Sprint 2 Planning Document

Team 11: Ryan Huff, Sam Kravitz, Akash Lankala, Raziq Raif Ramli, Tyler Stanish, Blake Steel

Sprint Overview

Our primary focus for this sprint will be using the foundation we made in the first sprint to implement most of the functionality of our game. In this sprint we will not be as tied down configuring hosting, deployment, ORM, and boilerplate code like we were last sprint, freeing up all our developers to work on code. This sprint will be very front-end heavy; we are going to implement all of the game pages that will be populated with data that was filled in our database last sprint. At the end of this sprint we plan on having all of our UI pages finished and a primitive version of the game completed, leaving sprint 3 open for refinement, debugging, and improving features.

Scrum Master: Ryan Huff

Meeting Plan:

Biweekly, Tuesday and Thursday evenings, as well as occasional peer coding Monday, Wednesday, and Friday mornings during what used to be class time.

Risks And Challenges:

We want to use what we learned from sprint 1 to improve in this sprint. We expect most of the challenges that occurred in sprint 1 to remain for this sprint, but after going through a complete sprint we will be better equipped to handle them. Some of these challenges include the division of tasks: we often found ourselves blocked by one another, waiting for one person to complete a story before another could start theirs. To improve this issue, we have assigned more

‘specialized’ stories this sprint, where we can go in depth on a specific section without waiting on another piece to be completed. Also, this sprint we will be committed to ensuring that all of the acceptance criteria in this document is completed for the sprint retrospective. We will modify the number of stories or the acceptance criteria to keep a manageable load while still completing a proper amount of work.

Finally, a challenge more related to the efforts of this sprint will be syncing historical cryptocurrency data with real time data. We will be using 2 different APIs to retrieve historical and live data, and finding the point where historical data ends and live data begins and seamlessly transitioning from one to the other will be a challenge.

Current Sprint Details

Landing Page and Global Game Stories

User Story # 13: View Prices of Coins On Landing Page

As a player, I would like to see a landing page that shows the current prices of several cryptocurrencies.

#	Description	Estimated Time	Owner
1	Create a component where pricing information about different cryptocurrencies can reside and can dynamically update based on an arbitrary choice of coins	3 hrs	Ryan
2	Fetch data from the database to serve the current prices of ten arbitrary coins	2 hrs	Ryan
3	Connect that data onto the created module to display onto the landing page	2 hrs	Ryan
4	Sort the listed coins based on price over a base currency (such as USD)	1 hr	Ryan
5	Create tests to ensure that an arbitrary number of information components are displayed on the landing page	1 hr	Ryan
6	Create automated tests to ensure displayed data matches current data.	2 hrs	Ryan

Acceptance Criteria:

- Given a functional connection to the database and correct data, when I navigate to the landing page, the current price of ten arbitrary coins will be displayed.
- Given a set of data for different coins, when the landing page gets populated with coins, it will display in order of price based on a base currency.
- Given an arbitrary set of coins, the layout of the page will be modified based on the number of coins in that set.

- Given a functional module for coin information, a suite of tests will ensure that data displayed matches current data.

User Story # 14: Landing Page Graphs

As a player, I would like to see a landing page that shows graphs of historical data of cryptocurrencies.

#	Description	Estimated Time	Owner
1	Create a module to graph historical data based on data from our database	4 hrs	Ryan
2	Create a backend module to request the relevant data from the database to use in our graphing module	2 hrs	Ryan
3	Ensure proper error handling when given bad or incomplete data from the database	2 hrs	Ryan
4	Add the created module onto the landing page dynamically based on an arbitrary number of coins	1 hr	Ryan
5	Create tests to check for error handling and displaying a blank graph if errors persist.	2 hrs	Ryan

Acceptance Criteria:

- Given historical cryptocurrency data in the database, when the landing page is visited, it will display a graph of historical data for each cryptocurrency on the page.
- Given data from the database, when a graphing module is created, it will be able to display data selected from our database.
- Given a working backend module, when erroneous data is passed to it, it will handle the data appropriately and retry requesting data. If error persists after an arbitrary number of tries, it will display a blank graph.

User Story # 15: Joining the Global Timed Game

As a player, I would like to be able to join the global timed game.

#	Description	Estimated Time	Owner
1	For development purposes, create an entry for the global timed game in the database each time the development server is run, via migration scripts.	1 hr	Ryan

2	When a user selects the global timed game option on the select a game page, create a GameProfile database entry for that user, if not yet already created	1 hr	Ryan
3	Once the global timed game is completed, automatically create a new global timed game and add it to the database.	2 hr	Ryan
4	Test that the creation of global timed game is automated and creates a game	1 hr	Ryan

Acceptance Criteria:

- Given a freshly started database during development, when the development server is started, a new global timed game will be added to the database.
- Given a completed global timed game, a new global timed game will be created automatically.
- Given a player has already joined a session of a global timed game, once they click on the global timed game option on the select a game page a new GameProfile database entry will not be created for them.

User Story # 16: Joining the Global Indefinite Game

As a player, I would like to be added to the global indefinite game, so that I can play and compete with strangers.

#	**Description**	**Estimated Time**	**Owner**
1	For development purposes, create an entry for the global indefinite game in the database each time the development server is run, via migration scripts	1 hr	Ryan
2	In the registration process, when a user registers for an account, create a GameProfile database entry for the global indefinite game of that user's game information (money, etc)	1 hr	Ryan
4	Test that a user is added to the global game when registering for an account	1 hr	Ryan

Acceptance Criteria:

- Given a freshly started database during development, when the development server is started, a new global indefinite game will be added to the database.
- Given a player completes registration for an account, once registration is completed they will be added to the global indefinite game
- Given a running global game, it will display information starting from the time it was created.

Select a Game Page ("Play" Page) and Join Page Stories

User Story # 17: Play Page

As a player, I would like to have a play page that displays the different types of games that I could play.

#	Description	Estimated Time	Owner
1.	Create a new dedicated page.	45 mins	Raziq
2.	Research about button components and create dedicated buttons to navigate to the global game, global timed game, and create game page.	1 hour	Raziq
3.	Write tests to ensure that all buttons are rendered.	30 mins	Raziq
4.	Write tests to ensure that each button redirects the player to the correct page.	1 hour	Raziq

Acceptance Criteria:

- Given the page was successfully created, when I try to access it, I will be able to see all the appropriate buttons.
- Given the page was successfully created, when a non-authenticated user tries to access it, the page will not return.
- Given a button was successfully rendered, when I click on it, I will be redirected to the correct page.

User Story # 18: Navigating Current Active Games

As a player, I would like to be able to navigate to any of my currently active games.

#	Description	Estimated Time	Owner
1.	Learn how data is structured in the backend and pull player's active games.	2 hours	Raziq
2.	Create dedicated buttons to navigate to all of the player's active games.	1 hours	Raziq

3.	Research suitable components to group a list of components into pages.	2.5 hours	Raziq
4.	Split the buttons into several pages if the player has many active games.	2 hours	Raziq
5.	Add a search bar to filter active games by name.	3 hours	Raziq
6.	Add a dropdown widget to sort the active games by name (other attributes can also be considered in the future, if needed).	2 hours	Raziq
7.	Write tests to ensure that correct results are returned when user filters games	1 hour	Raziq
8.	Write tests to ensure that clicking a button will redirect users to the correct page.	30 mins	Raziq
9.	Write tests to ensure that active game buttons are split into several pages.	1 hour	Raziq
10.	Write tests to ensure that active game buttons are sorted properly when the option is used.	1 hour	Raziq

Acceptance Criteria:

- Given a button for an active game was successfully rendered, when I click on it, I will be redirected to the correct game page.
- Given a search bar was successfully implemented, entering a string into it will allow me to only view active games that contain the string in their name.
- Given a button for an active game was successfully rendered, if the name of the game is too long, the name will be truncated.
- Given a button for an active game was successfully created, when I hover my cursor on that button, a tooltip displaying the game's full name will appear.
- Given that the sorting feature was successfully implemented, when I select an option, my list of games will be sorted according to that option.

User Story # 19: Joining Private Games

As a player, I would like to be able to join a private game that has been created.

#	Description	Estimated Time	Owner

1.	Create a button to join a game.	30 mins	Raziq
2.	Research suitable react components to be used and implement a popup page.	3 hours	Raziq
3.	Create a textbox inside the popup page for users to enter the game code.	30 mins	Raziq
4.	Request the game from the backend.	1 hour	Raziq
5.	Add the player into the game at the backend.	1.5 hours	Raziq
6.	Learn to write backend tests and test that the user was added into the game properly.	3.5 hours	Raziq
7.	Research a suitable react component to be used and display an error message if the user enters a code to a non-existing game.	1 hour	Raziq

Acceptance Criteria:

- Given that a button was successfully created, clicking on it will display a popup page.
- Given that a popup page was successfully rendered, clicking on the ‘X’ button or anywhere outside of the page will close it.
- Given that a textbox was successfully created, when I enter a correct game code, I will be redirected to the correct game page and be added into the game.
- Given that a textbox was successfully created, when I enter an incorrect game code, I will get an error message.

Game Screen Stories

User Story # 20: Game screen

As a player, I would like to have a dedicated game screen for every game that I join.

#	Description	Estimated Time	Owner
1	See title, current cash, current net worth, time remaining.	4 hrs	Sam
2	See liquidate button, time span button array	3 hrs	Sam
3	See “share” button (icon) which has a dropdown of shareable code and shareable link	3 hrs	Sam
4	Create deeply-rendered black box visual tests	3 hrs	Sam

Acceptance Criteria:

- Given the game screen appears, then I should see my current cash, my current net worth, and time remaining.
- Given the game screen appears, I should see and be able to click all available buttons.
- Given the code compiles, the game screen should be tested.

User Story # 21: Coin table

As a player, I would like to see a table of coins for any given game.

#	Description	Estimated Time	Owner
1	See table of information, graph not functional	4 hrs	Sam
2	Can type in amount to amount buyer, buy/sell sends that amount of that coin to backend.	3 hrs	Sam
3	Parse coin data and create graphs for each coin	3 hrs	Sam
4	Create deeply-rendered black box visual tests	2 hrs	Sam

Acceptance Criteria:

- Given the coin data, when I view the table, it should populate with coins.

- Given the table renders, when I type numbers (positive only) into the amount field and click buy, it should update my account with the appropriate amount.
- Given the coin data, when I view the table, it should create a graph for each coin.

User Story # 22: Get game

As a player, I would like to receive accurate game information.

#	Description	Estimated Time	Owner
1	Backend: Create “Get game by id” endpoint, which returns Title, current cash, current net worth, end date, and coins by user	5 hrs	Blake
2	Frontend: Parse data	3 hrs	Sam
3	Create black box integration tests	2 hrs	Blake

Acceptance Criteria:

- Given a request to the backend, when I receive a certain ID, I should return game data associated with that ID and the current user.
- Given the data from the backend, I should assign data to appropriate redux store variables.
- Given the code compiles, it should be tested by different parameters.

User Story # 23: Get Coins by game ID (filtered)

As a player, I would like to get all of a game’s active coins.

#	Description	Estimated Time	Owner
1	Backend: Take game ID, time span, and sort by categories (coin name, price, % change, your amount, page)	4 hrs	Blake
2	Backend: Return appropriate coins	2 hrs	Blake
3	Frontend: Send time span and above sort by categories	1 hr	Blake
4	Frontend: Send appropriate coins to table	2 hr	Blake
5	Create black box integration tests	3 hrs	Blake

Acceptance Criteria:

- Given a game ID, time span, and sort by categories, when I send this to the backend, it should return the matching coins.
- Given a set of returned coins, it should be assigned to the proper redux store variables.
- Given the code compiles, it should be tested sorting.

User Story # 24: Buy/Sell endpoint

As a player, I would like to be able to buy/sell a given coin with my chosen quantity.

#	Description	Estimated Time	Owner
1	Backend: create “Buy/sell currency” endpoint, which takes a coin id and amount and updates the user’s profile with that coin	3 hrs	Blake
2	Create black box integration tests	1 hr	Blake

Acceptance Criteria:

- Given a positive input to amount field, buying should trigger the endpoint “buy” and sell should trigger the endpoint “sell”
- Given a user has the proper cash, selecting buy with a positive amount, cash should be removed from a user and coins should be added.
- Given buy/sell endpoint created, buy and sell should work as intended, including erroring when attempting to make an invalid transaction and not allowing a negative buy or sell

User Story # 25: Liquefy endpoint

As a player, I would like to have a button to liquefy my assets.

#	Description	Estimated Time	Owner
1	“Liquefy” endpoint, which sells all of a users’ coins	3 hrs	Blake
2	Create black box integration tests	1 hr	Blake

Acceptance Criteria:

- Given the user sees the liquify button, they should be able to click and exchange all coins for cash.
- Given the user liquifies assets, the backend should compute the request accurately.

- Given the backend returns 200 (OK), the frontend should update to display updated data.

Leaderboard Page Stories

User Story # 26

As a developer, I would like to create a leaderboard in which I can view hiscores from other players as well as myself.

#	Description	Estimated Time	Owner
1	Create “Leaderboard” game page that can be accessed by players	4 hrs	Akash
2	Create text boxes to display leaderboard page information	3 hrs	Akash
3	Create SQL queries to fetch users net worth along with their usernames, sort by decreasing net worth to be placed in hiscores page	7 hrs	Akash
4	Display hiscores page to players	3 hrs	Akash
5	Display current leader of hiscores	1 hr	Akash
6	Create functionality to display separate box that displays hiscores by friends	1 hr	Akash
7	Create scripts to sort high scores by friends	2 hrs	Akash
8	Create tests to ensure correct values for hiscores leaderboard are displayed based on games	2 hrs	Akash
9	Write scripts to notify user if there are any large movements on the leaderboard	3 hrs	Akash
10	Write tests for security	2 hrs	Akash
11	Ensure smooth UI on leaderboard page	1 hr	Akash

Acceptance Criteria:

- Given a populated database with users high scores, leaderboard fetches top scores through querying.
- Given a leaderboard with hiscores, players’ usernames and current net values on the leaderboards should be displayed.
- Given a high scores page, a user should be able to sort high scores by friends.

Real-time Data Fetching and Serving Stories

User Story # 27

As a developer, I would like to create a service/daemon to fetch live cryptocurrency data.

#	Description	Estimated Time	Owner
1	Write service in background for fetching live data and algorithms for parsing data and then storing data into the database	7 hrs	Tyler
2	Search for websites serving real-time data and read API documentation	3 hrs	Tyler
3	Write stubbed implementation for development with randomly generated data	3 hrs	Tyler
4	Unit test ability to parse data	3 hrs	Tyler
5	Integration test service's ability to parse data together with ability to interact with database	3 hrs	Tyler

Acceptance Criteria:

- Given the background service is in the background, the service will be able to fetch data without interfering with the server's ability to serve requests.
- Given fetching the real-time data is implemented correctly, when we run the service it should insert data into the database.
- Given the real-time data service is implemented correctly, then it should be able to run as long as the server is running and should not crash.

User Story # 28

As a developer, I would like to serve live cryptocurrency prices (and potentially other cryptocurrency statistics) to the user with websockets.

#	Description	Estimated Time	Owner
1	Create websocket endpoints to serve cryptocurrency prices	5 hrs	Tyler

2	Create stubbed implementation for development	3 hrs	Tyler
3	Manually test that the websocket serves the correct prices and that it is generally working	2 hrs	Tyler
4	Smoke test that the websocket doesn't crash or stop the server from serving other requests	2 hrs	Tyler

Acceptance Criteria:

- Given serving live cryptocurrency prices is implemented correctly, when a user accesses a websocket endpoint, then the user should receive new price data in real-time.
- Given serving live cryptocurrency prices is implemented correctly, it should integrate with the service that fetches the data from external APIs.
- Given serving live cryptocurrency prices is implemented correctly, when the server is serving a websocket the server should also be able to handle other connections to the server.

Remaining from Sprint 1

User Story #4

As a developer, I would like to integrate historical cryptocurrency prices into the platform.

#	Description	Estimated Time	Owner
1	Obtain historical data files	0 hr *	Tyler
2	Parse historical data files	0 hr *	Tyler
3	Write script that inserts data into database	0 hr *	Tyler

* This story has 0 hours remaining because it was worked on between sprint review 1 and sprint 2 and is now completed and functional.

Acceptance Criteria:

- Given integrating historical data is successful, when I query the database, then I should get results of historical data.
- Given integrating historical data is successful, when I give the program a csv file, then it should populate the database.
- Given new data to integrate, when the program populates the database, then it should be retrievable via a query from the backend.

User Story #6

As a developer, I would like to only allow authenticated users to access some routes in Fortune.

#	Description	Estimated Time	Owner
1	Decorate route functions that we want to only be accessible to logged in users.	1 hr	Tyler
2	Persist user's token in localStorage on browser	2 hrs	Sam

Acceptance Criteria:

- Given the limited access to authenticated resources is implemented correctly, when a valid user tries to access the api at a certain authenticated route, then the user should be able to access that route.

- Given the Python decorator is implemented correctly, when an invalid user tries to access the api at a certain authenticated route, then the server should return an error.
- Given the Python decorator is implemented correctly, when a request is made to an authenticated route with no information or with poorly (or maliciously) formatted data, then the server should return an error.

User Story #11

As a player, I would like a choice of initial cash amount each player has.

#	Description	Estimated Time	Owner
3	Check for invalid input and alert the user when invalid input is entered	1 hr	Akash

- Given adding the choice of initial cash amount each player has is successful, when I create a new private game with the amount of initial cash specified, then all users should start with that amount.
- Given that a successful suite of tests have been written, when a user inputs an invalid input into the cash box, such as a negative number or a string, the input will be rejected.
- Given that the text box has been added successfully, it should appear in a visually pleasing way among the other data fields in the create game page.

Remaining Backlog

Green = On this Sprint
~~Strikethrough~~ = Completed Last Sprint

Functional Requirements

1. General

As a player,

a. I would like to see a landing page that includes the current prices of several cryptocurrencies.
b. As a player, I would like to see a landing page that shows graphs of historical data of cryptocurrencies.
c. I would like to have a play page that displays the different types of games that I could play.
d. I would like to have a dedicated game screen for every game that I join.
e. I would like to have a guest player option (optional).
f. I would like to have a notification popup bar where users can get notified of any new updates, friend requests, and game invitations (optional).
g. I would like to have a trophy system to keep track of my achievements and have some major goals to look forward to in this game (optional).
h. I would like to be given a set of goals that I can accomplish every week to motivate me to come back to the game from time to time (optional).
i. I would like to backtest trading bots to test my cryptocurrency trading models (if time permits and obtaining real-time cryptocurrency data is not feasible).

As a developer,

a. I would like to have an admin page to manage players and enforce game policies.
b. I would like to create a service/daemon to fetch live cryptocurrency data.
c. I would like to parse the raw data from (potentially) several APIs.
d. I would like to integrate historical cryptocurrency prices into the platform.
e. I would like to serve live cryptocurrency prices (and potentially other cryptocurrency statistics) to the user with websockets.
f. I would like to calculate a player's net worth, so that a player can see an accurate representation of their standing compared to others.

2. Player account

As a player,

~~a. I would like to be able to register for a Fortune account.~~

b. I would like to have a popup page that displays my profile's details.
c. I would like to be able to upload a profile picture (optional).
d. I would like to be able to log in and log out from my account.
e. I would like to be able to change my username (optional).
f. I would like to be able to reset my password (optional).
g. I would like to be able to add friends, so that I can compete with them and compare our progress (optional).
h. I would like to be able to view and manage my friends' list (optional).
i. I would like to link my account to Facebook, so that I can find my friends who also play this game easily (optional).
j. I would like to be able to register with my Google account so that the registration process could be done quicker (optional).

3. Creating a game

As a player,

~~a. I would like to create a private group game through the play page.~~
~~b. I would like a choice of game title.~~
~~c. I would like a choice of duration of the game.~~
~~d. I would like a choice of which cryptocurrencies (BTC, ETH, etc) are to be traded during the game.~~
~~e. I would like a choice of initial cash amount each player has.~~
f. I would like a shareable link to give to other players to invite them.
g. I would like a 4-digit code to give to other players to invite them.
h. I would like to invite my friends directly by entering their usernames (optional).

4. Joining / navigating to a game

As a player,

a. I would like to be able to join the global indefinite game, so that I can play and compete with strangers.
b. I would like to be able to join the global timed game.
c. I would like to be able to join a private game that has been created.
d. I would like to be able to navigate to any of my currently active games.

5. Playing a game

As a player

a. I would like to see the title of the game.
b. I would like to see a button that reveals the 4 digit code to join the game.
c. I would like to see my current cash within the current game.
d. I would like to see my current net worth.
e. I would like to see the time remaining in a game.

f. I would like to see a time series graph displaying the historical price of a cryptocurrency from the exchange(s) of my choice.
g. I would like to be able to switch the cryptocurrency that is being displayed in the graph.
h. I would like to modify the time span of crypto data to display (min/hr/day/month/yr), so that I may modify the data view to fit my buying and selling needs.
i. I would like to be able to buy mock cryptocurrency and have it attached to my account.
j. I would like to obtain more money if my net balance goes to zero (go into debt)
k. I would like to be able to liquefy all current assets immediately.
l. I would like to be able to buy/sell currency and view data about said currency (Price, history, % change, amount, min/max price on various exchanges)
m. I would like to have an option to buy coins from an exchange that offers the lowest price and sell coins to an exchange that offers the highest price at any given time.
n. I would like to see the current leader of the game.
o. I would like to be able to navigate to the leaderboard for the current game.
p. I would like to filter the leaderboard by friends (optional).
q. I would like to see players' usernames and current net values on the leaderboards.
r. I would like to receive in-app notifications for large movements in crypto price.
s. I would like to receive in-app notifications if there are any significant movements in the leaderboard.
t. I would like to see a widget that displays notification history throughout the game (optional).
u. I would like to chat with other players who are in the same game session with me (optional).
v. I would like to report players who make any form of communication abuse (optional).

Non-Functional Requirements

1. Architecture

~~a. As a developer, I would like to employ a client-server architecture for Fortune.~~
~~b. As a developer, I would like to develop the frontend with React and the server in Python with Flask.~~
~~c. As a developer, I would like to store user and cryptocurrency data with a hosted PostgreSQL database.~~

d. ~~As a developer, I would like to deploy Fortune through Amazon Web Services in order to integrate our frontend and backend easily.~~

2. Design

a. As a developer, I would like to make an intuitive, straightforward, and interactive interface.
b. As a developer, I would like the players to be able to navigate through the application and execute trades seamlessly.
c. As a developer, I would like to use a pleasing color scheme that will add to the aesthetic of the application. The color palette that I plan to use can be viewed here https://coolors.co/4aa7d6-2a628f-13293d-79b473-db504a.

3. Security

a. As a developer, I would like to secure the sensitive data that is stored in the database.
b. As a developer, I would like to only allow authenticated users to access some routes in Fortune.

4. Performance

a. As a developer, I would like the application to be responsive to all requests made by the user.
b. As a developer, I would like the application to be able to properly handle any raised errors.