

Project Plan for Expense Tracker

Group 8 Members (with GitHub username):

Kenée Shepherd (iamkenée)
Rafia Sajjad (rafiassajjad)
Shakeeb Rahman (shakeebur)
Hamza Srhiri (Hamzas61)
Sheeka Shee (SheekaShee)

Stakeholders

Professor. Essa for CIS 434

Requirements & Specifications

- Web Application using React for a user to track their expenses.
- Expense Tracker with features:
 - Running Balance
 - Income (in green / positive)
 - Expenses (in red / negative)
 - History (shows income and expenses history in green and red respectively)
 - Add new transaction section with fields for “Title” and “Amount”, and an “Add Transaction” button
- Other notes:
 - Initial screen shows \$0 for balance, income, and expenses as well as a clear history section.
 - Warning message for insufficient balance if deduction/expense is greater than the balance.
 - In the “Amount” field, if the sign is not specified then it is considered as positive income by default.
 - Web App should store a user’s data so if it is closed and reopened, previous data should be displayed.

Deliverables

Demonstration of Web App in class
Submitting all program files as one submission per group on Blackboard

Software & Programming Languages

Visual Studio Code
GitHub
JavaScript
React
Node.js
HTML
CSS

Group Member Task Assignment

Kenée and Rafia = Building and Testing app, Front-End design, Initial State
Shakeeb = Add Transactions, Modifying Items
Sheeka = Display Balance and Income
Hamza = Deduct Expenses, Removing Items

Timeline/Schedule

Weekly discussions will be held on discord or after class

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
01/30	01/31	02/01	02/02	02/03	02/04	02/05
02/06 Discussion of Project Plan	02/07	02/08 Tutorial on HTML & CSS	02/09	02/10 Tutorial on GUI Transaction feature added	02/11	02/12
02/13 Updated Project Plan Discussion Quiz	02/14	02/15 Tutorials on react for data storage Transaction name and math added	02/16 Added alert if expense entered when balance < 0	02/17 VS Code and Github access issues sorted Edits pushed to Git with update of readme	02/18	02/19
02/20 Discussion regarding Project 1 updates or issues	02/21 Local storage solution found to save transactions Front end styling	02/22 Expense and Income boxes Refine local storage options	02/23	02/24 Lab Quiz	02/25 Work on expense and income boxes Refine front end layout Add clear history button Allow decimal values	02/26
02/27	02/28 Add group member names to project layout Update readme	03/01	03/02 Master branch made the default branch in GitHub	03/03	03/04	03/05
03/06 Demo of completed web app and submit code files via blackboard (one submission per group)	03/07	03/08 Midterm	03/09	03/10	3/11	3/12

Milestones and Deadlines

Monday February 6th → Discussion of Project Plan

Monday February 13th → Chance to update Project Plan

Monday February 20th → Discussion of progress and issues with Web App

Monday March 6th → Demonstration of Completed Web Application

Reference Materials & Other Notes

Software requirements include Windows operating system, web browser, Java Runtime Environment (JRE), web application server such as Apache Tomcat, and Java Connector but we haven't ruled out utilizing Android studio to build the application as well.

We have assessed the engineering requirements process for this project and have decided that the elicitation and analysis include a web app to track user expenses from inputted information using react. We will design an application that allows a user to input their cash flow and outflow. We will use the UI for the appearance and layout of the web app and to handle the math equations and display the user's income, expenses, and current balance. The app must be responsive and resize correctly on mobile devices.

Specification requirements of "user login", "adding income", "subtracting expenses", "modifying items", "removing items" and "displaying current balance" while ensuring green transactions when adding and red transactions when deducting. We will implement warning statements to handle purchases that are larger than the balance and transactions denied from insufficient funds in a consistent location and style. Users should be able to obtain their expense history to check for accuracy of the data they have entered. We will add a navigation system to allow a user to move within the app and validate the form input fields.

Through requirements validation, we will conduct development testing of our model's realism, consistency, and completeness. We will ensure that the user design and functionality is appropriate and easy to use through system testing. We will test the model against different transaction scenarios to check that the model produces the appropriate results for acceptance testing.