Crypto coin ticker and dashboard portal

# Problem statement:

Create a simple cryptocoin ticker and dashboard portal with the following functions:

* Goals:
  + Integrate with open APIs like coinmarketcap.com, poloniex, etc
  + Pick 10 coins of choice
  + Show near-real-time movement of the coins (only value)
  + Show the visualization of 2-3 periods [1 day, 1 week, 1 month etc]
  + Store data that is fetched in a DB
  + Derive metrics from the stored data [e.g. peak to peak comparison in 24 hours, simple movie average over the period, candle stick representation for the chosen periods etc]
* Stretch goals:
  + Allow user to create a portfolio of coins of choice; users should be able to login and see the value of their portfolio as per current market prices
  + Convert this to an equivalent iOS app

# Use cases:

* Obtain Data every second from a selected API and update the database every second
* Show real time data in the form of both:
  + A graph
  + A ticker
* Allow the user to sign up and create his/her portfolio on the app
* Display the value and movement of value of the user’s portfolio
* Allow the user to edit the composition of the portfolio

## Obtain Data every second from a selected API and update the database:

* Obtain historical data from an API.
* Henceforth, update live data by running a cron job to make an API call and insert into DB

Things to figure out:

* Should we update /sec or /min
* How much historical data to store?
* Database Format

## Show real time data:

1. A graph:

* Pull data from the DB.
* Plot the obtained data for a desired interval of time on a graph.
* Two options: Simple Moving Averages and Japanese Candlesticks.
* 2-3 periods: 1minute, 1 hour, few hours.

1. Live Ticker:

* Get data from API/DB.
* Display a live ticker beside the graph to show the movement of values of the cryptocoins.
* Sort them in either descending/ascending order of their values.

Things to figure out:

* For the ticker, should we make an API call every second or just update it with the database data every minute?
* If we decide to make an API call instead, should we store the data obtained every second from the call in the database.

## Allow the user to sign up and create his/her portfolio on the app:

* Provide a signup button to direct the user to the signup page
* At the signup page, let the user choose a unique username and set a password.
* Once this is done, allow the user to make their portfolio. They can add coins from the 10 coins provided and adjust the volume they want to include in their portfolio.
* This portfolio can be edited later too.

## Display the value and movement of value of the user’s portfolio:

* Once the user is signed in, the graph should display the same metrics (simple moving average or candlesticks) for the user’s portfolio.
* The live ticker on the side can be sorted by displaying the data in descending order of the impact they have on the total value of the user’s portfolio. (important parameter: volume\*changeInValue)

## Allow the user to edit the composition of the portfolio:

* Allow the user to navigate to a portfolio “page”.
* Here, they can view their current composition of the portfolio.
* Allow an edit button that switches the app into edit mode.
* In this mode, the users can change the volume for each coin, add or remove a coin.
* Finally, they are provided the choice of either saving or cancelling the changes they have made.

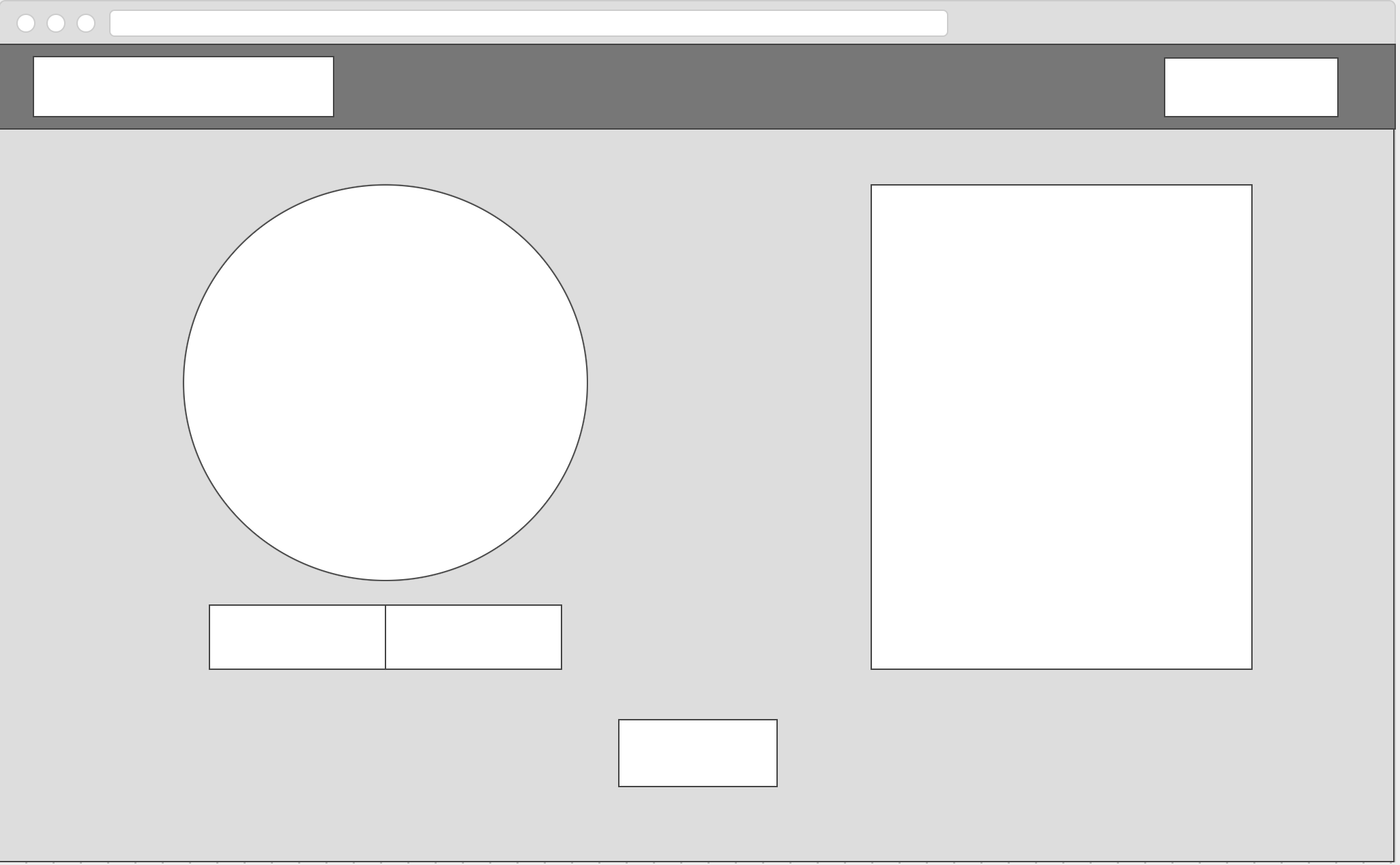
# A few layouts:



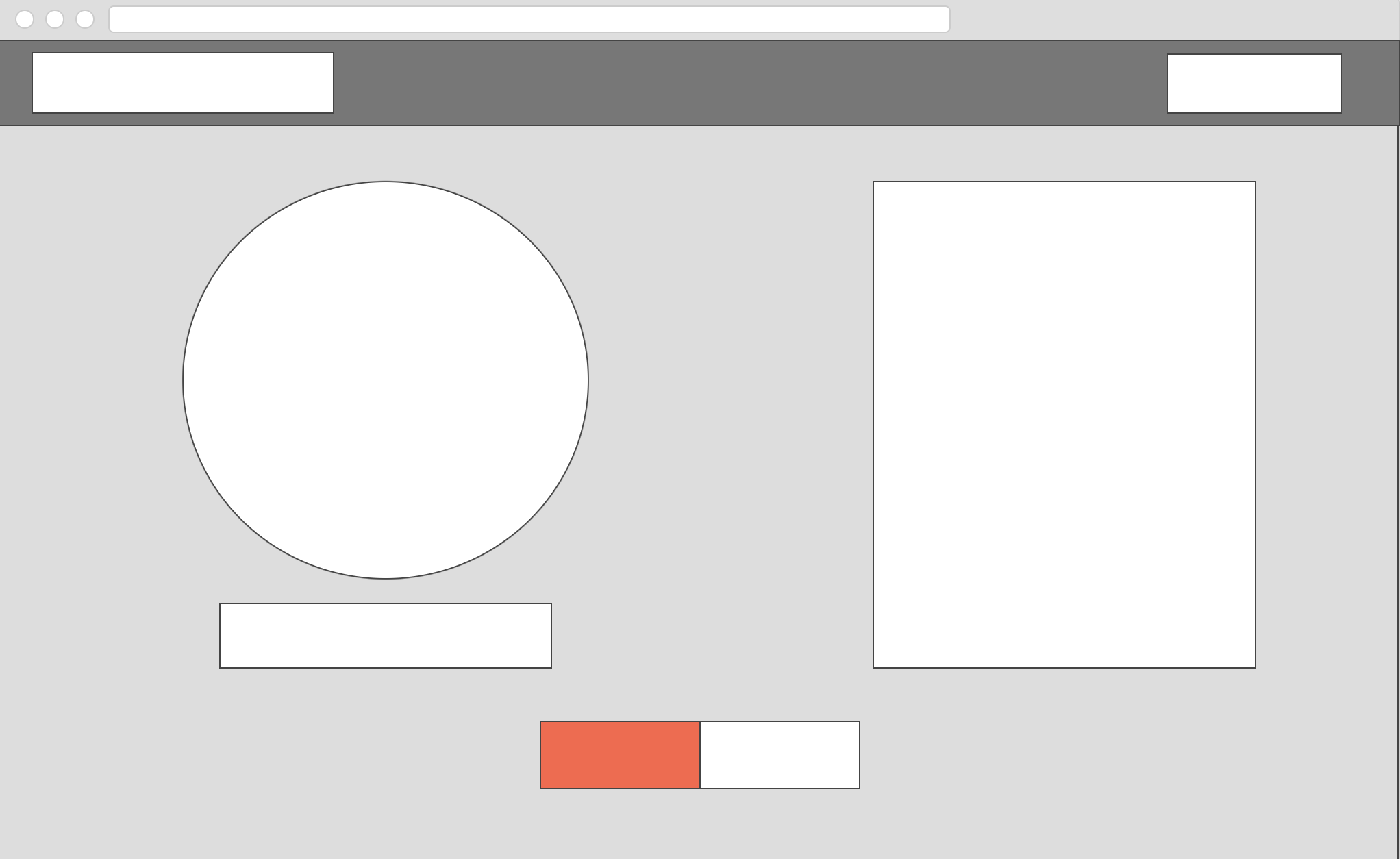
Homepage layout. Top right corner: signup button. Left half: graph, right half: live ticker.



homepage which signed in. topleft corner: portfolio and signout button. The graph and ticker remain the same.



portfolio page. Pie chart to display the portfolio composition either by volume or value.  
live ticker provided too. Bottom: edit button



portfolio page in edit mode: allow the user to add or remove coins in the ticker on the right half of the page. The ticker is dead now and sorts the coins in descending order of volume\*valueOfCoin. Bottom: save and cancel buttons.