

## SW Engineering CSC648-848-01 Summer 2025

### Limóney (Financial Budgeting Webapp)

#### Team 02

#### Milestone 4

8/5/25

Name	Role
Emily Perez (eperez@sfsu.edu)	Team Lead, Database Administrator, Github Master
Ishaank Zalpuri	Frontend Lead, UI Designer, Frontend Developer
Andrew Brockenborough	Technical Writer
Dani Luna	Frontend Lead, UI Designer, Frontend Developer
Jonathan Gonzalez	Software Architect, Frontend Lead, UX Specialist,
Gene Orias	Backend Lead

Milestone	Version	Date
Milestone 4	Version 2	8/5/25
Milestone 4	Version 1	7/31/25
Milestone 3	Version 2	7/31/25
Milestone 3	Version 1	7/22/25
Milestone 2	Version 2	7/10/25
Milestone 2	Version 1	7/3/25
Milestone 1	Version 2	7/2/25

Milestone 1	Version 1	6/17/25
-------------	-----------	---------

# Table of Contents

<b>Table of Contents.....</b>	<b>2</b>
<b>Product Summary.....</b>	<b>3</b>
Product Name:.....	3
Final Functional Requirements (P1 and P2).....	3
Unique Features.....	6
Deployment URL: <a href="https://limoney.org">https://limoney.org</a> .....	6
<b>Usability Test Plan.....</b>	<b>7</b>
Test Objectives.....	7
Test Setup.....	7
Test Functions.....	7
1. Reimbursement Effectiveness Table.....	8
2. Currency Converter Effectiveness Table.....	8
3. LemonAid AI Effectiveness Table.....	9
4. Budget Visualization Effectiveness Table.....	10
5. Account Tracking Effectiveness Table.....	10
6. Transactions Search Effectiveness Table.....	11
Likert User Satisfaction.....	11
<b>QA Test Plan.....</b>	<b>13</b>
Selected Non-Functional Requirements.....	13
QA Test Table 1. Performance.....	13
<b>Code Review.....</b>	<b>19</b>
Coding Standards.....	19
Github Code Review.....	19
External Review.....	27
<b>Security Practices.....</b>	<b>28</b>
Assets Protected.....	28
Password Encryption.....	28
Input Validation.....	28
<b>Self-Check: Adherence to Non-Functional Specs.....</b>	<b>31</b>
<b>List of Team Contributions.....</b>	<b>34</b>

[Limoney Inc.](#)

# Product Summary

Product Name:

Limoney

## Final Functional Requirements (P1 and P2)

### Priority 1

#### 1. User

- 1.1. A user shall be able to create an account.
- 1.2. A user shall be able to securely log in to their account.
- 1.3. A user shall be able to recover or reset their password.
- 1.7. A user shall be associated with one or more accounts.
- 1.10. A user shall be able to view and edit their financial data history

#### 2. Account

- 2.1. An account shall store income entries for a user.
- 2.2. An account shall store expense entries for a user.
- 2.4. An account shall track a balance value.
- 2.5. An account may be connected to one or more banks.
- 2.7. An account shall have a transaction history log.

#### 3. Bank

- 3.1. A bank shall be linked to one or more user accounts.
- 3.2. A bank shall provide transaction data to the system.

#### 4. Transaction

- 4.1. A transaction shall be classified as income or expense.

4.3. A transaction may be manually entered or synced from a bank.

## **5. Subscription**

5.1. A subscription shall be tied to a recurring transaction.

## **6. Budget**

6.1. A budget shall be created for one or more spending categories.

6.2. A budget shall belong to one and only one account.

6.3. A budget shall allow setting and tracking of goals.

## **13. ReimbursementRequest**

13.1. A reimbursement request may be a request for money to a valid user.

13.2. A reimbursement request may be sending money to a valid user.

13.2. A reimbursement request shall have a status: pending, approved, or rejected.

## **14. SupportTicket**

14.1. A support ticket shall be created by a user.

14.2. A support ticket shall include a subject, message, and status.

## **15. AccountBankLink**

15.1. An account may be linked to one or more banks.

15.2. A bank may be linked to one or more accounts via AccountBankLink.

## **16. Bank**

16.1. A bank shall store balance and credit data for users.

16.2. A bank shall provide transaction data to the system.

16.3. A bank shall be viewable in the user's linked accounts page.

## **17. Reviews**

17.1. A review shall be optionally submitted by a registered user.

17.2. A review shall include a rating value (1 to 5) and an optional message.

17.3. A review shall be timestamped at the time of creation.

17.5. A review may exist independently as general user feedback not tied to any LemonAid.

17.7. A review shall be categorized by type (chatbot or user\_experience).

## **Priority 2**

### **1. User**

1.4. A user shall be able to select between manual or bank-linked financial tracking.

1.15. A user that is an administrator shall be able to respond to support tickets.

### **2. Account**

2.6. An account shall be able to categorize transactions.

2.9. An account shall track subscriptions.

### **3. Bank**

3.3. A bank shall store balance and credit data for users.

### **4. Transaction**

4.7. A transaction may be associated with a reimbursement request.

### **5. Subscription**

5.2. A subscription shall be viewable in a centralized dashboard.

### **13. ReimbursementRequest**

13.3. A reimbursement request shall be visible to administrators for review.

### **14. SupportTicket**

14.3. A support ticket shall be responded to by an administrator.

14.4. A support ticket shall track the admin response and resolution state.

## Unique Features

- **LemonAid AI**

Our product uses AI to analyze spending behavior and create budgets for users.

- **Reimbursement Request System**

This feature helps track reimbursements. These transfers are designed for Limóney users to pay each other back. Users can send each other reimbursement requests on Limóney. Users can also receive these requests and pay back vice versa.

Overall, what makes them superior to competitors is that it is an all in one app used to make dealing with money easier for the common folk. These personalized insights from the AI will make users learn at a faster rate and save money along the way.

### Deployment URL:

<https://limoney.org>

# Usability Test Plan

## Test Objectives

Test the user interface intuitiveness, response flow, and ease of use for key features that differentiate Limóney from competitors. Necessary to ensure a positive user interaction with the app, encouraging them to come back because the app is easy to use and understand. To successfully test this, we will be testing five major functions from our superior features. To ensure accurate results, we enlisted the help of a tester with no prior knowledge of the application. Most of the tests will be performed by this user with little to no guidance (in the form of questions). We shall ask the user to navigate to a specific location or feature and ask them to interact with it before inquiring about their thoughts on it. The user shall be asked to locate the features for the reimbursement request and the currency converter, and what they think of these features. The user shall be pointed to LemonAid's pop-up present on the Budget page and asked what they think of it. The user shall be asked to locate and use the main feature of the app before being asked what they think said feature is and what they think of it.

## Test Setup

- **Users:** College students and young professionals aged 18-infinite
- **URL:** <https://limoney.org>

## Test Functions

1. Reimbursement Request
2. Currency Converter
3. LemonAid AI Insights
4. Budget Visualization
5. Account Tracking
6. Transactions Search



## 1. Reimbursement Effectiveness Table

Test/Use Case	% completed	errors	comments
Requested money from user	70%	no actual money involved reimbursement doesnt pop up intuitively	integrate api implement proper ranking of reimbursements
Sending money as new user(shouldnt allow because no accounts)	100%	N/A	
Sending money as used user (with accounts)	80%	No actual money involved Approving or denying does nothing right now	
Clicking the buttons approval/reject	50%	does nothing no real money involved	
Clicking flagged button	80%	shows up in admin page, but doesnt really allow for any further action	
Updating accounts	30%	doesnt updated accounts	

## 2. Currency Converter Effectiveness Table

Test/Use Case	% completed	errors	comments
Convert 10USD to Euro	85%	Not up to date Result: 8.50	

		Answer: 8,65	
Convert 10USD to USD	100%	wow so accurate	
Convert 10 USD to JPY	75%	Result: 1093.70 Answer: 1,471.91	
Convert 10 JPY to USD	88%	Result: 0.09 Answer:0.068	
Convert 10 JPY to Euro	88%	Result: 0.08 Answer: 0.059	
Convert 10 Euro to Jpy	79%	Result: 1302.56 Answer: 1,702.16	

### 3. LemonAid AI

#### Effectiveness Table

Test/Use Case	% completed	errors	comments
How can I cut down on food this month?	50%	None	Does not reference users' finances even when prompted to. Gives general advice.
Which subscriptions should I let go of?	50%	None	Does not reference users' finances and expects user to provide expenses. Gives general advice.
I want to save for a house. What should I do about my finances?	50%	Exception error	Does not work
Which subscriptions should I let go of? (budget page)	50%	Exception error	Does not work

Help me set realistic budget limits? (budget page)	50%	Exception error	Does not work
Give financial advice personalized	75%	Not the best formatting of writing	

#### 4. Budget Visualization Effectiveness Table

Test/Use Case	% completed	errors	comments
Ask for budget for past month	100%	N/A	
Ask for budget for current month	100%	N/A	
Ask for budget for future months	85%	doesn't adjust to new plans	Just wish that they was more creative freedom to changing on aspect of the budget plan
Budget easy to read?	75%	confusing for new users	
Budget ai	65%	could be refined . the outputs formats sometimes aren't good and they are too long to keep the attention of the common user	
usability	75%	typing on budget is kind of annoying because the initial number won't go away	

## 5. Account Tracking Effectiveness Table

Test/Use Case	% completed	errors	comments
Connecting plaid account	100%	N/A	
Creating account in private mode	100%	N/A	
Delete transaction	80%	Can only delete manual transactions	
Add transaction	100%	N/A	
Account Balance Updating when adding	100%	N/A	
Account Balance Updating when subtracting	100%	N/A	

## 6. Transactions Search Effectiveness Table

Test/Use Case	% completed	errors	comments
Search netflix	100%	N/A	
Search starbucks	100%	N/A	
Check list from high to low	100%	N/A	
Search 'nothing'	100%	N/A	
Check list for	100%	N/A	

flagged			
Check list from oldest to newst	100%	N/A	

## Likert User Satisfaction

### Scale:

- 1 – Strongly Disagree
- 2 – Disagree
- 3 – Neutral
- 4 – Agree
- 5 – Strongly Agree

### 1. Reimbursement Request

- 1. I found it easy to submit a reimbursement request through the system.
- 2. The system clearly communicated the status of my reimbursement requests.
- 3. I felt confident that the reimbursement process worked as expected.

### 2. Currency Converter

- 4. I found the currency converter feature easy to use.
- 5. The exchange rates provided by the system were clear and understandable.
- 6. The currency conversion results felt accurate and trustworthy.

### 3. LemonAid AI Insights

- 7. The AI insights I received were relevant to my financial habits.
- 8. The assistant's suggestions helped me improve my spending decisions.

9. I felt that the LemonAid assistant added meaningful value to the app.

#### **4. Budget Visualization**

10. The budget charts and graphs made it easy to understand my finances.
11. The visual layout of my budgets was clear and well-organized.
12. Tracking spending visually helped me stay on top of my budget.

#### **5. Account Tracking**

13. I could easily view and manage my linked accounts in one place.
14. The system correctly grouped my accounts by type (e.g., savings, checking).
15. I felt confident that my account balances were accurately reflected.

#### **6. Transactions Search**

16. I was able to quickly find specific transactions using the search feature.
17. The filters available in transaction search made it easier to narrow results.
18. I found the transaction search feature intuitive and helpful.

## **QA Test Plan**

### **Selected Non-Functional Requirements**

1. Performance
2. Usability

3. Load Tolerance
4. Reliability
5. Compatibility

## QA Test Table

### 1. Performance

Test Objectives: Ensure the system responds to user actions within 2 seconds under normal load.

HW and SW Setup:

-Hardware: AWS EC2 t2.micro, laptop

-Software: Ubuntu 22.04, [Node.js](#) backend, React frontend

-URL: <https://limoney.org/signin>

Feature to be Tested: Page and feature responsiveness (login, navigation, account linking)

QA Test Plan:

Test No.	Title	Task Description	Input	Expected Output	Result
1	Load Dashboard	User logs in and is redirected to dashboard	Login credentials	Dashboard UI loads and becomes interactive in $\leq 2$ seconds	<1 second <b>PASS</b>
2	Add New Account	User clicks link bank account and enters their bank credentials	Bank login credentials	about 15 seconds	Took a little while for the accounts to load about 20 seconds <b>FAIL</b>
3	Navigate to Accounts Page	User clicks on accounts page from dashboard	Click on navigation tab	Accounts page takes about 3 seconds to load all the accounts	about 1 second <b>PASS</b>

### 2. Usability

Test Objectives: Ensure the UI is intuitive and users can perform key tasks without documentation.

HW and SW Setup:

-Hardware: Laptop or mobile

-Software: React frontend tested on Chrome, Firefox, Safari

-URL: <https://limoney.org>

Feature to be Tested: Ease of use and understanding of the navigation and main pages

Test No.	Title	Task Description	Input	Expected Output	Result
1	Usage of Accounts page	User clicks on Accounts page in Navbar	Click accounts tab	User understands the purpose of the Accounts page	User understood majority purpose, was not immediately sure of benefit  <b>PASS</b>
2	Usage of Budgets page	User clicks on Budget page in Navbar	Click budgets tab	User understands purpose of the Budget page and how to create an budget	User struggled to understand the purpose of the page due to being able to change passed month's budgets  <b>FAIL</b>
3	Usage of Dashboard	User clicks on Dashboard in Navbar	Click dashboard tab	User understands purpose of Dashboard page as a summary of their financial status	User understood what the dashboard was for, had momentary confusion regarding create a reimburseme nt request



					PASS
--	--	--	--	--	------

### 3. Scalability

Test Objectives: Ensure AI tools respond within 5 seconds even under moderate system load.

HW and SW Setup:

-Hardware: AWS EC2

-Software: Node.js, AI recommendation module

-URL: <https://limoney.org>

Feature to be Tested: Response time of Budget AI and Lemonaid chatbot

Test No.	Title	Task Description	Input	Expected Output	Result
1	Budget AI Suggestion	AI generates budget from user's transactions	User types prompt to budget ai asking for budget to be generated	Ai recommends in less than 5 seconds	Makes budget in about 2 seconds PASS
2	Lemonaid Advice	AI gives personal financial advice	User types prompt asking lemonaid chat for financial help	Ai gives through and personal advice	Makes response in about 3-5 seconds PASS
3	Budget AI Save	User accepts AI suggestion, DB/UI updates	User accepts generated budget	It updates the db and ui for user	PASS

### 4. Reliability: The system shall maintain at least 99.5% uptime over a 30-day period. **DONE**

Test Objectives: Ensure the system maintains at least 99.5% uptime over a 30-day period.

HW and SW Setup:

-Hardware: AWS EC2 with uptime monitoring

-Software: PM2 (process manager), Nginx

-URL: <https://limoney.org>

Feature to be Tested: Service uptime consistency

Test No.	Title	Task	Input	Expected	Result
----------	-------	------	-------	----------	--------

		Description		Output	
1	Up for 5 days	Monitor uptime for 5 days	Up 5 days	System stays up 5 days	PASS
2	Up for 15 days	Monitor uptime for 15 days	Up 15 days	System stays up 15 days	PASS
3	Up for 30 days	Monitor uptime for 30 days	Up 30 days	System stays up 30 days	PASS

5. Compatibility: The system shall support the following browsers:

Test Objectives: Ensure the system supports modern browsers (Chrome, Firefox, Edge, Safari).

HW and SW Setup:

-Hardware: Laptop

-Software: Browsers: Chrome v80+, Firefox v75+, Edge v80+, Safari v11.1.1+

-URL: <https://limoney.org>

Feature to be Tested: Site compatibility with various browsers:

• Google Chrome v80+      • Mozilla Firefox v75+      • Microsoft Edge v80+      • Safari v11.1.1+

Test No.	Title	Task Description	Input	Expected Output	Result
1	Google Chrome	Open app on Google Chrome	User opens on google chrome	It works with all its functions	PASS
2	Mozilla Firefox	Open app on Firefox	User opens on mozilla firefox	It works with all its functions	PASS
3	Microsoft Edge	Open app on Edge	User opens on microsoft Edge	It works with all its functions	PASS

## Localization Testing

Tasks:

1. Home page
2. Budget page
3. About page

### **Translations: Spanish and Hindi**

#### **Cultural Nuances:**

- Spanish dialects containing different vocabulary and phrases for Hispanic users of different backgrounds
- Hindi and Hinglish for North Indians who speak other languages than Hindi, use English as a lingua franca, and those who can't read Hindi, speak pure Hindi or English.

#### **Regional Preferences:**

Argentina, Mexico, Spain, and North India.

#### **Sample Test Case:**

Page	Test	Translation OK?
Home.js	Test es to es-MX to es-AR and hi to hi-Latin	Yes
Budget.js	Test es to es-MX to es-AR and hi to hi-Latin	Yes
About.js	Test es to es-MX to es-AR and hi to hi-Latin	Yes

#### **Results**

Most UI text components passed the translation tests. Due to time constraints, there are some pages in user layout that were not translated such as the subpages in settings. Hinglish could use more English loan words and the hooks could use a different idiom that sounds more native in different languages.

# Code Review

## Coding Standards








Here are the coding/working standards I have on a doc shared with all of my team members to ensure unity and reflecting what is asked from the nonfunctional requirements.

Coding standards:

- Name variables under camelCase convention.
  - Ex:
  - camelCase
  - somethingCool
  - iDontKnowMan
- Always work within your own branch and have it be named after the feature you are creating. Try not to create many features at once!
- Never push code onto development branch with code that is not functional or that you are still testing.
- Make sure the commits have proper names!
- Make comments throughout the code so anyone looking at it can understand what is going on.
- Always encrypt vulnerable information like database information or passwords of users
- Never hardcode ports or http://localhost. There are environment variables made for that.





## Github Code Review


## MILESTONE 1:

 <b>create milestone1 folder for m1v1</b> #8 by emilyperez was merged on Jun 17	
 <b>Completion of the About Us Page - M1</b> #7 by emilyperez was merged on Jun 16	 1
 <b>Changed line to send user straight to about page for M1</b> #6 by emilyperez was merged on Jun 16	
 <b>Jonathan(AboutMe)</b> #5 by JGonzalez650 was merged on Jun 16	
 <b>Jonathan(AboutMe)</b> #4 by JGonzalez650 was closed on Jun 16	
 <b>Johns branch</b> #3 by JGonzalez650 was merged on Jun 16	

## Completion of the About Us Page - M1 #7


 Merged dluna2 merged 7 commits into `master` from `development` on Jun 16


 Conversation 1  Commits 7  Checks 0  Files changed 7










emilyperez commented on Jun 16


No description provided.






JGonzalez650 and others added 7 commits 2 months ago


-  Jonathan(AboutMe) 4f038ff
-  Merge pull request #5 from sfsu-joseo/john-branch2 Verified 6d7bbdc
-  Updated About Us ecf9994
-  added logo and description to my profile 6a68f14
-  added description about myself and put logo e6f4353
-  fixed typos 1ab3b5e
-  accepted Ishaank's push ea19457




dluna2 commented on Jun 16

will merge
































dluna2 merged commit 7ffb7e6 into `master` on Jun 16



**MILESTONE 2:**


## Limoney Inc.

<input type="checkbox"/>	 <b>Delete milestone2 directory</b> #24 by emilyperez was merged on Jul 2	
<input type="checkbox"/>	 <b>Updated Database Organization</b> ✓ #23 by izalpuri-creator was merged on Jul 1	
<input type="checkbox"/>	 <b>modified changes...deleted and kept</b> #22 by cheeto2003 was merged on Jun 30	
<input type="checkbox"/>	  <b>Update README.md into accessing server</b> #21 by cheeto2003 was closed on Jun 30	 1
<input type="checkbox"/>	  <b>Update README.md to accessing server</b> ✓ #20 by cheeto2003 was closed on Jun 30	 1
<input type="checkbox"/>	  <b>Update README.md to accessing server</b> ✓ #19 by cheeto2003 was closed on Jun 30 • Review required	 1
<input type="checkbox"/>	 <b>Created Dockerfile compose</b> #18 by emilyperez was merged on Jun 30	
<input type="checkbox"/>	  <b>Updating .gitignore in root</b> #17 by emilyperez was closed on Jun 27	
<input type="checkbox"/>	 <b>Updating global .gitignore</b> #16 by emilyperez was merged on Jun 27	
<input type="checkbox"/>	 <b>removed git idea</b> #15 by cheeto2003 was merged on Jun 27	
<input type="checkbox"/>	  <b>Gene6.27</b> #14 by cheeto2003 was closed on Jun 27	 1
<input type="checkbox"/>	  <b>Gene6.27</b> #13 by cheeto2003 was closed on Jun 27	 1
<input type="checkbox"/>	  <b>mongo -&gt; mysql</b> #12 by cheeto2003 was closed on Jun 27	 1
<input type="checkbox"/>	 <b>Clean up of code/directory + added signup page + search bar for milestone 2 (no functionalities)</b> #11 by emilyperez was merged on Jun 26	

## Gene6.27 #13


 Closed cheeto2003 wants to merge 8 commits into [development](#) from [gene6.27](#) 


 Conversation **1**  Commits **8**  Checks **0**  Files changed **15**




cheeto2003 commented on Jun 27

node modules removed






cheeto2003 added 8 commits 2 months ago




added back end communication with frontend, provided encryption and v...

684ca31




fixed mongo to mysql

8157131




Merge remote-tracking branch 'origin/development' into gene6.27

859a2bf




mongo to sql connection

6503a05




fixed mongo to sql

ac62c98




git ignore node modules

5148284




changed hardcoded info to env and deleted node\_modules

cc60b55




removed node\_modules changed so need an env file and info is not hard...

fdfedb3
















emilyperez commented on Jun 27

shared .env - bad for privacy



**MILESTONE 3:**



<input type="checkbox"/>	 <b>Budget dashboard animation</b> #92 by dluna2 was merged 2 weeks ago	
<input type="checkbox"/>	 <b>prettied up transaction page</b> #91 by cheeto2003 was merged 2 weeks ago	 1
<input type="checkbox"/>	 <b>added kebab functionality</b> #90 by cheeto2003 was closed 2 weeks ago	
<input type="checkbox"/>	 <b>lined up transaction page to look like figma</b> #89 by cheeto2003 was closed 2 weeks ago	
<input type="checkbox"/>	 <b>modified search for searching and loading transactions</b> #88 by cheeto2003 was merged 2 weeks ago	
<input type="checkbox"/>	 <b>Budget dashboard animation</b> #87 by dluna2 was merged 2 weeks ago	
<input type="checkbox"/>	 <b>SettingsPageUI</b> #86 by JGonzalez650 was merged 2 weeks ago	
<input type="checkbox"/>	 <b>transactoin and search is now live!</b> #85 by cheeto2003 was merged 3 weeks ago	
<input type="checkbox"/>	 <b>logout fix</b> #84 by andrewb-03 was closed 2 weeks ago	
<input type="checkbox"/>	 <b>Organized Directory</b> #83 by izalpuri-creator was merged 3 weeks ago	
<input type="checkbox"/>	 <b>Fix public pages</b> #82 by emilyperez was merged 3 weeks ago	
<input type="checkbox"/>	 <b>Modded accounts &amp; lemon aid page to look pretty and match color scheme :)</b> #81 by cheeto2003 was merged 3 weeks ago	

## prettied up transaction page #91

 Merged dluna2 merged 5 commits into [development](#) from [modded\\_search](#) 2 weeks ago

 Conversation 1  Commits 5  Checks 0  Files changed 3



cheeto2003 commented 2 weeks ago

...

prettied up transaction page




cheeto2003 and others added 5 commits 2 weeks ago



 lined up transaction page to look like figma

[0dd4290](#)



 added kebab functionality

[5a144a1](#)



 added kebab functionality

[fef229c](#)



 fixed search constantly searching for you

[802561b](#)



 Merge branch 'development' of github.com:sfsu-joseo/csc648-848-01-su2...

[b4c3b74](#)




dluna2 commented 2 weeks ago

...


great work. we need to change the trash icon for deleting.



 dluna2 merged commit [8e0d281](#) into [development](#) 2 weeks ago

Revert



 emilyperez deleted the [modded\\_search](#) branch 2 weeks ago

Restore branch

### MILESTONE 4:

## Sample of some of our pull requests:

<input type="checkbox"/>	<b>feat: enhance reimbursement system with send/request money options an...</b> #156 by andrewb-03 was merged 15 minutes ago	
<input type="checkbox"/>	<b>Mocked budget response</b> #154 by cheeto2003 was merged 4 hours ago	
<input type="checkbox"/>	<b>Cleaned up code</b> #153 by emilyperez was merged 4 hours ago	
<input type="checkbox"/>	<b>Fixed subscription table</b> #152 by emilyperez was merged 9 hours ago	
<input type="checkbox"/>	<b>Fixed categorizing of transactions</b> #151 by emilyperez was merged 10 hours ago	
<input type="checkbox"/>	<b>Fix/reimbursement table error</b> #150 by andrewb-03 was merged yesterday	1
<input type="checkbox"/>	<b>Polish UI and dashboard fetching data from user</b> #149 by dluna2 was merged yesterday	
<input type="checkbox"/>	<b>Js doc streamline design and complete responsivity</b> #148 by izalpuri-creator was merged 7 hours ago	
<input type="checkbox"/>	<b>Edit user</b> #147 by JGonzalez650 was closed yesterday	1
<input type="checkbox"/>	<b>Sub backend</b> #146 by JGonzalez650 was closed yesterday	2
<input type="checkbox"/>	<b>Feature/backend subscriptions lemonsaid + fix</b> #145 by andrewb-03 was closed yesterday	
<input type="checkbox"/>	<b>Moved signout to settings</b> #144 by emilyperez was merged yesterday	
<input type="checkbox"/>	<b>Fed lemon aid</b> ✓ #143 by cheeto2003 was merged yesterday	

## Sample of comments in a pull request:

Closed

Sub backend #146

JGonzalez650 wants to merge 3 commits into `development` from `subBackend`

JGonzalez650 commented yesterday

Subscription Backend. Users can now:

- Filter through subscriptions
- Edit their subscription notes
- Cancel Subscription

JGonzalez650 and others added 3 commits 2 days ago

userSavededit

b161c51

subscriptionBackend

303924d

Merge branch 'development' into subBackend

Verified 31bec8a

emilyperez commented yesterday • edited

It is repeating the same subscription in the subscription page. if it is the same subscription it doesnt need to be repeated. Can you also get rid of the other tabs in transactions and name the main tab subscriptions. Come back to me if you need me to explain further.

izalpuri-creator commented yesterday

Don't work brah

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

Unsubscribe

You're receiving notifications because you commented.

3 participants

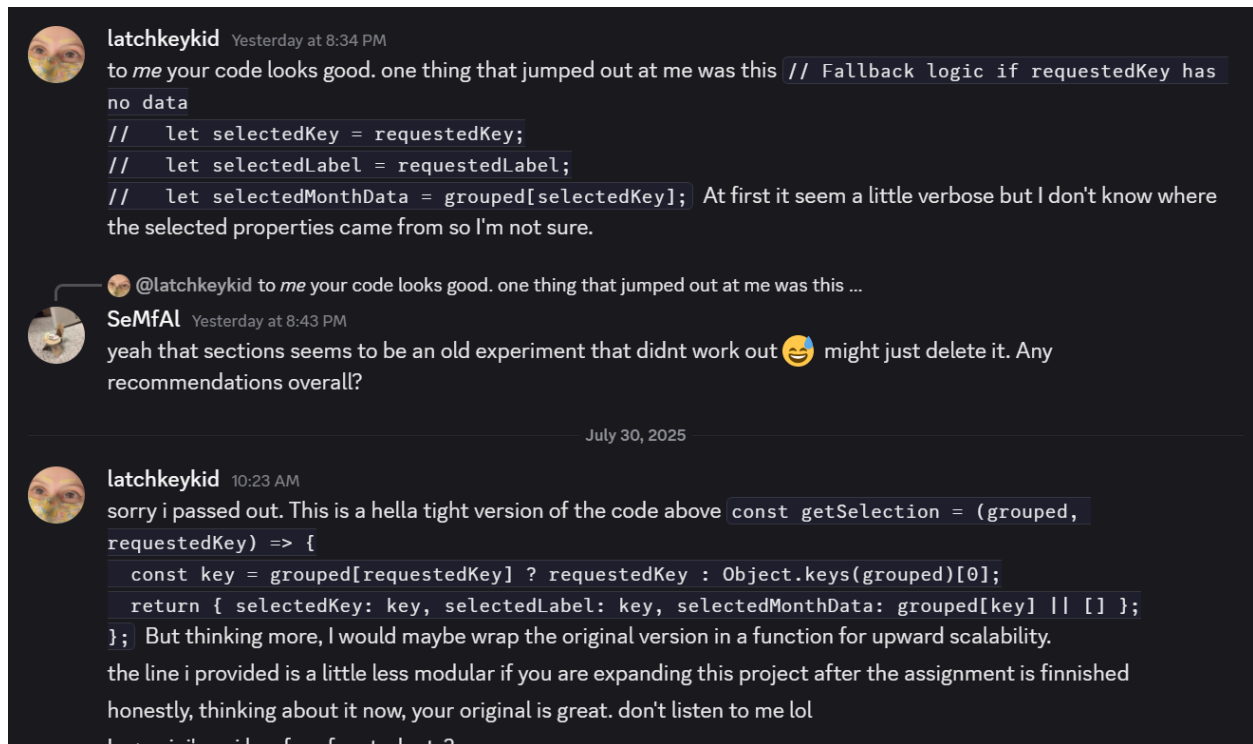
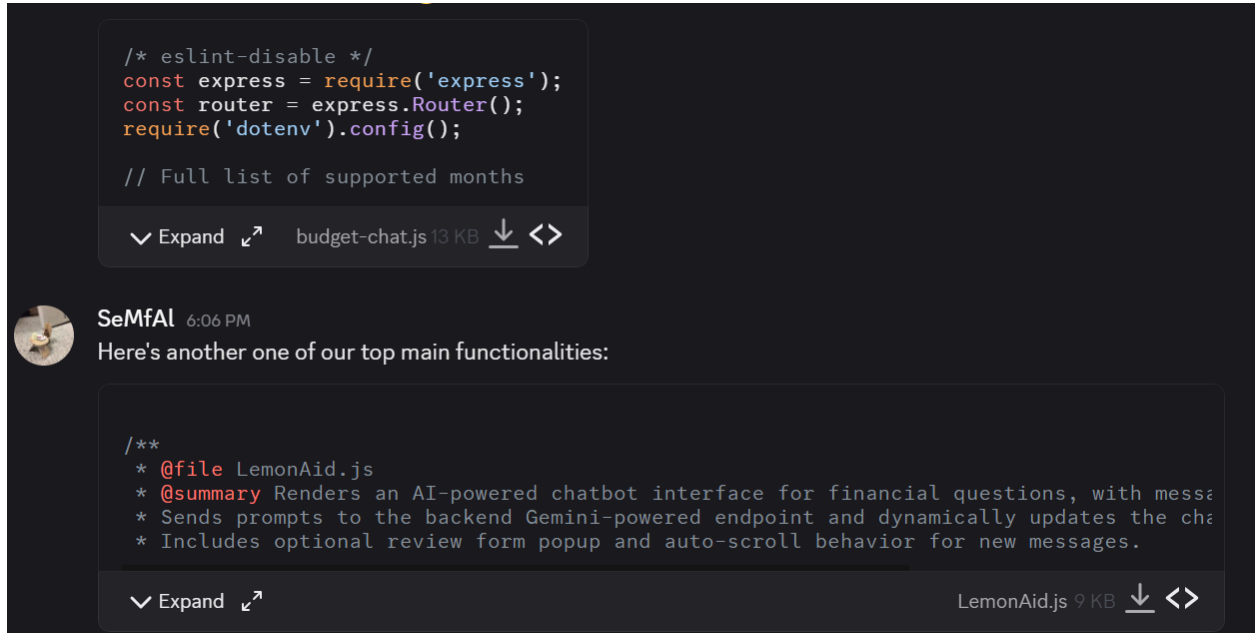
Lock conversation

## External Review

Team 5 reviewed our budget and budget ai backend code. We decided to message through discord because we couldn't open each other's files on our school emails.

- Names are too verbose. In milestone 5, we will clean and refine our code.

Photo proof:



# Security Practices

## Assets Protected

- User personal info
- Banking credentials
- Transaction history
- AI recommendations

## Password Encryption

- Bcrypt (12 salt rounds)
- Stored hashed in MySQL

```
mysql> SELECT * FROM User;
```

userId	name	email	password	userType
3	test2	test2@test.com	\$2b\$10\$Kj30Wcxjtucli.tM57TLWeVGYLQTd8D7M/R4tg0eo17jzZsZip4Ri	standard
4	test	test@pleasework.com	\$2b\$10\$L060/fSX.DWsEiqM3c8Kqud1iIOVvoqIaG6Ht9iXzJk51KkvcHmr0	standard
5	123	123@123.com	\$2b\$10\$vlj2lJXhQYCoU06g6HUWreTlHwxznsqEmTk6RZDsBH1cYuR/KCF0u	standard
6	rawr	rawr@oowow.com	\$2b\$10\$H4KDEN/gn0XR7PFcM/H0KeH3g0trVhQox6q0MMQSpYMrWo1W6J6bq	standard
8	testing	123@gmail.com	\$2b\$10\$FF7FzIFTcMe9rh1EY8Pb8uxLm12PjsaqQ6k4jG0.EkduTXS7P5Hcm	standard
9	test2	testing@gmail.com	\$2b\$10\$LRAm.bTu70CYbnMSdE1XWeKcqby5U1Ch52pDURo1E7BnsfEM787cW	standard
10	random	test@work.com	\$2b\$10\$bLYOVzNfISPSEQqGsr/Ate.U0ggIgr/qIvBcrw1VNxjWq9Yb5ydC2	standard
11	Ishaank	izalpuri@gmail.com	\$2b\$10\$5zCnglj4cnXFIDto2cMiiOUNwIl3fQZszvCZQeM7vP9CVk1zLJC6	standard
12	Alice Johnson	alice@example.com	hashedpass1	standard
13	Bob Lee	bob@example.com	hashedpass2	standard
14	Carol Kim	carol@example.com	hashedpass3	standard
15	fdafklldjasdas	sleep@sleep.com	\$2b\$10\$JgY9.ipudFRTn.2w/lo9wu9JMoK79sFEPv2FkqbBMK18X9n079Cty	standard
16	Dechen	dechen2020l@icloud.com	\$2b\$10\$uQoi0Gi8eQcMw1zzYgMD4uJfa5eqtzEq6fn/exUelOgZzJ88PNsC	standard
17	Jose	joseortizcosta@aol.com	\$2b\$10\$L7gaI2owfsayQUAjuhs1feP9YJZG5hjdttwemuoBZYSjoMQgeudS	standard
18	Hii	l@g.com	\$2b\$10\$HyQX719fSWYcX80Rf7eyVOPRJY6/MqWJMdCrrHITEv6HjPPIWY1K	standard
20	bla	bla@gmail.com	\$2b\$10\$c7cteCK0ZtgIFvIV08SX90.tgKoJoT/NJV5rVaxJIKc2DqD/9iQAY	standard
24	test2	testing2@gmail.com	\$2b\$10\$07VH/eHSQginFyqnnvERmODMU06LaYPYmi1D0G9zcVM0hyNjwhQHy	standard
25	test	test@test.comte	\$2b\$10\$Zu9fIob.XYSYoBnL/.Fmh.TenUV981d/E38WnqUCQWkDHQWtz/bQu	standard
27	usertestFP	usertestfp@example.com	\$2b\$10\$X08fdjFRDr4qyR40F97fAuCj2VT.r0z0kigmg/FWJy.raVNcIi5fy	standard
28	china	china@china.com	\$2b\$10\$rHFSQBxDt1fHDqEoD9z85.0oR4L.nhQR/KgWcH5hf9RcU3/eBigDG	standard
31	test3	test23@test.com	\$2b\$10\$zCs9WEs85zWvjh9XrCuYk.LgTBfG7vRzKNhqbUQJ0N/n0nbL4FiYa	standard
42	test	test1@test.com	\$2b\$10\$ldxmZa6EIVou1KV66Q7/ieuKbrvb8av7AYIXp8unjnK50w8gQztce	standard
55	test2	test@test.com	\$2b\$10\$H0EmdvTt9qVu9w3xhoL.SkeIZ65Z9Hr3.KXMzz85tgRDbEHE8TYQey	standard
57	123	21@test.com	\$2b\$10\$LvdffonsSnecnOjw.j0kG/h.D9mPaSsdHABep8forjT6/oEijbKAPPG	standard
58	testreimbursement	reimbursement@test.com	\$2b\$10\$Usk5TEGH/b9Y1BvmNPUQ8u/ardJkhCLBpMCGzKp4Ycws2jt5FPNfk	standard
59	Recipient User	recipient@test.com	\$2b\$10\$Me8gyTiMvLPgDEXY9aIvd.ae0oabH5Z.d7i6YCHSgwaNJSgXuHeii	standard
60	Sender User	sender@test.com	\$2b\$10\$dV7QVme0Wcj0ei3bMBLB6.EaOE4Rn0.7.1R65QoEYW4MxL7GCp04.	standard

## Input Validation

Validated inputs: email, password, search

- Signing up validation : email, password (at least 6 characters)

```
name = typeof name === 'string' ? name.trim() : '';
```

- ```
email = typeof email === 'string' ? email.trim().toLowerCase() : '';
```

```
• password = typeof password === 'string' ? password.trim() : '';
•
• if (!name || !email || !password) {
•   return res.status(400).json({ message: 'All fields are
required' });
• }
•
• const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
• if (!emailRegex.test(email)) {
•   return res.status(400).json({ message: 'Invalid email format'
});
• }
•
• if (password.length < 6) {
•   return res.status(400).json({ message: 'Password must be at
least 6 characters' });
• }
•
• try {
•   const existing = await UserModel.findUserByEmail(db, email);
•   if (existing) {
•     return res.status(409).json({ message: 'Email already
registered' });
•   }
•
•   const hashedPassword = await bcrypt.hash(password, 10);
•   await UserModel.createUser(db, name, email, hashedPassword);
•
•   console.log(`New user created: ${email}`);
•   res.json({ message: `Thanks for signing up, ${name}!` });
• } catch (err) {
•   console.error('Error saving user:', err);
•   res.status(500).json({ message: 'Server error', error:
err.message });
• }
• };
```

- Search:  
ensures an empty search isn't sent

```
if (query.trim()) params.append('query', query);
```

# Self-Check: Adherence to Non-Functional Specs

## List of Non-Functional Requirements

### 1. Performance

- 1.1. The system shall respond to user actions within 2 seconds under normal load. **DONE**
- 1.2. Backend APIs shall respond to simple requests in under 200 milliseconds. **DONE**
- 1.3. The dashboard shall load within 3 seconds over a stable 10 Mbps connection. **DONE**
- 1.4. AI recommendations shall be delivered within 5 seconds of user input. **DONE**
- 1.5. The system shall support real-time syncing of transactions without freezing the UI. **DONE??**

### 2. Security

- 2.1. All communications shall be encrypted using HTTPS and TLS protocols. **DONE**
- 2.2. All sensitive data, including user credentials, shall be encrypted. **DONE**
- 2.3. Passwords shall be hashed using bcrypt or equivalent secure hashing algorithms. **DONE**
- 2.4. Role-based access control shall be implemented to separate admin and user privileges. **DONE**
- 2.5. The system shall prevent common web vulnerabilities such as SQL injection, XSS, and CSRF. **DONE**
- 2.6. Only authenticated and authorized users shall be able to change, delete, or insert sensitive data. **DONE??**

### 3. Scalability

- 3.1. The system shall support up to 500 concurrent users without performance degradation. **DONE**
- 3.2. The database shall support horizontal scaling across distributed servers via Docker and AWS. **DONE**
- 3.3. The system shall efficiently scale with a 10x increase in user data and transaction volume. **DONE**
- 3.4. Database schema shall support sharding for large financial datasets. **DONE**

### 4. Usability

- 4.1. The UI shall be intuitive and require no documentation for key tasks such as setting up budgets or tracking expenses. **DONE**
- 4.2. The system shall support both dark mode and light mode themes. **DONE**
- 4.3. The AI assistant shall use natural, conversational language to support non-technical users. **DONE**
- 4.4. Navigation, task completion, and interaction shall follow established UX best practices. **DONE**

### 5. Reliability

- 5.1. The system shall maintain at least 99.5% uptime over a 30-day period. **DONE**
- 5.2. Transaction data shall be backed up every 12 hours automatically. **DONE**
- 5.3. Background processes such as syncing and reminders shall automatically retry on failure. **DONE**
- 5.4. Error handling shall provide feedback to users without crashing the system. **DONE**

### 6. Maintainability

- 6.1. The codebase shall follow JavaScript linting rules using ESLint and Prettier. **DONE**
- 6.2. Docker containers shall be used to ensure parity between local and production environments. **DONE**
- 6.3. The codebase shall follow a modular architecture to isolate and decouple components. **DONE**
- 6.4. Critical modules shall maintain a minimum of 70% unit test coverage. **DONE**



6.5. Version control shall be managed through Git and GitHub with frequent commits and reviews. **DONE**

## 7. Portability

7.1. Docker containers shall allow the app to run across local development, staging, and AWS cloud environments. **DONE**

7.2. The app shall run identically across Windows, macOS, and Linux-based systems. **DONE**

## 8. Compatibility

8.1. The system shall integrate with major U.S. banks using secure open banking APIs. **DONE**

8.2. The AI engine (for ex: ChatGPT API or Gemini API) shall be modular and replaceable without impacting core functionality. **DONE**

8.3. The system shall support the following browsers: **DONE**

• Google Chrome v80+      • Mozilla Firefox v75+      • Microsoft Edge v80+      • Safari v11.1.1+

## 9. Privacy

9.1. The app shall allow users to manually enter financial data without linking any bank accounts. **DONE**

9.2. User data shall not be sold to third-party services. **DONE**

9.3. All stored data shall remain local unless explicit user consent is given. **DONE**

9.4. Privacy mode shall restrict AI access to only local anonymized data. **DONE**

## 10. Compliance

10.1. The system shall comply with GDPR regulations for data privacy and user consent. **DONE**

10.2. Users shall be able to permanently delete their data on request. **DONE**

10.3. Consent prompts shall be clearly visible during signup. **DONE**

## 11. Supportability

11.1. System logs shall track user behavior, feature usage, and AI errors for debugging. **DONE**

11.2. An admin dashboard shall allow staff to review flagged transactions and feedback. **DONE**

11.3. The system shall support browser versions listed under Compatibility. **DONE**

11.4. Support documentation shall be maintained for installation, setup, and deployment. **DONE**

## 12. Efficiency

12.1. The system shall operate at no more than 60% CPU usage under normal load conditions. **DONE**

12.2. Memory usage shall not exceed 75% of allocated memory on the EC2 instance. **DONE**

12.3. Efficient logging shall be implemented to prevent log flooding and performance drag. **DONE**

## 13. Look and Feel

13.1. The system shall maintain a clean, minimal aesthetic with responsive design. **DONE**

13.2. UI elements shall follow a cohesive design language across all pages and devices. **DONE**

13.3. Accessibility standards (WCAG 2.1 AA) shall be considered for inclusive design. **DONE**

#### 14. Coding Standards

- 14.1. All backend code shall conform to Node.js and Express.js best practices. **DONE**
- 14.2. Frontend code shall follow React component structure and state management conventions. **DONE**
- 14.3. Continuous Integration tools shall run linting and testing on every merge request. **DONE**

#### 15. Environmental Sustainability

- 15.1. The system shall optimize backend server usage to reduce idle time and energy waste. **DONE**
- 15.2. Docker containers shall be configured to auto-scale down during periods of low usage. **DONE**
- 15.3. The app shall minimize data transfer size (e.g., compress JSON responses, lazy-load images) to reduce bandwidth and energy consumption. **DONE**
- 15.4. The frontend shall use efficient rendering practices in React (e.g., avoiding unnecessary re-renders, using memoization). **DONE**
- 15.5. System architecture shall consider the use of renewable-energy-based data centers (e.g., AWS sustainability zones). **DONE**
- 15.6. Logs and backups shall be archived using energy-efficient cold storage where applicable. **DONE**
- 15.7. Documentation shall be provided digitally only (no paper printing by default). **DONE**
- 15.8. The system shall avoid unnecessary background polling or constant sync unless essential for real-time use. **DONE**

## List of Team Contributions

| Name                  | Contributions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Score |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| Emily Perez           | <ul style="list-style-type: none"><li>• set up live compiling for easier coding process</li><li>• set up different environment for production vs development</li><li>• implemented sessions</li><li>• implemented plaid api</li><li>• made accounts page dynamic to user</li><li>• fixed transactions to where each transaction is put into the correct account</li><li>• Implemented database for usertransactions, plaiditem,</li><li>• deployed onto site</li><li>• dealt with pull requests and merging</li><li>• helped other debug their problems</li><li>• implemented categorization using plaid api</li><li>• implemented subscription into db using plaidapi</li></ul> | 9     |
| Ishaank Zalpuri       | <ul style="list-style-type: none"><li>• made all public pages responsive to various device sizes</li><li>• helped with m4v1</li><li>• attempted at localization</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | 8     |
| Andrew Brockenborough | <ul style="list-style-type: none"><li>• refactored <a href="#">server.js</a> so it is cleaner,</li><li>• helped with m4v1,</li><li>• added tools icon on userlogin navbar and connected the pages</li><li>• implemented backend for reimbursement request</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                              | 6     |

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |    |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
|                   | <ul style="list-style-type: none"><li>○ receive</li><li>○ send</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |
| Dani Luna         | <ul style="list-style-type: none"><li>● created todolist;</li><li>● inserted background colors for public layout, balance forecast, dashboard and lemonaid;</li><li>● turned off horizontal scrolling on budget page;</li><li>● helped with m4v1</li><li>● attempted at localization</li><li>● drew background for lemonaidchat;</li><li>● fixed cut off for piechart</li></ul>                                                                                                                                                                         | 7  |
| Jonathan Gonzalez | <ul style="list-style-type: none"><li>● implemented back end change email and name</li><li>● implement forgot password backend</li><li>● implemented forgot password frontend</li><li>● fixed ui for subscriptions</li></ul>                                                                                                                                                                                                                                                                                                                            | 7  |
| Gene Orias        | <ul style="list-style-type: none"><li>● added fake ssl certificates for coding environment</li><li>● implemented cookies,</li><li>● secured userloginpages for users only</li><li>● made userloginpages adjust to the user,</li><li>● implemented backend for settings:<ul style="list-style-type: none"><li>○ editing name</li><li>○ notifications</li><li>○ delete account</li></ul></li><li>● implemented backend for budget</li><li>● implemented backend for budget ai generator</li><li>● implemented backend for lemonaid general chat</li></ul> | 10 |

[Limoney Inc.](#)