Sprint 1 Planning Document

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

Sprint Overview

In the initial sprint, we will spend time getting accustomed to our tech stack and understanding how each piece of technology will be wired together to create an app. Our plan is to have a skeleton in place for our app by the end of the sprint, however much of the functionality is to be implemented in future sprints. For example, we may not have a playable game by the end of sprint 1, but we may have a create game screen set up with non-functional buttons. We also plan on having our backend set up and hosted with AWS by the end of sprint 1, so we can begin sending information back and forth between our frontend, backend, and database. Finally, we want to have authentication and profile creation in place this sprint. This sprint will mainly be used to set up a barebones version of our app, allowing us to flesh out functionality in sprints 2 and 3.

Scrum Master: Akash Lankala

Meeting Plan: Tuesday/Thursday at 7:00 pm

Risks And Challenges:

The main challenges we will encounter in this sprint will include learning new languages and technologies quickly enough to contribute useful code to the app. Another challenge we are

anticipating having is setting up a unified development environment amongst ourselves. We are still weighing our option for if we are each going to use an individually hosted local server, or set up a development specific server online that we can all connect to at once. Other than that, we are looking forward to a smooth and productive sprint.

Current Sprint Details

User Story #1

As a developer, I would like to develop the frontend with React and the server in Python with Flask

#	Description	Estimated Time	Owner
1	Set up Flask server and backend boilerplate	4 hrs	Tyler
2	Create create-react-app project	2 hrs	Blake Steel
3	Ensure UI Conformity between different pages of the game	3 hrs	Akash
4	Add Redux and state management skeleton?	5 hrs	Tyler
5	Create basic client routing and navigation	5 hrs	Blake Steel
6	Write/set up basic testing framework and smoke tests on the client	5 hrs	Akash
7	Read ORM documentation and translate database tables to Python ORM classes	5 hrs	Tyler

Acceptance Criteria:

- Given the backend is set up correctly, when I make an HTTP request to our backend, then I expect to get a JSON-formatted response.
- Given the client is set up correctly, when I go to the webpage, then I expect to be able to navigate around some basic pages.
- Given the database tables are represented in server code, when I write code on the backend, then I expect to be able to easily interact with database objects.

User Story #2

As a developer, I would like the application to be able to properly handle any raised errors.

#	Description	Estimated Time	Owner

1	Create JSON error handlers with Flask	3 hrs	Raziq
2	Create custom exceptions to raise like UnauthenticatedException or BadRequestException	4 hrs	Akash
3	Create catch-all error handler for unexpected server errors	1 hr	Raziq

Acceptance Criteria:

- Given basic error handling is done correctly, when a user sends an invalid request, then I expect the server to send a JSON-formatted error response.
- Given basic error handling is done correctly, when an unauthenticated user tries to access any authenticated routes, then I expect the server to send a JSON response stating that the user is not allowed to access this resource.
- Given basic error handling is done correctly, when the server throws any kind of unexpected exception, then I expect the server to return a 500 response.

User Story #3

As a developer, I would like to store user and cryptocurrency data with a hosted PostgreSQL database.

#	**Description**	**Estimated Time**	**Owner**
1	Register for AWS	1 hr	Ryan
2	Create AWS RDS instance for PostgreSQL	1 hr	Ryan
3	Create migration scripts	4 hrs	Akash
4	Migrate the database	2 hr	Sam
5	Set up database connectivity in Flask	5 hrs	Ryan

Acceptance Criteria:

- Given the database is up on a hosting platform, when I try to access/insert data from the backend, then I expect to be able to receive/insert data.
- Given the database is up on a hosting platform, when I get new data from the background worker, then I expect to be able to insert that data into the database.
- Given the database migrations are written correctly, when I apply the migrations to the database, I expect to be able to change the database's schema.

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	1 hr	Tyler
2	Parse historical data files	1 hr	Tyler
3	Write script that inserts data into database	4 hrs	Tyler

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 #5

As a player, I would like to be able to register for a Fortune account.

#	Description	Estimated Time	Owner
1	Create login page	6 hrs	Blake
2	Create registration page	6 hrs	Ryan
3	Create API endpoint for login	6 hrs	Sam
4	Create API endpoint for registration	6 hrs	Sam

Acceptance Criteria:

- Given creating a registration page is successful, when I register for an account, then the account will be added to the database if the username has not been taken yet and the password is strong.

- Given creating a login page is successful, when I log in, then I should be redirected to a different page.
- Given that my account has been successfully created, then I expect to be redirected to a different page.

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	Create Python authentication decorator	4 hrs	Tyler
2	Persist user's token in localStorage on browser	2 hrs	Sam
3	Write tests to ensure the authentication decorator only allows valid users with valid tokens	4 hrs	Tyler

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 #7

As a player, I would like to create a private group game through the play page.

#	Description	Estimated Time	Owner
1	Create "Create New Game" page	9 hrs	Blake
2	Create API endpoint for creating a new group game	4 hrs	Sam
3	Create API endpoint for updating a game	4 hrs	Sam
4	Write tests	5 hrs	Sam

Acceptance Criteria:

- Given creating a private group game is successful, when I create a new private game, then I would be redirected to the game's screen.
- Given the creation of API endpoints for creating a game are successful, when I create a game over the JSON API, then I expect to have created a game.
- Given the creation of the API endpoints for updating a game are successful, when I update a game through the web interface or through the API itself, then I expect to be able to have updated the game's information successfully.
- Given that creating a private group game is successful, when I create the game, I should be able to view this game on my profile page.

User Story #8

As a player, I would like a choice of game title.

#	Description	Estimated Time	Owner
1	Add text box to create game page with label to name private game	2hrs	Blake
2	Write tests	2 hrs	Blake
3	Check for invalid input and alert the user when invalid input is entered	1 hr	Blake

Acceptance Criteria:

- Given that the text box has been successfully added, it will appear in the create game screen with other data fields.
- Given that a game has been successfully created, the textbox input will be used as the title.
- Given an invalid input such as empty string is entered into the text box, when the user clicks the submit button, the input will be rejected.

User Story #9

As a player, I would like a choice of duration of the game.

#	Description	Estimated Time	Owner

1	Add calendar to create game page to select end time of game	3 hrs	Raziq
2	Display duration of selected end time	1 hrs	Raziq
3	Send data on save to game endpoint to update	2 hrs	Raziq
4	Write tests	5 hrs	Raziq

Acceptance Criteria:

- Given that the calendar has been added successfully, it will appear in the create game screen with other widgets.
- Given the create game screen, when I enter valid input into the dateTime picker, then submitting should send the datetime to the API.
- Given that a game has been successfully created, then it will end exactly at the date and time specified in the dateTime picker.
- Given that a successful suite of tests have been written, the user will be prevented from selecting an invalid date and time.

User Story #10

As a player, I would like a choice of which cryptocurrencies (BTC, ETH, etc) are to be traded during the game.

#	**Description**	**Estimated Time**	**Owner**
	Create dropdown list of coins on create game page to select from	2 hrs	Raziq
	Send data on save to game endpoint to update	2 hrs	Raziq
	Write tests to verify the dropdown shows each option and can select the option the user wants to choose	5 hrs	Raziq

Acceptance Criteria:

- Given that the dropdown list has been added successfully, it should appear in the create game page with all the other data fields.
- Given that creating a private group game is successful, when I create the game, only the selected coins should be available for trade.

- Given that a successful suite of tests have been written, then if the user made an invalid selection such as an empty selection, the input will be rejected.

User Story #11

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

#	Description	Estimated Time	Owner
1	Create text box on create game page with label to specify initial starting cash each player starts with	2 hrs	Akash
2	Write tests to verify the text box is visible and can update when the user types text into it	2 hrs	Akash
3	Check for invalid input and alert the user when invalid input is entered	1 hr	Akash

Acceptance Criteria:

- 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.

User Story #12

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

#	Description	Estimated Time	Owner
1	Deploy EC2 instance for production purposes	3 hrs	Ryan
2	Install and configure web and application hosting software such as nginx on production EC2 instance	3 hrs	Ryan
3	Use AWS Directory Service to join EC2 Instance and RDS database through the same domain.	2 hrs	Ryan

4	Point github.io domain of Github repository to EC2 instance for ease of connection	2 hrs	Ryan

Acceptance Criteria:

- Given a need to host our game, when we start up our web server, then the application will be available to access via the Internet.
- Given a working API for database calls, when we call to fetch data from the database, then the database will be accessed through the same domain.
- Given a running web server, when a user types in fortune.github.io into their browser, then they will be directed to our game.

Remaining Backlog

Functional Requirements

1. General

   As a player,

   a. I would like to see a landing page that includes the amount of money I currently have in the global game and the current prices of several cryptocurrencies.
   b. I would like to have a play page that displays the different types of games that I could play.
   c. I would like to have a dedicated game screen for every game that I join.
   d. I would like to have a guest player option (optional).
   e. I would like to have a notification popup bar where users can get notified of any new updates, friend requests, and game invitations (optional).
   f. 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).
   g. 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).
   h. 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

- 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.
- 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.
- 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.
- 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.