



# Frontend Team

## Week 5

Group Members:

Ayca Gökdağ

Doğa Güneş Karadağ

Salih Bulmaz

Semih Gençten

Nour Elhouda Znagui

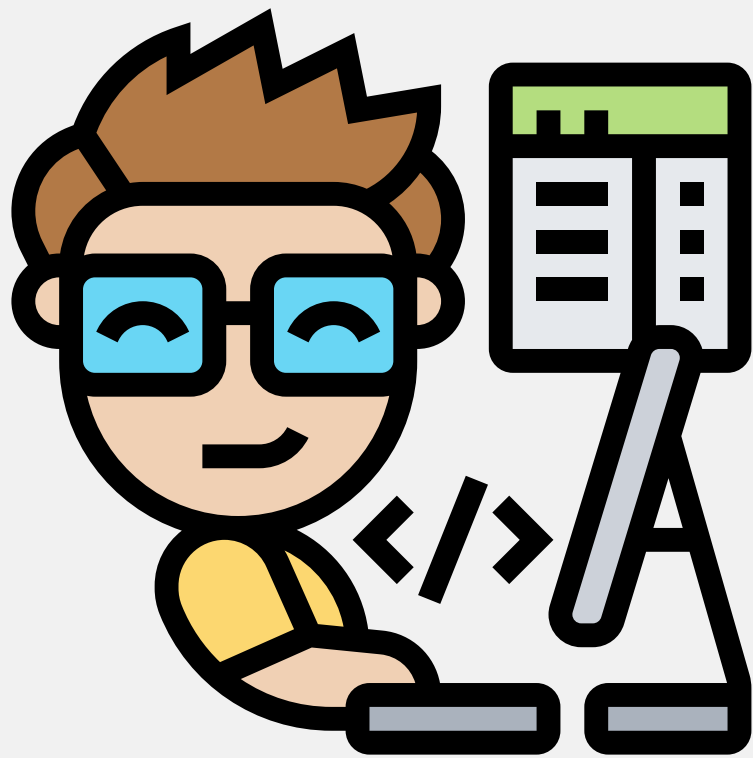
Baturalp İnce (Data Visualization)

Ayberk Bozkuş (Data Visualization)

Arda Altıntaş (System Integration)

Ubeydullah Önder (System Integration)

# About this week



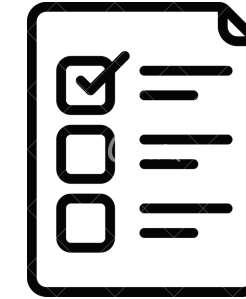
- Language Options
- Learning Process
- GitHub Workflow
- Task Division
- Website Development

- ▼ pages
  - > AnalysisPage
- ▼ FinancialsPage
  - > DividendsSubpage
  - > ProfileSubpage
  - > RiskSubpage
  - > StatemensSubpage
  - > StatisticSubpage
- > HomePage
- > MarketsPage
- > NewsPage
- > PortfolioPage
- > WatchlistPage

# Pages and Components

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}  
  
const element = <Welcome name="Faisal Arkan" />;  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```

# Task Division



- Tasks are divided into two main category (tentative):

## Business part (Redux)

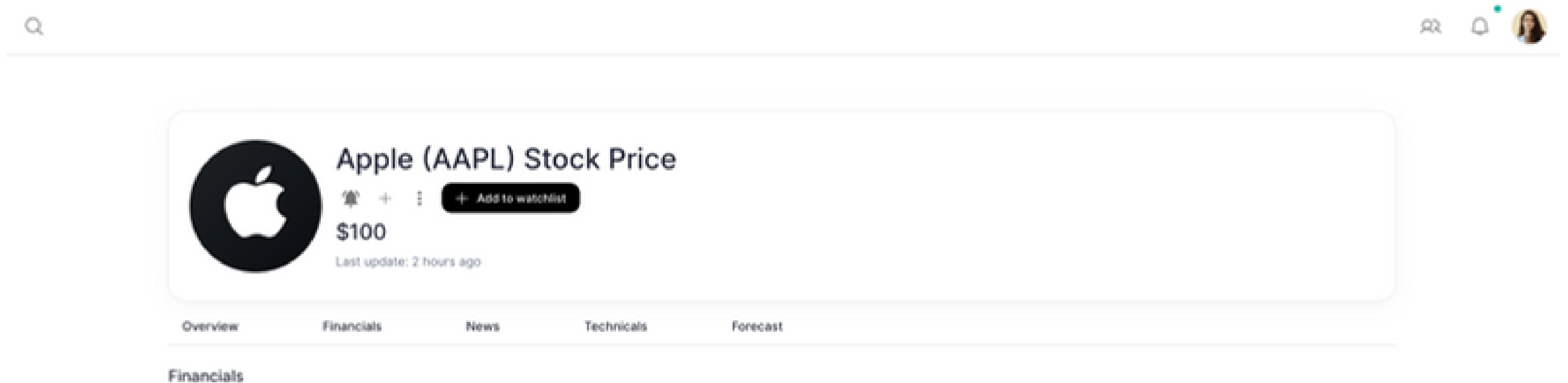
- Salih
- Ubeydullah

## UI Part

- Nour (Markets, Watchlist)
- Doğa (Financials)
- Ayca (Fincncials, Portfolio, Language options)
- Semih (Technicals, Overview)
- Arda (Charts, Peer Analysis)

- Researched about our tasks
- Tried to create some components while learning React
- Documentations for requested data will be prepared

# Attempting to create components...



# Multilanguage Website Feature



*For example, if you have an extracted message file en.json with the following contents:*

```
{  
  "greeting": "Hello, World!",  
  "buttonLabel": "Click me"  
}
```

*The translator for the French language would provide a translated file fr.json with the translated values:*

```
{  
  "greeting": "Bonjour tout le monde !",  
  "buttonLabel": "Cliquez ici"  
}
```



# Business Logic

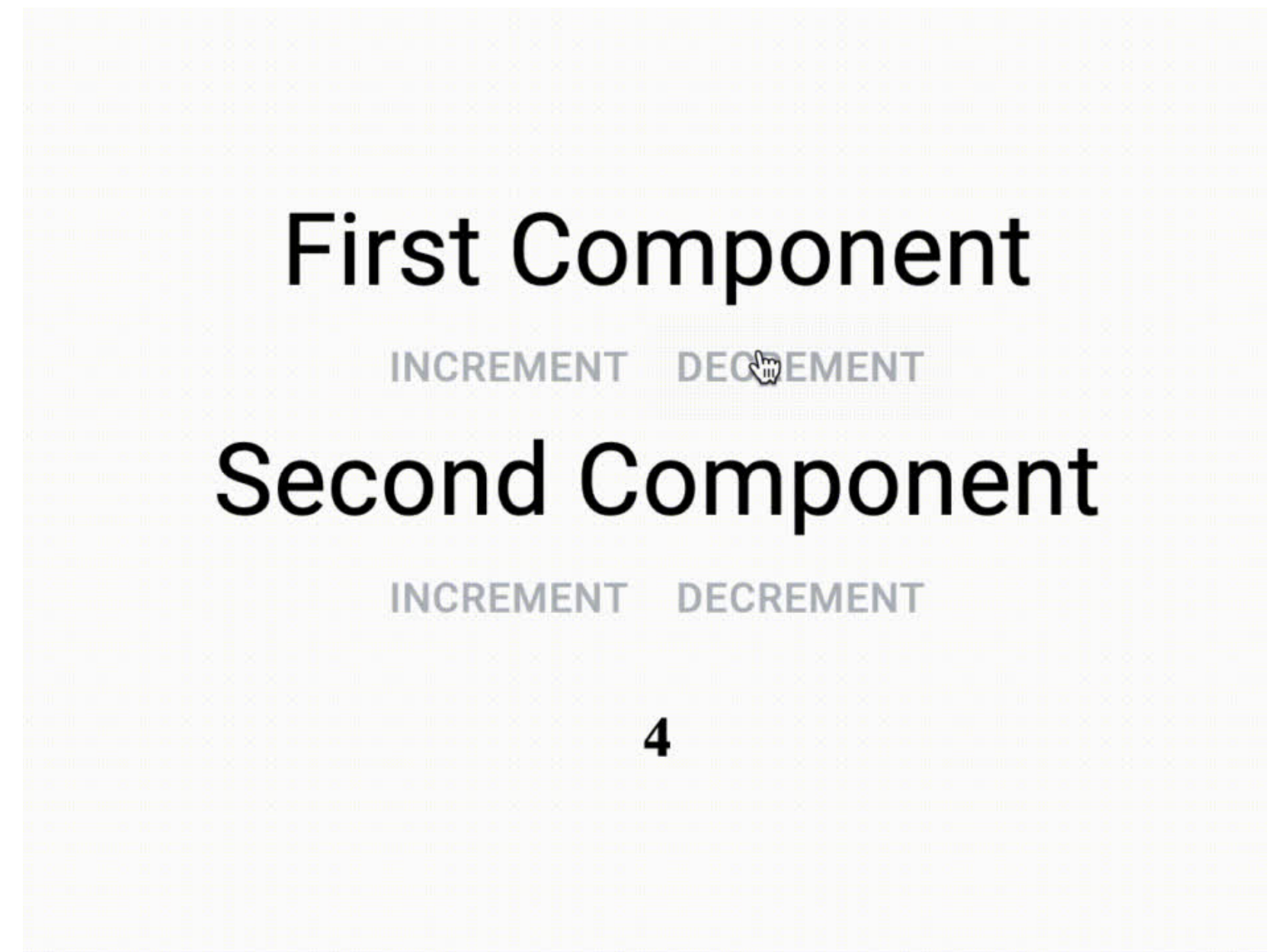
- Business logic refers to the processing and management of data, state, and backend interactions in the frontend of an application.
- We are using redux and redux toolkit to manage state globally, handle actions to interact with the backend and mutate the state accordingly.

## Pros of Using Redux

- Redux provides a single source of truth for the application's state, making it easier to manage and predict state changes.
- Redux centralizes the state management, which simplifies debugging and enhances maintainability, especially in larger applications.
- Redux has a large and active community, along with a rich ecosystem of libraries and tools, providing resources and support for developers.

# Examples of Using Redux

- Hello World of Redux is a counter. Let's assume we have a counter that has a state value and has actions like increment, decrement. Many pages and components can use the same reference and mutate the value with actions.
- In a blank react project two components are created. Both components can mutate the counter value. The value will react to both components and the state will be singular.



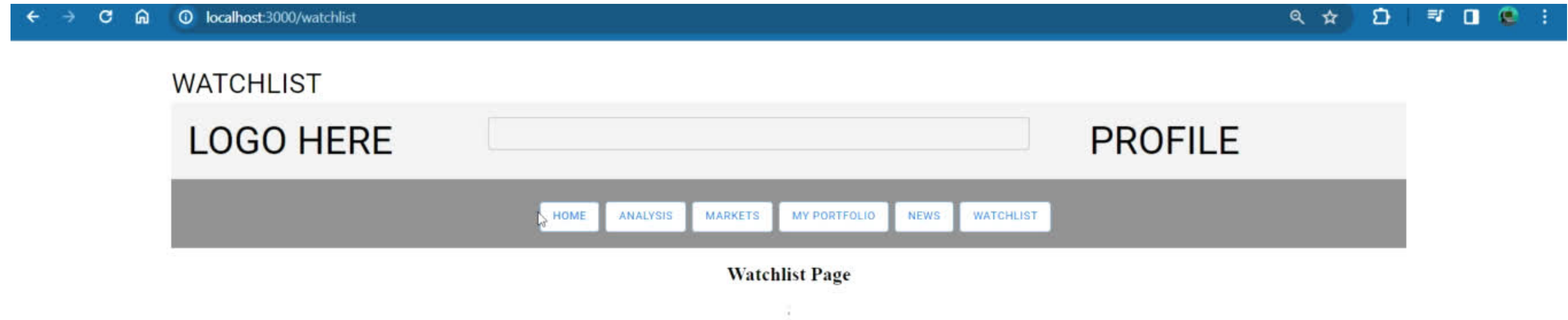


# The File Structure

- File structure is very essential in frontend projects. We have selected a structure that allows many people to work collaboratively without an issue.
- Short descriptions of each folder:
  - **Actions:** Redux actions
  - **Pages:** Page components along with any components used in those pages.
  - **Layouts:** Layout components
  - **Assets:** Svg, font, img files etc.
  - **Components:** Shared components
  - **Reducers:** Redux slices
  - **Store:** Redux store instance
  - **Utils:** Any helper function, class that is reused

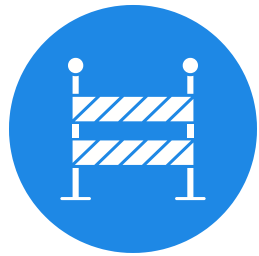
```
.
├── actions
│   └── companyActions.js
├── assets
│   └── react.svg
├── components
│   └── GeneralComponent.jsx
├── layouts
│   └── default
│       └── index.jsx
├── main.jsx
├── pages
│   ├── AnalysisPage
│   │   ├── index.jsx
│   │   └── sections
│   ├── HomePage
│   │   ├── index.jsx
│   │   ├── sections
│   │   │   ├── FirstComponent.jsx
│   │   │   └── SecondComponent.jsx
│   ├── MarketsPage
│   │   ├── index.jsx
│   │   └── sections
│   ├── NewsPage
│   │   ├── index.jsx
│   │   └── sections
│   ├── PortfolioPage
│   │   ├── index.jsx
│   │   └── sections
│   └── WatchlistPage
│       ├── index.jsx
│       └── sections
├── reducers
│   └── companySlice.js
├── store
│   └── index.js
├── utils
│   └── httpFetch.js
```

# First Prototype of Website



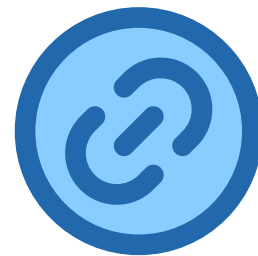
- General file structure established
- Navigation bar is developed
- Routing of main pages are done

# New Rules of The Project



## Opening an Issue

Open a new issue, give name to it and specify the issue.



## Binding Issue with New Branch

Open new branch from the issue, give name using format:

**<number\_of\_issue>\_<name\_of\_liable>**



## Tracking Tasks

Track the tasks from project tab

# Deneme #1

Edit

New issue

Open

Necro-U opened this issue 1 hour ago · 0 comments



Necro-U commented 1 hour ago



deneme



Necro-U self-assigned this 1 hour ago



Necro-U changed the title 1-Ubeydullah Deneme 1 hour ago

Assignees



Necro-U

Labels



None yet

Projects



None yet

Milestone



No milestone

Development



When branches are created from issues, their pull requests are automatically linked.

1-Ubeydullah  
Financial-Kit/Frontend

Notifications

Customize

Unsubscribe

You're receiving notifications because you authored the thread.



Add a comment

Write

Preview



Add your comment here...

Markdown is supported

Paste, drop, or click to add files

Close issue



Comment

Filter by keyword or by field

Backlog 1



This item hasn't been started

Frontend #1



Deneme

Ready 0



This is ready to be picked up

In progress 0



This is actively being worked on

In review 0

This item is in review





user story

back-end

Open

Ready

Front-end

QA

Bug

3-Steps

### OctoArcade Invaders

The PlanGame loop BacklogStandupNew view

Title	Team	Status
Prototype 3		
1 Game brief and go-no-go	Producers	Complete
2 Engine prototype (physics, rendering)	Engine	Complete
3 Initial concept art	Art	Complete
Beta 5		
4 Integrate with Leaderboard Service	Game Loop	Not Started
5 Creative design update to aliens for variety	Art	Planning
6 Updates to alien, beam, and cannon sprites	Art	Building
7 Update to collision logic	Engine	Building
8 Improve alien respawn rate	Game Loop	Behind
Launch 6		
9 Interviews with media outlets	Producers	Not Started
10 Save score across levels	Game Loop	Not Started

Now generally available

# Project planning for developers



## About Issues

Bug Tracking: Users can report bugs they encounter while using a project.

Discussion: Issues can also be used for general discussion, questions, or brainstorming sessions related to the project.

Organization: Issues can be labeled, assigned to specific contributors, and categorized using milestones, allowing for better organization and management of tasks within the project.

# Data Visualization

## HighCharts.js Progress

- Refactored the Graph Component.
  - Code is divided into separate and understandable functions. This separation will allow to handle dynamic data and dynamic user input regarding the indicators.
  - With current approach, it will be easier to add new indicators in a rapid way.
  - RSI and Volume bars could be viewed on any order on the bottom side of the main Graph. Each indicator graph will resize itself accordingly.
- This week's demo video link is below:  
<https://drive.google.com/file/d/1OqFPny0k82G9gkQrD8e7jTCkQsx3K3vl/view?usp=sharing>

# HighCharts.js - Demo



# Data Format

Data format remains the same compared to last week.

Candlestick Graph uses the following series format: [date, open, high, low, close] for each entry.

```
Candlestick:                                GraphHighCharts.js:
▼ Array(502) 1
  ▼ [0 ... 99]
    ► 0: (5) [1646663400000, 163.36, 165.02, 159.04, 159.3]
```

Volume Bar Graph uses the following series format: [date, volume] for each entry.

```
Volume Bar:
▼ Array(502) 1
  ▼ [0 ... 99]
    ► 0: (2) [1646663400000, 96418800]
```

- Data format structure could be changed based on Backend Team's design.
- date = unix timestamp

# Roadmap (until next week)

- Implementing the indicators that are displayed together with the main graph.
- Adding more indicators to the bottom side of the main graph like Volume Bar or RSI.
- Adding the Graph component to Frontend Team's individual components.
- Possibility to use mock data from the Backend Team.

## **This week,**

- We established our initial project repository and built the framework of the page.
- We experimented with various components and examined them.
- We defined the requirements for GitHub and established a working style.

## **What we are planning for the next week**

- Since we have finished distributing the pages, we will now begin working and committing individually.