JWT, or JSON Web Token, is a compact and secure way to transmit information between parties as a JSON object. It’s widely used for authentication and authorization in modern applications, especially in stateless systems like APIs.

How JWT Works:

1. Structure: A JWT has three parts, separated by dots (.):

• Header: Contains the token type (JWT) and the signing algorithm (e.g., HS256).

• Payload: Holds the claims, which are pieces of information like user ID, roles, or expiration time.

• Signature: Ensures the token’s authenticity by signing the header and payload with a secret key.

Example: Header.Payload.Signature

2. Authentication:

• When a user logs in, the server generates a JWT, signs it, and sends it to the client.

• The client stores this token (often in local storage or cookies) and includes it in the Authorization header of subsequent requests (e.g., Bearer <token>).

3. Stateless:

• JWTs eliminate the need for server-side sessions. All the user’s data and permissions are embedded in the token itself.

Advantages:

• Compact: Easy to pass in headers or URLs due to its small size.

• Stateless: Reduces server load as the token itself contains all necessary information.

• Interoperable: Can be used across different systems and platforms.

• Security: The signature ensures the token hasn’t been tampered with.