Backend Development

Day 1:

What is Backend ?

Backend is the **part of a website or app that works behind the scenes**.

When you use Instagram, Amazon, or any website — what you see (buttons, colors, images) is the **frontend**.

But…

When you log in
When your password gets checked
When data is saved in database
When products are fetched from server

That hidden logic? That's **backend**.

In Simple Word We Can say :

Frontend = What users see

Backend = The brain that makes everything work

Example

When you click **"Login"**:

1. Frontend sends your username & password.
2. Backend checks it in the database.
3. If correct → sends "Login Success".
4. If wrong → sends "Invalid Password".

You don't see this process. But backend handles it.

Why do we learn Backend?

- To make websites actually work (login, signup, payments, etc.)
- To store and manage data (users, products, orders)
- To control app logic (who can access what)
- To build real-world applications
- For good career opportunities

Note : Backend is basically:

"Taking request → Processing it → Sending response."

Popular Backend Technologies:

- Node.js
- Express.js
- Python (Django / Flask)
- Java (Spring Boot)
- PHP
- Databases like MySQL, MongoDB

Before Staring the Backend framework/technologies We should know about the basic.

✅ **What is:**

- Server
- Client
- Request
- Response
- HTTP
- API

1. Server

Server is a computer or system that stores data and sends it when requested.

Simple Understanding Purpose

Server = Gives information.

Real-Life Example:

Think of a restaurant kitchen.
The kitchen prepares food when customers order.

👉 Kitchen = Server

2. Client

Client is a user or device that sends request to the server to get data or service.

Simple Understanding Purpose

Client = Asks for information.

Real-Life Example:

You sitting in a restaurant and ordering food.

👉 You = Client

3. Request

A **request** is when the client asks the server for something.

Simple Understanding Purpose

Request = Asking.

Real-Life Example:

You tell waiter: “I want one pizza.”

👉 That order = Request

4.Response

A **response** is the reply from the server to the client.

Simple Understanding Purpose

Response = Answer.

Real-Life Example:

Waiter brings your pizza.

👉 Pizza delivery = Response

5.HTTP

HTTP is a rule that tells how client and server communicate.

Simple Understanding Purpose

HTTP = Communication language between client and server.

Real-Life Example:

It’s like the **restaurant rules**:

- Order properly
- Pay properly
- Follow format

Without rules, no proper communication.

6. API

API (Application Programming Interface) allows different software applications to communicate with each other.

Simple Understanding Purpose

API = Messenger between systems.

Real-Life Example:

Waiter in restaurant.

You don’t go to kitchen directly.
You tell waiter → waiter tells kitchen → kitchen sends food.

👉 Waiter = API

Final Quick Summary (Very Short)

- Client → Asks
- Server → Gives
- Request → Asking message
- Response → Reply message
- HTTP → Communication rule
- API → Messenger between systems