|  |  |  |
| --- | --- | --- |
| Academic Module | Portfolio Submission | Assessment Type |
| Collaborative Development (5CS024) |  | Individual Report |

[Expenses Tracker – Developer]

Student Information

Name: Supriya Gurung

Section: [L5CS26]

Project Group: [Group- 26]

Module Leader: [Yogesh Bikram Shah]

Assigned Tutor: [Ankit Tamrakar]

Date of Submission: [2025-05-19]

**Acknowledgement**   
I'd like to convey my heartfelt gratitude to our module leader and teachers for providing us with this wonderful opportunity to learn and grow via this project. The experience has considerably enhanced both my technical and human skills, such as time management, deadline compliance, and fruitful teamwork.

1. **Self-appraisal form**

|  |  |  |  |
| --- | --- | --- | --- |
| Student Number | 2431364 | Name | Supriya Gurung |
| Project | Expenses Tracker | Date | 2025-05- |
| Role | Developer | Team | L5CS26(Group-2) |
| Sprint( 1 or 2) | 2 |  |  |

# **Personal objectives – performance management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Objectives** | | **Evidence provided** |  | **Evaluation** | |
| 1.Implementing Functional Requirement | | The project was developed with features, including saving targets, daily savings, progress updates and targeted notifications which updated smoothly and automatically without having to refresh the page. The app uses API calls to access the data from the backend. Using Git made it simpler for me to manage and track all the changes made to the code, repair issues and improve how my team members worked together. |  |  |  |
|  | *Tutor feedback:* | | | | |
| 1. The program has two forms: one for updating existing goals by adding saved amounts to efficiently track progress, and another for establishing new financial target goals with parameters like name, amount, and date. | | By inputting information such the goal name, the desired savings amount, and the anticipated date of completion, the user can create a new financial objective. This aids them in establishing specific financial goals. Second, the user can contribute the money they have saved thus far to their current goals. This enables individuals to monitor their development over time and maintain motivation as they approach their financial goals. |  |  |  |
|  | *Tutor feedback:* | | | | |
| 1. Implement savings progress tracking logic | | The program's built-in logic allows the user to monitor their progress in saving. They can see how much they have saved and how much they still have left for each goal. The application automatically updates the progress when they contribute saved amounts to a goal. This promotes regular saving practices and makes it easy for users to understand how close they are to achieving their financial goals. |  |  |  |
|  | |  |  |  |  |
|  | *Tutor feedback:* | | | | |
| 1. Savings Goal Notifications   Implementing the design for notification system | | The user can create and get notifications regarding their savings goals with the "Savings Goal Notifications" feature. The user receives notifications from this system when significant events pertaining to their objectives take place, such as impending deadlines, progress reports, or reaching a milestone. A notification will appear on the device after the user reaches the desired amount. This helps the user stay motivated, track their progress, and stay on course to reach their financial goals. |  |  |  |
| 1. Bug Fixing | | A specific GitHub bug pertaining to the Savinggoals and dashbooard page was given to me. I successfully fixed the code bug after examining the problem and determining its underlying cause. I made sure the functionality now functions as intended by pushing the fix to the repository and committing the modifications. After testing and a code review confirmed the fix, the issue was closed on GitHub. |  |  |  |

# **Collaboration Document**

**Evidence of good collaboration**

## **Good communication and file sharing**

## The most practical and easy way for us to communicate with the team was through messages. We accomplished this by creating a group chat with our members in which we were free to share our ideas and opinions. We communicated most of the data and codes using Viber, and we utilised Jira software to convey our objectives and to-do activities. The file was posted to git for code review, merging, and finalisation when the task was finished.

A screenshot of a chat

AI-generated content may be incorrect.

***Talking about our project in a Google Space group chat***

A screenshot of a chat

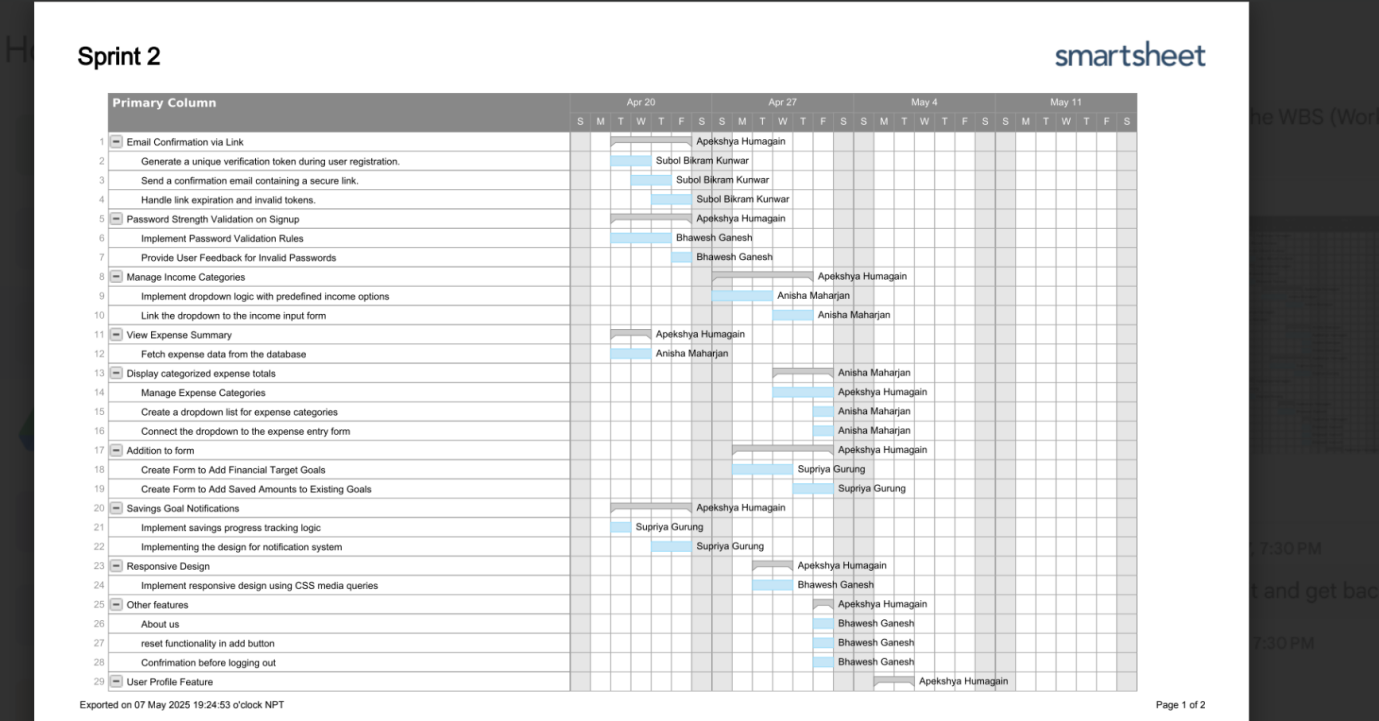
AI-generated content may be incorrect.

***Figure 8 Communication through google space***

A group of people in a classroom

AI-generated content may be incorrect.

***Figure 9 Communication through physical meeting***

******

***Figure 10 gantt chart***

It is simpler for us to monitor our work and project progress since we were told to use Jira for collaboration because all project details, such as the project manager's deadline and requirements, are displayed here.

A screenshot of a computer

AI-generated content may be incorrect.

***Figure 11 Jira backlog***

A screenshot of a computer

AI-generated content may be incorrect.

***Figure 12 Jira Board***

# **Appendix A**

## **Implementing functional requirements**

### **Use of Version control** **(screenshots of logs)**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.***Figure-1 git commit***

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

***Figure-2 Github files and folder***

A screenshot of a computer

AI-generated content may be incorrect.

***Figure-3 Github issue tracking***

A screenshot of a computer

AI-generated content may be incorrect.

***Figure-4 Github Branches***

# **Appendix B**

1. **Create form to add financial target goals**

**A screen shot of a computer program

AI-generated content may be incorrect.**

I handled the loading of the DOM content by arranging for the event listener to run once the DOM is ready. As a result, all the components needed are right at hand as you add functions. As soon as the page appears, loadGoals() and updateTotalSavings() action their code to display your goals and the total savings. Moreover, I set up event listeners so that when the "Add Goal" or "Add Saving" buttons are clicked, the app runs the corresponding function. I listened for clicks on the delete buttons of the table and the clicked value becomes the id for my deleteGoal function. I used e.preventDefault() to prevent the form from submitting and then saved the entered details for the goal name, target amount and deadline in the function addGoal(). If any entry is left blank, a warning shows the user and asks them to fill out all the details. After checking for a valid input, the data is delivered to the server using the fetch API by making a POST request. The information on ‘to save’ and ‘payment details’ are sent as JSON to SavingGoal.php?action=add. After a successful response, the input fields are set to blank and both methods are run to reload the UI. If anything goes wrong, the server will display an alert with the message. This allows goals to be added dynamically, providing users with extra convenience without having to refresh the page.

1. **Create form to add Saved Amounts to Existing Goals**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

# **Appendix C**

**Implement savings progress tracking logic**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

I implemented the addGoal and addSaving functions in my code to allow users to set up their saving goals and save money manually each day. I have not allowed the default button behavior within addGoal and instead got the input values from the goal name, target amount and deadline elements in the form. Should any field be left unfilled, the user gets a message to fill it in. In other cases, I will use an HTTP POST to transfer the goal details in JSON format to Savinggoal.php. When success is returned by the server, both the input fields are emptied and two functions are invoked: loadGoals() for a new goal list and updateTotalSavings() for an updated total savings. The purpose of addSaving is to allow the user to add how much money they want to save every day. Then it saves the amount and goal ID, ensures everything is filled in and sends data to the server. Upon being successful, the tool refreshes your goals and adds the amount saved to your total. The code clearly separates saving goals from entering daily amounts and it also checks and ensures any data entered does not cause problems.

# **Appendix D**

1. Savings Goals Notification

A screen shot of a computer screen

AI-generated content may be incorrect.

Within loadGoals(), some code has been added to notify users whenever a savings goal is achieved. The function grabs a list of previously alerted goals from the browser’s localStorage by using the key alertedGoals. It is used to ensure that a user does not get the same notification twice. While the code does not define how the check is completed, this setup suggests elsewhere there are conditions that monitor the progress a user has and the time remaining before the end date. After the condition is met, a notification (for example, an alert or message) can occur and the goal ID should be placed in the alertedGoals array in localStorage. Therefore, the user only gets one timely notification for every goal discussed. As a result, users are engaged to keep saving and recognize their achievements without feeling annoyed by too many texts.

1. Implementing the design for notification system

A screenshot of a computer

AI-generated content may be incorrect.

A computer code on a black background

AI-generated content may be incorrect.

The code here ensures that a celebratory alert is shown on the screen after the user manages to save the set goal. After receiving the goals data and processing it, the code cycles through every goal with a forEach loop. The app reviews if more has been paid up than needed, so it announces that the goal has been met. It confirms that this goal’s ID does not exist in the alerted array which comes from localStorage. Should the user manage to achieve their goal without being alerted, a notice pops up to celebrate what they got done. The ID of the goal is put into an array and saved in localStorage so the alert only displays for each goal once. Because of this, alerts appear less frequently and motivate people to keep using the app when they notice their achievements.

**Appendix E**

Bug Fixing

Bug No.1

A screenshot of a computer

AI-generated content may be incorrect.

The total savings amount was not showing up on the total saving goal page since the id of the total savings amount was not defined in the goals.js file.

A close-up of a person

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

I have therefore changed the goals.js file's id name. This resolved the issue with the saving amount not updating.

Bug No.2

A screenshot of a computer

AI-generated content may be incorrect.

My problem was that I don't have a specific location to save individual contributions, so when users attempted to add a manual saving, the amount wasn't being recorded appropriately. Instead of updating the goal's total, each manual deposit just disappeared.

A screen shot of a computer code

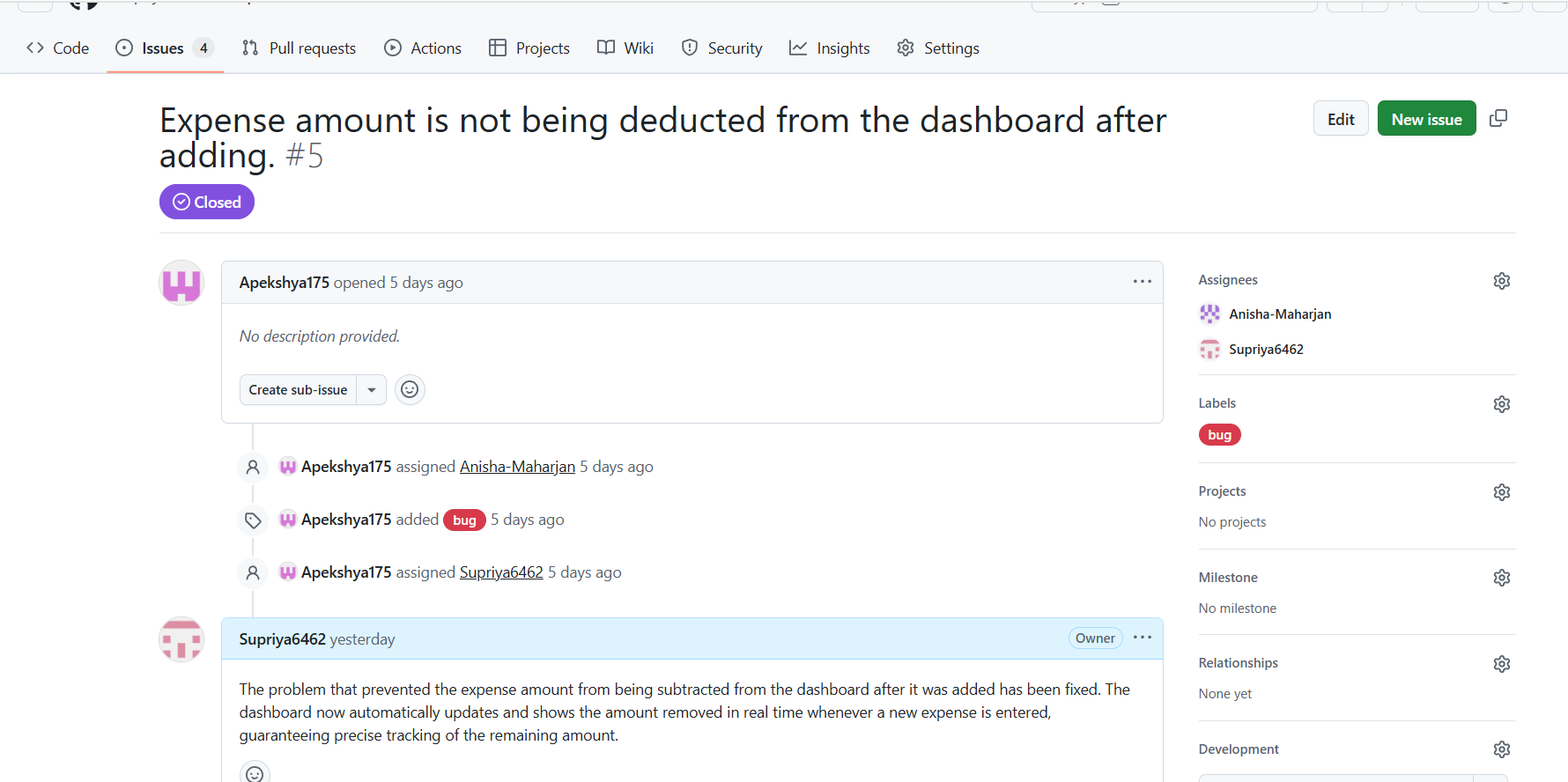
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

My PHP now grabs my goal ID and amount and adds them as a new row into the contributions table whenever I click "Add Saving," which is how I resolved the issue. To put it another way, each manual saving is treated as a separate contribution entry rather than attempting to update the goal record directly; hence, the total savings for any goal is simply the sum of all of its contribution rows. This greatly simplifies and improves the reliability of adding, monitoring, and auditing every deposit.

Bug No.3



The problem was that` i never deducted the complete expenses; therefore, the "budget" value was always too high, and the dashboard's total budget merely displayed income less savings.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

In order to ensure that the reported total budget accurately includes all outflows, I modified the budget computation in the updateSummary() function to deduct both expenses and savings from income (const budget = income - expense - saving;).

Personal Development Document: