**What is JWT?**

* JWT stands for **JSON Web Token**.
* It is a compact, URL-safe token format used to represent claims securely between two parties.
* Commonly used for authentication and authorization in web applications.

**JWT Process Flow**

1. Client sends username and password to the server.
2. Server validates the credentials and generates a JWT.
3. JWT is returned to the client.
4. Client includes the JWT in the Authorization header for future requests (Bearer <token>).
5. Server validates the JWT in each request to ensure authenticity.

**JWT Structure**

A JWT is composed of three parts separated by dots (.):

Header.Payload.Signature

**1. Header**

Specifies the algorithm and token type.  
Example:

{

"alg": "HS256",

"typ": "JWT"

}

**2. Payload**

Contains application-specific data (claims), such as user information.  
Example:

{

"sub": "user",

"role": "USER",

"iat": 1516239022

}

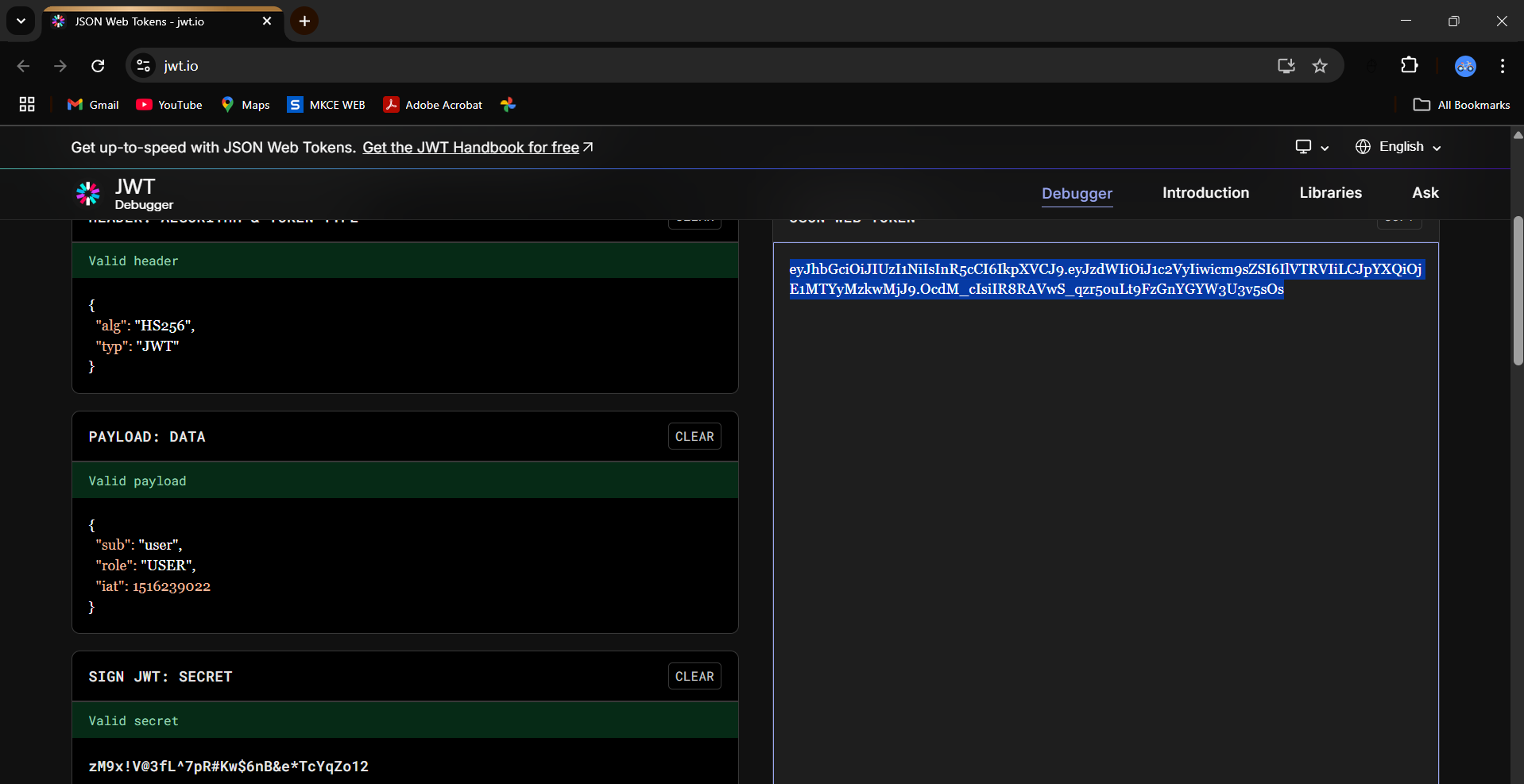
**3. Signature**

Generated using the header, payload, and a secret key. It ensures the integrity and authenticity of the token.

**Hands-On Activity Steps**

1. Open the Wikipedia page on JWT: [JWT Structure](https://en.wikipedia.org/wiki/JSON_Web_Token#Structure)
2. Open <https://jwt.io> in another tab.
3. Copy the **Header** and **Payload** examples from Wikipedia and paste them into the respective fields on jwt.io.
4. In the **Verify Signature** field, type "secretkey" (without quotes).
5. Check if the generated token in the Encoded section matches the one on Wikipedia.

Note: You may see a warning saying:  
**"A key of 256 bits or larger MUST be used with HS256 as specified in RFC 7518."**  
This means your secret key should be at least 32 characters long for better security.

****