1. What is different between web server and application server?

Web Server	Application Server
Handles HTTP requests and serves static content.	Manages business logic and serves dynamic content.
Static content like HTML, CSS, JS, images.	Dynamic content like database-driven responses.
Focuses on delivering web pages to users.	Provides services for running business logic and APIs.
Apache HTTP Server, Nginx, Microsoft IIS.	Tomcat, JBoss, WebLogic, WebSphere.
Limited to web-specific protocols (HTML, HTTP).	Supports multiple languages and frameworks (Java, .NET).
Primarily HTTP, HTTPS.	HTTP, HTTPS, RMI, JMS, SOAP, REST, etc.
Simpler and faster.	More complex, providing additional services.
Serves as a gateway between user and web content.	Handles application logic, transactions, and communication.
Limited or no session management.	Provides session management and transaction support.
Limited integration with backend services.	Integrates with databases, messaging, and enterprise services.

2. Difference between GET and POST method:

GET	POST
Used to request data from a server.	Used to send data to a server to create/update resources.
Data is appended to the URL and visible in the browser's address bar.	Data is included in the request body and not visible in the URL.
Limited by the URL length (varies by browser).	No restrictions on data length (within server limits).
Can be cached by browsers.	Not cached by default.
Less secure, as data is exposed in the URL.	More secure for sensitive data like passwords, as it's in the request body.
Idempotent (same request will not change server state).	Not necessarily idempotent.

3. What is MIME Type?

MIME (Multipurpose Internet Mail Extensions) Type is a standard way to indicate the nature and format of a file or data transmitted over the internet. It helps the browser or client understand how to process the content.

Example MIME Types:

- text/html - HTML document
- application/json - JSON data
- image/png - PNG image
- application/pdf - PDF file

4. What is a web application and its directory structure?

Web Application: A web application is software that runs on a web server and is accessed through a web browser. Examples include e-commerce websites, online banking systems, and social networking sites.

Directory Structure: A standard web application directory structure might look like this:

MyWebApp

— WEB-INF/

– web.xml (Deployment descriptor file)

– classes/ (Compiled servlet classes)

– lib/ (JAR files for dependencies)

– META-INF/ (Meta-information for the application)

– index.html (Static content like HTML)

— css/ (Stylesheets)

— js/ (JavaScript files)

— images/ (Images used in the application)

5. What is a servlet?

A servlet is a Java program that runs on a server and handles requests and responses for web-based applications. It is a part of Java EE and is used to create dynamic web content, such as data processing and database interaction.

6. Advantages of Servlet over CGI:

Servlet	CGI (Common Gateway Interface)
Better performance (runs within the JVM).	Slower, as each request spawns a new process.
Uses a single process and handles multiple requests with threads.	No multithreading; creates a new process for each request.
Platform-independent (Java-based).	Depends on server configuration and scripts.
Code can be reused and maintained easily.	Less modular and harder to maintain.
Provides robust security features like HTTPS and session management.	Limited security capabilities.

7. Common tasks performed by Servlet Container:

A Servlet Container (e.g., Apache Tomcat) is responsible for:

1. Lifecycle Management: Loading, initializing, and destroying servlets.
2. Request Handling: Receiving HTTP requests and sending them to the appropriate servlet.
3. Thread Management: Creating and managing threads for concurrent request handling.
4. Session Management: Managing user sessions through cookies or URL rewriting.
5. Security Enforcement: Handling authentication and authorization for web applications.
6. Resource Management: Managing resources like database connections, file systems, and memory.
7. Deployment: Supporting the deployment of web applications packaged as WAR files.

8. What is ServletConfig object?

The ServletConfig object is used for passing initialization parameters to the servlet. It provides configuration information specific to a single servlet. It is created by the container during servlet initialization.

9. What is ServletContext object?

ServletContext object is used for sharing information among servlets in a web application. It represents the web application environment and provides access to resources and application-wide parameters.

10. What is Request Dispatcher?

The RequestDispatcher is used to forward a request to another resource (like a servlet, JSP, or HTML) or include the content of another resource in the response. It can be obtained using getRequestDispatcher() or getNamedDispatcher() methods.

11. What is PrintWriter?

PrintWriter is a class intended for character-based output to the client. It is retrieved from the HttpServletResponse object using the getWriter() method and is used to write either HTML or text responses.

12. How do we call one servlet from another servlet?

One servlet can invoke another servlet by using RequestDispatcher to forward the request or include the response or by sending an HTTP redirect to the other servlet.

13. How can we invoke another servlet in a different application?

You can invoke a servlet in a different application using an HTTP request via HttpURLConnection or a web service call since RequestDispatcher cannot be used across applications.

14. Why is HttpServlet declared as an abstract class?

The HttpServlet class is declared abstract because it provides default implementations for HTTP methods such as doGet(), doPost(), etc., which are typically overridden by developers to provide their own custom behavior. It ensures that the class cannot be instantiated directly but will be used as a base class.

15. What are life cycle methods of a servlet?

Life cycle methods of a servlet are:

1. init(): Initializes the servlet on its first load.

2. service(): Takes the requests from the clients and creates responses.

3. destroy(): Cleans up resources before servlets are destroyed.

16. What are different methods of session management in servlets?

1. Cookies: Stores session information on the client side as small text files.

2. URL Rewriting: Appends session ID to the URL as a query string.

3. HTTPSession: Stores session information on the server side.

4. Hidden Form Fields: It embeds session data in the hidden fields of HTML forms.

17. What is URL Rewriting?

URL rewriting is a technique for managing sessions, in which session ID is appended to the URL as a query string . This keeps session state alive when cookies are disabled.

18. How does Cookies work in Servlets?

Cookies are small text files sent by the server to the client's browser. The browser stores these cookies and sends them back to the server with subsequent requests. This allows the server to identify users and maintain session states.

19. How to notify an object in session when the session is invalidated or timed-out?

In the object, implement the HttpSessionBindingListener or HttpSessionActivationListener interface. The valueUnbound() method of the HttpSessionBindingListener will be called when the session is invalidated or the object is removed from the session.

The HttpServlet class is declared abstract because it provides default implementations for HTTP methods such as doGet(), doPost(), etc., which are typically overridden by developers to provide their own custom behavior. It ensures that the class cannot be instantiated directly but will be used as a base class.

20. what is deployment descriptor?

Deployment Descriptor: A Blueprint for Application Deployment. A deployment descriptor is essentially a configuration file that outlines how an application should be deployed to a specific container or environment. It provides crucial information to the deployment engine, enabling it to understand the application's structure, dependencies, and configuration settings.

21.How to get the actual actual path of the servlet in server?

The actual path of a servlet within a server, it's recommended to use the ServletContext object and relative paths. This approach is more portable and reliable, especially when dealing with WAR file deployments. Avoid using getRealPath(), as it might not work consistently across different environments.

22.How to get the server information in a servlet?

To obtain server information within a servlet, you can leverage the ServletContext object. This object provides access to various information about the servlet's environment, including the server name and version.

23.What is MVC?

MVC (Model-View-Controller) is a software design pattern that separates an application into three interconnected components:

Model: Represents the data and business logic of the application. It manages the data, performs calculations, and interacts with the database.

View: Represents the user interface. It displays the data to the user and allows them to interact with the application.

Controller: Handles user input and updates the model and view accordingly. It receives user requests, processes them, and determines which view to display.

This separation of concerns makes applications more modular, maintainable, and testable. It also promotes code reusability and allows for easier modification and updates.

24. What is the diff between doget() and dopost()?

doGet() and doPost() are two HTTP methods used in servlets to handle client requests. They differ in how they transmit data and their intended use:

doGet():

* Data Transmission: Data is appended to the URL in the form of query parameters.

* Idempotency: Safe and idempotent, meaning repeated requests with the same parameters have the same effect.

* Use Cases: Retrieving data, performing searches, or loading static content.

doPost():

* Data Transmission: Data is sent in the request body, hidden from the URL.

* Side Effects: Not idempotent, meaning repeated requests can have different effects, such as updating data or submitting forms.

* Use Cases: Submitting forms, uploading files, or performing actions that modify server-side state.

25.How do servlets differ from regular Java classes?

* Deployment Descriptor

* Getting Server Information

* MVC Pattern

* doGet() vs doPost()

* Servlets vs Regular Java Classes