



## Sprint 2 Retrospective

Team 8: Hugh Bromund, Peyton Williams, Aditya Naik, Daniel Joseph, Nathan Ashta, Niyati Sriram

# What Went Well?

## **Overall:**

**In general,** we greatly expanded the functionality of our website during this sprint. We completed all User Stories and all acceptance criteria for those user stories. We focused the majority of our efforts this sprint on building the initial budgeting system and expanding our already existing stock systems. At the end of this sprint, users can now create and manage their budgets as well as view more detailed information about stocks and cryptocurrencies.

## User Story #1

As a user, I would like to be able to switch between dark mode and a light mode so that I can have an easier time viewing the application

Description	Estimated Time	Owner
Create UI toggle allowing the user to switch between light and dark mode	1 hr	Aditya
Handle toggle in CSS for all pages	1 hr	Aditya
Allow the site to store light and dark mode on refresh	2 hrs	Aditya
Make request to backend to update dark vs. light	1 hr	Aditya
Update from the backend to the database for dark vs. light	1 hr	Niyati
Test to detect light and dark mode change for each page	0.5 hrs	Aditya, Niyati

**Completed:** The user is able to toggle between dark and light mode in their account menu. All pages will update to reflect this change. Additionally, when the user refreshes the page, the state will be retained and the page will be rendered in whatever mode was last active. Lastly, if loading the page for the first time, the user will load into light mode.

## User Story #2

As a user, I would like the application to be aesthetically pleasing.

Description	Estimated Time	Owner
Create color selector on the frontend for the user	5 hrs	Nate
Post to the backend a change in the color	1 hr	Nate
Handle the color change and edit the database from the backend	2 hrs	Niyati

**Completed:** The app, by the team's standards, is aesthetically pleasing. The user is also able to change their personal colors from their account menu. These colors will update the graph accordingly. These colors will be stored in the backend.

### User Story #3

As a user, I would like to be able to save the stocks I'd like to watch so that I can quickly access the stocks that I have interest in.

Description	Estimated Time	Owner
Create a page for users to choose and view saved stocks	3 hrs	Aditya
Allow a user to save a stock by clicking a button and saving the stock ID to the database	2 hrs	Niyati
Allow a user to un-save a stock that they no longer want to track	2 hrs	Niyati
Setup routing so the user can navigate to the saved stocks from the navigation bar	1 hr	Aditya
Testing	1 hr	Aditya, Niyati

**Completed:** The user can successfully follow a stock when logged in. A button is available for them to press on the stock's information page. Additionally, when they are following a stock, the button will update to an unfollow button and a list of their followed stocks appears at the bottom of the page. The user can click these and the site will redirect to that stock's information page. These changes are stored as an array on the backend in the database.

### User Story #4

As a user, I would like to be able to create a monthly budget

Description	Estimated Time	Owner
Create a UI Page/Form that allows users to set up their monthly budget	8 hrs	Hugh
Setup routing so that the User can navigate to the budget page from the Navigation Bar	1 hr	Hugh
Integrate with the backend so that the User's	3 hrs	Niyati

budget is stored in the database		
Handle Logic to ensure that only valid budgets can be created by the User	2 hrs	Peyton
Include Unit Tests for frontend and backend	1 hr	Hugh, Niyati

**Completed:** The user is able create a monthly budget on the budget creation page. This page will ask the user for a total budgeted amount for the month, and then they must fill in a budgeted amount for each category (food, transportation, housing, etc.). If the categories do not add up to the total, the budget cannot be submitted. Once submitted, the budget is stored in the database and a mirror for storing the spending in each category is created.

## User Story #5

As a user, I would like to be able to reset my password if I forget it

Description	Estimated Time	Owner
Create a form on the frontend to reset the password	3 hrs	Hugh
Handle routing to this component	1 hr	Hugh
Send notification to the backend	1 hr	Hugh
Backend handles logic and sends an email to update the password	4 hrs	Peyton
Password is updated in the database	1 hr	Peyton
Testing	0.5 hrs	Peyton

**Completed:** The user is able to reset their password through a portal linked to on the login page. The user provides the linked email to their account, and if it is found in the database, a randomly generated 12 character password is emailed to them from [mail.finex.finex@gmail.com](mailto:mail.finex.finex@gmail.com). This password is also updated in the database.

## User Story #6

As a user, I would like to be able to edit my personal information

Description	Estimated Time	Owner
-------------	----------------	-------

Create a form on the frontend to input updated personal information	3 hrs	Hugh
Handle routing to this component	1 hr	Hugh
Send information to backend	1 hr	Hugh
Backend handles routing and logic	1 hr	Peyton
Information updated in the database	1 hr	Peyton
Testing	1.5 hrs	Peyton

**Completed:** The user is able to update their name, email, and password on their account page once they are logged in. Each field has an update button next to it which then allows the user to fill in a field and submit the change. These changes are then updated in the database.

## User Story #7

As a user, I would like to be able to have a dashboard of all of my personal finances.

Description	Estimated Time	Owner
Create a UI page/form that allows the user to see their personal finances	8 hrs	Hugh
Create routing from Navigation Bar to this component	1 hr	Hugh
Integrate this page with the backend so data is pulled from the database	4 hrs	Peyton
Routing for backend	1 hr	Peyton
Create unit tests for frontend and backend	1 hr	Hugh, Peyton

**Completed:** The user is able to view a personal finances page which displays each expense and income for the month, as well as their monthly budget which includes allocated and spent for each category, as well as an overall total. The user can view a graphical representation of (spent / allocated) for each category in the budget.

## User Story #8

As a user, I would like a monthly analysis of where I am spending the most money (food, transportation, rent, etc.)

Description	Estimated Time	Owner
Create a form allowing user to input a category for the transaction	5 hr	Nate
Create page allowing user to view monthly analysis of spending	2 hr	Hugh
Store category data in the database	2 hr	Niyati
Create test to ensure that category data is correctly stored	1 hr	Hugh, Niyati

**Completed:** If a user has a transaction they wish to enter, when they view the form, then there exists a form for them to input a category. If a user has entered a transaction, when they submit it, the associated category is stored in the database. When a user enters the page to view a monthly summary, when they enter the page, they are able to view a graphical representation of spending between categories

## User Story #9

As a user, I would like to keep track of my actual spending as compared to my projected budget plan

Description	Estimated Time	Owner
Create a separate budgeting page that shows the user's actual and projected spending	3 hrs	Aditya
Create a pie chart and transaction table for the user's projected budget plan and actual spending	5 hrs	Aditya
Allow user to navigate to a page, allowing them to enter more budget information	1 hr	Aditya
Allow users to navigate to a page to create their budget plan	1 hr	Aditya

Allow users to change their timeframe of budgeting	2 hrs	Aditya
Pull applicable transactions from the database	3 hrs	Peyton
Routing for transactions from the database	1 hr	Peyton

**Completed:** For the given month, the user can navigate to their spending summary page where they will be shown a table dedicated to their spending per category. They will also be shown a pie chart of this spending.

## User Story #10

As a user, I would like the server to support fast server-client communication

Description	Estimated Time	Owner
Ensure that the frontend application will not crash on error on calling the backend	3 hrs	Aditya, Nate, Hugh
Ensure that the backend application will not crash on incorrect call from the frontend	5 hrs	Peyton
Ensure speed on calls from frontend to the backend with no blockers	2 hrs	Everyone

**Completed:** If the user submits some request on the frontend, they will receive a response shortly. If an error occurs on the backend, it will not cause unhandled errors on the frontend. If an error occurs on the frontend, it will not cause unhandled errors on the backend. If the frontend is waiting for a response, the fields should display an appropriate temporary value.

## User Story #11

As a user, I would like to be able to view stock data in variable timeframes

Description	Estimated Time	Owner
Support endpoints for accessing data from different timeframes and different amounts	5 hrs	Daniel
Support Standardized JSON output across all API requests	10 hrs	Daniel



Display information on stock page from parsed JSON	5 hr	Nate
Create buttons to allow users to toggle between graphs	5 hr	Nate
Testing	3 hrs	Daniel

**Completed:** The user should be able to toggle between different timeframes on the stock page and the correct scale is displayed on the stock graph. If there is an error with the Alpha Vantage API, a JSON error object is returned to the frontend. If the user selects a stock to view, the backend will execute several API requests to get the necessary data

## User Story #12

As a user, I would like to be able to view a cryptocurrency's current price

Description	Estimated Time	Owner
Add a page on the frontend to connect with the stock page to display information on cryptocurrencies	3 hrs	Nate, Aditya
Add routing for this component to other components	1 hr	Aditya
Integrate component with backend	4 hrs	Daniel
Create form with dropdown to search for cryptocurrencies	1 hr	Aditya
Backend autocomplete for searched crypto names	2 hrs	Peyton
Handle backend logic to get the cryptocurrency price	3 hrs	Daniel

**Completed:** When a user switches to search for cryptocurrency, they will be able to see the search box filled with crypto auto generated results. When they select an index, they will see the graph, all analytical data, and the toggle between intraday and daily data.

## User Story #13

As a user, I would like to be able to add transactions to my finances so that I can manage my budget effectively.

Description	Estimated Time	Owner
Add a form so the user can input transactions to the budget	5 hr	Nate
On submit, send information to the backend	1 hr	Nate
Handle routing to this component	1 hr	Nate
On backend, update the database so that the budget is correct.	2 hrs	Niyati

**Completed:** The user is able to add transactions (expense or income) which are added to the total spent and compared to the monthly budget the user created. Transactions must have a non-zero value and all fields must be present.

## User Story #14

As a user, I would like to be able to add income to my finances so that I can manage my budget effectively.

Description	Estimated Time	Owner
Add a form so the user can input income to the budget	5 hr	Nate
On submit, send information to the backend	1 hr	Nate
Handle routing to this component	1 hr	Nate
On backend, update the database so that the budget is correct.	2 hrs	Niyati

**Completed:** The user is able to navigate to the “add income” page from the personal finance home page and input their income. FINEX will only allow valid inputs on this page and will ask for input to classify the income in different categories. When the income is submitted, it is stored in the database and displayed on the personal finance home page.

## User Story #15

As a user, I would like to be able to view deep analytics on a stock so that I can better make buying and selling decisions

Description	Estimated Time	Owner
Support endpoints for accessing analytical data	4hrs	Daniel
Algorithms to determine analytical scores	8hrs	Daniel
Display information on stock page from parsed JSON	4 hrs	Aditya, Hugh
Create unit tests to ensure functionality	3hrs	Daniel
Testing	12 hrs	Daniel

**Completed:** The deep analytics are displayed on the stock page and errors are printed to the user. The user can change the stock they are looking at and view updated data. On the event that the Alpha Vantage API returns malformed JSON or fails to return, the router will return a JSON error object.

## Sprint 1 - Incomplete User Stories, Acceptance Criteria

### User Story #1

As a user, I would like to be able to register for an account

Description	Estimated Time	Owner
Handle registration logic and validate the request	3hr	Hugh

**Completed:** When the user creates an account, a bar showing the strength of their password is shown below the field and changes dynamically.

### User Story #2

As a user, I would like to be able to login to my account

Description	Estimated Time	Owner
-------------	----------------	-------

Integrate the login page with the rest of the application	3hr	Hugh
Handle login logic and validate the request	5hr	Peyton, Niyati

**Completed:** When the user attempts to log in to an account and types in an incorrect password 4 times, they will be locked out of their account for 30 minutes.

## User Story #7

As a user, I would like to be able to view a stock's trend/range

Description	Estimated Time	Owner
Display information on stock page from parsed JSON	2hr	Aditya

**Completed:** The user is now able to view a stock's trend and range on the stock page. The color will be updated based on the trend and what colors the user has thematically chosen.

# What did not go well?

## **Overall:**

**In general,** this sprint had only a few minor problems. One of the biggest issues we ran into was the transition to working remotely. This transition caused a few scheduling issues at the beginning, but over the course of the sprint, those issues were sorted out. Additionally, transitioning to virtual communication caused the many different meetings that we have to become longer. This was because the delay in these services caused communication between members to be slower than it was in person. There is little that we can do to improve this, but we have figured out how to schedule these meetings better in the future.

Another issue we ran into was not properly separating tasks in our user stories. While we all had a good idea of the overall features that needed to be implemented, the specific responsibilities in order to complete these tasks could have been more clearly defined.

**There were no unsuccessful user stories for this sprint.**

# How can we improve?

Sprint Two ended on a positive note, with all user stories and acceptance criteria being finished before the sprint review. That being said, there are always ways to improve and we came up with some ways to make Sprint Three our best yet. First and foremost, we want to work on the way we divide up the work. As we move into the last sprint, it's become apparent that the frontend is faced with more work than the backend. To improve the speed at which we can progress through user stories, we can shift someone from the backend side of development to the frontend side of development. This will make sure that the end of the sprint doesn't become a *sprint* through the last acceptance criteria.

One thing that we could improve upon is communication between the frontend and backend. A couple of times, the backend would change the requirements of the body of an HTML request or URL and the communication may be lost or forgotten easily and this caused a few hiccups and bugs as a result. We can fix this by getting in writing sooner some of the changes being made between endpoints. Overall, this was not a huge issue but it happened occasionally, and it is a fairly easy fix too.

Another one of the issues that we can improve is the frontend design. Some of the initial design patterns that the frontend created during Sprint 1 and 2 are now starting to show their weaknesses. Specifically, the way that we store pages and their associated CSS and tests is no longer a viable system. In Sprint 3, we will be refactoring the frontend to create a better design that supports scaling in a more long term way.

Additionally, we faced some issues with understanding and properly implementing tests. One of the initial problems was a lack of knowing exactly what was required for the testing hours we assigned. Because of this, we delayed writing most of our tests until the end of the sprint which increased the stress to finish them due to a lack of time remaining. Once we realized, we had to adjust our testing hours to fit the amount of tests we felt appropriate. Now that we know exactly what is required, we can plan testing hours accordingly.

Another area of improvement for our team is prioritizing user stories between sprints. One challenge we faced was having to go back to existing, seemingly finalized pages and reimplement CSS for them to include dark mode functionality. In the future, we would likely consider the benefit of implementing a user story early on rather than midway through the process of coding. While this may be difficult to fix on our last sprint, we can change how we prioritize stories during the sprint, making sure that backtracking will be minimized.

Finally, we can better address roadblocks by more clearly defining what tasks are needed to complete each user story. Clearly listing out these tasks and assigning them will ensure that everyone is aware of what is expected of them during the sprint and will enable them to better manage their time, as well as be able to prioritize tasks that could be roadblocks for others if they are not completed in a timely fashion.