1:) What is a Program?

ANS) A program is a set of instructions written in a programming language that tells a computer what to do.

2:) What is Programming?

ANS) Programming is the process of writing, testing, and maintaining a set of instructions that a computer can follow to perform a specific task.

KEY STEPS :-

I) Problem Definition
II) Planning and Design
III) Coding
IV) Testing and Debugging
V) Documentaion
VI) Execution
VII) Maintenance

3:) Types of Programming Languages :-

ANS)

High-level and low-level programming languages differ in several key aspects:

(Abstraction Level)

=>High-level languages provide a higher level of abstraction from the hardware, using human-readable syntax and constructs.
=>Low-level languages are closer to machine code, offering minimal abstraction and requiring detailed hardware management.

(Ease of Use)

=>High-level languages are easier to learn and use, with simpler syntax (e.g., Python, Java, C++).
=>Low-level languages are harder to learn as they require understanding of hardware architecture(e.g.,Assembly, Machine Code).

(Portability)

=>High-level languages are platform-independent and can run on different systems with minimal modification.

=>Low-level languages are platform-dependent and often need to be rewritten for different architectures.

(Performance & Efficiency)

=>High-level languages can be slightly slower due to the overhead of interpretation or compilation.

=>Low-level languages provide higher performance and efficiency as they interact directly with the hardware.

(Memory Management)

=>High-level languages often handle memory management automatically (e.g., garbage collection in Java, Python).

=>Low-level languages require manual memory management, allowing more control but also increasing complexity.

4:) Describe the roles of the client and server in web communication

ANS) In web communication, the client requests information from the server, and the server responds with the requested data. This is called the client-server model.

Client :-

> The client is the device that requests information from the server.
> The client can be a computer, such as a laptop, smartphone, or desktop.
> The client is also known as the service requester.
> The client initiates communication with the server by sending a request.

Server:-

> The server is the device that provides information to the client.
> The server can be a computer that stores and manages files.
> The server handles tasks such as data storage, security, and resource management.
> The server processes the client's request and sends a response back to the client.

5:) Explain the function of the TCP/IP model and its layers.

ANS) function of the TCP/IP model and its layers:-

1. Network Interface (Link) Layer
Function: Handles physical transmission of data over the network.

2.Internet Layer

Function: Routes data packets between devices across different networks.

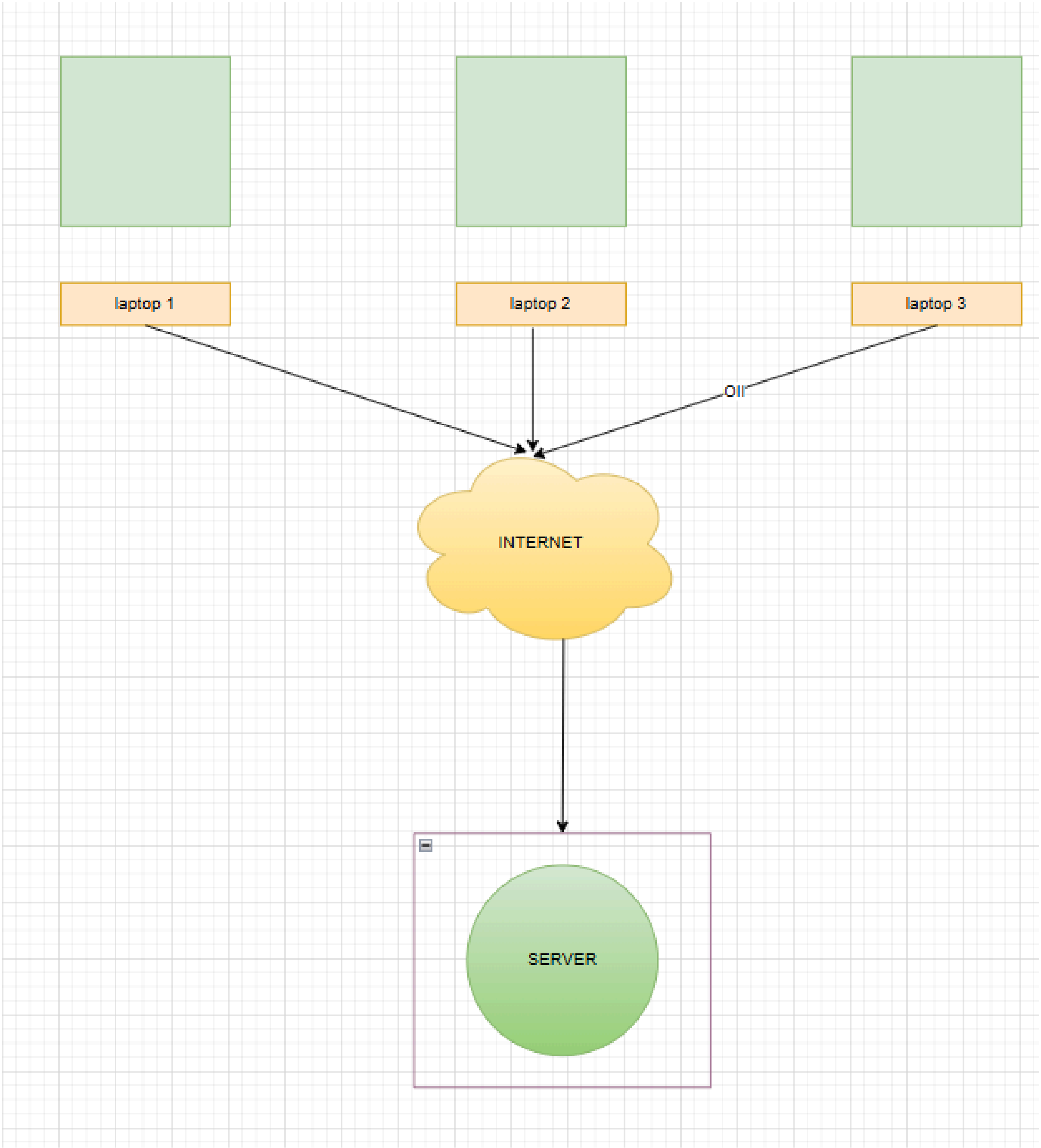
3. Transport Layer
Function: Ensures reliable or fast data delivery between applications.

4. Application Layer
Function: Provides network services to end-users and applications.

6:) Explain client server communication

ANS) Client-server communication is a model where a client requests services, and a server processes these requests and provides responses. This model is widely used in networks, including the internet.

Example of client server communication:-



7:) What are the differences between HTTP and HTTPS protocols?

ANS) HyperText Transfer Protocol (HTTP)

> HyperText Transfer Protocol (HTTP) is a protocol using which hypertext is transferred over the Web.
> Due to its simplicity, HTTP has been the most widely used protocol for data transfer over the Web.
> The connection between the web browser and the server ends after the transaction is finished.

Hypertext Transfer Protocol Secure (HTTPS)

> Hypertext Transfer Protocol Secure (HTTPS) is an extended version of the Hypertext Transfer Protocol (HTTP).
> In HTTPS, the communication protocol is encrypted using Transport Layer Security.
> While HTTPS guarantees data security, the HTTP protocol does not provide data security.

8:) What is the role of encryption in securing applications ?

ANS) Encryption plays a critical role in securing applications by converting sensitive data into an unreadable format, ensuring that only authorized parties can access it.

9:) What is the difference between system software and application software?

ANS) > System Software: The foundational software that manages hardware and provides a platform for running applications.Example,Operating Systems (Windows),
Device Drivers

> Application Software: Programs designed for end-users to perform specific tasks. Example ,word processing, browsing, or gaming.

10:) What is the significance of modularity in software architecture ?

ANS) Modularity is a key principle in software architecture that involves breaking a system into smaller, independent, and reusable components.

11:) Why are layers important in software architecture ?

ANS) Layers in software architecture refer to the logical separation of concerns within an application, where each layer is responsible for a specific function and interacts with adjacent layers in a structured manner.

12:) Explain the importance of a development environment in software production.

ANS) A development environment is a set of tools, configurations, and resources that enable software developers to write, test, and debug applications efficiently.

KEYS OF development environment in software production:-

I) Facilitates Code Writing & Management

II) Ensures Code Consistency & Standardization

III) Enables Efficient Debugging & Testing

IV) Enhances Collaboration Among Teams

V) Speeds Up Development & Deployment

13:) Why is version control important in software development?

ANS) Version control is a system that records changes to files over time, allowing developers to track modifications, collaborate efficiently, and revert to previous versions if needed.

14:) : What are the benefits of using Github for students?

GitHub offers several benefits for students, helping them learn, collaborate, and build a strong portfolio. Some key advantages include:

I) Access to Open Source Projects

II) Project Collaboration

III) Networking with Developers

IV) Internship & Job Opportunities

15:) How does GIT improve collaboration in a software development team ?

ANS) Git enhances teamwork by providing an efficient, structured, and reliable way to manage code. Here's how:

I) Version Control & Tracking:-
Git tracks all changes made to the code, allowing developers to review history and revert to previous versions if needed.

II) Branching & Merging:-

Developers can create separate branches to work on features or bug fixes without affecting the main codebase.

III) Parallel Development:-
Multiple team members can work on different parts of the project at the same time.

IV) Code Review & Quality Control:-
Git allows for pull requests and code reviews, enabling teammates to review changes before merging.

V) Backup & Recovery:-
Since the repository exists both locally and remotely (e.g., GitHub, GitLab, Bitbucket), there's always a backup.

16:) What is the role of application software in businesses?

Application software plays a vital role in businesses by improving efficiency, automating processes, and enhancing decision-making.

17:) What are the main stages of the software development process?

ANS)

The Software Development Life Cycle (SDLC) consists of several key stages that guide the creation of high-quality software. The main stages include:
The main stages of the software development process:-

I) Analysis:-

> Understanding the project needs and defining clear requirements.

II) System Design:-

> Translating requirements into a technical blueprint.

III) Implementation:-

> Writing clean, modular, and efficient code.

IV) Testing & Quality Assurance:-

> Identifying and fixing bugs before deployment.

V) Deployment:-

> Deploying the software in a production environment.
> Performing final testing and system monitoring.

VI) Maintenance & Updates:-

> Providing bug fixes, security patches, and performance optimizations.
> Ensuring long-term compatibility with evolving technologies.

18:) Why is the requirement analysis phase critical in software development?

ANS)
IN Software Development Life Cycle (SDLC) it lays the foundation for the entire project. Mistakes in this phase can lead to costly rework, project delays, or even failure.

19:)What is the role of software analysis in the development process?

ANS)
Software analysis is a crucial step in the Software Development Life Cycle (SDLC) as it helps in understanding, planning, and defining the project requirements.

20:) What are the key elements of system design?

ANS) System design is a crucial phase in software development that defines the architecture, components, and data flow of a system.

KEY ELEMENTS:-

I) Architectural Design
II) Software Components Design
III) Database Design
IV) Data Flow & Process Design
V) Security Design

21:) Why is software testing important? ?

ANS:) Software testing is important because it helps ensure that software is secure, reliable, and performs well. It also helps to improve customer satisfaction and save money.

22:)What types of software maintenance are there ?

ANS) Types of Software Maintenance

I) Corrective Maintenance (Bug Fixing)
Fixes bugs, errors, and defects discovered after software release.

II) Adaptive Maintenance (System Updates & Compatibility)
Modifies software to work with new hardware, operating systems, or third-party services.

III) Perfective Maintenance (Performance & Feature Enhancements)
Improves performance, user experience, and efficiency.

IV) Preventive Maintenance (Future-Proofing & Security)
Identifies and addresses potential risks before they cause issues.

23:)What are the advantages of using web applications over desktop applications?

ANS) The advantages of using web applications:-

I) Accessibility from Anywhere
Web apps can be accessed anywhere, anytime using a web browser.

II)Cross-Platform Compatibility
Works on Windows, macOS, Linux, iOS, and Android without separate versions.

III) No Installation or Updates Required
Users don't need to download or update software manually.

IV) Easier Collaboration & Sharing
Enables real-time collaboration, file sharing, and communication.

V) Saves Storage Space
Web apps store data in the cloud, reducing the need for local storage.

24:) What role does UI/UX design play in application development?

ANS)

UI (User Interface) and UX (User Experience) design are critical in creating applications that are intuitive, visually appealing, and user-friendly. A well-designed UI/UX ensures users can efficiently navigate, interact, and achieve their goals within the application.

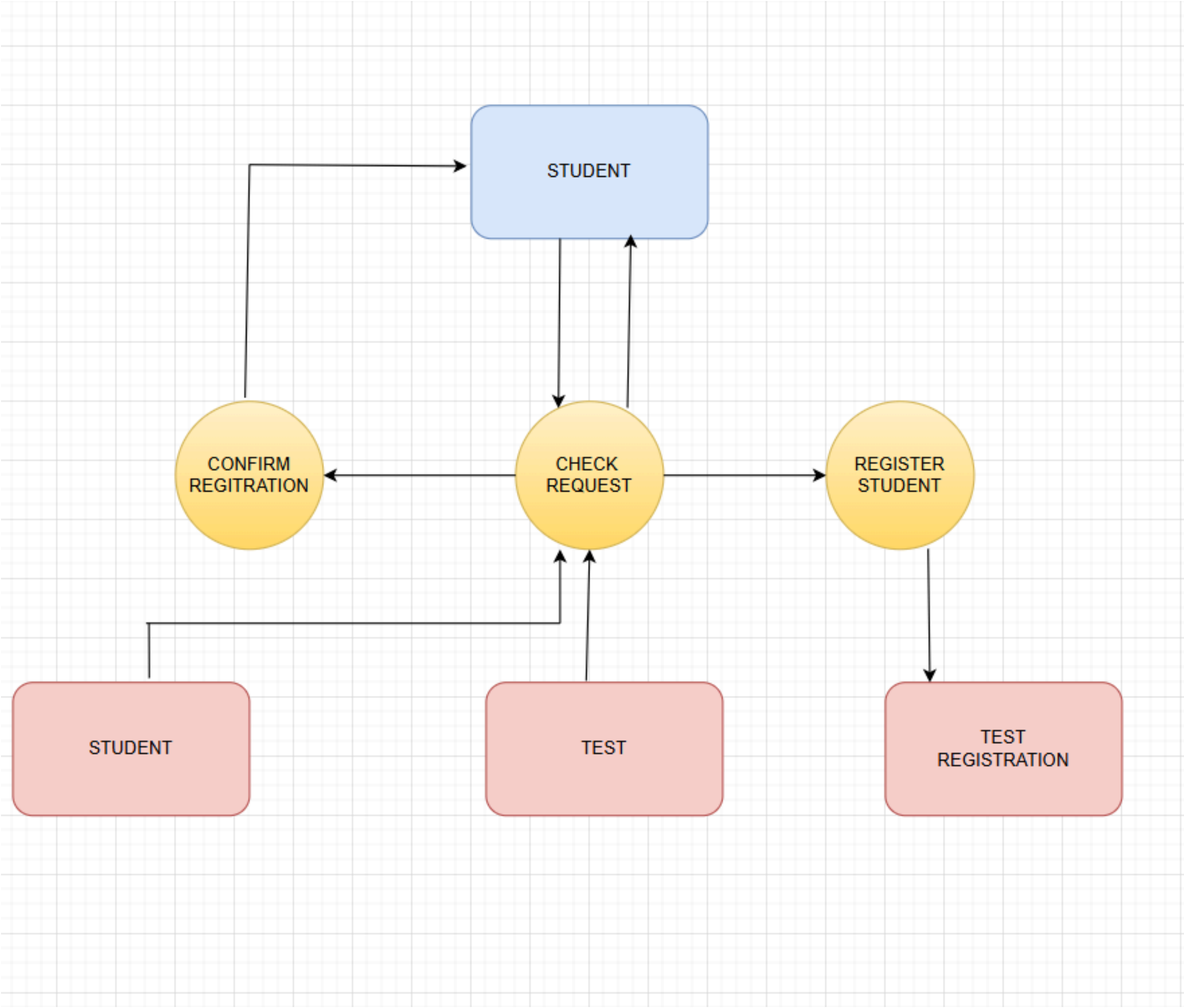
KEY ELEMENTS:-

I) Consistency
II)Simplicity
III)Responsiveness
IV)Accessibility

25:) What is the significance of DFDs in system analysis?

ANS) A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system. It illustrates how data is processed, stored, and transferred between different components of the system. DFDs help in understanding, analyzing, and designing the structure of a system in a clear and logical manner.

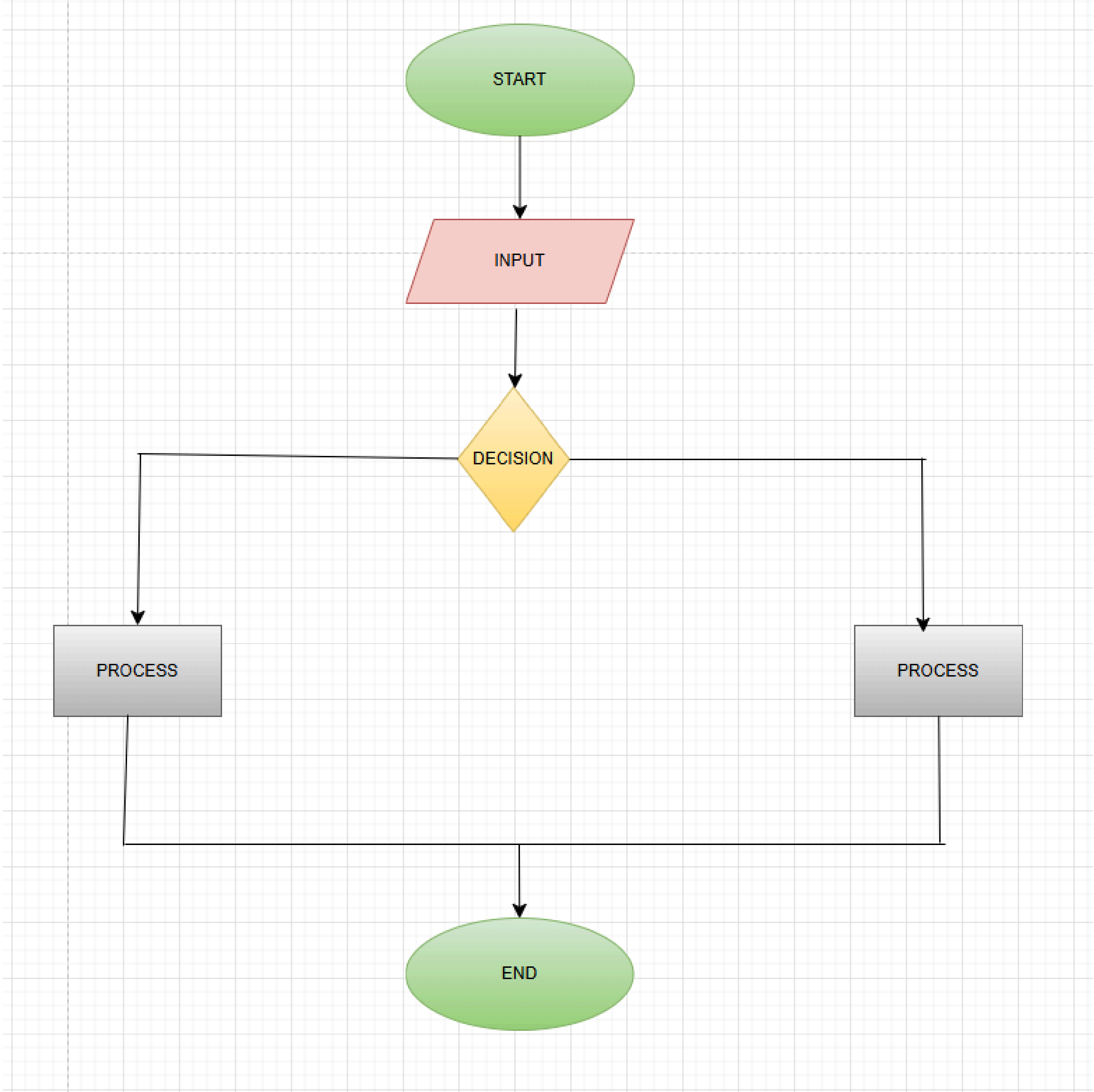
Example fo DFD ➖



26:) How do flowcharts help in programming and system design?

ANS) The flowcharts are simple visual tools that help us understand and represent processes very easily. They use shapes like arrows, rectangles, and diamonds to show steps and decisions clearly. If someone is making a project or explaining a complex task, flowcharts can make complex ideas easier to understand.

FLOW CHART EXAMPLE:-



27:)What is the difference between source code and machine code?

ANS)Source Code:

Definition: Source code is the human-readable instructions written by programmers using a high-level or assembly programming language.
Readability: It is written in a structured, understandable format with keywords, variables, and functions

Machine Code:

Definition: Machine code is the binary code (0s and 1s) that a computer's CPU can directly execute.

Readability: It is not human-readable and consists of numeric instructions specific to the processor architecture.

28:) How does broadband differ from fiber-optic internet?

ANS)

Broadband is a general term that refers to high-speed internet access that is always on and faster than traditional dial-up. It includes multiple types of internet connections, such as DSL, cable, satellite, and fiber-optic.

29:)What are the differences between open-source and proprietary software ?

ANS) The differences between open-source and proprietary software:-

I) Source Code Access

Open-Source Software (OSS): The source code is freely available for anyone to view, modify, and distribute.
Proprietary Software: The source code is closed and controlled by the developer or company, meaning users cannot modify or access it.

II) Licensing

Open-Source: Distributed under licenses like GPL, MIT, or Apache, which allow modification and redistribution.
Proprietary: Requires users to purchase a license or agree to strict terms of use; modifications and redistribution are prohibited.

III)Cost

Open-Source: Often free to use, though some versions may have paid support or premium features.
Proprietary: Typically requires payment (one-time, subscription, or per-user licensing fees).

IV) Customization & Flexibility

Open-Source: Highly customizable, allowing users to tweak the software to fit their needs.
Proprietary: Limited customization, as users must rely on the developer for updates or feature changes.

V) Examples
Open-Source: Linux, Apache, LibreOffice, Mozilla Firefox, WordPress.
Proprietary: Windows, Microsoft Office, Adobe Photoshop, macOS.

30) What are the key differences between web and desktop applications ?

ANS) The key differences between web and desktop applications

I)Deployment & Accessibility

Web Applications: Run in a web browser (e.g., Chrome, Firefox) and require an internet connection.
Desktop Applications: Installed and run directly on a computer’s operating system (Windows, macOS, Linux).

II)Internet Dependency

Web Apps: Typically require an internet connection, although some offer offline functionality.
Desktop Apps: Can work entirely offline, depending on the application.

III)Performance

Web Apps: Dependent on browser and internet speed; may be slower for resource-intensive tasks.
Desktop Apps: Generally faster since they use the device’s full processing power.

IV)Updates & Maintenance

Web Apps: Updates are done server-side, so users always have the latest version.
Desktop Apps: Users may need to manually update or install new versions.

V) Examples
Web Apps: Gmail, Google Docs, Facebook, Trello.
Desktop Apps: Microsoft Word, Adobe Photoshop, VLC Media Player.

31) What are the differences between native and hybrid mobile apps?

ANS) The differences between native and hybrid mobile apps:-

I) Development Approach:-

Native Apps: Built specifically for a single platform using platform-specific programming languages.
Hybrid Apps: Developed using web technologies and wrapped in a native container.

II)User Experience (UX)

Native Apps: Offer a seamless and highly responsive UI/UX that aligns with platform guidelines.
Hybrid Apps: May have minor UI/UX inconsistencies due to cross-platform compatibility issues

III)Platform Compatibility

Native Apps: Limited to the platform they are built for (iOS or Android).
Hybrid Apps: Work on both iOS and Android with minimal changes.

IV)Updates & Maintenance

Native Apps: Requires separate updates for each platform.
Hybrid Apps: Easier and faster updates since a single codebase is used.

V) Examples
Native Apps: Instagram (iOS & Android versions built separately), WhatsApp.
Hybrid Apps: Twitter, Uber, Instagram (some parts), and Airbnb.

32) What are the pros and cons of desktop applications compared to webapplications ?

ANS) The pros and cons of desktop applications compared to webapplications

Pros:

> Better Performance – Utilizes the full power of the device's hardware, making it faster for intensive tasks.
> Offline Access – Works without an internet connection.
> Enhanced Security – Data is stored locally, reducing exposure to online threats.
> Better Integration – Can fully integrate with system resources (e.g., file system, GPU, peripherals).
> More Stable – Less dependent on server uptime and internet speed.

Cons:

> Platform-Specific – Often requires different versions for Windows, macOS, and Linux.
> Manual Updates – Users must download and install updates.
> Limited Remote Access – Cannot be accessed from multiple devices unless cloud synchronization is available.
> Installation Required – Takes up storage space and may require admin permissions to install.