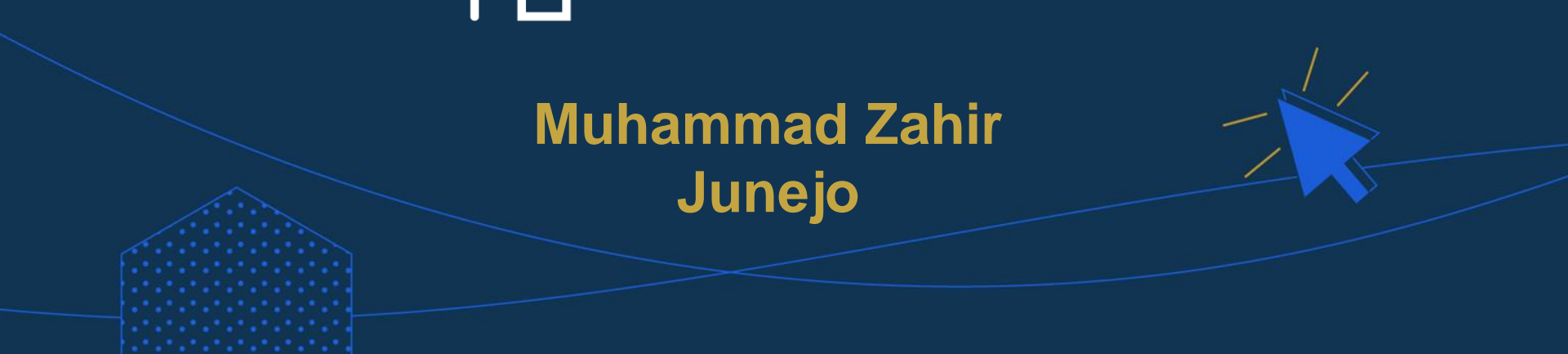




# Security



**Muhammad Zahir  
Junejo**



# Lecture – Housekeeping

- ❑ The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
  - ❑ Please review Code of Conduct (in Student Undertaking Agreement) if unsure
- ❑ No question is daft or silly - **ask them!**
- ❑ Q&A session at the end of the lesson, should you wish to ask any follow-up questions.
- ❑ Should you have any questions after the lecture, please schedule a mentor session.
- ❑ For all non-academic questions, please submit a query: [www.hyperiondev.com/support](https://www.hyperiondev.com/support)

# Lecture Objectives

1. Introduction to Security
2. Overview of JSON Web Tokens (JWT)
3. How to Implement JWT for Application Security
4. Other Security Measures for Web Resources

# Introduction to Security

- ❑ Definition of Security in Software Development: Security in software development involves protecting applications and data from unauthorized access, breaches, and threats.
- ❑ Importance of Security: Security is crucial to safeguard sensitive information, maintain user trust, and prevent financial losses.
- ❑ Types of Threats: Common threats include data breaches, denial of service attacks, and unauthorized access.
- ❑ Consequences of Security Breaches: Breaches can result in data leaks, financial losses, and damage to a company's reputation.

# Security Best Practices

- ❑ The Principle of Least Privilege: Limit user and system access to only what is necessary.
- ❑ Regular Updates and Patch Management: Keep software and systems up-to-date to address vulnerabilities.
- ❑ Data Encryption: Encrypt data both at rest and in transit to protect it from unauthorized access.
- ❑ Password Policies and User Authentication: Enforce strong password policies and multi-factor authentication to enhance user security.

# Overview of JSON Web Tokens (JWT)

- ❑ What is JWT?
  - ❑ JWT is a compact, self-contained mechanism for securely transmitting information between parties as a JSON object.
- ❑ JWT Structure (Header, Payload, Signature):
  - ❑ Header: Contains the type of token and the signing algorithm.
  - ❑ Payload: Contains claims (e.g., user information).
  - ❑ Signature: Ensures the token authenticity and integrity.
- ❑ Advantages of JWT for Authentication and Authorization:
  - ❑ Stateless: No need to store session state on the server.
  - ❑ Compact and self-contained.

# How JWT Works

- ❑ **JWT Generation:** After user authentication, generate a JWT token.
- ❑ **Token Signing and Verification:** Sign the token with a secret key and verify it on subsequent requests.
- ❑ **Use of JWT in Stateless Authentication:** Use JWT to authenticate and authorize users without the need for server-side session management.

# Other Security Measures for Web Resources

- ❑ Cross-Origin Resource Sharing (CORS):
  - ❑ CORS is a security feature that controls web resource access across different domains.
  - ❑ It prevents malicious websites from making unauthorized requests to your backend by specifying which domains are allowed to access your resources.
  - ❑ Implementing CORS headers properly helps prevent cross-site request forgery (CSRF) attacks and protects sensitive data.
- ❑ Input Validation and Sanitization:
  - ❑ Always validate and sanitize user inputs to prevent injection attacks.
  - ❑ Input validation ensures that the data provided by users adheres to expected formats and ranges.
  - ❑ Sanitization cleans and removes potentially malicious or unsafe characters from input data.



# Other Security Measures for Web Resources

- ❑ HTTP Security Headers:
  - ❑ These are HTTP response headers that provide an extra layer of security to your web applications.
  - ❑ Some essential HTTP security headers include:
    - ❑ Content Security Policy (CSP): Specifies which resources are allowed to be loaded, helping prevent code injection attacks like Cross-Site Scripting (XSS).
    - ❑ X-Content-Type-Options: Prevents browsers from interpreting files as something other than their declared content type, reducing the risk of certain attacks.

# References

- ❑ <https://www.passportjs.org/>
- ❑ <https://letsencrypt.org/getting-started/>
- ❑ <https://www.synopsys.com/glossary/what-is-cross-site-scripting.html#:~:text=Definition,the%20user%20to%20click%20it.>
- ❑ <https://www.synopsys.com/glossary/what-is-csrf.html#:~:text=Definition,has%20in%20an%20authenticated%20user.>
- ❑ <https://jwt.io/introduction>



# Questions and Answers





**Thank You!**

