AUTHENTICATION & SECURITY

AUTHENTICATION & SECURITY

Authentication and security are important parts of any web system. They protect the system, the users, and the data from being misused or accessed by unauthorized people.

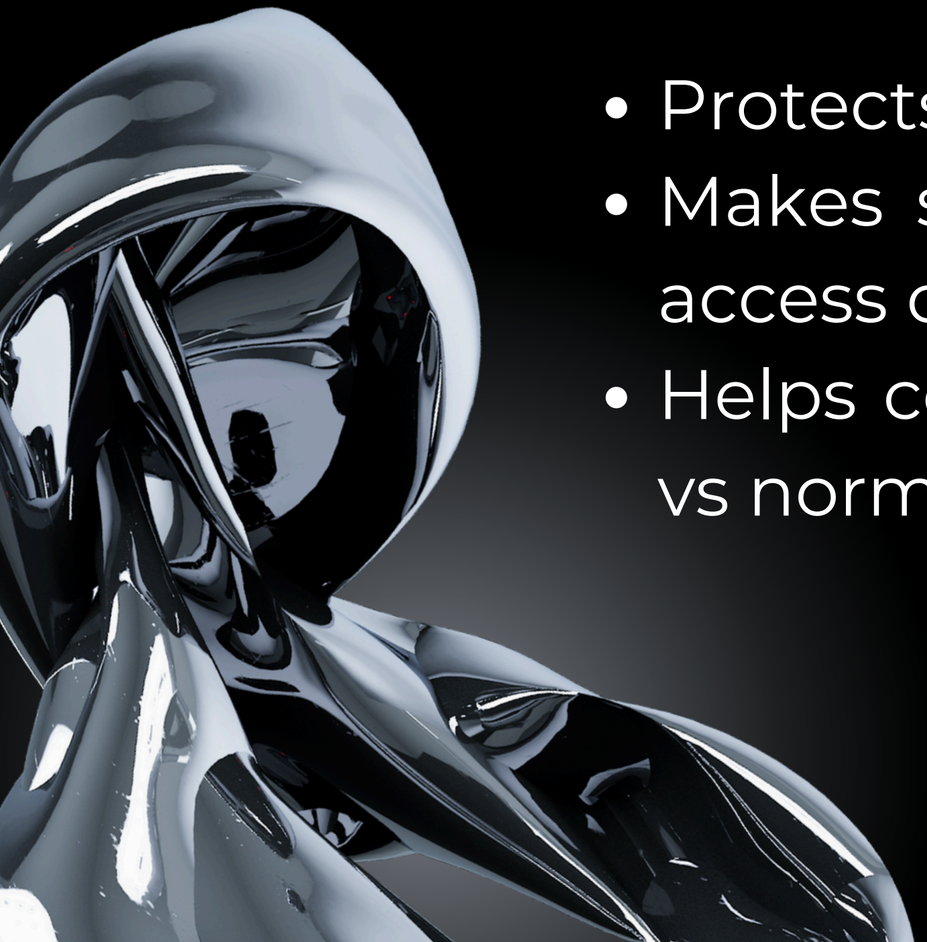
WHAT IS AUTHENTICATION?

Authentication means checking if a **user is who they say they are.**

This usually happens through:

- **Login forms** (username and password)
- **User accounts**
- **Sessions** (keeping users logged in)

WHY IT MATTERS:



- Protects private information
- Makes sure only authorized users can access certain pages
- Helps control user permissions (admin vs normal user)

WHAT IS SECURITY?

Security means protecting the system from attacks, errors, and abuses.

This includes:

- Protecting passwords
- Preventing hackers from entering the system
- Securing the database from harmful code
- Keeping user information safe

AUTHENTICATION IN CODEIGNITER

CodeIgniter makes authentication easier by using:

A. PASSWORD HASHING

- Passwords are not saved as plain text.
- Instead, CodeIgniter uses functions like: password_hash() and password_verify()
- This protects user passwords even if the database is hacked.

AUTHENTICATION IN CODEIGNITER

B. SESSIONS

- After login, CodeIgniter stores a small session data.
- This session keeps the user “logged in” until they log out.

C. FORM VALIDATION

- Checks if user input is correct and safe.
- Prevents empty or harmful inputs.

SECURITY FEATURES IN CODEIGNITER

a. CSRF Protection (Cross-Site Request Forgery)

- Protects forms from unwanted or fake submissions.
- CodeIgniter creates a special token that must match during form submission.

b. XSS Filtering (Cross-Site Scripting)

- Removes harmful scripts or codes from user input.
- Prevents hackers from injecting JavaScript into your site.

SECURITY FEATURES IN CODEIGNITER

c. SQL Injection Protection

- CodeIgniter uses **Query Builder**.
- This avoids direct SQL queries and makes database actions safer.

d. Input Validation

- Ensures only safe, expected input is accepted.

e. Encryption

- CodeIgniter can encrypt data to make it unreadable to attackers.

COMMON SECURITY PRACTICES

- Never store passwords in plain text.
- Always validate user input.
- Use HTTPS (secure connection).
- Apply user roles (admin, user, guest).
- Keep CodeIgniter updated.

EXAMPLE

1. User enters email and password (View)
2. Controller checks form → sends data to Model
3. Model verifies password using **password_verify()**
4. If correct → Controller creates a session
5. View shows dashboard page

If not correct → user sees an error message.

THANK YOU