Instructions

* Write a code to the registration page and the login page. The registration page contains a form for the user to enter full name(varchar), address(varchar), date of birth(date), and password(varchar).
* This page also contains a login button. On entering the details and clicking on the register button start the initial counter of the user id(int) from 1 and update it for every registration. This will be the unique user id that is unique to each user.
* Now create a SQL table with userId, fullName, address, dob and ethereumId that stores alphanumeric characters of length 42 as columns. On clicking the register button, auto generates a random but unique 42 character alphanumeric ethereumId and. Show this ethereumId and the userId as a pop up to the user and store all these details in the user table. Now, when the user clicks on the login button, redirect him to the login page.
* The login page contains two parts. One side contains login to an age gated service and the other one is a non age gated service. Both of them take userId and the ethereumId from the user to login.
* For the age gated service, when the user tries to login, calculate age from the date of birth provided while registering and if the user is above 21, and the given ethereumId and userId matches with values in the database, log him in and give a pop up saying login is successful.
* If the age is not above 21, give a pop up saying you need to be above 21 to be able to use this service.
* If the userId and the ethereumId does not match with values in the database, show a pop up saying check your userId or ethereumId and try to login again.
* For the whole verification process of the age and userId and the ethereumId write a smart contract. Also include some logic in the same smart contract for the non age gated service login.
* This takes userId, ethereumId from the user, checks if they match with userId and ethereumId from the table in the database and if they match say that the login is successful else show a pop up saying check your userId or ethereumId and try to login again.
* Now, write a code to the registration.html, registration.js, login.html,login.js, smart contract and the migration file to the smart contract. Consider writing this code in Visual Studio Code with an inbuilt MySQL database