

Homework 3: Ajax, JSON, Responsive Design and Node.js

Stock Search

(AJAX/JSON/HTML5/Bootstrap/Angular /Node.js/Cloud Exercise)

Grading Guidelines (aka Rubric)

Total Points: 15

Deduction of points in each category will NOT exceed the total points of the category.

- **Navbar (0.50 points in total)**
 - The styling, highlighting and hover effect of the navbar should be similar to the reference video, **0.25 points** will be deducted otherwise.
 - The positioning of each element should be as shown in the reference video, **0.25 points** will be deducted otherwise.
- **Footer (0.25 point in total)**
 - Footer should be as mentioned in the description with a hyperlink and must stay at the bottom of the page, **0.15 points** will be deducted otherwise.
- **Autocomplete (1 point in total)**
 - The placeholder value and the auto suggesting values should be displayed similar to the reference video; **1 point** will be deducted if auto-complete doesn't work.
 - Poor performance of Autosuggest will result in a deduction of **0.5 points**.
 - While loading the suggestions, the loading spinner must appear, **0.25 points** will be deducted otherwise.
- **Routing and state management (1.25 point in total)**
 - If routes in the address bar are not as mentioned in the homework description, **0.5 points** will be deducted.
 - If the previous stock search results are not retained during navigation between routes or a new backend call is initiated for the retained ticker when navigating back to the search results page, **0.5 points** will be deducted
 - Clicking the clear icon (cross) inside the search bar should clear out the loaded stock's search results. **0.25 points**

- **Search Details Route (6.5 points in total)**

- Add/Remove stock to/from watchlist with the star color updating accordingly - **0.25 points**
- Error Alert must be displayed, if the wrong ticker/empty input is entered manually by the user in the search bar – **0.15 points**
- Overall Styling of Search Details Route must be the same as shown in video and screenshots, **0.15 points** will be deducted otherwise.
- Auto-fetching the details of the stock every 15 seconds, when the market is open and constant when closed – **0.30 points**
- Color of details according to the price change – **0.25 points**
- Adding /removing a stock from the watchlist and buying/selling stock from Search Details Route should display a pop-up message. Styling should be similar with auto-closing - **0.3 points**
- Stock logo should be displayed with proper styling - **0.1 points**
- **Trading Functionality (0.75 points)**
 - Disable Buy/Sell button inside Modal if the input is invalid along with an error message– **0.25 points.**
 - Dynamic Sell button if the stock already exists - **0.25 points**
 - Total must update based on the quantity entered by the user. – **0.1 points**
 - Modal design must be the same as shown in video and screenshots – **0.15 points**
- **Summary Tab (1 point):**
 - Display correct data values for corresponding fields – **0.25 points**
 - Color of the chart based on change value – **0.15 points**
 - Clicking on the links to company peers should navigate to search results of that ticker - **0.3 points**
 - Chart must be of the last working day with correct properties – **0.3 points.**
- **Top News Tab (1 points):**
 - Styling of news article cards should be similar to the reference video; **0.25 points** will be deducted otherwise.
 - Modal must open, containing more details as shown in video and screenshots, by clicking on the news card. – **0.25 points**
 - Clicking on more details must load the respective news article in the new tab – **0.2 points**
 - The sharing functionality should work as shown for both Facebook and Twitter buttons, **0.3 points** will be deducted otherwise.
- **Charts Tab (0.75 points):**
 - Type of chart and its properties must be the same as shown in video and screenshots – **0.5 points**

- **Insight Tab (1.25 points):**
 - MSPR Table as shown in video and screenshots – **0.5 points**
 - Type of the two charts and their properties must be the same as shown in video and screenshots – **0.75 points**
- **Watchlist Route (1 point in total)**
 - Clicking the stock card from the watchlist, if any, should redirect to the details route of the stock - **0.25 points**
 - Styling of the watchlist card should be similar to the reference video, **0.25 points** in total.
 - If watchlist is empty, display alert message – **0.25 points**
 - Live changes in My Watchlist after removing stock from watchlist by clicking remove button – **0.25 points.**
 - Watchlist is properly retrieved from MongoDB Atlas, when running on a different browser – **0.25 points.**
- **Portfolio Route (2 points in total)**
 - Live changes in My Portfolio list after performing Trading functions – **0.4 points.**
 - If portfolio is empty, display alert message – **0.1 points**
 - **Trading Functionalities (1.5 points)**
 - Display and Update Money correctly in the Wallet. – **0.5 points**
 - Disable Buy/Sell button if input is invalid. Form shouldn't get submitted. Appropriate error message– **0.25 points**
 - Total must update based on Quantity entered by the user. – **0.25 points**
 - Design and Appearance must resemble as shown in video and screenshots – **0.25 points**
 - Portfolio is properly retrieved from MongoDB Atlas, when running on a different browser – **0.25 points.**
- **Loading Screen (0.5 point in total)**
 - The styling and position of the loading spinner while loading search results should be as shown in the reference video, **0.25 points** will be deducted.
 - Loading spinner must be present while loading the suggestions in autocomplete, **0.25 points** will be deducted.
- **Responsive (2 points in total)**

- In the mobile browser, navbar, footer and all routes should be the same as the screenshots provided in the description and the reference video, **1 point** will be deducted otherwise.
- In the mobile browser, all functionalities must work exactly, **1 point** will be deducted otherwise.
- **Use of ng-bootstrap**
 - Ng-bootstrap is recommended. Implementations without it will not result in a penalty.
- **Use of highcharts-angular**
 - All the charts must be implemented using the highcharts-angular package. Implementing without it will result in a **2-point penalty**.
- **Use of Google Cloud/AWS/Azure**
 - You should implement Assignment 3 using AWS/Google Cloud/Azure cloud services as described in the **Cloud Setup (NodeJS)** available on D2L Brightspace. This will be verified with an additional link in the Table of Assignments, showing a sample API call to your “cloud” service and verifying the corresponding JSON result. This link would guide graders to a page where it displays the JSON data returned by your Node.js script running on the Google App Engine/AWS/Azure. Failing to provide a valid link will result in a **4-point penalty**.
- **Additional Requirements**
 - The entire assignment is to be implemented using Angular, Node.js, ng-bootstrap and highcharts-angular. Use of other technologies will result in a **5-point penalty**.
 - Using **React** instead of Angular is allowed, with no penalty, but the look and feel of the app needs to be the same. Otherwise a **1-point penalty** will be applied.
 - The use of jQuery is allowed, with no penalty.
 - All API requests should be made from the server-side, as outlined in the homework description. Any violation will result in a **4-point penalty**.
 - This application is a “1-page web app.” The browser window should not reload (i.e., “refresh”) for any kind of data request. This assignment is focused on building single-page applications. Refreshing the page will result in a **3-point penalty**.
 - The program must work in Chrome and Firefox on desktop, and Safari and Chrome on mobile devices.
 - Watchlist and Portfolio data should be stored and retrieved from MongoDB Atlas, on the same cloud service used for the backend.