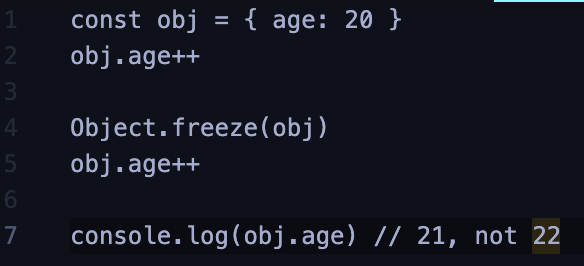
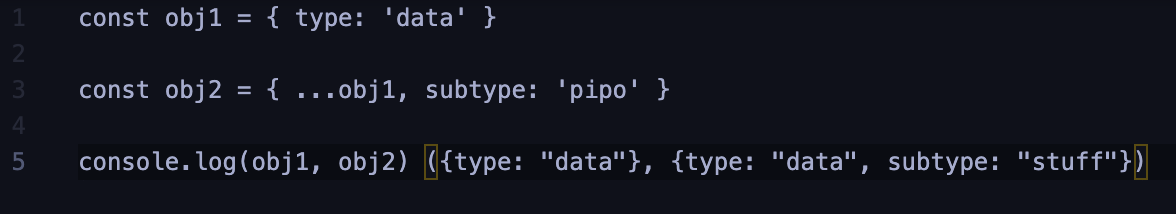
1. JWT  
     
   eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJzb21lb25lQGV4YW1wbGUubmV0IiwiYWRtaW4iOmZhbHNlLCJ2YWxpZF91bnRpbCI6IldlZCBEZWMgMzEgMjM6NTk6NTkgQ0VTVCAxOTY5In0.4bl2puoaRetNjO1GsweKOnnQsYgwNa9bQIC-WQZkuNo  
     
   {  
   "sub": "someone@example.net",  
   "admin": true,  
   "valid\_until": "Wed Dec 31 23:59:59 CEST 1969"  
   }  
     
   - An email address is a personal identifiable information (PII), it would be better to use a UUID in the sub.  
   - It is also not advisable to add admin role entry in a token.
2. Describe two attack vectors:
   1. Client-side: HTML messages received from client should always be sanitized, preventing cross-site scripting (DomSanitizer).
   2. Server-side: As the backend receives client input, apply scripting filter to what is valid and expected.
3. Mutable and immutable objects  
   A mutable object: If the value can change after it is created, the object is called mutable. An immutable object: If the value cannot change after it is created, the object is called immutable.   
   We have primitive types and reference types in JavaScript. Numbers, strings, boolean, null, and undefined are examples of primitive types.
   1. Example: let myName = 'Spider-man';
   2. Pros:  
      - Because variables stay the same, code is simpler to understand  
      - Its easier to test your code, since the state of immutable object do not change  
        
      Cons:  
      - Takes more effort to write immutable code
   3. The easiest way of achieving immutability is to use:
      1. the JavaScript method Object.freeze()   
         
      2. Spread method  
         
4. Speed up the loading of our current web-application  
   There are multiple ways of optimizing an angular web-accplication:
   1. Checking if the --prod is used to build the production build
   2. Is Ahead Of Time (AOT) compilation model is used:  
      There are 2 compilation models in Angular: AOT and JIT (Just In Time). JIT compiles the web-application at runtime, whereas AOT anticipated compilation at build time.
   3. Using lazy loading module:  
      Loading pages on demand, it is only loaded when the user navigates to the route.
   4. Using OnPush change detection:  
      OnPush only reacts to changes make in the @Input parameters.
5. In this case I would talk to the other team members. The flow of work should be the same, but if I had to make a decision I would go for option A.   
   A mediocre laptop will only slowdown the process of debugging etc.