Design pattern

The design pattern we chose for our stock prediction application is the MVC (Model-View-Controller) design pattern.

How we used each component:

- Model – The Model is where we defined our data and managed interactions with the database. For example, the StockItem class represents the data structure we use for predictions, storing all relevant information like issuer codes and stock prices.
- Controller – The Controller enables communication between the Model and the View. For example, in our PredictionController, we handle the logic for running technical analysis and passing the results to the View so that users can see them.
- View – The View displays the information that users see and interact with. For example, the Prediction page shows a dropdown list of issuer codes, allowing users to select a specific issuer and view the predictions.

Reasons why we chose to use this pattern:

- The code is clean and organized which makes maintainability easier.
- The MVC pattern is great for separation of concerns because it separates the application into components, each with their own responsibility.
- Each of the components is reusable.
- It helps with testing, since each of the components has its own logic and can be tested separately. Then we can test the interaction of the components as a whole system.