Sprint 3 // Testing Document

User Story 1: As a user, I would like to be able to ass a profile picture

- **Test:** Reset Profile Picture Page renders without crashing
  - Nature: Automatic Testing
  - Expected: No Errors are thrown when the test is run. The component is rendered correctly
- **Test:** Initial Page State is set correctly
  - Nature: Automatic Testing
  - Expected: All initial state variables match the expect values. The Aspect ratio is 1:1 for cropping

User Story 2: As a user, I would like to be able to view personalized advice on how to manage my money so that I can make informed decisions

- **Test:** backend endpoint (not logged in)
  - Nature: Manual Testing
  - Expected: Responds with 400, and message “not logged in” in json
- **Test:** backend endpoint (logged in, account without budget)
  - Nature: Manual Testing
  - Expected: Responds with 400, and message “Budget has not been created for this month.” in json
- **Test:** backend endpoint (logged in, account with spending over budget)
  - Nature: Manual Testing
  - Expected: Responds with 200, and object list, each containing: trigger, advice, and isBudget in json
- **Test:** backend endpoint (logged in, new account with empty budget)
  - Nature: Manual Testing
  - Expected: Responds with 200, and empty list (assuming a good budget).
- **Test:** Frontend renders the Budget Advice without crashing
  - Nature: Automatic
  - Expected: No Errors are thrown when the test is run. The component is rendered correctly
- **Test:** Initial Page state is set correctly.
  - Nature: Automatic
  - Expected: When props are passed to the component, the state of the component is set correctly based on those props.

User Story 3: As a user, I would like to generate a savable document of my finances so that I can access the information offline

- Test: Component shows an error message if there is an error loading the budget data
  - Nature: Manual Testing

    - Expected: Upon the backend returning a status 400, the frontend will display a message showing that no budget was found for the given user
- Test: The finance document renders correctly
    - Nature: Automatic Testing
    - Expected: A shallow version of the component renders correctly with no errors
- Test: Initial states of finance document are set correctly upon creating a new instance
    - Nature: Automatic Testing
    - Expected: Upon creating any instance of the finance document, the initial values of the states are set correctly

User Story 5: As a user, I would like to be able to be educated on how to create a budget so that I can make more responsible decisions in my daily life

- **Test:** The component renders correctly without crashing
    - Nature: Automatic Testing
    - Expected: No Errors are thrown when the test is run. Component is rendered correctly
- **Test:** The initial state of the component is set correctly when the component is loaded.
    - Nature: Automatic
    - Expected: All initial values match the expected values.

User Story 6: As a user, I would like to be able to compare one stocks data to another's

- **Test:** The component renders correctly
    - Nature: Automatic Testing
    - Expected: No Errors are thrown when the test is run. Component is rendered correctly
- **Test:** The state of the component is set correctly when component is loaded
    - Nature: Automatic Testing
    - Expected: All initial values match the expected values.
- **Test:** The dropdown correctly shows the list of followed stocks
    - Nature: Manual Testing
    - Expected: The dropdown correctly shows the user's list.
- **Test:** The graphs render correctly in compare mode.
    - Nature: Manual Testing
    - Expected: The graph correctly displays the current stock information in comparison mode.
- **Test:** The colors render correctly in compare mode.
    - Nature: Manual Testing
    - Expected: The graph colors line up with the user's current color scheme.
- **Test:** The stock info renders correctly in compare mode.
    - Nature: Manual Testing
    - Expected: The stock info correctly displays the current stock information in comparison mode.
- **Test:** The deep analytics renders correctly in compare mode.

    - Nature: Manual Testing
    - Expected: The deep analytics correctly displays the current deep analytics in comparison mode.
- **Test:** The graphs are togglable between followed stocks.
    - Nature: Manual Testing
    - Expected: The user can toggle between followed stocks and all data is updated correctly.
- **Test:** The date renders correctly in compare mode.
    - Nature: Manual Testing
    - Expected: The stock info correctly displays the current date in comparison mode.
- **Test:** The graph is togglable to intraday in compare mode.
    - Nature: Manual Testing
    - Expected: The graph can be toggled to show intraday and renders correctly.

User Story 7: As a user, I would like to be educated on building a strong stock portfolio so that I can make more informed purchases

- **Test:** The component renders correctly
    - Nature: Automatic Testing
    - Expected: No Errors are thrown when the test is run. Component is rendered correctly

User Story 8: As a user, I would like to be informed of the risks involved in financial investments so that I can make informed decisions.

- **Test:** The component renders correctly
    - Nature: Automatic Testing
    - Expected: No Errors are thrown when the test is run. Component is rendered correctly
- **Test:** The state of the component is set correctly when component is loaded
    - Nature: Automatic Testing
    - Expected: All initial values match the expected values.

User Story 9: As a user, I would like to be educated on risk management for investing so that I can make safer decisions

- **Test:** The component renders correctly
    - Nature: Automatic Testing
    - Expected: No Errors are thrown when the test is run. Component is rendered correctly

User Story 10: As a user, I would like to be able to simulate stock investments so I can see how much net gain or loss I would incur before investing in the actual market

- **Test:** getPortfolio endpoint (logged in, no portfolio created)
    - Nature: manual Testing
    - Expected: returns 400 with message: “Portfolio not created”

- **Test:** getPortfolio endpoint (logged in, portfolio created)
    - Nature: manual Testing
    - Expected: returns 200, with following fields: wallet, investing, and stocks. Stocks is an object list each containing: code, buyPrice, buyValue, price, value, quantity
- **Test:** createPortfolio endpoint (logged in)
    - Nature: manual Testing
    - Expected: returns 200, with message: “success!”. Wallet is set to 5000 to start
- **Test:** createPortfolio endpoint (logged in, already have portfolio)
    - Nature: manual Testing
    - Expected: returns 400, with message: “Portfolio already created”
- **Test:** buyStock endpoint (logged in, have money)
    - Nature: manual Testing
    - Expected: returns 200, with message: “Successfully bought for $x”, and cost = x
- **Test:** buyStock endpoint (logged in, not enough money)
    - Nature: manual Testing
    - Expected: returns 400, with message: “not enough money”
- **Test:** sellStock endpoint (logged in, have stock)
    - Nature: manual Testing
    - Expected: returns 200, with message: “Successfully sold for $x”, and value = x
- **Test:** sellStock endpoint (logged in, don’t have stock)
    - Nature: manual Testing
    - Expected: returns 400, with message: “(code) not found in portfolio”

User Story 11: As a user, I would like to be able to share my stocks to social media outlets

- **Test:** The component renders correctly
    - Nature: Automatic Testing
    - Expected: No Errors are thrown when the test is run. Component is rendered correctly
- **Test:** The state of the component is set correctly when component is loaded
    - Nature: Automatic Testing
    - Expected: All initial values match the expected values.
- **Test:** The stock can be shared to Facebook.
    - Nature: Manual Testing
    - Expected: The facebook window will show up with the correct information.
- **Test:** The stock can be shared to Twitter.
    - Nature: Manual Testing
    - Expected: The twitter window will show up with the correct information.
- **Test:** The stock can be shared to Reddit.
    - Nature: Manual Testing
    - Expected: The Reddit window will show up with the correct information.
- **Test:** The stock that is shared will depend on the current stock.
    - Nature: Manual Testing
    - Expected: The current stock will be the one that is shared, not another stock.

User Story 12: As a user, I would like to be able to view the website on my phone and have it format correctly so that I can view my information from multiple devices

- **Test:** The proper components appear correctly on mobile
  - Nature: Manual Testing
  - Expected: Components are formatted for mobile