Groundspeed Analytics Take Home Challenge

In this challenge we're interested in seeing your thought process for how you solve problems and how you develop in a professional setting. We'll be looking at your code for correctness, efficiency, ease of use and clarity. You can choose any language and framework you'd like, though we would recommend JavaScript, Node.js and React for the front end and Python for backend services. We highly recommend using tools you would expect to use in your day to day professional environment. We use GitHub, Docker and we test everything as part of our deliverables.

We'll review your code and provide any feedback we might have. You do not need to finish all aspects of this coding challenge. Feel free to focus only on areas of interest and expertise.

Deliverables

- Working code
- Short README on how to run the code

Problem statement

People are at the heart of health insurance policy. Personally identifiable information is often collected and stored during the procurement process along with aggregate and policy information. In this problem you will design a data structure to hold the information for consumers and some aggregate metrics for the policies. Please create an interface for managing the below data.

1. Design a schema for policyholders that
    - Has a unique id per policy holder
    - Contains the properties
        - Gender, date of birth, Social Security Number, smoking status, allergies, medical conditions
    - Contains insured events associated to unique policy holders
        - Health history details will have a date of incidence, type of issue, billed amount, covered amount
2. Design a data structure that
    - Contains aggregated metrics for all insured people
        - Total covered amount for all claims
        - Claims per year
        - Average age of insured
3. Write methods that
    - Adds an insured individual and returns the unique identifier of that individual
    - Adds an insurance event for a specific user identified by unique identifier
    - List all insured individuals
    - List all events associated with a specific user by unique identifier

4. Implement a simple user interface to call the methods in #3 and display the results
5. What considerations should you take into account when using or storing information such as Social Security Numbers?