

SECURE THE VAULT

YOU ARE THE DEFENDER



DETECT

Detect the threat and understand the security concept



RESPOND

Respond by choosing the correct answer to handle the situation



MITIGATE

Mitigate the risk and move forward when you answer correctly



BREACH

Breach occurs when you answer incorrectly, allowing the hacker to advance

SAFEGUARD

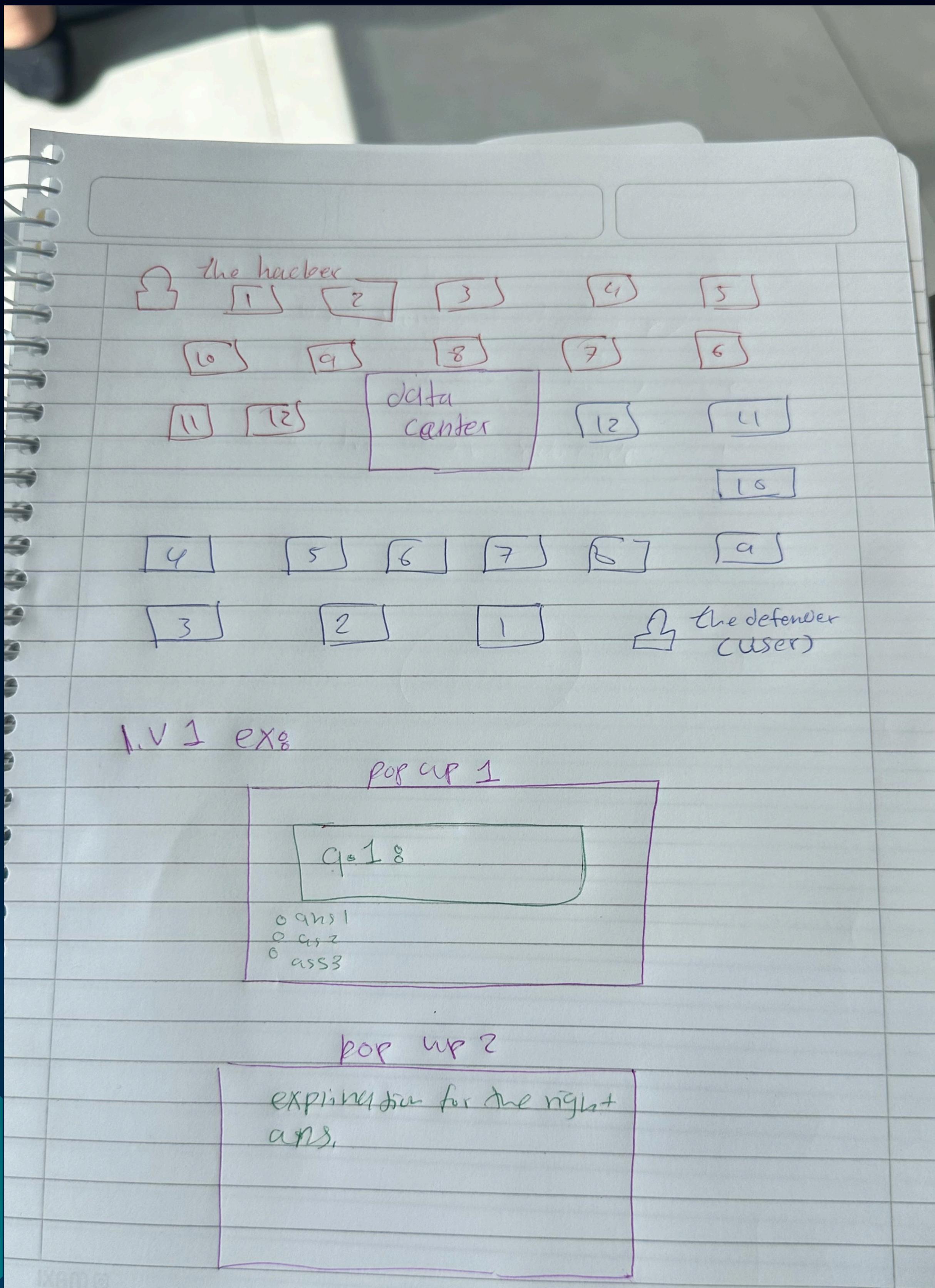
Safeguard the data by reinforcing your defenses and keeping the vault fully protected



USER STORIES

- As a player, I want a start screen that explains the game's concept, so that I understand the goal before playing.
- As a player, I want to see a progress track showing both me Defender and the Hacker, so that I can visually see who is closer to the company's data vault.
- As a player, I want a short explanation before each question, so that I can learn the security concept before answering.
- As a player, I want to respond using either multiple-choice options or a text input field, so that each question can be answered in the most suitable and interactive way.
- As a player, I want my Defender to move one step forward when I answer correctly, so that I feel rewarded for correct answers.
- As a player, I want the Hacker to move one step forward when I answer incorrectly, so that I understand the consequences of wrong answers.
- As a player, I want to see a clear winning message (Data Protected) when I reach the data vault first, so that I know I successfully defended the company.
- As a player, I want to see a clear losing message (Data Breached) when the Hacker reaches the vault first, so that I understand the impact of my mistakes.
- As a player, I want immediate feedback after each question, so that I can learn the correct answer and improve.
- As a player, I want a "Play Again" button at the end of the game, so that I can retry and improve my score.

MOCKUP



@MAXI

PSEUDOCODE

```
// 1) Define the required variables used to track the state of the game.  
  
// 2) Store cached element references.  
  
// 3) Upon loading, the game state should be initialized, and a function should  
//    be called to render this game state.  
  
// 4) The state of the game should be rendered to the user.  
  
// 5) Handle the game over logic.  
  
// 6) Handle each instance of a player clicking a button with the use of a  
//    `handleClick()` function.  
  
// 7) Create reset functionality.
```