

# Expensify Take-Home Challenge

## Author/Developer Details

**Name:** Jerry Allan Akshay  
**Email:** [jakshay@usc.edu](mailto:jakshay@usc.edu)  
**Contact Number:** +1 (213)-522-6313  
**Portfolio:** [Github Portfolio](#)

## Link to access the hosted website

[Jerry Allan's Expensify Take-Home Project](#)

## Steps to use the website

1. Login using the following credentials:
  1. Username: `expensifytest@mailinator.com`
  2. Password: `hire_me`
2. You can use any login credentials you like as long as it is stored on Expensify's server.
3. Once you login, you'll have the option to logout too if you so desire. Once you logout, you cannot automatically log back in again as the `authToken` will be lost.
4. If you don't logout, you can just refresh the page and it will take you right to the transaction page.
5. The transaction page contains all of the transactions recorded by the user.
6. You can also see your current total balance on the bottom left corner of the page above the footer.
7. In order to add a transaction, click on the 'Add Transaction' button and fill in the necessary details.
8. Note that you cannot add a future transaction.
9. Once you've added the transaction, you'll be able to see the transaction on the top of the table with a green background. The green background indicates the transactions that have been added in the current user session.
10. Once you refresh the page, the newly added transactions will be placed in the table according to their chronological order.
11. Log out once you're done. Or just stay logged in. The website takes care of all the edge cases.
12. Don't forget to test out the 404 page functionality. You'll be surprised! Just type in any random target link on the same url (or you can use this [link](#) to access it).

## Time Measurements

Sl. No.	Task	Time Taken
1	Hosting a PHP environment on Heroku	1.5 hours
2	Login portion of the task (Frontend and Backend)	3 hours
3	404 Page	1 hour
4	Transaction Page	6 hours
5	'Add Transaction' Form	2 hours
6	Refactoring and Code Cleanup	4 hours
7	Adding in Comments	1 hour
8	Documentation Referencing	4 hours

## Problems I encountered, and how I overcame them

1. The very first issue was the usage of PHP. I had never used PHP before, so I had to read up on the documentation. It was quite simple to grasp and very easy to use. I had no troubles later on with the usage of PHP.
2. While creating the 404 page, I did face an issue where the web server didn't redirect the user to the right 404 error page. I had to search online and then figured out that the `.htaccess` file was required with the right configuration set in order for the 404 redirection to work.
3. The third issue I came across was a relatively faster way to display the large table. I did look up online and came across this [link](#). The author of this blog had tried various approaches and concluded (with conclusive test results) that the fastest way to display a HTML table with a large dataset is it use the `node (documentFragment)` approach. I later used that same methodology in the webpage.
4. One issue I came across while testing was that the Expensify API `GET https://www.expensify.com/api?command=Get` made some errors in storing the transaction amount. The API stored the negative of the transaction amount for some reason. Since I don't have access to the Expensify servers (obviously), I used a correction parameter of `-1` (`transactionAmount * -1`) on the client side while sending the data to the Expesify API.

## Issues with the Expensify API

1. The POST API `GET https://www.expensify.com/api?command=Get` to create a transaction creates an entry with the negative of the amount. This issue needs to be fixed.

## References

---

1. [Markdown Reference](#)
2. [Creating a Heroku server to deploy PHP](#)
3. [PHP Documentation](#)
4. [Heroku Documentation](#)
5. [Fastest way to populate a large table](#)
6. [CSS Reference](#)
7. [HTML Reference](#)
8. [jQuery Documentation](#)
9. [AJAX Documentation](#)