Take-Home Assignment - Web2

Hi, and welcome to the web2 take-home assignment. Your task is to implement a dashboard for our triaging team with the list of all the submissions reported to Immunefi.

First, a quick walkthrough of our system:

- A report is a collection of information related to a security vulnerability
- A report is created by a whitehat and sent to one of our customers
- All reports are identified uniquely by a numeric ID.
- Immunefi classifies reports into five severity levels:
    - Critical
    - High
    - Medium
    - Low
    - None
- Immunefi receives reports for vulnerabilities in the following categories:
    - Blockchain/DLT
    - Smart Contrat
    - Website and Applications
- A report can be in one of the following statuses:
    - Reported
    - Escalated
    - Confirmed
    - Paid
    - Closed
- Immunefi hosts the bug bounty programs of the following customers:
    - Wallet Wonders
    - Encrypted Exchange
    - Hash Hacienda
    - Public Key Party
    - Satoshi Boulevard
    - Key Keepers
    - Big-Endians
    - Small-Endians
- Immunefi tracks the following data from our white hats:
    - Email
    - Username

All the data mentioned above is stored in a PostgreSQL database.

Hint: The database schema was initially designed by a fictional junior DBA who has moved on to new adventures. So, it might be a good idea for you to check that the database schema is in

good shape. Please make any change you think will help the company and make sure not to lose any of our precious data.

For this code assignment, we need you to code a dashboard that will list all the reports submitted to Immunefi.

Our triaging team has requested some filters they consider helpful to review the data:

- Status
- Severity
- ID
- Hacker username or email address
- Report type
- Project

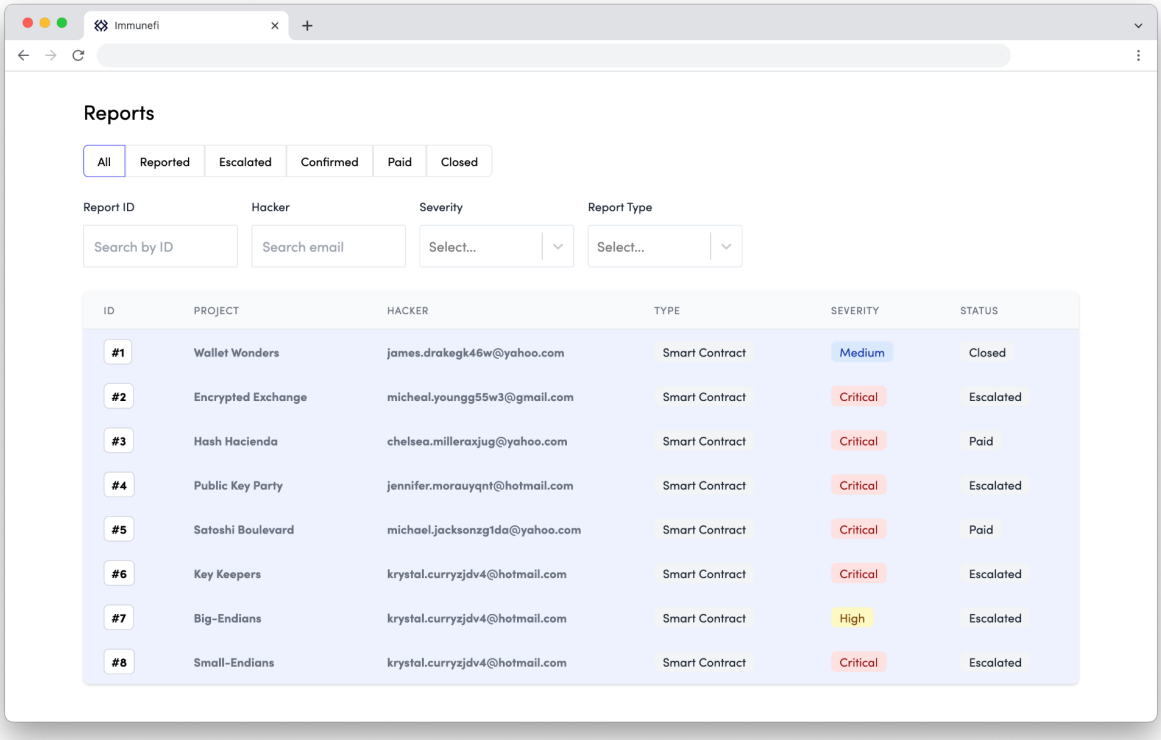
Soon, you will receive access to our code base, a Next.js web application. Feel free to install any package that will make your task easier. However, we love querying the database through Prisma, so do not replace this package.

Our security department has requested to fetch the report's data exclusively using the `GET /api/reports` API endpoint because they are continuously monitoring it; You will need to secure this API endpoint with an API key.

Finally, find attached the expected design for the report list. Our fictional designer was distracted while working on the dashboard designs and forgot to add the report's submission date to the table and the project filter. Please include them in the final version.

Best of luck,

DESIGN:


Immunefi
Reports
All
Reported
Escalated
Confirmed
Paid
Closed
Report ID
Search by ID
Hacker
Search email
Severity
Select...
Report Type
Select...
ID
PROJECT
HACKER
TYPE
SEVERITY
STATUS
#1
Wallet Wonders
james.drakegk46w@yahoo.com
Smart Contract
Medium
Closed
#2
Encrypted Exchange
micheal.youngg55w3@gmail.com
Smart Contract
Critical
Escalated
#3
Hash Hacienda
chelsea.milleraxjug@yahoo.com
Smart Contract
Critical
Paid
#4
Public Key Party
jennifer.morauyqnt@hotmail.com
Smart Contract
Critical
Escalated
#5
Satoshi Boulevard
michael.jacksonzg1da@yahoo.com
Smart Contract
Critical
Paid
#6
Key Keepers
krystal.curryzjdv4@hotmail.com
Smart Contract
Critical
Escalated
#7
Big-Endians
krystal.curryzjdv4@hotmail.com
Smart Contract
High
Escalated
#8
Small-Endians
krystal.curryzjdv4@hotmail.com
Smart Contract
Critical
Escalated