

[Home](#) [Features](#) [Flow](#) [Code](#) [FAQ](#) [Compare](#)

Server-Side Engineering Fundamentals

Master the backend. Build scalable systems.

Switch to Light Mode

Introduction to Server Side Engineering

Server-side engineering involves building and maintaining the part of a web application that runs on a server. It includes processing business logic, handling databases, managing user authentication, and delivering dynamic content to the client.

Technologies like **Node.js**, **Express.js**, and **Django** are commonly used. Server-side development focuses on security, scalability, and performance optimization.

Made with ❤️ by Ashwinth B — 2025 | React Project on Server Side Engineering

Key Server-Side Features



Authentication

Secure login systems using JWT, OAuth, or session-based auth.



Database Operations

Interacting with databases like MongoDB or PostgreSQL via backend APIs.



Routing & APIs

Defining server routes that respond to client requests using REST or GraphQL.



Middleware Functions

Reusable functions for logging, auth-checking, and error handling in the request cycle.

Server Communication Flow

Client Sends Request

The client (browser/app) sends an HTTP request to the server.

← Prev

Next →



[Home](#)[Features](#)[Flow](#)[Code](#)[FAQ](#)[Compare](#)

Sample Express Server

```
const express = require('express');
const app = express();

app.get('/', (req, res) => {
  res.send('Hello from the server!');
});

app.listen(3000, () => {
  console.log('Server running on port 3000');
});
```

Made with ❤️ by Ashwinth B — 2025 | React Project on Server Side Engineering

localhost:3000/code

💡 Frequently Asked Questions

What is middleware?

Middleware is a function that runs before the request reaches the final route handler.

What is REST API?

Why is server-side security important?



Client vs Server Comparison

Aspect	Client Side	Server Side
Execution	In browser	On server
Security	Low	High
Languages	HTML, CSS, JS	Node.js, PHP, Python
Focus	UX & Interactions	Logic, Data, Auth