Follow Up Questions

How Long did you spend on the code test?

I would estimate that I spent around 10 hours on the code test. This includes the time spent becoming familiar with the google charts API and the time spent actually finding the right library to use for generating the chart.

What went well?

I manged to get a chart to display which shows how a client's lump sum and monthly investments grow over time based on a given risk level/monthly interest rate. I didn't match the requirements entirely due to some confusion over the terms wide and narrow bounds. These aren't terms I have come across before and after spending some time researching the terms online, I decided it would be better to drop that area of functionality. This meant I could get something delivered within a reasonable time scale and allowed me to focus on demonstrating my skills as a developer.

Was there anything that was attempted but was not possible to get working in the time so is not visible in the code?

When it came to displaying the target value and end date on the chart, I wanted to set a marker line for this as detailed in the requirements. I struggled to figure out how to use the google chart API to do this within the given timescale. In the end I managed to get the target value/end date to display on the chart but as a single dot marker instead of a marked line. If I had more time, I may have used an overlay to achieve this functionally.

What would you do to improve it / continue development?

If I were to continue development, I would add server-side validation for the fields that are submitted to the controller. I would also add some handling for edge cases such as entering a target amount that is less than the initial investment. This would make for a better user experience.

Currently it is possible to force an exception to be thrown if the amounts you enter are too large. This is due to the max value for the C# decimal type being exceeded. Decimal should be used for financial systems where possible but using the "double" data type could prevent the exception from being thrown. This is because double has a greater maximum range than the decimal data type. Although you could potentially lose some decimal precision if you do this.

I would also ensure that the application is covered in unit tests. I added some tests to demonstrate that I can write unit tests and also to help ensure the server-side code was working as expected when performing the calculation. However more coverage is needed.