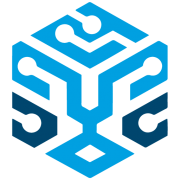
****

**BÁO CÁO THỰC TẬP SINH FRONTEND REACTJS**

**Công ty: DEEP TECH ASIA CORPORATION**

**Ứng viên thực hiện: LÊ QUỐC THỊNH**

**Ngày 13 tháng 10 năm 2022**



Mục lục

[1. Phân tích business của game (Break down các chức năng cần làm) 1](#_Toc116623984)

[1.1 Call api 1](#_Toc116623985)

[1.2 Xáo bài 1](#_Toc116623986)

[1.3 Chia bài 1](#_Toc116623987)

[1.4 Tính điểm 2](#_Toc116623988)

[1.5 Tìm điểm lớn nhất 3](#_Toc116623989)

[1.6 Tính coins của mỗi người chơi 3](#_Toc116623990)

[1.7 Reset bàn chơi 3](#_Toc116623991)

[1.8 Hiển thị các lá bài 4](#_Toc116623992)

[1.9 Úp bài lúc chia 4](#_Toc116623993)

[1.10 Ngửa bài lúc xét điểm 4](#_Toc116623994)

[1.11 Hiểm thị thông tin người chơi 4](#_Toc116623995)

[1.12 Xử lý trường hợp người chơi hết coins 4](#_Toc116623996)

[1.13 Popup thông báo 5](#_Toc116623997)

[1.14 Tìm người chiến thắng 5](#_Toc116623998)

[1.15 Xáo bài khi bộ bài đã hết 5](#_Toc116623999)

[2. Làm Basic UI cho game 6](#_Toc116624000)

[1.16 Màn hình bắt đầu 6](#_Toc116624001)

[1.17 Khi thực hiện chia bài 6](#_Toc116624002)

[1.18 Khi thực hiện kiểm bài 7](#_Toc116624003)

[1.19 Popup thông báo 7](#_Toc116624004)

## Phân tích business của game (Break down các chức năng cần làm)

### Call api

|  |  |
| --- | --- |
| Yêu cầu | * Lấy dữ liệu từ API cho sẵn |
| Cách thực hiện | * Chia thành component call api riêng biệt để tái sử dụng * Sử dụng async await cho từng phương thức gọi api * Sử dụng axios |

### Xáo bài

|  |  |
| --- | --- |
| Yêu cầu | * Call api lấy dữ liệu đã được xáo trộn của một bộ bài bất kì * Xáo bài sẽ không làm ảnh hưởng đến điểm đã đạt được và coins của người chơi |
| Cách thực hiện | * Chỉ lấy một bộ bài 52 lá (param deck\_count = 1) * Khi xáo bài sẽ cần reset lại lá bài của mỗi người (cards = []) * Set state của deck\_id và remaining cards |

### Chia bài

|  |  |  |
| --- | --- | --- |
| Yêu cầu | * Chia bài cho từng người theo thứ tự từ phải sang trái sao cho mình là người được chia cuối cùng trong mỗi lần chia * Các lá bài sẽ được úp xuống * Chia mỗi người ba lá | |
| Cách thực hiện | * Chạy một vòng lặp đầu tiên với số lần lặp là 3 (Mỗi người 3 lá bài) * Bên trong tiếp tục sử dụng vòng lặp với số lần lặp là số người chơi hiện có (lúc bắt đầu là 4) * Gọi api random từng lá bài trong một bộ bài (param count = 1 vì chỉ 1 lá cho mỗi người trong một lần chia) * Áp dụng chia lấy phần dư * Push lá bài random từ api vào mảng card của từng người chơi (theo thứ tự từ phải qua trái, mỗi người 1 lá trong 1 lần chia) | |
|  | | |
| Các lỗi mắc phải | | Cách xử lý |
| * Lúc chia bài trong tay người chơi vẫn còn 3 lá bài của ván trước đó | | * Sử dụng vòng lặp và phép chia lấy dư để thực hiện update lại số lá bài trong tay của mỗi người thành rỗng |
| * Số lá bài trong bộ bài không đủ để chia cho những người ngồi trong bàn | | * Kiểm tra điều kiện tính toán lấy phần dư (số lá bài cuối cùng còn dư lại) nếu số lá bài còn dư lại nhỏ hơn tổng số lá cần để chia cho tất cả người chơi thì thông báo và chọn một bộ bài xáo trộn khác |
| * Lúc chọn bộ bài xáo trộn khác trong tay mỗi người vẫn còn 3 lá bài | | * Sử dụng useState, rest, destructuring để update lại bộ bài trong tay của mỗi người thành rỗng |

### Tính điểm

|  |  |  |
| --- | --- | --- |
| Yêu cầu | * Điểm của người chơi trong mỗi ván là số lẻ của tổng điểm ba lá bài. * Số điểm tương ứng con số. * Các là bài: J, Q, K thì mỗi lá được 10 điểm. Lá A = 1 điểm | |
| Cách thực hiện | * Sử dụng reduce để tính tổng điểm của các lá bài * Sử dụng phương thức Number.isNaN để kiểm tra giá trị (giá trị hiện tại đang là số hay chữ vì A = 1, J, Q, K = 10), * Nếu giá trị là NaN và là "ACE" thì cộng 1 * Ngược lại nếu là NaN mà khác "ACE" thì cộng 10 * Ngược lại cộng với chính giá trị đó. * Sử dụng useState, rest, destructuring để gán điểm vào cho từng người chơi | |
|  | | |
| Các lỗi mắc phải | | Cách xử lý |
| * API trả về dạng String | | * Sử dụng parseInt để chuyển String thành số |

### Tìm điểm lớn nhất

|  |  |
| --- | --- |
| Yêu cầu | * Kiểm tra điểm lẻ lớn nhất trong số điểm của các người chơi |
| Cách thực hiện | * Tạo ra một biến tạm maxValue với giá trị đầu tiên là 0. * Trong lúc tính tổng sẽ đồng thời kiểm tra maxValue với số điểm tính được nếu số điểm lớn hơn maxValue thì gán maxValue = điểm. * Phần điểm và maxValue sẽ được chia lấy dư cho 10 để so sánh phần số lẻ |

### Tính coins của mỗi người chơi

|  |  |
| --- | --- |
| Yêu cầu | * Mỗi người bắt đầu với 5000 coins * Bị trừ 900 coins cho mỗi lần thua |
| Cách thực hiện | * Khi ván bài chưa kết thúc (chưa có người thắng cuộc cuối cùng) sử dụng useState, rest, destructuring để trừ 900 coins của mỗi người * Nếu số điểm của người đó chia lấy dư cho 10 khác với giá trị maxValue đã tính được chia lấy dư cho 10 thì coins sẽ bị trừ 900 (Chia lấy dư cho 10 bởi vì chỉ xét số lẻ VD: 19 và 29 hai người đều thắng cuộc) |

### Reset bàn chơi

|  |  |  |
| --- | --- | --- |
| Yêu cầu | * Reset tất cả giá trị trong bàn chơi về giá trị ban đầu | |
| Cách thực hiện | * Sử dụng useState hook để set mỗi state về trạng thái ban đầu * Lựa chọn một bộ bài xáo trộn mới bằng cách gọi api * Set lại state của deck\_id và remaining cards | |
|  | | |
| Các lỗi mắc phải | | Cách xử lý |
| * Sau khi tìm được người chơi cuối cùng là reset bàn chơi thì không thể cập nhật điểm của mỗi người | | * Set lại state finish là false |

### Hiển thị các lá bài

|  |  |
| --- | --- |
| Yêu cầu | * Hiển thị các lá bài lên màn hình cho người chơi |
| Cách thực hiện | * Chia lá bài thành một component riêng sau đó dùng vòng lặp đổ dữ liệu vào |

### Úp bài lúc chia

|  |  |
| --- | --- |
| Yêu cầu | * Úp bài lúc chia (Không để người chơi thấy giá trị của lá bài) |
| Cách thực hiện | * Khi người dùng nhấn vào nút chia bài - state checkCard sẽ được set là false * Khi render sẽ căn cứ vào state này để úp bài và lật bài. * Khi state này là false thì mặt lưng của lá bài sẽ được render * Ngược lại mặt giá trị của lá bài sẽ được render |

### Ngửa bài lúc xét điểm

|  |  |
| --- | --- |
| Yêu cầu | * Cho người chơi thấy được giá trị của 3 lá bài khi nhấn vào nút xét điểm |
| Cách thực hiện | * Dựa vào state checkCard đã được thực hiện ở lúc chia bài state checkCard lúc này sẽ là true - Các lá bài sẽ được render mặt giá trị |

### Hiểm thị thông tin người chơi

|  |  |
| --- | --- |
| Yêu cầu | * Thực hiện hiển thị thông tin của người chơi lên màn hình như: Tên, ID, Điểm, Coins |
| Cách thực hiện | * Sử dụng map để render giá trị trong list đã được set sẵn |

### Xử lý trường hợp người chơi hết coins

|  |  |
| --- | --- |
| Yêu cầu | * Không cho người chơi thực hiện hành động (tham gia ván chơi) khi đã hết coins hoặc số coins < 900 |
| Cách thực hiện | * Sử dụng filter để loại bỏ người chơi có số coins < 900 trong danh sách người chơi * Xử lý khi thực hiện chia bài |

### Popup thông báo

|  |  |
| --- | --- |
| Yêu cầu | * Hiển thị thông báo đến người dùng |
| Cách thực hiện | * Được chia ra thành một component riêng để tái sử dụng * Sử dụng phương pháp truyền dữ liệu từ cha sang con để truyền các tham số cũng như các giá trị |

### Tìm người chiến thắng

|  |  |  |
| --- | --- | --- |
| Yêu cầu | * Tìm ra người chiến thắng trong ván chơi | |
| Cách thực hiện | * Trong lúc tính toán điểm lớn nhất nếu điểm của người chơi lớn hơn maxValue thì setWinner là người chơi đó | |
| Các lỗi mắc phải | | Cách xử lý |
| * Trường hợp có nhiều hơn một người có số điểm bằng maxValue | | * Kiểm tra nếu số điểm người chơi bằng maxValue và không tồn tại trong state winner thì dùng destructuring để thực hiện thêm người chơi vào state winner * Ngược lại thì update |

### Xáo bài khi bộ bài đã hết

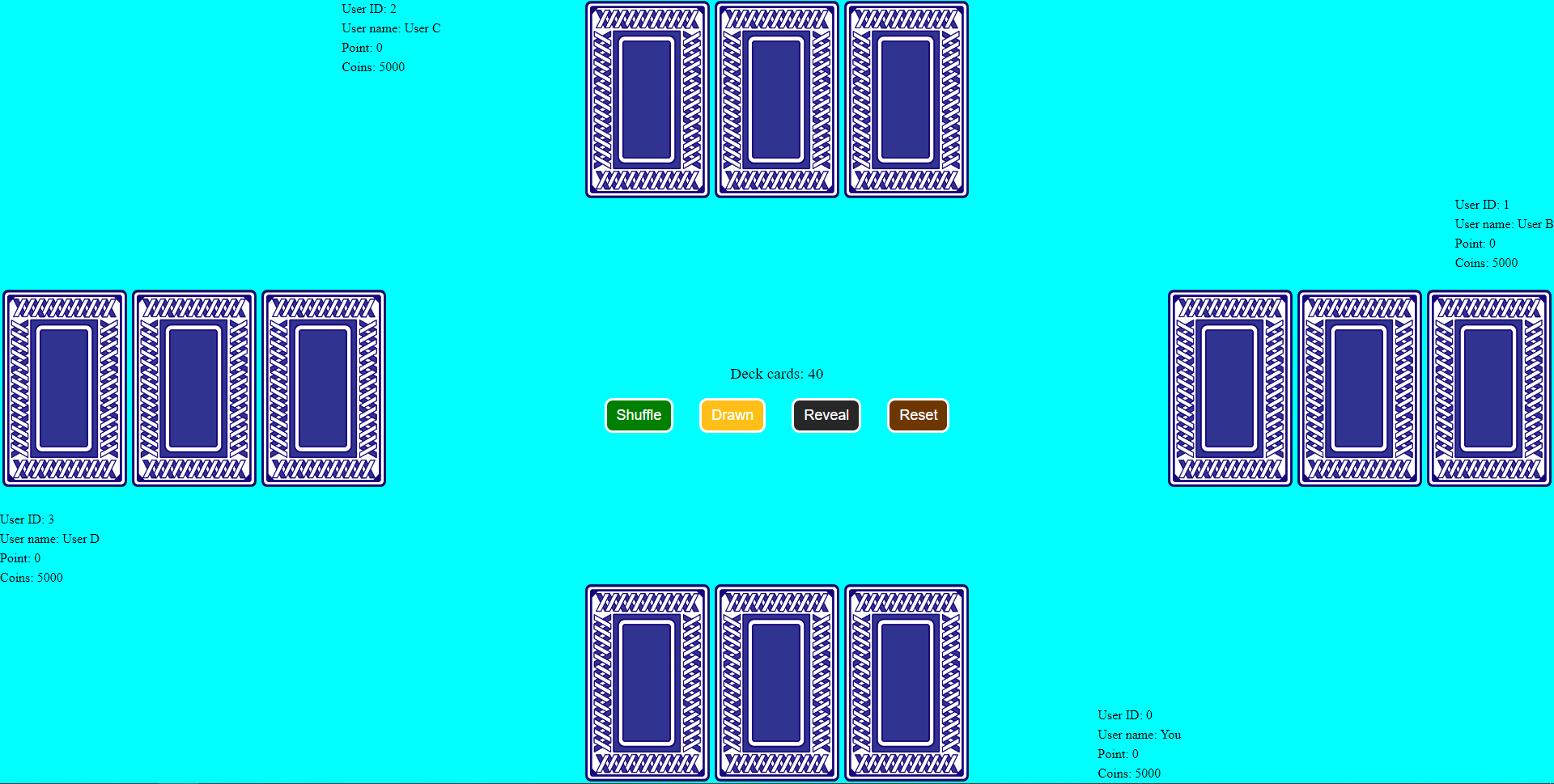
|  |  |
| --- | --- |
| Yêu cầu | * Xáo bài khi bộ bài đã hết |
| Cách thực hiện | * Bắt sự kiện button close kiểm tra remaining cards nếu remaining cards nhỏ hơn số dư của bộ bài (52 lá) chia cho tổng số người chơi trên bàn (Mỗi người 3 lá) thì thực hiện xáo bài |

## Làm Basic UI cho game

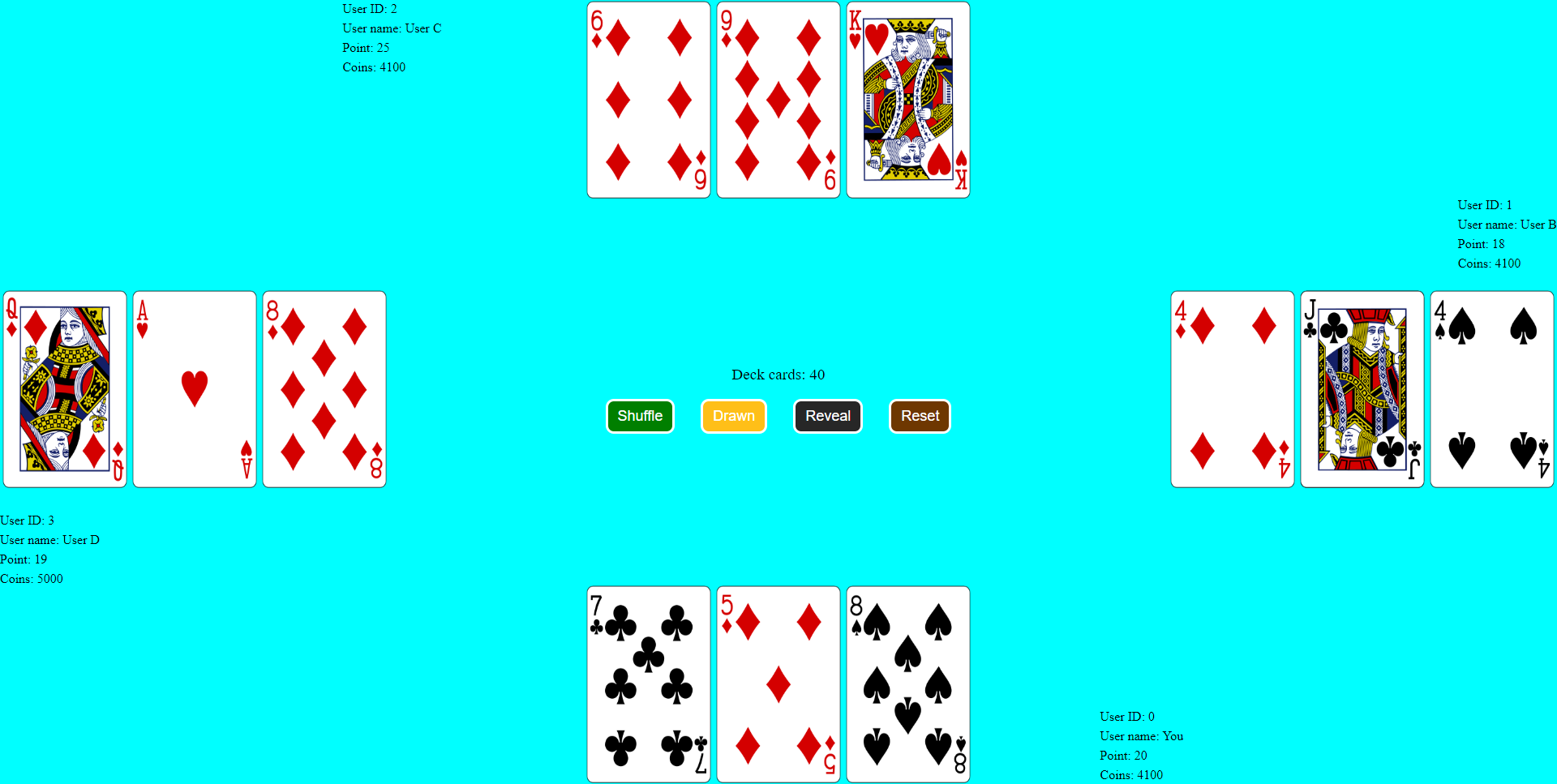
### Màn hình bắt đầu



### Khi thực hiện chia bài



### Khi thực hiện kiểm bài



### Popup thông báo

