

createroom.test.js

Function: `players(numberOfPlayers)`

Test Case 1: Sets the number of players in `sessionStorage` and redirects to `/game-mode`.

Purpose: Ensure that the correct number of players is stored and the redirection occurs as expected.

Expected Outcome:

`sessionStorage` should store `cryptid-num-players` with the correct number of players.

`window.location.href` should be `/game-mode`.

Code:

```
test('sets the number of players in sessionStorage and redirects to /game-mode', () => {  
  const numberOfPlayers = 4;  
  players(numberOfPlayers);  
  expect(sessionStorage.getItem('cryptid-num-players')).toBe(numberOfPlayers.toString());  
  expect(window.location.href).toBe('/game-mode');  
});
```

navigation.test.js

Function: `switch_page(mode)`

Test Case 1: Sets game mode to intro and redirects to `/game` if `cryptid-num-players` is set.

Purpose: Ensure that the game mode is set to intro and redirection to `/game` occurs.

Expected Outcome:

`sessionStorage` should store `cryptid-game-mode` as intro.

`window.location.href` should be `/game`.

Code:

```
test('sets game mode to intro and redirects to /game if cryptid-num-players is set', () => {  
  sessionStorage.setItem('cryptid-num-players', '4');  
  switch_page(0);  
});
```

```
    expect(sessionStorage.getItem('cryptid-game-mode')).toBe('intro');  
    expect(window.location.href).toBe('/game');  
  });
```

Test Case 2: Sets game mode to intro and redirects to /waiting if cryptid-num-players is not set.

Purpose: Ensure that the game mode is set to intro and redirection to /waiting occurs.

Expected Outcome:

sessionStorage should store cryptid-game-mode as intro.

window.location.href should be /waiting.

```
test('sets game mode to intro and redirects to /waiting if cryptid-num-players is not set', () => {  
  switch_page(0);  
  expect(sessionStorage.getItem('cryptid-game-mode')).toBe('intro');  
  expect(window.location.href).toBe('/waiting');  
});
```

Test Case 3: Sets game mode to normal and redirects to /game if cryptid-num-players is set.

Purpose: Ensure that the game mode is set to normal and redirection to /game occurs.

Expected Outcome:

sessionStorage should store cryptid-game-mode as normal.

window.location.href should be /game.

```
test('sets game mode to normal and redirects to /game if cryptid-num-players is set', () => {  
  sessionStorage.setItem('cryptid-num-players', '4');  
  switch_page(1);  
  expect(sessionStorage.getItem('cryptid-game-mode')).toBe('normal');  
  expect(window.location.href).toBe('/game');  
});
```

Test Case 4: Sets game mode to normal and redirects to /waiting if cryptid-num-players is not set.

Purpose: Ensure that the game mode is set to normal and redirection to /waiting occurs.

Expected Outcome:

sessionStorage should store cryptid-game-mode as normal.

window.location.href should be /waiting.

```
test('sets game mode to normal and redirects to /waiting if cryptid-num-players is not set', () => {  
  switch_page(1);  
  expect(sessionStorage.getItem('cryptid-game-mode')).toBe('normal');  
  expect(window.location.href).toBe('/waiting');  
});
```

game.test.js

Functions: create(), join(), browse()

Test Case 1 (create): Sets session storage and redirects.

Purpose: Ensure that the session is set up for creating a game and redirection occurs.

Expected Outcome:

sessionStorage should store cryptid-game-action as create.

sessionStorage should remove cryptid-num-players.

window.location.href should be /game-mode.

Code:

```
test('create function sets session storage and redirects', () => {  
  sessionStorage.setItem('cryptid-num-players', '4');  
  create();  
  expect(sessionStorage.getItem('cryptid-game-action')).toBe('create');  
  expect(sessionStorage.getItem('cryptid-num-players')).toBe(null);  
  expect(window.location.href).toBe('/game-mode');  
});
```

Test Case 2 (join): Sets session storage and redirects.

Purpose: Ensure that the session is set up for joining a game and redirection occurs.

Expected Outcome:

sessionStorage should store cryptid-game-action as join.

window.location.href should be /join-room.

```
test('join function sets session storage and redirects', () => {
```

```
join();  
  
expect(sessionStorage.getItem('cryptid-game-action')).toBe('join');  
  
expect(window.location.href).toBe('/join-room');  
  
});
```

Test Case 3 (browse): Logs a message and shows an alert.

Purpose: Ensure that the appropriate log message and alert are shown.

Expected Outcome:

console.log should log "coming soon".

An alert should show "Coming soon...".

Code:

```
test('browse function logs message and shows alert', () => {  
  
  browse();  
  
  expect(console.log).toHaveBeenCalledWith('coming soon');  
  
  expect(global.alert).toHaveBeenCalledWith('Coming soon...');  
  
});
```

getBoards.test.js

Function: getAllJsons(directory)

Test Case 1: Fetches directory contents successfully.

Purpose: Ensure that JSON files are fetched successfully from the specified directory.

Expected Outcome:

The result should be an array with the fetched files.

fetch should be called with the correct directory.

Code:

```
test('should fetch directory contents successfully', async () => {  
  
  global.fetch = jest.fn(() =>  
  
    Promise.resolve({  
  
      ok: true,
```

```

        status: 200,
        json: () => [{ file: 'example.json' }]
    })
};

const result = await getAllJsons('pressed');
expect(result).toEqual([{ file: 'example.json' }]);
expect(fetch).toHaveBeenCalledWith('/maps/pressed');
});

```

Test Case 2: Handles failed fetch.

Purpose: Ensure that errors are handled gracefully when the fetch operation fails.

Expected Outcome:

The result should be an empty array.

An error message should be logged to the console.

```

test('should handle failed fetch', async () => {
    global.fetch = jest.fn(() => Promise.reject(new Error('Network error')));
    const result = await getAllJsons('pressed');
    expect(result).toEqual([]);
    expect(console.error).toHaveBeenCalled();
});

```

home.test.js

Functions: online_play(), local_play(), custom()

Test Case 1 (online_play): Sets session storage and redirects to /game-mode.

Purpose: Ensure that the session is set up for online play and redirection occurs.

Expected Outcome:

sessionStorage should store cryptid-game-action as play.

sessionStorage should remove cryptid-num-players.

window.location.href should be /game-mode.

Code:

```
test('online_play function sets sessionStorage and redirects to /game-mode', () => {  
  sessionStorage.setItem("cryptid-num-players", "3");  
  online_play();  
  expect(sessionStorage.setItem).toHaveBeenCalledWith("cryptid-game-action", "play");  
  expect(sessionStorage.getItem("cryptid-num-players")).toBe(null);  
  expect(window.location.href).toBe("/game-mode");  
});
```

Test Case 2 (local_play): Sets session storage and redirects to /create-room.

Purpose: Ensure that the session is set up for local play and redirection occurs.

Expected Outcome:

sessionStorage should store cryptid-game-action as local.

window.location.href should be /create-room.

javascript

Copy code

```
test('local_play function sets sessionStorage and redirects to /create-room', () => {  
  local_play();  
  expect(sessionStorage.setItem).toHaveBeenCalledWith("cryptid-game-action", "local");  
  expect(window.location.href).toBe("/create-room");  
});
```

Test Case 3 (custom): Redirects to /custom-room.

Purpose: Ensure that redirection to the custom room creation page occurs.

Expected Outcome:

window.location.href should be /custom-room.

```
test('custom function redirects to /custom-room', () => {  
  custom();  
  expect(window.location.href).toBe("/custom-room");  
});
```

index.test.js

Function: login()

Test Case: Redirects to /users.

Purpose: Ensure that redirection to the users page occurs after login.

Expected Outcome:

window.location.href should be /users.

Code:

```
describe('login function', () => {  
  test('login function redirects to /users', () => {  
    login();  
    expect(window.location.href).toBe('/users');  
  });  
});
```

join_room.test.js

Function: join()

Test Case: Sets session storage and redirects to /waiting.

Purpose: Ensure that the necessary session variables are set and redirection occurs.

Expected Outcome:

sessionStorage should store cryptid-game-action as join.

sessionStorage should store cryptid-game-room-number with the room number from the input field.

window.location.href should be /waiting.

Code:

```
describe('join function', () => {
```

```
beforeEach(() => {  
  global.document.getElementById = jest.fn((id) => ({  
    value: "123"  
  }));  
});
```

```
delete global.window.location;  
global.window.location = {  
  href: "  
};  
});
```

```
test('join function sets sessionStorage and redirects to /waiting', () => {  
  join();  
  expect(sessionStorage.setItem).toHaveBeenCalledWith("cryptid-game-action", "join");  
  expect(sessionStorage.setItem).toHaveBeenCalledWith("cryptid-game-room-number", "123");  
  expect(window.location.href).toBe("/waiting");  
});  
});
```

These test cases verify that the functions in each module behave as expected, ensuring proper session management, redirection, data fetching, and user notifications.