

## Laboratory Practice-II (AI) Oral Questions (1 to 20)

### 1. Differentiate between DFS and BFS Algorithms.

**DFS (Depth First Search)** explores as far as possible along each branch before backtracking. DFS explores deep paths using **stack**,

**BFS (Breadth First Search)** explores all the nodes at the present depth level before moving on to the nodes at the next depth level. BFS explores level-wise using **queue**.

Feature	DFS (Depth First Search)	BFS (Breadth First Search)
Traversal Type	Level-wise (goes broad first)	Level-wise (goes broad first)
Data Structure Used	stack	queue
Memory Usage	less	more
Backtracking	Yes	no

### 2. Write down the Time and Space Complexity of DFS and BFS.

- **DFS:** Time –  $O(V + E)$ , Space –  $O(V)$
- **BFS:** Time –  $O(V + E)$ , Space –  $O(V)$  Where V = vertices, E = edges

### 3. Which data structure is used by DFS and BFS?

- DFS uses a stack (can be implemented using recursion).
- BFS uses a queue.

### 4. Differentiate between Best First Search and A\* algorithm.

- **Best First Search:** Uses only heuristic function  $h(n)$ .



**Study material provided by: Vishwajeet Londhe**

**Join Community by clicking below links**



**Telegram Channel**



**[https://t.me/SPPU\\_TE\\_BE\\_COMP](https://t.me/SPPU_TE_BE_COMP)**

(for all engineering Resources)



**WhatsApp Channel**

(for all Engg & tech updates)



**<https://whatsapp.com/channel/0029ValjFrilCVfpcV9HFc3b>**



**Insta Page**

(for all Engg & tech updates)



**@SPPU\_ENGINEERING\_UPDATE**

**[https://www.instagram.com/sppu\\_engineering\\_update](https://www.instagram.com/sppu_engineering_update)**

- **A\* Algorithm:** Uses both heuristic  $h(n)$  and cost so far  $g(n)$ .  $A^* = g(n) + h(n)$

Feature	Best First Search	A* Algorithm
Evaluation Function	$f(n) = h(n)$	$f(n) = g(n) + h(n)$
<b>Heuristic Used</b>		
Optimality	Not guaranteed to be optimal	Guaranteed optimal if $h(n)$ is admissible
Completeness	Not complete	Complete (if step cost $\geq \epsilon$ and finite branching)
Search Strategy	Greedy (chooses node that appears closest to goal)	Informed (balances cost so far and estimated cost to goal)
Speed	Generally faster but less accurate	Slower but finds optimal and accurate path
Use Case	When fast, approximate solutions are acceptable	When optimal solution is important
Path Cost Consideration	Ignores path cost so far ( $g(n)$ )	Considers both path cost so far and estimated cost to goal

## 5. Solve this problem using A\* algorithm.

You'll need initial, goal state, heuristic and cost function. If you have a specific example, I can solve it step-by-step.

## 6. What is the drawback to solve 8 Puzzle problem with a non-heuristic method?

It will take more time and memory as it blindly explores all paths, not guiding towards the goal.

It becomes **very slow** and explores **many unnecessary paths**, making it **inefficient**

It is time consuming process

### 7. What is the time and space complexity of selection sort?

Selection Sort is a sorting algorithm that works by repeatedly finding the smallest element in the unsorted part of a list and swapping it with the first unsorted element. This process is repeated until the entire list is sorted.

- **Time:**  $O(n^2)$
- **Space:**  $O(1)$  – works in-place

### 8. Sort [6, 1, 9, 10] using selection sort, stepwise.

1. [1, 6, 9, 10]
2. [1, 6, 9, 10]
3. [1, 6, 9, 10] (Final Sorted)

### 9. Maximum comparisons in one iteration for array of size N?

An **array** is a data structure that **stores a collection of elements**, typically of the same type, in a specific order. These elements can be accessed using an index or a key. Arrays are commonly used in programming to handle multiple values under one variable name, making it easier to manage and manipulate large sets of data.

It is  $(N - 1)$  comparisons.

### 10. Comparison chart for sorting techniques:

Sort	Time	Space	Stable
Selection	$O(n^2)$	$O(1)$	No
Bubble	$O(n^2)$	$O(1)$	Yes
Merge	$O(n \log n)$	$O(n)$	Yes
Quick	$O(n \log n)$	$O(\log n)$	No

**11. (Combined with Q10)**

**12. What is a Stable Sort Algorithm? Is selection sort stable?**

- Stable sort keeps equal elements in the same order.
- **Selection Sort is not stable** by default.

**13. Constraints to solve N-Queen Problem?**

No two queens should be in the **same row, column or diagonal**.

**14. Compare Backtracking and Branch and Bound.**

- **Backtracking**: Tries all possibilities, skips invalid ones.
- **Branch and Bound**: Uses cost to cut down unnecessary paths.

**15. What is a Constraint Satisfaction Problem (CSP)?**

Problems with rules/conditions (constraints) that must be satisfied.

Example: Sudoku, N-Queen.

**16. Use of a chatbot?**

Used to answer queries, give support, and automate conversations.

Chatbots automate responses to user queries like in customer support or virtual assistants.

**17. Explain dialog flow.**

It is how a chatbot **understands, processes, and responds** to users. Includes **intents, entities, and responses**.

Dialog flow is a platform to build chatbots that understand user input using NLP(**Natural language processing**) and provide responses.

#### **18. Requirements to develop a chatbot:**

1. **NLP Engine:** Understands and processes human language (e.g., recognizing intent, extracting info).
2. **User Interface:** The front-end where users interact with the chatbot (like a chat window on a website or app).
3. **Backend System:** Handles logic, manages workflows, and connects everything together behind the scenes.
4. **Database:** Stores user data, chat history, and other information the bot needs to work properly.

#### **19. How do you evaluate chatbot performance?**

Using accuracy, response time, user satisfaction, fallback rates. Task completion rate

#### **20. How to improve chatbot accuracy?**

- Use more training data
- Refine intents and entities
- Add fallback messages and context

---

## Laboratory Practice-II (Cloud Computing)

### Oral Questions (1 to 40)

#### 1. What is Cloud Computing with example?

Cloud computing is using internet to access servers, storage, and software.

It means storing and accessing data/apps on the internet.

**Example:** Google Drive, Dropbox

#### 2. Who is the father of cloud computing?

**Dr. Joseph Carl Robnett Licklider** is the father of cloud computing.

#### 3. Basic characteristics of cloud computing:

1. **On-demand self-service:** You can get resources whenever you need them without asking anyone.
2. **Scalability:** You can increase or decrease resources easily based on need.
3. **Availability:** Resources are always available with minimal downtime.
4. **Flexibility:** You can choose different types of resources and adjust them as needed.
5. **Pay-as-you-go:** You only pay for the resources you actually use.
6. **Accessibility & platform Independent :** You can access services from anywhere using the internet.
7. **Resource pooling:** Multiple users share the same resources, but it's securely managed.
8. **Rapid elasticity:** Resources can be quickly added or removed as demand changes.
9. **Measured service:** Your usage is tracked and billed accurately based on what you use.

#### 4. Vertical vs Horizontal scaling:

- **Vertical Scaling:** Adding more power (like CPU, RAM) to a single server to handle more load. Think of it as upgrading your computer to make it faster and stronger.
- **Horizontal Scaling:** Adding more servers to handle the load by spreading it across many machines. It's like adding more workers to get the job done faster instead of making one worker do everything.

## 5. Services provided by cloud computing:

- **IaaS (Infrastructure as a Service):** Provides virtualized computing resources like servers, storage, and networks over the internet.
- **PaaS (Platform as a Service):** Offers a platform allowing developers to build, deploy, and manage applications without worrying about the underlying infrastructure.
- **SaaS (Software as a Service):** Delivers software applications over the internet, where users can access and use them directly, without installation or maintenance.

## 6. Deployment models in cloud:

- **Public cloud:** Cloud services shared with everyone, run by a third party.
- **Private cloud:** Cloud used only by one organization for more control.
- **Hybrid cloud:** Combination of both public and private clouds.
- **Community cloud:** Cloud shared by a group of similar organizations.

## 7. Platforms used for large scale cloud computing:

- **Hadoop:** A framework that allows you to store and process large datasets across many computers, making it easier to handle big data.
- **Spark:** A fast, open-source engine that processes large amounts of data much quicker than Hadoop, often used for real-time data processing.
- **Kubernetes:** An open-source platform used to automate the deployment, scaling, and management of containerized applications (applications packaged in containers for easy deployment).

#### 8. Large cloud providers and databases:

##### Large Cloud Providers:

- **AWS (Amazon Web Services)**: A major cloud provider offering a wide range of cloud services like computing power, storage, and databases.
- **Azure**: Microsoft's cloud platform providing services for computing, storage, networking, and more.
- **Google Cloud**: Google's cloud platform offering various services including data storage, computing, and machine learning tools.

##### Databases:

- **MongoDB**: A NoSQL database that stores data in flexible, JSON-like documents.
- **BigTable**: A scalable NoSQL database from Google, designed for large analytical and operational workloads.
- **Firebase**: A cloud-based platform from Google that offers real-time databases, authentication, and hosting for apps.

#### 9. Difference between cloud and traditional datacenters:

- **Cloud**: Virtual, easily scalable, and you pay only for what you use.
- **Traditional Datacenters**: Physical hardware, expensive to maintain, and less flexible for growth.

#### 10. Components in cloud computing:

- **Clients**: Devices (like computers, phones) that access cloud services over the internet.
- **Datacenters**: Physical facilities that house cloud servers and store data.
- **Distributed Servers**: Servers spread across multiple locations, working together to provide cloud services efficiently and reliably.

#### 11. Platforms in cloud architecture:

- **Management Platform:** Tools and services used to monitor, control, and manage cloud resources and services.
- **Storage Platform:** Cloud services that provide storage space for data, allowing easy access and management.
- **Compute Platform:** Cloud services that provide computing power to run applications, process data, and perform tasks.

## 12. What is a cloud service?

A service like storage, database, or computing provided over the cloud.

1. **IaaS (Infrastructure as a Service):** Provides virtualized computing resources over the internet.  
*Example:* AWS EC2, Google Compute Engine.
2. **PaaS (Platform as a Service):** Offers a platform to build, deploy, and manage applications.  
*Example:* Google App Engine, Microsoft Azure App Service.
3. **SaaS (Software as a Service):** Delivers software applications over the internet.  
*Example:* Google Workspace, Microsoft 365.
4. **Storage Services:** Provides cloud storage to store and manage data. *Example:* Amazon S3, Google Cloud Storage.
5. **Database Services:** Cloud-based databases for data management and storage.  
*Example:* Amazon RDS, Microsoft Azure SQL Database.
6. **CDN (Content Delivery Network):** Distributes content to users from the nearest location for faster access.  
*Example:* Amazon CloudFront, Cloudflare.
7. **Compute Services:** Provides computing power to run applications and process data.  
*Example:* AWS Lambda, Azure Functions.
8. **Networking Services:** Cloud solutions for network connectivity, security, and load balancing.  
*Example:* Azure Virtual Network, Google Cloud VPC.

9. **AI/ML Services:** Provides tools for building and deploying AI and machine learning models.

*Example:* AWS SageMaker, Google AI Platform.

10. **Backup and Disaster Recovery:** Cloud services to back up data and recover from failures.

*Example:* AWS Backup, Azure Site Recovery.

### 13. Basic clouds in cloud computing:

1. **Public Cloud:** Cloud services offered to the general public, shared by multiple users, and managed by third-party providers.

*Example:* AWS, Microsoft Azure, Google Cloud.

2. **Private Cloud:** Cloud infrastructure dedicated to a single organization, offering more control and security, not shared with others.

*Example:* A company's own data center or private cloud services like VMware.

3. **Hybrid Cloud:** A mix of public and private clouds, allowing data and applications to be shared between them for greater flexibility and optimization.

*Example:* A company using both its own private cloud and public cloud services like AWS or Azure.

### 14. Issues with cloud computing:

1. **Security:** Risks of data being hacked or accessed by unauthorized users.

2. **Privacy:** Concerns about who controls and accesses your sensitive data.

3. **Downtime:** Cloud services may go offline, causing disruptions.

4. **Vendor Lock-in:** It's hard to switch to another cloud provider once you're using one.

### 15. Services of Windows Azure:

1. **App Services:** A platform to build, deploy, and scale web applications and APIs without managing the underlying infrastructure.

2. **Azure Functions:** A serverless compute service that allows you to run code in response to events without managing servers.
3. **Virtual Machines:** Provides scalable virtualized computing resources that let you run applications and services just like on a physical server.

## 16. What is cloud computing architecture?

**Cloud computing architecture** refers to the design and structure of the components and services that work together to deliver cloud computing services.

1. **Front-end:** The client side, such as user devices or web applications that interact with the cloud services.
2. **Back-end:** The cloud infrastructure that includes data storage, servers, databases, and services, managed by cloud providers.
3. **Middleware:** Software that connects front-end and back-end systems, enabling communication and data exchange.
4. **Cloud services:** The actual services (IaaS, PaaS, SaaS) provided to users for computing, storage, networking, and more.

## 17. Services by AWS:

1. **EC2 (Elastic Compute Cloud):** Scalable virtual servers to run applications and workloads.
2. **S3 (Simple Storage Service):** Object storage service for storing and retrieving large amounts of data.
3. **RDS (Relational Database Service):** Managed database service for SQL databases like MySQL, PostgreSQL, and SQL Server.
4. **Lambda:** Serverless computing service that lets you run code in response to events without managing servers.
5. **VPC (Virtual Private Cloud):** Enables you to create a private, isolated network within AWS for securely launching resources.

6. **DynamoDB**: Fully managed NoSQL database service for fast and flexible data storage.
7. **CloudFront**: Content Delivery Network (CDN) for fast delivery of websites, videos, and other content to users globally.
8. **Elastic Load Balancing**: Automatically distributes incoming traffic across multiple servers to ensure high availability.
9. **SQS (Simple Queue Service)**: Managed message queuing service to decouple and scale microservices.
10. **SNS (Simple Notification Service)**: Messaging service for sending notifications to subscribers via email, SMS, etc.

#### **18. Services by Microsoft Azure:**

1. **Azure Virtual Machines**: Scalable virtual servers to run applications and workloads.
2. **Azure Blob Storage**: Object storage service for storing large amounts of unstructured data like documents and media files.
3. **Azure SQL Database**: Managed relational database service for SQL Server databases.
4. **Azure Functions**: Serverless computing that lets you run event-driven code without managing infrastructure.
5. **Azure Active Directory**: Identity and access management service for secure user authentication and access to resources.
6. **Azure Kubernetes Service (AKS)**: Managed service for deploying and managing containerized applications using Kubernetes.
7. **Azure App Service**: Platform for building, hosting, and scaling web applications and APIs.
8. **Azure DevOps Services**: Tools for automating software development processes, including continuous integration and delivery (CI/CD).
9. **Azure Cognitive Services**: A set of APIs for adding AI capabilities such as vision, speech, language, and decision-making to applications.

10. **Azure Virtual Network:** Networking service that allows you to create secure, isolated networks for your resources.

## 19. What is virtualization?

- Creating virtual version of hardware, OS, or storage.
- **Virtualization** is the process of creating a virtual version of a physical resource, such as a server, storage device, or network. It allows multiple virtual instances (virtual machines, virtual storage, etc.) to run on a single physical machine, optimizing resource use and enabling efficient management.
- virtualization lets you run multiple "virtual" systems on one physical machine, making better use of hardware resources.

## 20. What is EC2 instance?

A virtual server in AWS cloud.

An **EC2 instance** (Elastic Compute Cloud) is a virtual server provided by Amazon Web Services (AWS) to run applications and workloads in the cloud.

1. **Launch Virtual Servers:** EC2 instances are virtual machines that can be configured with different amounts of CPU, memory, and storage.
2. **Scalability:** You can easily scale up or down based on your computing needs.
3. **Cost-Effective:** You pay only for the compute resources you use, with options for on-demand, reserved, or spot pricing.
4. **Flexibility:** You can choose from a variety of operating systems (Linux, Windows) and software configurations.
5. **Customizable:** You can select instance types based on performance requirements like compute power, memory, or storage.

## 21. What is AMI in EC2?

Amazon Machine Image: Pre-configured OS + app environment to launch EC2.

An **AMI (Amazon Machine Image)** is a template that contains everything needed to launch an EC2 instance, including the operating system and software.

1. **Pre-configured:** Includes the OS and software setup.
2. **Customizable:** You can create your own AMI with specific apps or settings.
3. **Fast Setup:** Helps quickly launch EC2 instances with the same setup.

## **22. Launching on-demand EC2 instance:**

Use AWS Console > EC2 > Launch Instance > Select AMI > Configure > Launch.

## **23. What is Google App Engine?**

Platform to develop and host web applications on Google's infrastructure.

**Google App Engine** is a fully managed **Platform-as-a-Service (PaaS)** offering by Google Cloud that enables developers to build, deploy, and scale web applications and APIs without managing the underlying infrastructure.

**Key points :** Automatic Scaling, Managed Environment, Multi-language Support, Integrated Services

## **24. Advantages of Google App Engine:**

1. **Auto-scaling:** Adjusts resources based on app traffic.
2. **No server management:** Google handles the infrastructure.
3. **Pay-as-you-go:** You only pay for what you use.
4. **Multiple languages:** Supports Python, Java, Go, and more.
5. **Integrated services:** Comes with built-in tools like storage and databases.
6. **Cost saving:** No need to invest in physical servers or infrastructure; you only pay for usage.

7. **Performance:** High performance with automatic scaling and load balancing to handle more traffic efficiently.

## 25. Steps to install Google App Engine:

1. Install SDK: Download and install Google Cloud SDK.
2. Setup project: Initialize with `gcloud init` and set project ID.
3. Deploy app: Use `gcloud app deploy` in project root directory.

## 26. What is Apex?

Apex is a strongly typed, object-oriented programming language used to develop custom business logic on the Salesforce platform.

Apex is a programming language used in Salesforce.

## 27. Features of Apex:

1. **Object-Oriented** – Supports classes, inheritance, and interfaces.
2. **Strongly Typed** – Variables must be declared with data types.
3. **Integrated with Salesforce** – Direct access to Salesforce objects and database.
4. **Built-in DML & SOQL** – Native support for data manipulation and queries.
5. **Triggers Support** – Can be used to write triggers for automation.
6. **Batch & Scheduled Processing** – Supports asynchronous execution.
7. **Test Classes** – Enforces unit testing with code coverage requirements.
8. **Version Control** – Code can be versioned and deployed easily.

9. **Multitenant Aware** – Designed to work securely in shared environments.
10. **Governor Limits** – Ensures efficient use of shared resources.

## 28. Application of Apex:

1. **Custom Business Logic** – Implement complex workflows, validation rules, and calculations.
2. **Triggers** – Automate record updates and actions (e.g., after insert, update, delete).
3. **Web Services** – Expose custom RESTful or SOAP web services for integration with other systems.
4. **Batch Processing** – Handle large volumes of data asynchronously with batch jobs.
5. **Scheduled Jobs** – Automate tasks on a defined schedule (e.g., nightly data cleanup).
6. **Custom APIs** – Build custom APIs to interact with external systems.
7. **Data Manipulation** – Perform bulk updates, inserts, and deletes of Salesforce records.
8. **Test Automation** – Write unit tests to ensure code reliability and meet deployment requirements.
9. **Integration** – Integrate Salesforce with external applications and services.
10. **Custom Controllers** – Create custom Visualforce pages and Lightning components with Apex controllers.

## 29. Apex Code Development Tools:

1. **Salesforce Developer Console** – A web-based tool within Salesforce for writing, testing, and debugging Apex code directly.
2. **VS Code with Salesforce Extensions** – A powerful local development environment with Salesforce extensions (e.g., Salesforce CLI, Apex Language Server) for efficient coding, debugging, and deployment.

### 30. Steps to create app in Apex:

1. **Setup Salesforce Org** – Create and configure a Salesforce Developer or Production Org.
2. **Write Apex Code** – Develop custom logic using the Apex editor in Salesforce or VS Code with Salesforce extensions.
3. **Test and Deploy** – Write test classes, ensure code coverage, and deploy using Salesforce tools like the Developer Console or VS Code.

### 31. What is Salesforce.com Inc.?

A cloud-based CRM platform.

Salesforce.com Inc. is a cloud-based SaaS provider offering a comprehensive CRM platform with integrated solutions for sales automation, marketing, customer service, analytics, and application development, leveraging multi-tenant architecture and a highly scalable, customizable environment for business process automation and data management.

### 32. What is Lightning Platform?

Salesforce platform for building apps with UI and backend logic.

The **Lightning Platform** (formerly Salesforce1 Platform) is a cloud-based development platform by Salesforce that allows users to build and deploy custom applications. It provides a set of tools, including Lightning components, App Builder, and Apex, for creating responsive, scalable, and interactive applications with a focus on user experience and seamless integration with Salesforce CRM data.

### 33. Create Custom App using Salesforce Classic:

1. **Go to Setup** – Click on your name → Setup.
2. **Navigate to Apps** – Under "Build", select *Create* → *Apps*.
3. **Click New** – In the Apps section, click the *New* button.
4. **Choose Custom App** – Select *Custom App* and click *Next*.
5. **Enter App Details** – Provide App Label, Name, and upload logo (optional).

6. **Select Tabs** – Choose the tabs you want in your app (e.g., Accounts, Contacts).
7. **Assign Profiles** – Choose which user profiles can access the app.
8. **Save** – Click Save to create the custom app.

#### **34. Difference: Custom vs Console app:**

- Custom app: For regular use
- Console app: For customer support (multi-tabbed)

Feature	Custom App	Console App
UI Type	Standard tab-based interface	Tab-based with split-view for multitasking
Use Case	General business processes	High-volume support/sales scenarios
Navigation	One record at a time	Multiple records in sub-tabs within a workspace
Efficiency	Less efficient for multitasking	Designed for fast, multi-record access
Best For	Regular users and managers	Call centers, service agents, sales reps

#### **35. Steps to create custom app:**

Same as Q33.

1. **Go to Setup** – Click the gear icon → *Setup*.
2. **Search "App Manager"** – In Quick Find, type *App Manager*.
3. **Click "New Lightning App"** – For Lightning, or *New* under Classic Apps.
4. **Enter App Details** – Name, description, and branding options.
5. **Choose Navigation Style** – Standard or Console (for Lightning).

6. **Add Items (Tabs)** – Select objects like Accounts, Contacts, etc.
7. **Assign Profiles** – Choose which user profiles can access the app.
8. **Review & Save** – Confirm setup and click *Finish* to create the app.

### **36. What is a hypervisor?**

Software that creates and runs virtual machines.

A **hypervisor** is a virtualization layer that enables multiple guest operating systems to run concurrently on a single physical host by abstracting and managing the underlying hardware resources such as CPU, memory, and I/O devices.

### **37. Load Balancing in Cloud Computing:**

Distributing workloads across servers to ensure efficiency and avoid overload.

**Load Balancing in Cloud Computing** is the process of distributing incoming network traffic or application workloads across multiple servers or resources to ensure optimal performance, high availability, fault tolerance, and efficient resource utilization.

### **38. Open-source cloud databases:**

1. **MySQL** – Popular relational database, widely used in web applications.
2. **PostgreSQL** – Advanced relational database with strong support for complex queries.
3. **MongoDB** – NoSQL database for storing JSON-like documents.
4. **Cassandra** – Distributed NoSQL database for handling large volumes of data.
5. **Redis** – In-memory key-value store used for caching and real-time analytics.
6. **MariaDB** – Fork of MySQL with additional features and performance improvements.

7. **CouchDB** – NoSQL database that stores data in JSON documents with sync support.

#### **39. Recent trends in cloud computing:**

1. **AI in the Cloud** – Companies use ready-made AI tools from cloud providers without building their own.
2. **Hybrid & Multi-Cloud** – Businesses use a mix of cloud services (like AWS, Azure, Google) for better flexibility.
3. **Edge Computing** – Data is processed closer to where it's created (like in IoT devices) to reduce delays.
4. **Serverless Computing** – Developers focus on code only; the cloud handles servers automatically.
5. **Green Cloud (Sustainability)** – Cloud companies are working to reduce energy use and be eco-friendly.
6. **Quantum Cloud** – Early access to powerful quantum computers through the cloud.
7. **Smart Cloud Management** – AI helps manage and optimize cloud performance and resources.
8. **Distributed Cloud** – Cloud services are spread across many locations for speed and data compliance.
9. **Stronger Security** – New tools like zero-trust and AI-powered protection are making the cloud safer.
10. **Cloud-Native Apps** – Apps built using modern tech like containers and microservices for speed and scalability.

#### **40. Applications of cloud computing:**

1. **Online Storage** – Services like Google Drive, Dropbox, and iCloud provide scalable, remote storage for personal and business data.

2. **Streaming** – Platforms like Netflix, Spotify, and YouTube use cloud computing for streaming videos and music to millions of users.
3. **E-commerce** – Cloud infrastructure supports online stores (like Amazon, eBay) by providing scalable resources for handling traffic and transactions.
4. **Healthcare Systems** – Cloud-based healthcare solutions enable secure patient data storage, telemedicine services, and real-time access to medical records across platforms.