**19. What is Full Stack Development?**Full Stack Development refers to the development of both the front-end (client-side) and back-end (server-side) parts of a web application. A Full Stack Developer is someone who is skilled in both areas and can manage the entire stack of technologies needed to build and run a web application.

**Components of Full Stack Development:**

1. **Front-End Development (Client-Side):**
   * Focuses on user interfaces and user experiences.
   * Technologies:
     + HTML, CSS, and JavaScript.
     + Frameworks/Libraries like React, Angular, or Vue.js.
2. **Back-End Development (Server-Side):**
   * Focuses on database interactions, server logic, and application functionality.
   * Technologies:
     + Programming languages like Node.js, Python, Java, PHP.
     + Frameworks like Express.js, Django, Flask, or Spring.
3. **Database Management:**
   * Storing, managing, and retrieving data.
   * Examples: MySQL, MongoDB, PostgreSQL.
4. **DevOps & Hosting:**
   * Deployment, hosting, and scaling applications.
   * Examples: Docker, AWS, Heroku.

**Example: Full Stack Development**

**Tech Stack:**

* **Front-End:** React.js
* **Back-End:** Node.js with Express
* **Database:** MongoDB

**Code Example:**

1. **Front-End (React.js):**

*// App.js*

import React, { useState, useEffect } from 'react';

*const* App = () => {

*const* [data, setData] = useState([]);

useEffect(() => {

fetch('http://localhost:5000/api/data')

.then((*response*) => response.json())

.then((*data*) => setData(data));

}, []);

return (

<div>

<h1>Full Stack Example</h1>

<ul>

{*data*.*map*((*item*) => (

<li key={item.id}>{item.name}</li>

))}

</ul>

</div>

);

};

export default App;

Back-End (Node.js + Express):

*// server.js*

*const* express = require('express');

*const* cors = require('cors');

*const* app = express();

app.use(cors());

*const* data = [

{ id: 1, name: 'Alice' },

{ id: 2, name: 'Bob' },

];

app.get('/api/data', (*req*, *res*) => {

res.json(data);

});

app.listen(5000, () => {

console.log('Server running on http://localhost:5000');

});

Database (MongoDB Example):

*// Connecting to MongoDB*

*const* mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/fullstack', {

useNewUrlParser: true,

useUnifiedTopology: true,

});

*const* schema = new mongoose.Schema({ name: String });

*const* User = mongoose.model('User', schema);

app.get('/api/users', *async* (*req*, *res*) => {

*const* users = await User.find();

res.json(users);

});