**PenChecks Coding Exercises**

**Part 1:**

Create a basic CRUD application that behaves as described below. Write the application in a manner that you would in your daily work.

**Foundation**

Build a REST API on the backend using .NET Core. Consume the API by a web frontend using a JavaScript library of your choice. Use any additional libraries on backend or frontend that support the outcome. Do not worry about authorization/authentication.

**Entities**

**Retirement Plan** - A retirement plan (ex. a 401k plan) offers a way to save for retirement. A retirement plan has a display name that is required. A retirement plan can have 0 or more participants.

**Participant** - A participant contributes to a retirement plan. A participant has a first name, last name and social security number that are all required and an email address that is optional.

**Functionality**

- Add the ability to Create, Edit and Delete a Plan

- Add the ability to add a participant to an existing plan

- Add the ability to delete a participant from a plan

**Rules**

- A plan can only be deleted when it doesn't have any participants

- A social security number must start with a number from 0-8 and must be exactly 9 digits long (11 with dashes)

- when an email address is entered it must be valid

**Persistence**

Write objects to a file in a format of your choice. At application start read objects from that file.

**Quality Assurance**

Add tests where you feel they are necessary or explain why they aren't.

**Documentation**

Briefly explain some of the design choices you have made.

**Timeframe**

Spend about 2 to 3 hours on the problem and submit the full code base in a zip file.

**Part 2:**

Create a UI component using the JavaScript framework of your choice.

Follow the specifications here for this fictional UI component:

<https://xd.adobe.com/view/d2c0b8b7-f64a-49f1-998e-1cdeec92b361-8793/specs/>

**Timeframe**

Spend about 1 hour on the problem and submit the full code base in a zip file.