" placeholder="e.g. 1000 or -30000" /></td>
    <td><input type="text" placeholder="e.g. Bought a car" /></td>
  </tr>
  <tr>
    <td><input type="text" /></td>
    <td><input type="number" /></td>
    <td><input type="text" /></td>
  </tr>
  <tr>
    <td><input type="text" /></td>
    <td><input type="number" /></td>
    <td><input type="text" /></td>
  </tr>
</tbody>
<tfoot>
  <tr class="total-row">
    <td colspan="2">Total P/L:</td>
    <td><input type="text" placeholder="e.g. 52557.86 USD" /></td>
  </tr>
```

<- this piece of code creates a basic table for the user to input their currency type, amount, and add a description, with an entry set aside to calculate the profits/losses according to the user's input info

Code & UI Explanation: Varun Pininty

```html
<div class="chart-container">
    <div class="pie-chart"></div>
    <div class="pie-chart-legend">
        <h3>Categories</h3>
        <ul>
            <li><span class="category-a"></span>A: Food</li>
            <li><span class="category-b"></span>B: Entertainment</li>
            <li><span class="category-c"></span>C: Travel</li>
            <li><span class="category-d"></span>D: Utilities</li>
            <li><span class="category-e"></span>E: Investment</li>
            <li><span class="category-f"></span>F: Clothing</li>
            <li><span class="category-g"></span>G: Work</li>
        </ul>
    </div>
</div>
```

The code on the left are the key parts of the code for the pie chart and the legend. In the first image you can see how I created the legend with multiple categories while the second image is the css for the pie chart. An issue I faced was trying to get separate colors for each “pie slice”, as I found it difficult to split the large pie into individual slices with unique colors.

```css
.pie-chart {
    width: 200px;
    height: 200px;
    border-radius: 50%;
    background: conic-gradient(
        #ff9999 0% 15%,  /* A */
        #66b3ff 15% 30%, /* B */
        #99ff99 30% 45%, /* C */
        #ffcc99 45% 60%, /* D */
        #c2c2f0 60% 75%, /* E */
        #ffb3e6 75% 90%, /* F */
        #ff6666 90% 100% /* G */
    );
    margin-right: 20px;
}
```

The code on the right is part of the code for creating the expense table in which we have four main categories: expense, price, date, and category in which every row represents the data of a purchase made

```html
<h3>Detailed Expenses Table</h3>
<table>
    <tr>
        <th>Expense</th>
        <th>Price</th>
        <th>Date</th>
        <th>Category</th>
    </tr>
    <tr>
        <td>Purchase 1</td>
        <td>$10.00</td>
        <td>3/9/25</td>
        <td>A</td>
    </tr>
    <tr>
        <td>Purchase 2</td>
        <td>$9.45</td>
        <td>3/12/25</td>
        <td>B</td>
```

Challenges & Insights: Kendrick Ong

Reflection: I thought it was difficult to not have an entire roadmap planned out and only just pieces of the project. I didn’t think it was wise to not use any JS since our project should have both together, it just makes more work to integrate everything later.

Key Takeaways: Having a big picture idea is good to know where you are going with the project and how everything fits together. This ensures everyone on the team is on the same page.

Challenges & Insights: Brian Tan

Reflection: I found it difficult to add files and work with Git but as I started to use it more it became easier to share files with my team and communicate with them more effectively.

Key Takeaways: Working in a collaborative team allows for more efficient work flow because if I get stuck I can ask the team members for input. It also helps to work on separate files so that multiple tasks can be worked on the same time without waiting on other components.

Challenges & Insights: Nicholas Brojek

Reflection: I am pretty inexperienced with git in vscode as well so I had a bit of trouble. But once I got the hang of it, it was pretty fun to create the branches, commit my changes to the branch and then send a pull request to main & merge it.

Key Takeaways: Git is actually really fun & conducive to productivity . But coordinating a large project does take a good team with goals laid out well.

Challenges & Insights: Varun Pininty

Reflection: I think one thing that hindered me was focusing too much on the end goal. In my head I was thinking about how the pie chart data and expense data table should be linked with data and how tables/graphs would change depending on the button. I think simply starting to write code and going through the iterations helped me focus on first ensuring that I have the basic UI completed and I could move on.

Key Takeaways: Having my teammates to talk to actually helped me a lot in not only validating ideas that I had, but also getting new ideas from them when I was stuck allowed me to progress faster and not get stuck on one particular thing.

Future Improvements & Next Steps: Kendrick Ong

Improvements: In the login page, it should properly check a database for valid user credentials.

The login page should include a link to a page that they can reset their password on.

We should add an about us to tell users about the project on a page a user can access without logging in.

Valid Login Link: https://github.com/brtan8/CS326/issues/27

Forgot Password Link: https://github.com/brtan8/CS326/issues/28

About Us Link: https://github.com/brtan8/CS326/issues/29

Future Improvements & Next Steps: Brian Tan

Improvements: In the expense input, it should default .00 at the end if it is not specified.

We should be able to make an excel sheet to organize the data for the user. In order to export it, we would need an excel sheet export button.

If we are organizing the data we should show a list of expense data to the user within the project.

Expense Input Improvement Link: https://github.com/brtan8/CS326/issues/19

Export Excel Sheet Task Link: https://github.com/brtan8/CS326/issues/20

Data Sheet Display Task Link: https://github.com/brtan8/CS326/issues/21

Future Improvements & Next Steps: Nicholas Brojek

Improvements: Now that the basic table UI is set, the backend mechanics that convert different currencies with their amounts to other currencies needs to be setup

A basic financial rating scale should also be added to let the user know if they are on the right track in terms of progressing with their financial management

Also an option to pause all financial management should be incorporated just in case of the user wanting to take a break

Cross Currency Conversion backend link: https://github.com/brtan8/CS326/issues/32

Finance Rating Scale: https://github.com/brtan8/CS326/issues/33

Pausing Finance Management: https://github.com/brtan8/CS326/issues/34

Future Improvements & Next Steps: Varun Pininty

Improvements: Simply having a pie graph doesn't make it easier for the user to gain information, by having more graphs like line graphs that demonstrate spending overtime

We could also allow for quick edits to the expense table such as changing the category or changing amount, for ease of use

Also add a comparison tool to see how your spending was compared to set budget limits to visualize amount of money spent and budget

Adding More Graphs Link: https://github.com/brtan8/CS326/issues/36

Quick Edits to Expense Table Link: https://github.com/brtan8/CS326/issues/37

Comparison Tracker for Budget Limit vs Actual Spending Link: https://github.com/brtan8/CS326/issues/38