CSE446: Project Part 2

Submission Deadline: 23rd April, 2025

Missing Persons Management System

In the first part of the project, you developed the smart contract for managing records of missing individuals and facilitating their recovery. Now, your task is to build a complete decentralized application (DApp) to allow users to interact with your smart contract via an intuitive interface.

There will be three types of users in the system: **1. Admin**, **2. Reporter** (Someone who reports a missing person), **3. Investigator** (Assigned by admin to track a case).

- **User Registration**
  There should be a registration form for Admins, Reporters, and Investigators to register with using the smart contract.

- **Add Missing Person**
  There should be a form through which only **reporters** can add missing person information using the smart contract function of *adding a missing person*. Make sure the entries get a unique serial number. As there is age data, your system should assign the urgency level based on that to every instance of a missing person report.
  Like: High = age**<18**, **Medium** = age**>50**, and **Low** is in between these ages.

- **Update Missing Person Status**
  In part 1, you have developed an update function. Now, the interface should have a form to update missing person information using that smart contract function. Remember, only Admin should be able to see and use this. Using the function, the admin can update the status of the smart contract. If the Admin enters any wrong value other than the data mentioned *in Project Part-1*, the data will not get updated, and appropriate messages should be shown.

- **Assign Investigator**
  Admins can assign an investigator to any missing person case. You can assume Admin first looks at the address of the investigator and then assigns the missing task based on that, but, no need to add any constraint to that.

- **Search Missing Person**
  Users can search for missing persons, and it should show the count of missing persons area-wise in a sorted list.

- **Book an Investigation Slot**
  There should be a form through which only **Reporters** can book an **Investigator**'s appointment. Payment should be deducted from the reporter's metamask account. One

of the few admins will receive it. You can put an option in the form where reporters can select any available admin to transfer the currency. Reporters should be able to see available booking slots. After a successful transaction, there should be a “Success” message in the interface. This should be automated, meaning you cannot manually refresh the interface/browser to show the “success” message.

- **View Appointment Schedule**

  There should be a separate section where all the users of the system should be able to see the appointment schedule of every investigator.

  Bonus: *Investigators can let the Admin know if a missing person is found so that the Admin can update the Missing Person Status. Implement it in your own way.*

NOTE:

- You cannot use any database, e.g., MongoDB, MySQL, etc. All the data must be stored via a smart contract.

- You must use Ganache and Truffle for the development.

- For the frontend, you can use JavaScript like we did in the lab. If you already have experience in frameworks/libraries such as ReactJS, AngularJS, and VueJS, you can use them. If you want, you can take help from the lab files also. But remember, all the data must be stored and updated via a smart contract.

- Focus on the features, don’t waste time on designing a good-looking user interface. There are no additional marks for the design part.