Exercise: Fetching User Data with Error Handling

Objective: Create a function that fetches user data from a public API. Use `async/await` for handling asynchronous operations and a `try...catch` block to handle potential errors.

Instructions:

1. Write an asynchronous function named `fetchUserData` that takes a `userId` as a parameter.
2. Inside the function, use `fetch` to get user data from the API https://jsonplaceholder.typicode.com/users/{userId} where `{userId}` is replaced with the value of the `userId` parameter.
3. Use `await` to wait for the response and then convert it to JSON.
4. If the user is not found (e.g., the API responds with a `404` status), throw an error with a message like `"User not found"`.
5. If the fetch operation fails for any reason, the `catch` block should log an error message to the console.
6. Log the user's name to the console if the data is successfully retrieved.

Hint: You can simulate errors by using an invalid user ID (e.g., a very high number) or by disconnecting from the internet.

Example Usage:

```
fetchUserData(1);  // Should fetch and log the name of the user with ID 1
fetchUserData(999); // Should log "User not found" error in the console
```

Expected Output

Here's what students should expect to see based on different scenarios:

Example 1: Valid User ID

```
fetchUserData(1);
```

Console Output:

```
User Name: Leanne Graham
```

Example 2: Invalid User ID

```
fetchUserData(999);
```

Console Output:

```
User not found
```

Example 3: Network Error

If the network is disconnected or the API is unreachable:

Console Output:

```
Network error occurred
```