1. **What is Multer?**
   - A) A middleware for handling JSON data
   - B) A middleware for handling file uploads
   - C) A database management system
   - D) A testing framework

   Answer: B

   Explanation: Multer is specifically designed for handling `multipart/form-data`, which is used for file uploads in web applications.
2. **Which method is used to configure Multer in an Express application?**
   - A) multer.init()
   - B) multer.configure()
   - C) multer()
   - D) multer.set()

   Answer: C

   Explanation: The `multer()` function initializes Multer middleware with the specified storage and file handling options.
3. **What type of data does Multer handle?**
   - A) application/json
   - B) application/xml
   - C) multipart/form-data
   - D) text/plain

   Answer: C

   Explanation: Multer is designed to handle `multipart/form-data`, which is primarily used for file uploads.
4. **Which of the following is a valid option for the `storage` property in Multer?**
   - A) memoryStorage
   - B) fileStorage
   - C) dataStorage
   - D) cloudStorage

   Answer: A

   Explanation: `memoryStorage` is a valid option that stores files in memory as Buffer objects.
5. **What does `upload.single('file')` do in a route?**
   - A) Uploads multiple files
   - B) Uploads a single file
   - C) Downloads a file
   - D) Deletes a file

   Answer: B

   Explanation: The `upload.single('file')` method is used to handle a single file upload where `'file'` is the name of the file input field.
6. **Which of the following is the default storage engine for Multer?**
   - A) Cloud storage

- B) File system storage
- C) Memory storage
- D) No storage engine

  Answer: C

  Explanation: The default storage engine for Multer is `memoryStorage`, which stores uploaded files in memory.

7. **How do you specify the destination directory for uploaded files in disk storage?**
   - A) `destination: './uploads'`
   - B) `destination: 'uploads/'`
   - C) `dest: './uploads'`
   - D) `dest: 'uploads/'`

     Answer: B

     Explanation: The correct way to specify the destination is using a callback in the `destination` property of disk storage.
8. **What is the purpose of the `filename` property in the Multer storage configuration?**
   - A) To define the path of the file
   - B) To set the file name upon upload
   - C) To set the file size limit
   - D) To specify the file type

     Answer: B

     Explanation: The `filename` property is used to set the name of the uploaded file, often ensuring uniqueness.
9. **Which of the following is a limitation of using memory storage with Multer?**
   - A) Limited file types
   - B) Cannot handle large files
   - C) Slow performance
   - D) Requires disk space

     Answer: B

     Explanation: Memory storage is not suitable for large file uploads as it can lead to high memory usage.
10. **How do you access the uploaded file in the request object?**
    - A) `req.file`
    - B) `req.files`
    - C) `req.body.file`
    - D) `req.uploadedFiles`

      Answer: A

      Explanation: The uploaded file is accessible via `req.file` when using `upload.single()`.
11. **What do you need to do before using the `uploads` directory for storing files?**
    - A) Create the directory
    - B) Set permissions

- C) Add files manually
- D) Configure server settings

  Answer: A

  Explanation: You need to create the `uploads` directory if it does not exist, or else Multer will throw an error.

12. **How can you ensure that files are uniquely named when uploaded?**
    - A) Use a timestamp in the filename
    - B) Append the original name
    - C) Ignore renaming
    - D) Use a random number

      Answer: A

      Explanation: Using a timestamp or a unique identifier in the filename helps prevent naming collisions.
13. **What happens if you do not specify a storage option in Multer?**
    - A) An error occurs
    - B) The default memory storage is used
    - C) No files can be uploaded
    - D) Files are stored in a temporary folder

      Answer: B

      Explanation: If no storage option is specified, Multer defaults to using memory storage.
14. **Which of the following is a method to limit file size using Multer?**
    - A) maxFileSize
    - B) limits: { fileSize: ... }
    - C) fileLimit
    - D) sizeLimit

      Answer: B

      Explanation: You can limit the file size by specifying a `limits` object in the Multer configuration.
15. **What is the purpose of `resave` in the session middleware?**
    - A) To always save the session
    - B) To update the session data
    - C) To create a new session
    - D) To prevent saving the session

      Answer: A

      Explanation: The `resave` option, when set to true, forces the session to be saved back to the session store even if it was never modified during the request.
16. **What is the role of `saveUninitialized` in the session middleware?**
    - A) To prevent saving new sessions
    - B) To save only modified sessions
    - C) To save uninitialized sessions
    - D) To reset session data

      Answer: C

Explanation: When `saveUninitialized` is true, it saves uninitialized sessions to the store.

17. **How can you restrict file types when using Multer?**
    - A) Using the `limits` option
    - B) Using the `fileFilter` function
    - C) By renaming the files
    - D) It's not possible

    Answer: B

    Explanation: You can use the `fileFilter` function to control which file types are accepted.
18. **Which of the following file types can be uploaded using Multer by default?**
    - A) Only images
    - B) Any file type
    - C) Only text files
    - D) Only JSON files

    Answer: B

    Explanation: Multer can handle any file type unless restricted by a file filter.
19. **In the Multer configuration, how can you handle errors during file uploads?**
    - A) Using a callback function
    - B) Throwing exceptions
    - C) Setting a global error handler
    - D) It's not possible

    Answer: A

    Explanation: You can handle errors by using a callback function in the Multer middleware.
20. **What is the purpose of `path.extname(file.originalname)` in the filename function?**
    - A) To get the file path
    - B) To extract the file extension
    - C) To rename the file
    - D) To validate the file type

    Answer: B

    Explanation: `path.extname()` extracts the extension from the original filename, allowing you to preserve the file type.
21. **Which middleware is commonly used with Multer for processing file uploads?**
    - A) body-parser
    - B) cookie-parser
    - C) express-session
    - D) morgan

    Answer: A

    Explanation: `body-parser` is often used to parse incoming request bodies, though as of Express 4.16.0, it has been integrated into Express itself.
22. **What should you do to secure file uploads in a production application?**

- A) Limit file types and sizes
- B) Store files in public directories
- C) Allow any file upload
- D) Do nothing, it's secure by default

  Answer: A

  Explanation: Limiting file types and sizes helps protect against malicious uploads and ensures system integrity.

23. **What is a common way to structure your project for file uploads with Multer?**
    - A) All files in the root directory
    - B) Separate directories for uploads and routes
    - C) No specific structure needed
    - D) Store uploads in the `node_modules` directory

      Answer: B

      Explanation: Organizing uploads and routes into separate directories promotes maintainability and clarity.

24. **How can you delete uploaded files after processing?**
    - A) Using fs.unlink()
    - B) It's not possible
    - C) Automatically handled by Multer
    - D) By renaming files

      Answer: A

      Explanation: You can use the `fs.unlink()` method to delete files from the server after they have been processed.

25. **Which of the following tools can be used for testing file uploads?**
    - A) Postman
    - B) JUnit
    - C) Mocha
    - D) Selenium

      Answer: A

      Explanation: Postman is a popular tool for testing APIs, including file uploads.

26. **In a production environment, where should uploaded files typically be stored?**
    - A) On the same server
    - B) In cloud storage (e.g., AWS S3)
    - C) In the application directory
    - D) In memory

      Answer: B

      Explanation: Using cloud storage provides scalability and reliability for handling file uploads.

27. **What is the best way to ensure user privacy when handling file uploads?**
    - A) Use public directories
    - B) Avoid validation
    - C) Implement access controls and encryption
    - D) Store files in the root directory

      Answer: C

Explanation: Implementing access controls and encryption helps protect sensitive user data.

28. **How can you handle multiple file uploads using Multer?**
    - A) Use upload.single('file')
    - B) Use upload.array('files', maxCount)
    - C) Use upload.multi('files')
    - D) It's not possible

    Answer: B

    Explanation: The `upload.array('files', maxCount)` method allows for multiple file uploads, where `maxCount` specifies the maximum number of files.
29. **What type of function should you provide to validate uploaded files?**
    - A) A synchronous function
    - B) An asynchronous function
    - C) A callback function
    - D) A promise

    Answer: C

    Explanation: You should provide a callback function for file validation in the `fileFilter` option.
30. **Which storage option is best for applications requiring temporary file uploads?**
    - A) memoryStorage
    - B) diskStorage
    - C) cloudStorage
    - D) No specific storage needed

    Answer: A

    Explanation: `memoryStorage` is suitable for temporary file uploads due to its fast access speed.
31. **What type of error handling is recommended for Multer file uploads?**
    - A) Global error handler
    - B) Inline error handling
    - C) Asynchronous error handling
    - D) No error handling

    Answer: A

    Explanation: Using a global error handler can help manage errors consistently across your application.
32. **What is a common approach to logging file upload errors?**
    - A) Console.log()
    - B) Using a logging library (e.g., Winston)
    - C) Ignoring errors
    - D) Emailing errors to admin

    Answer: B

    Explanation: Using a logging library provides better control over logging levels and destinations.
33. **What should you do if an upload exceeds the maximum file size?**
    - A) Ignore the error

- B) Send an error response to the client
- C) Automatically resize the file
- D) Store the file anyway

  Answer: B

  Explanation: It's important to notify the client when an upload exceeds the allowed size.

34. **Which option in the Multer configuration can you use to restrict file types?**
    - A) limits
    - B) fileFilter
    - C) maxFileSize
    - D) allowedTypes

      Answer: B

      Explanation: The `fileFilter` option allows you to specify which file types are accepted.
35. **If a user tries to upload a file with an unsupported type, what should your application do?**
    - A) Accept the file anyway
    - B) Reject the file and respond with an error message
    - C) Rename the file
    - D) Log the attempt

      Answer: B

      Explanation: Your application should provide feedback and reject unsupported file types.
36. **Which response code should be used for a successful file upload?**
    - A) 200
    - B) 201
    - C) 400
    - D) 500

      Answer: B

      Explanation: A response code of 201 indicates that a resource has been successfully created, which is suitable for file uploads.
37. **What middleware is often used in conjunction with Multer for additional request handling?**
    - A) cors
    - B) body-parser
    - C) helmet
    - D) morgan

      Answer: B

      Explanation: `body-parser` middleware is commonly used to parse incoming request bodies.
38. **How can you ensure the security of uploaded files on your server?**
    - A) Store files in a publicly accessible directory
    - B) Perform virus scanning on files
    - C) Allow any file type

- D) Ignore security measures
  Answer: B
  Explanation: Implementing virus scanning helps protect against malicious files being uploaded.

39. **Which of the following can be a potential security risk with file uploads?**
    - A) File size limits
    - B) Executable files
    - C) Image files
    - D) Text files
      Answer: B
      Explanation: Executable files pose a significant security risk if allowed to be uploaded.
40. **What is a common strategy for organizing file uploads in an application?**
    - A) All uploads in one folder
    - B) Organizing uploads by user or date
    - C) No specific organization
    - D) Storing in memory
      Answer: B
      Explanation: Organizing uploads by user or date helps manage files more effectively.
41. **How can you provide feedback to the user after a successful file upload?**
    - A) Redirect to another page
    - B) Display a success message
    - C) Log the success
    - D) Do nothing
      Answer: B
      Explanation: Providing a success message improves user experience by confirming the upload.
42. **Which package can be used with Multer to handle file storage in a cloud environment?**
    - A) multer-s3
    - B) multer-cloud
    - C) multer-file
    - D) multer-ftp
      Answer: A
      Explanation: The `multer-s3` package integrates Multer with AWS S3 for file uploads.
43. **What is a benefit of using a cloud storage solution for file uploads?**
    - A) Limited storage capacity
    - B) Easy scalability
    - C) Increased server load
    - D) Complexity in setup
      Answer: B

Explanation: Cloud storage solutions allow for easy scaling as storage needs grow.

44. **In a production application, which option should be set to true in the Multer configuration for security?**
    - A) resave
    - B) secure
    - C) preservePath
    - D) isValid

    Answer: B

    Explanation: Setting secure options helps protect against certain attacks, especially in production environments.
45. **What is the use of `multer-gridfs-storage`?**
    - A) To store files in memory
    - B) To store files in a grid file system using MongoDB
    - C) To store files in a local directory
    - D) To upload files to an FTP server

    Answer: B

    Explanation: `multer-gridfs-storage` allows you to store files directly into MongoDB's GridFS.
46. **How can you manage file versioning during uploads?**
    - A) Overwrite existing files
    - B) Append a version number to filenames
    - C) Ignore versioning
    - D) Use the same filename

    Answer: B

    Explanation: Appending a version number helps keep track of different file versions.
47. **Which HTTP method is typically used for file uploads?**
    - A) GET
    - B) POST
    - C) PUT
    - D) DELETE

    Answer: B

    Explanation: The POST method is used for submitting data to be processed, including file uploads.
48. **What is the primary purpose of the `fileFilter` function in Multer?**
    - A) To set file size limits
    - B) To determine which files to accept or reject
    - C) To rename files
    - D) To set the upload directory

    Answer: B

    Explanation: The `fileFilter` function allows you to control which files are accepted during the upload process.
49. **How can you limit the number of files a user can upload using Multer?**

- A) Set a limit in the server configuration
- B) Use the `limits` option in Multer
- C) Implement client-side restrictions
- D) It's not possible

  Answer: B

  Explanation: You can limit the number of files by using the `limits` property in the Multer configuration.

50. **What happens when a user tries to upload a file that exceeds the allowed file size?**
    - A) The file is accepted but ignored
    - B) An error is thrown
    - C) The file is automatically resized
    - D) The application crashes

      Answer: B

      Explanation: When the file size exceeds the limit, Multer will throw an error, which should be handled appropriately.
51. **Which method can be used to clear uploaded files from a temporary storage?**
    - A) fs.delete()
    - B) fs.unlink()
    - C) fs.clear()
    - D) fs.remove()

      Answer: B

      Explanation: `fs.unlink()` is used to delete files from the filesystem.
52. **What is a common way to handle file upload progress in a client application?**
    - A) Using XMLHttpRequest
    - B) Using fetch
    - C) Using WebSocket
    - D) It's not possible

      Answer: A

      Explanation: You can use XMLHttpRequest to track the progress of file uploads.
53. **What is the purpose of setting `storage` to `diskStorage` in Multer?**
    - A) To handle files in memory
    - B) To store files on the disk
    - C) To process files in real-time
    - D) To compress files

      Answer: B

      Explanation: Setting `storage` to `diskStorage` allows files to be stored directly on the server's filesystem.
54. **How can you set specific file names when using disk storage in Multer?**
    - A) Use a timestamp in the filename function
    - B) Ignore the filename option
    - C) Store the original filename
    - D) Use random strings

      Answer: A

Explanation: You can modify the filename in the `filename` function to include timestamps or unique identifiers.

55. **What type of validation can be implemented in the `fileFilter`?**
    - A) Type validation
    - B) Size validation
    - C) Custom validation logic
    - D) All of the above

    Answer: D

    Explanation: The `fileFilter` function can implement various types of validation, including type, size, and custom rules.
56. **Which of the following is true about Multer's `memoryStorage`?**
    - A) Files are stored on the disk
    - B) Files are stored in RAM
    - C) Files cannot be uploaded
    - D) Files are automatically deleted

    Answer: B

    Explanation: `memoryStorage` stores files in RAM, which is fast but not persistent.
57. **Which command is used to install Multer in a Node.js application?**
    - A) npm install multer
    - B) npm install express
    - C) npm install file-upload
    - D) npm install body-parser

    Answer: A

    Explanation: `npm install multer` is the command to install Multer in your Node.js project.
58. **How do you specify a maximum file size limit in Multer?**
    - A) In the fileFilter function
    - B) By setting maxFileSize in the storage options
    - C) By using the limits property
    - D) It's not possible

    Answer: C

    Explanation: You can set a maximum file size by specifying the `limits` property in Multer's configuration.
59. **Which response should be sent back to the client after a successful upload?**
    - A) 404 Not Found
    - B) 200 OK
    - C) 201 Created
    - D) 500 Internal Server Error

    Answer: C

    Explanation: A 201 Created response indicates that the file was successfully uploaded and a resource was created.
60. **What is a common practice for naming uploaded files?**

- A) Use random characters
- B) Use the original name without changes
- C) Combine user ID and timestamp
- D) Use only file extensions

  Answer: C

  Explanation: Combining user ID and timestamp ensures uniqueness and prevents overwriting files.

61. **Which application could benefit from file uploads using Multer?**
    - A) A chat application
    - B) A blogging platform with image uploads
    - C) A static website
    - D) A calculator app

      Answer: B

      Explanation: A blogging platform often requires users to upload images and files, making Multer suitable.
62. **In a profile picture upload feature, how can you ensure that only image files are uploaded?**
    - A) By limiting the size of uploads
    - B) By implementing a file type filter
    - C) By renaming files
    - D) By ignoring file types

      Answer: B

      Explanation: Implementing a file type filter ensures that only valid image files are accepted.
63. **How can an application handle large file uploads effectively?**
    - A) Use memory storage
    - B) Implement chunked uploads
    - C) Ignore file size limits
    - D) Always reject large files

      Answer: B

      Explanation: Implementing chunked uploads allows users to upload large files without hitting size limits.
64. **What is a common reason for implementing file uploads in a web application?**
    - A) To create static pages
    - B) To allow user-generated content
    - C) To send emails
    - D) To fetch data from a database

      Answer: B

      Explanation: File uploads enable users to contribute content, such as images or documents, to the application.
65. **Which of the following is a recommended practice when storing uploaded files?**
    - A) Store them in the application root directory
    - B) Use unique names to avoid collisions
    - C) Allow public access to all uploads

- D) Store in the `node_modules` directory
  Answer: B
  Explanation: Using unique names for uploaded files helps avoid collisions and ensures data integrity.

66. **How can you enhance performance when uploading large files?**
    - A) Compress files before upload
    - B) Always reject large files
    - C) Store them in memory
    - D) Use slow network connections
      Answer: A
      Explanation: Compressing files can significantly reduce upload times and improve performance.
67. **Which of the following describes a use case for Multer in an e-commerce application?**
    - A) Product reviews
    - B) User profile updates with images
    - C) Payment processing
    - D) Inventory management
      Answer: B
      Explanation: Users often upload images when updating their profiles in e-commerce applications.
68. **What is a common way to notify users of successful file uploads?**
    - A) Silent success
    - B) Using alerts or notifications
    - C) Redirecting without feedback
    - D) No feedback needed
      Answer: B
      Explanation: Using alerts or notifications helps users understand that their file upload was successful.
69. **Which approach can be taken to handle sensitive data in uploaded files?**
    - A) Ignore security
    - B) Encrypt files before upload
    - C) Allow any file type
    - D) Store in public folders
      Answer: B
      Explanation: Encrypting sensitive files helps protect user data during uploads.
70. **In a social media application, what kind of files might users upload?**
    - A) Only text files
    - B) Images, videos, and documents
    - C) Only images
    - D) Only audio files
      Answer: B
      Explanation: Social media applications typically support various file types, including images, videos, and documents.

71. **Why is it important to validate file uploads?**
    - A) To make uploads more complex
    - B) To ensure data integrity and security
    - C) To increase server load
    - D) To ignore user input

    Answer: B

    Explanation: Validating file uploads helps protect against malicious files and ensures that only acceptable content is processed.
72. **What is the role of middleware in Express applications?**
    - A) To enhance performance
    - B) To modify request and response objects
    - C) To block all requests
    - D) To create routes

    Answer: B

    Explanation: Middleware functions in Express can modify the request and response objects and are used for various tasks, including handling file uploads.
73. **How can you ensure that file uploads do not affect application performance?**
    - A) Limit file sizes and types
    - B) Allow unlimited uploads
    - C) Store all files in memory
    - D) Avoid validations

    Answer: A

    Explanation: Limiting file sizes and types helps prevent performance issues caused by excessive resource usage.
74. **What happens if you configure Multer to accept files with the wrong MIME type?**
    - A) The files are accepted
    - B) An error is thrown
    - C) The application crashes
    - D) The files are renamed

    Answer: B

    Explanation: If a file with an unsupported MIME type is uploaded, Multer will throw an error if configured to filter types.
75. **Which of the following is NOT a valid file type for uploads?**
    - A) JPEG
    - B) PNG
    - C) EXE
    - D) PDF

    Answer: C

    Explanation: Executable files (EXE) are generally not allowed due to security risks.
76. **How does Multer determine the storage location for uploaded files?**
    - A) Based on the file size
    - B) Through the storage option specified in the configuration
    - C) Randomly

- D) It cannot determine the location
  Answer: B
  Explanation: Multer uses the specified storage option in the configuration to determine where to store uploaded files.

77. **In an application where users upload documents, what should be prioritized?**
    - A) Aesthetic design
    - B) Security and validation
    - C) File size
    - D) Network speed
      Answer: B
      Explanation: Security and validation are crucial when handling document uploads to prevent malicious files.
78. **What is a good practice for storing user-uploaded images?**
    - A) Store in a public directory
    - B) Use a CDN (Content Delivery Network)
    - C) Store them in the application root
    - D) Keep them in the database
      Answer: B
      Explanation: Using a CDN can improve performance and provide better access to user-uploaded images.
79. **What is the impact of allowing too large of file uploads?**
    - A) Improved user experience
    - B) Increased server load and potential crashes
    - C) No impact at all
    - D) Enhanced performance
      Answer: B
      Explanation: Allowing large file uploads can lead to increased server load, slower performance, and potential crashes.
80. **Why should file uploads be logged in an application?**
    - A) For user engagement
    - B) To monitor and track usage and errors
    - C) To clutter the application
    - D) It's unnecessary
      Answer: B
      Explanation: Logging file uploads helps monitor application usage and can assist in troubleshooting errors.
81. **What is the primary purpose of session management in web applications?**
    A) To store static data
    B) To maintain user state across requests
    C) To enhance page load speed
    D) To manage server resources
    Answer: B
    Explanation: Session management is crucial for maintaining user state across multiple requests in web applications.

82. **Which module is commonly used for session management in Node.js applications?**
    A) cookie-parser
    B) express-session
    C) body-parser
    D) multer
    Answer: B
    Explanation: `express-session` is the middleware specifically designed for managing sessions in Express applications.
83. **How are session IDs typically stored on the client side?**
    A) In local storage
    B) In cookies
    C) In the URL
    D) In session storage
    Answer: B
    Explanation: Session IDs are usually stored in cookies, allowing the server to recognize the user on subsequent requests.
84. **Which of the following is NOT a common method of storing session data?**
    A) Memory
    B) File system
    C) Database
    D) Global variable
    Answer: D
    Explanation: Using global variables is not reliable for session data storage, especially in a multi-user environment.
85. **What does `express-session` use by default to store session data?**
    A) Memory store
    B) Redis
    C) MongoDB
    D) File storage
    Answer: A
    Explanation: By default, `express-session` uses a memory store to store session data.
86. **Which option in `express-session` controls whether to save uninitialized sessions?**
    A) resave
    B) saveUninitialized
    C) secret
    D) cookie
    Answer: B
    Explanation: The `saveUninitialized` option determines whether to save uninitialized sessions to the store.

87. **What does the `resave` option in `express-session` do?**
    A) Prevents uninitialized sessions from being saved
    B) Forces session to be saved even if not modified
    C) Deletes expired sessions
    D) Allows sharing sessions across servers
    Answer: B
    Explanation: The `resave` option forces the session to be saved back to the store even if it hasn't changed.
88. **Which attribute of cookies helps to prevent cross-site scripting attacks?**
    A) secure
    B) httpOnly
    C) sameSite
    D) domain
    Answer: B
    Explanation: The `httpOnly` attribute prevents JavaScript from accessing the cookie, thus mitigating XSS attacks.
89. **What is a common technique to protect against session fixation attacks?**
    A) Regenerate the session ID on login
    B) Use a long session duration
    C) Allow multiple active sessions
    D) Store session IDs in the URL
    Answer: A
    Explanation: Regenerating the session ID upon login helps prevent session fixation.
90. **What is the effect of setting `cookie.secure` to true?**
    A) Cookies can be accessed via JavaScript
    B) Cookies will only be sent over HTTPS
    C) Cookies will expire immediately
    D) Cookies will be stored in memory
    Answer: B
    Explanation: Setting `cookie.secure` to true ensures that cookies are only sent over secure HTTPS connections.
91. **Which method is used to destroy a session in `express-session`?**
    A) req.session.destroy()
    B) req.session.clear()
    C) req.session.remove()
    D) req.session.end()
    Answer: A
    Explanation: The `req.session.destroy()` method is used to terminate a session and delete its data.
92. **What is the purpose of the `maxAge` option in cookie settings?**
    A) To set the domain of the cookie
    B) To specify how long the cookie should last
    C) To control cookie access

D) To secure the cookie
Answer: B
Explanation: The `maxAge` option determines how long the cookie is valid before it expires.

93. **What does the `sameSite` attribute on cookies help to prevent?**
A) Cross-Origin Resource Sharing (CORS)
B) Cross-Site Request Forgery (CSRF)
C) Session hijacking
D) SQL injection
Answer: B
Explanation: The `sameSite` attribute is designed to help prevent CSRF attacks by controlling when cookies are sent with cross-origin requests.
94. **What is a potential drawback of using in-memory session storage?**
A) High speed
B) Session data is lost on server restart
C) Easy to implement
D) Secure storage
Answer: B
Explanation: In-memory storage leads to loss of session data if the server restarts, which can be problematic in production environments.
95. **What type of session storage is recommended for distributed applications?**
A) File-based storage
B) In-memory storage
C) Redis or MongoDB
D) Local storage
Answer: C
Explanation: Using Redis or MongoDB allows session data to persist and be accessed across multiple instances of the application.
96. **Which session management option can enhance security by regenerating the session ID?**
A) rolling
B) resave
C) saveUninitialized
D) secret
Answer: A
Explanation: The `rolling` option can be set to true to regenerate the session ID on each request, enhancing security.
97. **How can you check if a session exists in a route handler?**
A) if (req.session) { ... }
B) if (req.sessionID) { ... }
C) if (session.exists) { ... }
D) if (req.session == null) { ... }

Answer: B

Explanation: You can verify the existence of a session by checking `req.sessionID`.

98. **Which option helps to ensure that a session cookie is only sent over secure connections?**
    A) cookie.sameSite
    B) cookie.secure
    C) cookie.maxAge
    D) cookie.domain
    Answer: B
    Explanation: Setting `cookie.secure` ensures the cookie is only transmitted over HTTPS.
99. **What does the `saveUninitialized` option do in `express-session`?**
    A) Prevents uninitialized sessions from being saved
    B) Saves new sessions that have not been modified
    C) Deletes expired sessions
    D) Allows sharing sessions across servers
    Answer: B
    Explanation: The `saveUninitialized` option determines whether new sessions that have not been modified should be saved to the store.
100. **Which of the following is a session management best practice?**
    A) Use long session durations
    B) Store sensitive data in sessions
    C) Regenerate session IDs on login
    D) Use secure cookie settings
    Answer: C
    Explanation: Regenerating session IDs upon login helps to enhance security against session hijacking.
101. **How does using a session store improve user experience?**
    A) By reducing login times
    B) By maintaining user preferences across sessions
    C) By avoiding session expirations
    D) By increasing page load speed
    Answer: B
    Explanation: A session store can maintain user preferences and session data, enhancing user experience.
102. **What happens if you do not set a `secret` for `express-session`?**
    A) The session will not work
    B) The session data will be encrypted
    C) It will use a default secret
    D) Cookies will not be sent
    Answer: A
    Explanation: The session will not function properly without a defined secret, which is essential for signing the session ID cookie.

103. **What is a potential consequence of not using a session store in production?**
A) Increased security
B) Session data loss on server restart
C) Improved performance
D) Simplified architecture
Answer: B
Explanation: Not using a session store can lead to loss of session data if the server restarts.

104. **Which session management feature can enhance scalability?**
A) Using in-memory sessions
B) Storing sessions in a distributed database
C) Ignoring session management
D) Limiting session duration
Answer: B
Explanation: Storing sessions in a distributed database like Redis enhances scalability by allowing multiple servers to access the same session data.

105. **What is the role of middleware in an Express application?**
A) To handle static files
B) To manage session state and requests
C) To connect to a database
D) To perform error handling
Answer: B
Explanation: Middleware in Express is used to manage session state, handle requests, and execute additional processing.

106. **How can you check if a session has expired?**
A) Check the session ID
B) Compare the session's last access time with the current time
C) Use a specific middleware
D) Always assume the session is active
Answer: B
Explanation: You can check if a session has expired by comparing the last access time with the current time.

107. **What is the purpose of the `cookie` option in `express-session`?**
A) To store user preferences
B) To configure how the session cookie behaves
C) To set session timeouts
D) To encrypt session data
Answer: B
Explanation: The `cookie` option is used to configure the behavior and attributes of the session cookie.

108. **How can session data be shared across subdomains?**
A) By using a different session ID
B) By setting the cookie domain to the parent domain
C) By enabling cross-origin requests

D) By using a unique storage for each subdomain

Answer: B

Explanation: Setting the cookie domain to the parent domain allows the cookie to be accessible across subdomains.

109. **What is one disadvantage of using client-side session storage?**

A) Higher performance

B) Limited storage capacity

C) Security vulnerabilities

D) Easier to implement

Answer: C

Explanation: Client-side session storage can expose session data to potential security vulnerabilities.

110. **What does the `rolling` option do in session configuration?**

A) Prevents session expiration

B) Regenerates the session ID on every request

C) Increases the duration of sessions

D) Saves uninitialized sessions

Answer: B

Explanation: The `rolling` option regenerates the session ID on each request, which enhances security.

111. **How can you prevent session hijacking?**

A) Use long session durations

B) Regenerate session IDs frequently

C) Store session data in cookies

D) Limit the number of active sessions

Answer: B

Explanation: Regenerating session IDs frequently can help prevent session hijacking.

112. **Which of the following can be stored in a session?**

A) User preferences

B) Passwords

C) Credit card information

D) All of the above

Answer: A

Explanation: User preferences are appropriate to store in a session, while sensitive information like passwords and credit card details should not be stored in sessions.

113. **What is the impact of setting `cookie.httpOnly` to true?**

A) Increases cookie size

B) Prevents client-side scripts from accessing the cookie

C) Allows cookies to be sent over insecure connections

D) Enables cross-site requests

Answer: B

Explanation: Setting `cookie.httpOnly` to true prevents client-side scripts from accessing the cookie, enhancing security.

114. **How does session storage differ from local storage?**
A) Session storage persists after the browser is closed
B) Local storage is limited to a single session
C) Session storage is temporary and limited to the session
D) Local storage cannot be accessed by scripts
Answer: C
Explanation: Session storage is temporary and only lasts for the duration of the page session, while local storage persists beyond that.

115. **What happens to a session when the user closes the browser?**
A) The session is automatically saved
B) The session is destroyed if using session storage
C) The session remains active
D) The session data is sent to the server
Answer: B
Explanation: Sessions using session storage are typically destroyed when the browser is closed.

116. **What is the purpose of the `expires` option in session cookies?**
A) To define when the session should expire
B) To specify the maximum size of the cookie
C) To indicate the domain for the cookie
D) To control whether the cookie is secure
Answer: A
Explanation: The `expires` option defines when the session cookie should expire.

117. **What is one benefit of using Redis for session management?**
A) Reduced security
B) High speed and scalability
C) Complexity in setup
D) Inability to share sessions
Answer: B
Explanation: Redis provides high speed and scalability, making it a popular choice for session management.

118. **Which method is called to initialize a session in an Express application?**
A) req.session.start()
B) req.session.init()
C) app.use(session())
D) session.create()
Answer: C
Explanation: The method `app.use(session())` initializes session management in an Express application.

119. **What does the `secure` cookie flag do?**
A) Prevents the cookie from being accessed by JavaScript
B) Ensures the cookie is sent only over HTTPS
C) Allows the cookie to be shared across domains
D) Increases cookie size

Answer: B

Explanation: The `secure` flag ensures that the cookie is only sent over secure HTTPS connections.

120. **Which storage method should be avoided for sensitive session data?**

A) Encrypted database
B) In-memory store
C) Local storage
D) Encrypted file storage

Answer: C

Explanation: Local storage is accessible via client-side scripts, making it unsuitable for storing sensitive session data.

121. **What does the `key` option in `express-session` specify?**

A) The session storage method
B) The name of the session ID cookie
C) The maximum session duration
D) The secret for signing cookies

Answer: B

Explanation: The `key` option specifies the name of the cookie used to store the session ID.

122. **How does session expiration affect user experience?**

A) It has no impact
B) It can cause users to be logged out unexpectedly
C) It enhances security
D) It increases server load

Answer: B

Explanation: Session expiration can lead to unexpected logouts, negatively affecting user experience.

123. **What should you do when a user logs out?**

A) Keep the session active
B) Destroy the session
C) Change the session ID
D) Reset the cookie

Answer: B

Explanation: Destroying the session when a user logs out is essential to prevent unauthorized access.

124. **Which database is often used for session storage in Node.js applications?**

A) SQLite
B) MongoDB
C) MySQL
D) All of the above

Answer: D

Explanation: Any of these databases can be configured for session storage in Node.js applications.

125. **What does the `maxAge` setting control in session cookies?**
A) The size of the cookie
B) The duration before the cookie expires
C) The security level of the cookie
D) The domain of the cookie
Answer: B
Explanation: The `maxAge` setting specifies how long the cookie will remain valid before it expires.

126. **How can you mitigate the risk of CSRF attacks?**
A) Use session cookies without validation
B) Implement same-site cookie attributes
C) Ignore session management
D) Allow cross-origin requests
Answer: B
Explanation: Implementing same-site cookie attributes can help mitigate the risk of CSRF attacks.

127. **What does the `uninitialized` session mean?**
A) A session that has been destroyed
B) A new session that has not been modified
C) A session with expired data
D) A session that is fully configured
Answer: B
Explanation: An uninitialized session refers to a new session that has not been modified.

128. **What is the recommended way to handle sensitive data in sessions?**
A) Store plain text passwords
B) Use encryption
C) Store data directly in cookies
D) Avoid using sessions
Answer: B
Explanation: Sensitive data should be encrypted before being stored in sessions to enhance security.

129. **Which of the following should be avoided in session storage?**
A) User preferences
B) Authentication tokens
C) Plain text passwords
D) Session IDs
Answer: C
Explanation: Storing plain text passwords in session storage is a security risk.

130. **What does the `httpOnly` flag do for cookies?**
A) Makes the cookie available to JavaScript
B) Prevents the cookie from being sent over secure connections
C) Protects the cookie from being accessed by JavaScript
D) Increases the size of the cookie

Answer: C

Explanation: The `httpOnly` flag prevents JavaScript from accessing the cookie, providing an additional layer of security.

131. **What happens to a session when a user closes the browser?**

A) It remains active

B) It is saved for the next session

C) It is typically destroyed if using session storage

D) It automatically regenerates

Answer: C

Explanation: Sessions using session storage are usually destroyed when the browser is closed.

132. **What is a primary benefit of using a session store like Redis?**

A) It increases latency

B) It allows sessions to persist across server restarts

C) It reduces security

D) It limits session access

Answer: B

Explanation: Using a session store like Redis allows sessions to persist even after server restarts.

133. **How can you retrieve session data in an Express route?**

A) session.get()

B) req.session.data

C) req.session.get()

D) req.data

Answer: B

Explanation: Session data can be accessed using `req.session.data`.

134. **What is a common security measure for managing session IDs?**

A) Use short session durations

B) Use predictable session IDs

C) Regenerate session IDs frequently

D) Store session IDs in local storage

Answer: C

Explanation: Frequently regenerating session IDs can help improve security.

135. **What should be done if a session is found to be compromised?**

A) Ignore it

B) Notify the user

C) Destroy the session and regenerate the ID

D) Increase session duration

Answer: C

Explanation: If a session is compromised, it should be destroyed, and a new session ID should be generated to secure the user's session.

136. **Which storage option provides the fastest session retrieval?**

A) Disk storage

B) In-memory storage

C) Redis
D) Local storage
Answer: B
Explanation: In-memory storage provides the fastest session retrieval due to its access speed.

137. **How do you enable session management in an Express app?**
A) app.use(express.json())
B) app.use(session({ secret: 'your-secret' }))
C) app.use(cookie-parser())
D) app.use(body-parser())
Answer: B
Explanation: To enable session management, you use `app.use(session({ secret: 'your-secret' }))`.

138. **What does `req.session.user` represent?**
A) The current user's password
B) The session storage method
C) The data associated with the current user's session
D) The session ID
Answer: C
Explanation: `req.session.user` typically holds data associated with the current user's session.

139. **Which middleware can be used to parse cookies in Express?**
A) express-session
B) body-parser
C) cookie-parser
D) multer
Answer: C
Explanation: `cookie-parser` is used to parse cookies in Express applications.

140. **How can you set a session variable?**
A) req.session.variableName = value;
B) session.set(variableName, value);
C) req.session.set(variableName, value);
D) session.variableName = value;
Answer: A
Explanation: You can set a session variable using the syntax `req.session.variableName = value;`.

141. **What is the default expiration time for cookies if not set?**
A) 1 hour
B) Session-based (until browser is closed)
C) 24 hours
D) No expiration
Answer: B

Explanation: By default, cookies without an expiration time are session-based and last until the browser is closed.

142. **What is a potential risk of storing sensitive data in a session?**
A) Faster access
B) Security vulnerabilities
C) Improved user experience
D) Easy implementation
Answer: B
Explanation: Storing sensitive data in a session can lead to security vulnerabilities if not handled properly.

143. **How can you implement a session timeout feature?**
A) Check session expiration on each request
B) Use a default session expiration time
C) Ignore session management
D) Keep sessions active indefinitely
Answer: A
Explanation: Implementing a session timeout feature involves checking the session expiration on each request.

144. **What type of data is suitable for storing in sessions?**
A) User preferences
B) Large binary files
C) Application source code
D) System configurations
Answer: A
Explanation: User preferences are appropriate for storing in sessions, while large binary files should not be stored there.

145. **Which option helps to prevent CSRF by checking the origin of requests?**
A) cookie.secure
B) sameSite
C) saveUninitialized
D) resave
Answer: B
Explanation: The `sameSite` attribute helps prevent CSRF by controlling when cookies are sent with cross-origin requests.

146. **What is the effect of `cookie.signed` being set to true?**
A) The cookie is encrypted
B) The cookie is secure
C) The cookie is signed with a secret
D) The cookie is accessible via JavaScript
Answer: C
Explanation: Setting `cookie.signed` to true means the cookie is signed with a secret, adding a layer of integrity verification.

147. **How can you limit the number of active sessions per user?**
A) Increase session duration

B) Use a session store that tracks active sessions
C) Ignore session management
D) Allow unlimited sessions
Answer: B
Explanation: Using a session store that tracks active sessions can help limit the number of concurrent sessions per user.

148. **What is the advantage of using JWTs instead of traditional sessions?**
A) Increased security
B) Stateless and scalable
C) Simplified implementation
D) Improved performance
Answer: B
Explanation: JSON Web Tokens (JWTs) provide a stateless and scalable method for session management.

149. **What does `req.session.destroy()` do?**
A) Deletes the session data
B) Resets the session ID
C) Clears the session store
D) Saves uninitialized sessions
Answer: A
Explanation: The method `req.session.destroy()` is used to delete the session data.

150. **How can you prevent users from accessing expired sessions?**
A) Automatically log them out
B) Allow them to continue using the session
C) Notify them of expiration
D) Ignore session management
Answer: A
Explanation: Automatically logging users out when sessions expire prevents unauthorized access.

151. **What should be done when a session is created?**
A) Store sensitive data
B) Set default values
C) Leave it empty
D) Ignore session management
Answer: B
Explanation: Setting default values when a session is created helps maintain a consistent user experience.

152. **What is a common method to handle session state in a distributed system?**
A) Use in-memory storage
B) Store sessions in a centralized database
C) Limit to a single server
D) Use local storage
Answer: B

Explanation: In a distributed system, storing sessions in a centralized database allows multiple servers to access the same session data.

153. **What is a potential downside of using cookies for session management?**
A) They are secure
B) They can be manipulated by users
C) They are persistent
D) They improve performance
Answer: B
Explanation: Cookies can be manipulated by users if not properly secured, posing a risk to session integrity.

154. **How can you check if a user is authenticated in a session?**
A) req.session.isAuthenticated
B) req.session.authenticated
C) req.user.isAuthenticated
D) req.session.userExists
Answer: A
Explanation: You can check if a user is authenticated by examining a property like `req.session.isAuthenticated`.

155. **What does the `uninitialized` session state indicate?**
A) A session that has expired
B) A session that has not been modified after creation
C) A fully active session
D) A session that is invalid
Answer: B
Explanation: An uninitialized session state indicates that a session has been created but not modified yet.

156. **What should be avoided when designing a session management strategy?**
A) Regularly rotating session secrets
B) Using predictable session IDs
C) Validating user input
D) Using secure cookie attributes
Answer: B
Explanation: Using predictable session IDs can lead to security vulnerabilities.

157. **What is the purpose of `req.session.cookie`?**
A) To store user preferences
B) To manage cookie settings related to the session
C) To track session IDs
D) To encrypt session data
Answer: B
Explanation: `req.session.cookie` is used to manage cookie settings related to the session.

158. **How can you securely transmit session IDs?**
A) Over HTTP
B) Over HTTPS

C) In plain text
D) Via email
Answer: B
Explanation: Securely transmitting session IDs should always be done over HTTPS to prevent interception.

159. **What is a typical characteristic of session data?**
A) It is permanent
B) It is typically short-lived
C) It is stored in the database
D) It is publicly accessible
Answer: B
Explanation: Session data is typically short-lived and intended for temporary use during a user session.

160. **What is the main purpose of a session?**
A) To store permanent user data
B) To manage user authentication and state across requests
C) To enhance application performance
D) To replace database storage
Answer: B
Explanation: The main purpose of a session is to manage user authentication and maintain state across multiple requests.

161. **What is authentication?**
A) Verifying user identity
B) Granting permissions to users
C) Logging user activity
D) Encrypting user data
Answer: A
Explanation: Authentication is the process of verifying the identity of a user.

162. **What is authorization?**
A) Checking user credentials
B) Granting access to resources
C) Storing user data
D) Encrypting passwords
Answer: B
Explanation: Authorization is the process of determining whether a user has permission to access certain resources.

163. **Which of the following is a common method for user authentication?**
A) IP address
B) Username and password
C) Device type
D) Account creation date
Answer: B
Explanation: Username and password are commonly used for authenticating users.

164. **What is the purpose of a session token?**
A) To store user credentials
B) To authenticate users
C) To track user activity across requests
D) To encrypt data
Answer: C
Explanation: A session token is used to track user activity across multiple requests.

165. **What does OAuth primarily provide?**
A) User authentication
B) Resource authorization
C) Password storage
D) Data encryption
Answer: B
Explanation: OAuth is primarily used for authorization, allowing users to grant third-party applications access to their resources without sharing their passwords.

166. **What is two-factor authentication (2FA)?**
A) Using two usernames
B) Verifying identity with two methods
C) Changing passwords twice
D) Logging in twice
Answer: B
Explanation: Two-factor authentication requires two different methods to verify a user's identity, enhancing security.

167. **Which of the following is a secure way to store passwords?**
A) Plain text
B) Encrypted format
C) Hashed with a salt
D) Base64 encoded
Answer: C
Explanation: Passwords should be hashed with a salt to ensure they are stored securely and to prevent easy recovery.

168. **What is a common exception that can occur during authentication?**
A) Access denied
B) Invalid credentials
C) Resource not found
D) Timeout
Answer: B
Explanation: An invalid credentials exception occurs when the user provides incorrect authentication information.

169. **Which status code is typically returned when authentication fails?**
A) 200
B) 401
C) 403
D) 404

Answer: B
Explanation: A 401 status code indicates that authentication has failed or has not been provided.

170. **What is the purpose of role-based access control (RBAC)?**
A) To encrypt data
B) To manage user roles and permissions
C) To store user credentials
D) To log user activity
Answer: B
Explanation: RBAC is used to assign permissions to users based on their roles within an organization.

171. **What is an access token?**
A) A temporary password
B) A credential used to access resources
C) A unique user ID
D) A session ID
Answer: B
Explanation: An access token is a credential that allows access to a resource or service.

172. **What happens when a user attempts to access a resource they are not authorized to?**
A) They are logged out
B) They are shown an error page
C) They are granted access
D) They are redirected to the login page
Answer: B
Explanation: When a user attempts to access a restricted resource, they typically receive an error indicating they do not have permission.

173. **Which of the following is a method of handling authentication exceptions?**
A) Ignoring the error
B) Logging the error
C) Displaying a generic error message
D) All of the above
Answer: D
Explanation: Various methods can be used to handle authentication exceptions, including logging errors and displaying messages.

174. **What does the `403 Forbidden` status code indicate?**
A) User is not authenticated
B) User is authenticated but not authorized
C) Resource was not found
D) Invalid credentials
Answer: B
Explanation: A 403 status code indicates that the user is authenticated but does not have permission to access the requested resource.

175. **What is a common approach to secure APIs?**
A) Using basic authentication
B) Implementing rate limiting
C) Using JWT (JSON Web Tokens)
D) Disabling CORS
Answer: C
Explanation: Implementing JWT is a common method to secure APIs, as it allows for stateless authentication.

176. **What is the purpose of a refresh token?**
A) To access protected resources
B) To renew access tokens
C) To log users out
D) To authenticate users
Answer: B
Explanation: Refresh tokens are used to obtain new access tokens without requiring the user to log in again.

177. **Which authentication method requires users to provide a code sent to their mobile device?**
A) Single sign-on (SSO)
B) OAuth
C) Two-factor authentication (2FA)
D) Basic authentication
Answer: C
Explanation: Two-factor authentication often involves sending a code to the user's mobile device for verification.

178. **What is the purpose of input validation during authentication?**
A) To improve performance
B) To enhance user experience
C) To prevent injection attacks
D) To store user data securely
Answer: C
Explanation: Input validation is critical for preventing injection attacks and ensuring that only valid data is processed.

179. **Which of the following is a method for handling authorization exceptions?**
A) Redirecting to the login page
B) Logging the user out
C) Returning a 403 Forbidden response
D) Displaying user roles
Answer: C
Explanation: Returning a 403 Forbidden response is a common way to handle authorization exceptions.

180. **What does it mean for an application to be stateless?**
A) It does not remember user sessions
B) It does not perform authentication

C) It only uses cookies
D) It does not store any data
Answer: A
Explanation: A stateless application does not retain user session information between requests, often using tokens for authentication.

181. **What is the effect of using HTTPS for authentication?**
A) Slower connection
B) Enhanced security for transmitted data
C) Inability to use tokens
D) None of the above
Answer: B
Explanation: HTTPS encrypts data during transmission, significantly enhancing security for authentication processes.

182. **Which of the following is a characteristic of a strong password?**
A) It is easy to remember
B) It contains special characters and numbers
C) It is the same as the username
D) It is short
Answer: B
Explanation: A strong password should include a mix of letters, numbers, and special characters to enhance security.

183. **What is the purpose of a password reset token?**
A) To change a user's username
B) To allow users to reset their passwords securely
C) To log users out
D) To encrypt passwords
Answer: B
Explanation: A password reset token is used to verify the identity of a user requesting a password change.

184. **Which of the following can be a consequence of poor authentication practices?**
A) Improved user experience
B) Data breaches
C) Increased application performance
D) Enhanced security
Answer: B
Explanation: Poor authentication practices can lead to data breaches and unauthorized access to sensitive information.

185. **What does the acronym SSO stand for in authentication?**
A) Single Sign-On
B) Secure Session Object
C) Simple Security Option
D) Standard Sign-Out
Answer: A

Explanation: SSO stands for Single Sign-On, allowing users to log in once and access multiple applications without re-authenticating.

186. **Which exception might be thrown if a user exceeds maximum login attempts?**
A) Too Many Requests
B) Invalid Credentials
C) Unauthorized Access
D) User Not Found
Answer: A
Explanation: A "Too Many Requests" exception is typically thrown if a user exceeds the allowed number of login attempts.

187. **What is a common method for logging authentication attempts?**
A) Writing to a database
B) Sending an email
C) Displaying on the user interface
D) Ignoring them
Answer: A
Explanation: Logging authentication attempts in a database can help track successful and failed login attempts.

188. **What does a 401 Unauthorized status code indicate?**
A) User is not logged in
B) User does not have permission
C) Resource is not found
D) Invalid token
Answer: A
Explanation: A 401 status code indicates that the user is not logged in or that authentication is required.

189. **How can you ensure secure token storage on the client side?**
A) In local storage
B) In session storage
C) In cookies with the HttpOnly flag
D) In plain text files
Answer: C
Explanation: Storing tokens in cookies with the HttpOnly flag helps protect them from being accessed by JavaScript.

190. **What is the purpose of logging failed authentication attempts?**
A) To increase performance
B) To identify potential security threats
C) To enhance user experience
D) To track user activity
Answer: B
Explanation: Logging failed authentication attempts can help identify and respond to potential security threats.

191. **What is the impact of using a brute-force attack?**
A) It improves authentication speed

B) It compromises user accounts
C) It enhances security
D) It has no impact
Answer: B
Explanation: A brute-force attack attempts multiple password combinations to compromise user accounts.

192. **Which of the following is a strategy for preventing brute-force attacks?**
A) Allow unlimited login attempts
B) Implement account lockout mechanisms
C) Use predictable passwords
D) Ignore failed attempts
Answer: B
Explanation: Implementing account lockout mechanisms can help prevent brute-force attacks by temporarily disabling accounts after several failed attempts.

193. **What is the role of middleware in handling authentication in web applications?**
A) To manage server resources
B) To process user requests before reaching the main application logic
C) To store session data
D) To encrypt communication
Answer: B
Explanation: Middleware processes requests and responses, allowing for authentication checks before reaching the main application logic.

194. **What type of information is typically included in an access token?**
A) User credentials
B) User permissions and roles
C) Sensitive data
D) Session history
Answer: B
Explanation: Access tokens often contain user permissions and roles to determine what resources the user can access.

195. **What should be done if an authentication exception occurs?**
A) Redirect the user to a different page
B) Notify the user and log the exception
C) Ignore the exception
D) Retry the authentication
Answer: B
Explanation: Notifying the user and logging the exception is essential for both user experience and security monitoring.

196. **Which of the following can be a method of user authorization?**
A) Checking user roles
B) Validating input data
C) Storing passwords
D) Sending emails
Answer: A

Explanation: Checking user roles is a common method for determining user authorization levels for accessing resources.

197. **What does the acronym RBAC stand for?**
A) Role-Based Access Control
B) Random Binary Access Control
C) Required Basic Access Control
D) Rapid Backup and Access Control
Answer: A
Explanation: RBAC stands for Role-Based Access Control, a method of restricting system access based on user roles.

198. **What is a possible consequence of not properly handling exceptions during authentication?**
A) Improved application performance
B) Increased security risks
C) Better user experience
D) Reduced server load
Answer: B
Explanation: Improperly handled exceptions can lead to increased security risks and vulnerabilities.

199. **Which technique can help mitigate session hijacking?**
A) Using unencrypted cookies
B) Setting secure cookie flags
C) Allowing all origins
D) Ignoring session management
Answer: B
Explanation: Setting secure cookie flags helps mitigate session hijacking by ensuring cookies are only sent over secure connections.

200. **What is the role of CAPTCHA in authentication?**
A) To enhance security by verifying human users
B) To store passwords
C) To encrypt data
D) To improve user experience
Answer: A
Explanation: CAPTCHA is used to verify that a user is human, helping prevent automated attacks during authentication.

201. **What does the `401 Unauthorized` response typically indicate?**
A) Authentication is required but has not been provided
B) The request is valid but the user does not have permissions
C) The resource could not be found
D) The server is busy
Answer: A
Explanation: A 401 response indicates that the user must authenticate to access the resource.

202. **What should be included in an authorization failure response?**
A) A success message
B) A detailed error description
C) A 403 Forbidden status code
D) Redirecting to the homepage
Answer: C
Explanation: A 403 Forbidden status code is typically returned to indicate that the user is authenticated but not authorized.

203. **Which exception handling strategy is best for user feedback during failed authentication?**
A) Displaying a detailed error message
B) Logging the error only
C) Hiding error messages to prevent information leakage
D) Ignoring the error
Answer: C
Explanation: It's best to provide generic error messages to avoid giving attackers information about which part of the authentication failed.

204. **What is a potential risk of using JSON Web Tokens (JWT)?**
A) They are always secure
B) They can be tampered with if not properly signed
C) They are too large
D) They cannot be used for authorization
Answer: B
Explanation: JWTs can be tampered with if they are not properly signed and validated.

205. **What is the purpose of an authorization header in HTTP requests?**
A) To encrypt data
B) To provide user credentials
C) To specify the method of the request
D) To identify the type of content
Answer: B
Explanation: The authorization header is used to provide credentials to authenticate a user for accessing resources.

206. **What is the main purpose of using a secure password policy?**
A) To simplify user registration
B) To enhance security by enforcing strong passwords
C) To speed up the login process
D) To reduce server load
Answer: B
Explanation: A secure password policy enhances security by ensuring that users create strong and difficult-to-guess passwords.

207. **What is a security best practice for handling authentication tokens?**
A) Store them in local storage
B) Transmit them over HTTP
C) Set expiration times for tokens

D) Share them with third parties
Answer: C
Explanation: Setting expiration times for tokens helps reduce the risk of them being used after a user logs out or after a session ends.

208. **How can you prevent unauthorized access to sensitive resources?**
A) Using public links
B) Implementing strict access controls
C) Allowing anyone to access
D) Ignoring permissions
Answer: B
Explanation: Strict access controls are essential for preventing unauthorized access to sensitive resources.

209. **What is the role of logging in tracking authentication attempts?**
A) It reduces performance
B) It provides insights for security audits
C) It replaces user feedback
D) It serves no purpose
Answer: B
Explanation: Logging authentication attempts provides valuable insights for security audits and can help identify patterns of abuse.

210. **What happens if a user forgets their password?**
A) They cannot log in
B) They can reset their password using a secure process
C) They are logged out permanently
D) They must create a new account
Answer: B
Explanation: Users can typically reset their passwords through a secure process to regain access.

211. **What is an invalid token exception?**
A) A successful authentication
B) A failure to authenticate due to a revoked or expired token
C) A request for user information
D) An error in logging
Answer: B
Explanation: An invalid token exception occurs when a token is expired or has been revoked, preventing successful authentication.

212. **What is the main advantage of using LDAP for authentication?**
A) It is user-friendly
B) It allows for centralized user management
C) It encrypts passwords automatically
D) It requires less configuration
Answer: B
Explanation: LDAP provides centralized user management, making it easier to authenticate users across multiple services.

213. **Which of the following is a method to improve user experience during authentication?**
A) Implementing CAPTCHA on every login
B) Allowing social media logins
C) Using complex passwords only
D) Ignoring user feedback
Answer: B
Explanation: Allowing social media logins can streamline the authentication process and improve user experience.

214. **What is the purpose of token expiration?**
A) To limit access to resources
B) To enhance security by reducing the window of opportunity for misuse
C) To simplify token management
D) To track user activity
Answer: B
Explanation: Token expiration reduces the risk of token misuse by limiting the time they can be used.

215. **Which of the following is a sign of a potential security breach?**
A) Multiple failed login attempts from different IPs
B) Successful logins
C) Regular user activity
D) Routine password changes
Answer: A
Explanation: Multiple failed login attempts from different IP addresses may indicate a potential security breach or attack.

216. **What does the term "least privilege" mean in access control?**
A) Users should have the maximum amount of access possible
B) Users should only have the access necessary to perform their jobs
C) Users should have no access
D) Users can share their access
Answer: B
Explanation: The principle of least privilege states that users should only have the permissions necessary to perform their specific roles.

217. **What does an authorization middleware function do in an Express app?**
A) Logs all requests
B) Validates user roles and permissions before proceeding to the next middleware
C) Encrypts data
D) Sends responses to clients
Answer: B
Explanation: An authorization middleware function checks user roles and permissions to ensure they are allowed to access certain routes.

218. **What is the purpose of an audit log?**
A) To track user preferences
B) To record actions taken by users for security review

C) To improve application speed
D) To store user credentials
Answer: B
Explanation: Audit logs are used to record actions taken by users, helping in security reviews and identifying potential abuse.

219. **Which of the following is true about hashed passwords?**
A) They can be easily decrypted
B) They are stored in plain text
C) They are one-way and cannot be reversed
D) They are not secure
Answer: C
Explanation: Hashed passwords are one-way functions, meaning they cannot be reversed to retrieve the original password.

220. **What does the `401 Unauthorized` error usually signify?**
A) The user is authenticated
B) The user needs to provide credentials
C) The resource is forbidden
D) The request is malformed
Answer: B
Explanation: A 401 Unauthorized error indicates that the user must provide valid authentication credentials to access the resource.

221. **How should sensitive information be transmitted over the network?**
A) Using plain HTTP
B) Using FTP
C) Using HTTPS
D) Using local storage
Answer: C
Explanation: HTTPS should be used for transmitting sensitive information to ensure data is encrypted during transmission.

222. **What does it mean to "revoke" a token?**
A) To delete it permanently
B) To invalidate it, preventing further use
C) To extend its expiration
D) To share it with another user
Answer: B
Explanation: Revoking a token invalidates it, preventing further use for authentication.

223. **What is a session fixation attack?**
A) Capturing user credentials
B) Exploiting a user's session ID
C) Brute-forcing passwords
D) Injecting malicious scripts
Answer: B
Explanation: A session fixation attack involves exploiting a user's session ID to gain unauthorized access.

224. **What role does input sanitization play in authentication?**
A) It speeds up authentication
B) It prevents injection attacks
C) It simplifies code
D) It is not necessary
Answer: B
Explanation: Input sanitization is crucial for preventing injection attacks, ensuring that only valid data is processed during authentication.

225. **Which of the following is a disadvantage of basic authentication?**
A) It is secure
B) It requires less bandwidth
C) Credentials are sent in plain text
D) It is easy to implement
Answer: C
Explanation: Basic authentication sends credentials in plain text, making it vulnerable to interception.

226. **What is the purpose of a secure token?**
A) To store user data
B) To represent user sessions securely
C) To log user actions
D) To manage resources
Answer: B
Explanation: A secure token represents a user session and allows access to protected resources without exposing user credentials.

227. **What is an essential aspect of API authentication?**
A) Using a single username
B) Providing a simple login form
C) Using standard protocols like OAuth
D) Ignoring security
Answer: C
Explanation: Standard protocols like OAuth provide a framework for secure API authentication and authorization.

228. **What is the effect of enabling CORS (Cross-Origin Resource Sharing) for an API?**
A) It prevents all external requests
B) It allows requests from specified origins
C) It increases API load times
D) It is unrelated to security
Answer: B
Explanation: Enabling CORS allows requests from specified origins, facilitating safe interactions between different domains.

229. **Which type of attack is commonly mitigated by using secure tokens?**
A) SQL injection
B) Cross-Site Scripting (XSS)

C) Session hijacking
D) Brute-force attacks
Answer: C
Explanation: Secure tokens help mitigate session hijacking by ensuring that session information is not easily accessible.

230. **What does the term "token revocation" refer to?**
A) Granting new tokens
B) Disabling previously issued tokens
C) Creating new user accounts
D) Ignoring token expiration
Answer: B
Explanation: Token revocation refers to the process of disabling tokens that were previously issued to prevent further access.

231. **What does an `invalid_grant` error indicate in OAuth?**
A) The token is valid
B) The credentials provided are incorrect
C) The user is authorized
D) The request is malformed
Answer: B
Explanation: An `invalid_grant` error indicates that the provided credentials are incorrect or have expired.

232. **What is the purpose of user roles in authorization?**
A) To increase server load
B) To define permissions and access levels
C) To reduce user complexity
D) To eliminate authentication
Answer: B
Explanation: User roles define what permissions and access levels a user has within an application.

233. **What is a common method for detecting unauthorized access?**
A) Ignoring failed attempts
B) Implementing logging and monitoring
C) Allowing unlimited access
D) Hiding error messages
Answer: B
Explanation: Logging and monitoring are common methods used to detect unauthorized access attempts.

234. **What should a web application do when a user is logged out?**
A) Clear all user data
B) Maintain the session
C) Invalidate the session token
D) Redirect to the home page
Answer: C

Explanation: When a user logs out, the application should invalidate the session token to prevent further access.

235. **Which of the following is a way to improve the security of passwords?**
A) Allowing short passwords
B) Using multi-factor authentication
C) Storing passwords in plain text
D) Reusing old passwords
Answer: B
Explanation: Multi-factor authentication enhances password security by adding an additional layer of verification.

236. **What is a common method for logging out users?**
A) Deleting cookies
B) Ignoring logout requests
C) Redirecting to the login page without invalidating sessions
D) Keeping the session active
Answer: A
Explanation: Logging out users typically involves deleting cookies or invalidating session tokens to ensure that access is removed.

237. **What does a 200 OK status code indicate during authentication?**
A) User is not authenticated
B) Authentication was successful
C) Resource was not found
D) User needs to log in
Answer: B
Explanation: A 200 OK status code indicates that the authentication process was successful.

238. **What is the purpose of logging authentication failures?**
A) To ignore user actions
B) To track potential security threats
C) To reduce server load
D) To enhance user experience
Answer: B
Explanation: Logging authentication failures helps track potential security threats and identifies patterns of unauthorized access attempts.

239. **How can session expiration enhance security?**
A) By allowing unlimited session duration
B) By reducing the risk of unauthorized access
C) By simplifying user login
D) By ignoring user activity
Answer: B
Explanation: Session expiration reduces the risk of unauthorized access by limiting the time a session can be active.

240. **What is the primary purpose of implementing HTTPS?**
A) To improve loading speed

B) To encrypt data in transit
C) To simplify web development
D) To allow more traffic
Answer: B
Explanation: The primary purpose of HTTPS is to encrypt data in transit, providing security for information exchanged between clients and servers.

241. **What is an example of a secure method for sending authentication credentials?**
A) Sending credentials in a query string
B) Sending credentials in a JSON body over HTTPS
C) Using plain HTTP
D) Storing credentials in cookies
Answer: B
Explanation: Sending credentials in a JSON body over HTTPS is a secure method as it ensures that data is encrypted in transit.

242. **What does the `403 Forbidden` response indicate?**
A) The user is not logged in
B) The user is authenticated but does not have permission to access the resource
C) The resource is unavailable
D) The request was malformed
Answer: B
Explanation: A 403 Forbidden response indicates that the user is authenticated but lacks permission to access the requested resource.

243. **What is the function of OAuth 2.0?**
A) To provide a framework for authorization
B) To encrypt user data
C) To store session information
D) To handle user authentication
Answer: A
Explanation: OAuth 2.0 provides a framework for authorization, allowing users to grant third-party applications limited access to their resources without sharing credentials.

244. **Which of the following can be a consequence of insufficient authentication security?**
A) Improved user engagement
B) Increased risk of account takeover
C) Enhanced application performance
D) Simplified user login
Answer: B
Explanation: Insufficient authentication security increases the risk of account takeover and unauthorized access.

245. **What is the purpose of a refresh token?**
A) To change the user's password
B) To allow a user to obtain new access tokens without re-authenticating
C) To log the user out

D) To store user preferences
Answer: B
Explanation: A refresh token allows a user to obtain new access tokens without needing to re-authenticate, maintaining a seamless user experience.

246. **What is a common practice for securing APIs?**
A) Using basic authentication only
B) Implementing API keys and OAuth
C) Allowing open access
D) Ignoring rate limiting
Answer: B
Explanation: Implementing API keys and OAuth is a common practice for securing APIs against unauthorized access.

247. **What is an example of a session management vulnerability?**
A) Long session expiration
B) Lack of secure cookie flags
C) Strict access controls
D) Frequent password updates
Answer: B
Explanation: Lack of secure cookie flags is a session management vulnerability that can lead to session hijacking.

248. **What does a 404 Not Found status code indicate?**
A) The user is authenticated
B) The requested resource does not exist
C) The user has insufficient permissions
D) The server is busy
Answer: B
Explanation: A 404 Not Found status code indicates that the requested resource could not be found on the server.

249. **What is a secure method for password storage?**
A) Storing passwords in plain text
B) Hashing passwords with a strong algorithm
C) Using reversible encryption
D) Using weak hashing algorithms
Answer: B
Explanation: Hashing passwords with a strong algorithm (e.g., bcrypt) is a secure method for storing passwords.

250. **What is the consequence of not validating input during authentication?**
A) Increased performance
B) Vulnerability to injection attacks
C) Improved user experience
D) Simplified coding
Answer: B
Explanation: Not validating input can lead to vulnerabilities, including injection attacks.

251. **What is the benefit of using a token-based authentication system?**
A) Reduced user privacy
B) Stateless sessions, which can improve scalability
C) Complexity in implementation
D) Increased server load
Answer: B
Explanation: Token-based authentication allows for stateless sessions, which can improve scalability by reducing server-side session management.

252. **What does a `429 Too Many Requests` response indicate?**
A) The server is overloaded
B) The user is authenticated
C) The client has sent too many requests in a given timeframe
D) The resource is unavailable
Answer: C
Explanation: A `429 Too Many Requests` response indicates that the client has sent too many requests in a specified timeframe.

253. **What is the role of password hashing in authentication?**
A) To encrypt passwords for storage
B) To ensure passwords are not stored in plain text
C) To simplify password retrieval
D) To make passwords more memorable
Answer: B
Explanation: Password hashing ensures that passwords are not stored in plain text, enhancing security.

254. **What does a `302 Found` status code indicate?**
A) The resource has been permanently moved
B) The resource is temporarily located at a different URI
C) The user is authenticated
D) The request was not valid
Answer: B
Explanation: A `302 Found` status code indicates that the resource is temporarily located at a different URI.

255. **What is a common way to enhance API security?**
A) Using public endpoints
B) Implementing rate limiting
C) Ignoring security protocols
D) Using plain HTTP
Answer: B
Explanation: Implementing rate limiting helps enhance API security by preventing abuse and protecting against denial-of-service attacks.

256. **What is the purpose of two-factor authentication (2FA)?**
A) To simplify login
B) To add an extra layer of security beyond just a password
C) To reduce server load

D) To allow multiple logins

Answer: B

Explanation: Two-factor authentication (2FA) adds an extra layer of security by requiring a second form of verification in addition to a password.

257. **What is an example of a secure coding practice for handling user input?**

A) Ignoring user input

B) Validating and sanitizing all input

C) Allowing all input types

D) Using input as is

Answer: B

Explanation: Validating and sanitizing all user input is a secure coding practice that helps prevent various types of attacks, including injection attacks.

258. **What does the acronym CSRF stand for?**

A) Cross-Site Request Forgery

B) Client-Side Rendering Framework

C) Common Security Response Framework

D) Cross-Site Reference Format

Answer: A

Explanation: CSRF stands for Cross-Site Request Forgery, a type of attack that tricks a user into executing unwanted actions on a web application.

259. **What does a 500 Internal Server Error indicate?**

A) The request is invalid

B) An unexpected condition was encountered

C) The user is not authenticated

D) The resource is forbidden

Answer: B

Explanation: A 500 Internal Server Error indicates that the server encountered an unexpected condition that prevented it from fulfilling the request.

260. **What is the purpose of logging successful authentication attempts?**

A) To ignore user actions

B) To track user behavior and audit access

C) To reduce server load

D) To simplify the login process

Answer: B

Explanation: Logging successful authentication attempts is important for tracking user behavior and auditing access for security purposes.

261. **What is JWT?**
    - A) JSON Web Token
    - B) Java Web Token
    - C) JavaScript Web Token
    - D) None of the above

      Answer: A

      Explanation: JWT stands for JSON Web Token, a compact and self-contained way for securely transmitting information between parties.

262. **What does the JWT payload contain?**
    - A) User's password
    - B) Claims
    - C) Encrypted data
    - D) None of the above

    Answer: B

    Explanation: The payload of a JWT contains claims, which are statements about an entity (usually the user) and additional metadata.
263. **Which part of a JWT is used for verification?**
    - A) Header
    - B) Payload
    - C) Signature
    - D) None of the above

    Answer: C

    Explanation: The signature is used to verify the integrity and authenticity of the token.
264. **What is the typical structure of a JWT?**
    - A) Header.Payload.Signature
    - B) Signature.Payload.Header
    - C) Payload.Signature.Header
    - D) None of the above

    Answer: A

    Explanation: A JWT is structured as three parts: Header, Payload, and Signature, separated by dots.
265. **Which algorithm is commonly used to sign a JWT?**
    - A) SHA-256
    - B) HMAC SHA256
    - C) RSA
    - D) Both B and C

    Answer: D

    Explanation: Both HMAC SHA256 and RSA are commonly used algorithms for signing JWTs.
266. **What is the purpose of the 'exp' claim in a JWT?**
    - A) To define the issuer
    - B) To specify the expiration time
    - C) To indicate the audience
    - D) To include user roles

    Answer: B

    Explanation: The 'exp' claim indicates when the token should expire, helping to limit its validity.
267. **What library is commonly used for creating and verifying JWTs in Node.js?**
    - A) express-jwt
    - B) jsonwebtoken
    - C) passport-jwt

- D) cookie-parser

  Answer: B

  Explanation: The `jsonwebtoken` library is widely used for creating and verifying JWTs in Node.js applications.

268. **How is a JWT typically transmitted in a web application?**
   - A) As a URL parameter
   - B) In the HTTP headers
   - C) In the body of a POST request
   - D) All of the above

     Answer: B

     Explanation: JWTs are commonly transmitted in the HTTP Authorization header as a Bearer token.

269. **What is the main advantage of using JWTs for authentication?**
   - A) They are stateless and can be easily verified
   - B) They require less server storage
   - C) They can be shared across domains
   - D) All of the above

     Answer: D

     Explanation: JWTs are stateless, require no server storage, and can be shared across domains, making them versatile for authentication.

270. **What does the 'iss' claim in a JWT represent?**
   - A) The audience
   - B) The issuer
   - C) The subject
   - D) The expiration

     Answer: B

     Explanation: The 'iss' claim indicates the issuer of the token, which is the entity that generated the JWT.

271. **What is bcrypt commonly used for?**
   - A) Generating JWTs
   - B) Hashing passwords
   - C) Encrypting tokens
   - D) None of the above

     Answer: B

     Explanation: Bcrypt is specifically designed for hashing passwords securely.

272. **What is a major benefit of using bcrypt over other hashing algorithms?**
   - A) It is faster
   - B) It allows for adjustable work factors
   - C) It generates shorter hashes
   - D) It does not require a salt

     Answer: B

     Explanation: Bcrypt allows for adjustable work factors, making it more secure against brute-force attacks.

273. **What is the recommended minimum length for a bcrypt hash?**

- A) 60 characters
- B) 32 characters
- C) 128 characters
- D) 256 characters

  Answer: A

  Explanation: Bcrypt hashes are typically 60 characters long.

274. **What is the purpose of a refresh token?**
   - A) To replace expired access tokens
   - B) To provide access to sensitive data
   - C) To invalidate other tokens
   - D) To store user credentials

     Answer: A

     Explanation: A refresh token is used to obtain a new access token when the current access token expires.

275. **How do you typically store JWTs on the client side?**
   - A) In local storage
   - B) In session storage
   - C) In cookies
   - D) All of the above

     Answer: D

     Explanation: JWTs can be stored in local storage, session storage, or cookies, depending on the security requirements.

276. **Which security risk is associated with storing JWTs in local storage?**
   - A) XSS attacks
   - B) CSRF attacks
   - C) Man-in-the-middle attacks
   - D) SQL injection

     Answer: A

     Explanation: Storing JWTs in local storage makes them vulnerable to XSS attacks.

277. **What is the primary purpose of JWT authentication?**
   - A) To create user accounts
   - B) To authorize access to resources
   - C) To log user activity
   - D) To encrypt sensitive data

     Answer: B

     Explanation: JWT authentication is primarily used to authorize access to protected resources.

278. **What does the 'sub' claim in a JWT signify?**
   - A) The issuer
   - B) The subject (user ID)
   - C) The expiration
   - D) The audience

     Answer: B

Explanation: The 'sub' claim typically signifies the subject, often the user ID, that the token is associated with.

279. **How do you ensure that a JWT is valid?**
    - A) By checking the payload
    - B) By verifying the signature
    - C) By checking the expiration time
    - D) Both B and C

    Answer: D

    Explanation: A JWT is validated by verifying the signature and checking the expiration time.

280. **What is a potential drawback of using JWTs for session management?**
    - A) Tokens cannot be revoked easily
    - B) They require server-side storage
    - C) They are insecure
    - D) They are too large

    Answer: A

    Explanation: JWTs are stateless, making it challenging to revoke tokens before their expiration.

281. **Which HTTP status code is commonly used to indicate an unauthorized request?**
    - A) 403
    - B) 401
    - C) 404
    - D) 500

    Answer: B

    Explanation: The 401 status code indicates that authentication is required and has failed.

282. **What is a common way to protect JWTs from CSRF attacks?**
    - A) Use secure cookies
    - B) Store them in local storage
    - C) Send them as URL parameters
    - D) Use plain text

    Answer: A

    Explanation: Storing JWTs in secure cookies can help protect against CSRF attacks.

283. **How can you refresh an expired JWT without requiring user re-authentication?**
    - A) By using a refresh token
    - B) By changing the secret key
    - C) By creating a new token manually
    - D) By invalidating the old token

    Answer: A

    Explanation: A refresh token allows the application to obtain a new access token without re-authenticating the user.

284. **What should be done if a user's JWT is compromised?**

- A) Notify the user
- B) Invalidate the token
- C) Change the user's password
- D) All of the above

  Answer: D

  Explanation: If a JWT is compromised, the user should be notified, the token invalidated, and the password changed if necessary.

285. **What is the purpose of the 'aud' claim in a JWT?**
   - A) To specify the audience for the token
   - B) To define the issuer
   - C) To indicate the expiration
   - D) To list permissions

     Answer: A

     Explanation: The 'aud' claim specifies the intended audience for the JWT.

286. **What happens if you do not include an 'exp' claim in a JWT?**
   - A) The token will never expire
   - B) The token is considered invalid
   - C) The token will expire immediately
   - D) The token will last for a fixed time

     Answer: A

     Explanation: Without an 'exp' claim, the token is effectively set to never expire.

287. **What is the role of middleware in an Express.js application using JWT?**
   - A) To manage database connections
   - B) To handle authentication and authorization
   - C) To serve static files
   - D) To implement business logic

     Answer: B

     Explanation: Middleware is often used to handle authentication and authorization in Express.js applications.

288. **What should you do if a user changes their password while logged in?**
   - A) Invalidate existing tokens
   - B) Do nothing
   - C) Force logout immediately
   - D) Allow the token to remain valid

     Answer: A

     Explanation: Existing tokens should be invalidated to prevent unauthorized access after a password change.

289. **What does it mean for a JWT to be stateless?**
   - A) The server does not store any session information
   - B) The JWT cannot be modified
   - C) The JWT is always secure
   - D) The JWT has no expiration

     Answer: A

Explanation: Being stateless means that the server does not need to store any session information; all the required data is in the token.

290. **What type of claim is the 'iat' claim in a JWT?**
    - A) Registered claim
    - B) Public claim
    - C) Private claim
    - D) None of the above

    Answer: A

    Explanation: The 'iat' (issued at) claim is a registered claim indicating when the token was issued.

291. **What is the recommended approach for storing secret keys in a Node.js application?**
    - A) Hard-code them in the source code
    - B) Store them in environment variables
    - C) Save them in a public repository
    - D) Use them directly in front-end code

    Answer: B

    Explanation: Storing secret keys in environment variables helps keep them secure.

292. **What is one way to protect JWTs from being hijacked?**
    - A) Use short expiration times
    - B) Store them in local storage
    - C) Avoid encryption
    - D) Share them publicly

    Answer: A

    Explanation: Using short expiration times limits the window for potential hijacking.

293. **What is a common method for implementing user roles with JWTs?**
    - A) Include roles in the JWT payload
    - B) Use separate tokens for each role
    - C) Manage roles in a database only
    - D) Use static roles

    Answer: A

    Explanation: Including roles in the JWT payload allows for easier management of user permissions.

294. **What is a disadvantage of using long-lived access tokens?**
    - A) They can be easily generated
    - B) They pose a security risk if compromised
    - C) They are not user-friendly
    - D) They cannot be used across multiple services

    Answer: B

    Explanation: Long-lived tokens increase the risk of being compromised, as they can be used for an extended period.

295. **How can you implement logging out a user when using JWTs?**

- A) By deleting the JWT from local storage
- B) By invalidating the JWT on the server
- C) By changing the user's role
- D) Both A and B

  Answer: D

  Explanation: Logging out can involve deleting the JWT from the client and invalidating it on the server.

296. **What is the primary purpose of hashing passwords?**
   - A) To make them easy to read
   - B) To ensure they are stored securely
   - C) To encrypt them
   - D) To increase their length

     Answer: B

     Explanation: Hashing passwords helps ensure that they are stored securely and are not easily accessible.

297. **Why is salting used in password hashing?**
   - A) To make hashes shorter
   - B) To add randomness and protect against rainbow table attacks
   - C) To speed up the hashing process
   - D) To encrypt the password

     Answer: B

     Explanation: Salting adds randomness to the password hash, making it more secure against precomputed attacks like rainbow tables.

298. **What is the main benefit of using a library like bcrypt?**
   - A) It provides easy-to-read hashes
   - B) It allows for slow hashing to deter brute-force attacks
   - C) It requires no configuration
   - D) It stores passwords in plain text

     Answer: B

     Explanation: Bcrypt is designed to be slow, which helps deter brute-force attacks.

299. **What is an effective strategy for user authentication using JWTs?**
   - A) Only using a username and password
   - B) Using multi-factor authentication (MFA)
   - C) Storing all user data in the JWT
   - D) Allowing unlimited login attempts

     Answer: B

     Explanation: Implementing multi-factor authentication adds an extra layer of security to user authentication.

300. **What is the purpose of the 'nbf' claim in a JWT?**
   - A) To specify the not before time
   - B) To define the expiration time
   - C) To indicate the issuer

- D) To list permissions

  Answer: A

  Explanation: The 'nbf' (not before) claim specifies the time before which the token should not be accepted for processing.

301. **Which HTTP method is typically used for logging in a user?**
   - A) GET
   - B) POST
   - C) PUT
   - D) DELETE

   Answer: B

   Explanation: The POST method is commonly used for logging in a user as it involves sending sensitive data.

302. **What can happen if you expose your JWT secret key?**
   - A) Tokens can be forged
   - B) Tokens will expire immediately
   - C) Tokens will be more secure
   - D) None of the above

   Answer: A

   Explanation: If the secret key is exposed, attackers can forge tokens, compromising the security of the system.

303. **How should you handle password resets securely?**
   - A) Send the new password via email
   - B) Use a one-time token for verification
   - C) Allow the user to change their password directly
   - D) Do nothing

   Answer: B

   Explanation: A one-time token ensures that only the rightful user can reset their password.

304. **What is a key benefit of using JSON Web Tokens over traditional session IDs?**
   - A) JWTs require server-side storage
   - B) JWTs can be used across different domains
   - C) JWTs are always encrypted
   - D) JWTs are less secure

   Answer: B

   Explanation: JWTs can be easily shared across different domains, making them versatile for authentication in microservices.

305. **What is the impact of using the wrong algorithm to sign a JWT?**
   - A) The token will be invalid
   - B) The token will work fine
   - C) The token will always expire
   - D) The token will be secure

   Answer: A

   Explanation: Using the wrong algorithm will result in an invalid token since the verification process will fail.

306. **What is one way to enhance the security of JWTs?**
    - A) Use a static secret key
    - B) Implement token revocation lists
    - C) Make tokens longer
    - D) Reduce payload size

    Answer: B

    Explanation: Implementing token revocation lists helps manage security by allowing the invalidation of tokens.

307. **How does the OAuth 2.0 framework relate to JWTs?**
    - A) OAuth 2.0 does not use tokens
    - B) OAuth 2.0 can use JWTs as access tokens
    - C) OAuth 2.0 only uses session IDs
    - D) OAuth 2.0 is incompatible with JWTs

    Answer: B

    Explanation: OAuth 2.0 can use JWTs as access tokens to facilitate secure API access.

308. **What is a common practice to mitigate the risk of token theft?**
    - A) Use short-lived tokens
    - B) Use static tokens
    - C) Store tokens in local storage
    - D) Allow unlimited token reuse

    Answer: A

    Explanation: Using short-lived tokens reduces the risk associated with token theft.

309. **What claim type indicates that a token is meant for a specific recipient?**
    - A) aud
    - B) iss
    - C) exp
    - D) sub

    Answer: A

    Explanation: The 'aud' (audience) claim indicates the intended recipient of the token.

310. **What should be done when implementing JWT expiration?**
    - A) Set a long expiration time
    - B) Implement refresh tokens
    - C) Ignore expiration completely
    - D) Use the same expiration for all users

    Answer: B

    Explanation: Implementing refresh tokens allows for maintaining user sessions without long-lived access tokens.

311. **What is a typical approach to protecting against XSS attacks when using JWTs?**
    - A) Store tokens in local storage
    - B) Use secure, HttpOnly cookies

- C) Avoid token storage
- D) Use public tokens

  Answer: B

  Explanation: Storing tokens in secure, HttpOnly cookies helps protect them from XSS attacks.

312. **How does one typically create a JWT in Node.js?**
   - A) Using a plain string
   - B) Using the jsonwebtoken library
   - C) By handcoding the structure
   - D) Using URL encoding

     Answer: B

     Explanation: The jsonwebtoken library provides a straightforward way to create JWTs in Node.js applications.

313. **What is the function of the 'alg' parameter in a JWT header?**
   - A) To define the audience
   - B) To specify the signing algorithm
   - C) To indicate the expiration
   - D) To list claims

     Answer: B

     Explanation: The 'alg' parameter in the header specifies the algorithm used to sign the JWT.

314. **Which of the following can be a consequence of failing to validate a JWT?**
   - A) Unauthorized access
   - B) Increased security
   - C) Improved performance
   - D) No consequences

     Answer: A

     Explanation: Failing to validate a JWT can lead to unauthorized access, as tokens might not be legitimate.

315. **What should you do with an expired JWT?**
   - A) Continue using it
   - B) Refresh it with a refresh token
   - C) Store it indefinitely
   - D) Delete it without any checks

     Answer: B

     Explanation: An expired JWT can be refreshed using a refresh token to obtain a new access token.

316. **Which claim indicates the time at which the JWT was issued?**
   - A) exp
   - B) iat
   - C) nbf
   - D) sub

     Answer: B

Explanation: The 'iat' (issued at) claim indicates the time when the JWT was issued.

317. **What is a benefit of using HTTPS when transmitting JWTs?**
    - A) It prevents token expiration
    - B) It protects against man-in-the-middle attacks
    - C) It makes the token smaller
    - D) It eliminates the need for tokens

    Answer: B

    Explanation: Using HTTPS encrypts the data in transit, protecting against man-in-the-middle attacks.

318. **What is a common vulnerability related to JWTs?**
    - A) SQL injection
    - B) Token forgery
    - C) Directory traversal
    - D) Buffer overflow

    Answer: B

    Explanation: Token forgery is a risk if the secret used to sign the token is compromised.

319. **What is the maximum payload size for a JWT?**
    - A) 256 bytes
    - B) 1 KB
    - C) 8 KB
    - D) No specific limit, but smaller is better

    Answer: D

    Explanation: There is no strict limit, but it's best to keep payload sizes small for efficiency.

320. **What is the role of client-side libraries when working with JWTs?**
    - A) They create JWTs on the server
    - B) They store user credentials
    - C) They handle the storage and transmission of tokens
    - D) They manage database connections

    Answer: C

    Explanation: Client-side libraries help manage the storage and transmission of JWTs.

321. **What is the main goal of token-based authentication?**
    - A) To manage user sessions
    - B) To facilitate single sign-on (SSO)
    - C) To secure APIs
    - D) All of the above

    Answer: D

    Explanation: Token-based authentication serves various goals, including session management, SSO, and API security.

322. **Which claim can be used to specify user permissions in a JWT?**
    - A) sub

- B) exp
- C) custom claim
- D) iat

  Answer: C

  Explanation: Custom claims can be added to specify user permissions or roles within the JWT.

323. **How often should secret keys be rotated in JWT implementations?**
   - A) Never
   - B) Regularly, based on security policies
   - C) Only during major updates
   - D) Randomly

     Answer: B

     Explanation: Regularly rotating secret keys enhances security.

324. **What is a common strategy to protect JWTs in transit?**
   - A) Use unencrypted HTTP
   - B) Use HTTPS
   - C) Compress the token
   - D) Avoid using tokens

     Answer: B

     Explanation: Using HTTPS encrypts the token during transit, protecting it from interception.

325. **What should be the maximum lifetime of an access token?**
   - A) 1 hour
   - B) 24 hours
   - C) It depends on the application's security requirements
   - D) Indefinitely

     Answer: C

     Explanation: The maximum lifetime should depend on the application's specific security requirements and use case.

326. **How can you implement rate limiting for JWT authentication?**
   - A) By checking JWT expiration
   - B) By limiting login attempts
   - C) By restricting API calls based on IP address
   - D) All of the above

     Answer: D

     Explanation: Rate limiting can involve various strategies, including checking expiration, limiting logins, and restricting API calls.

327. **What is the significance of the 'jti' claim in a JWT?**
   - A) It indicates the user ID
   - B) It is a unique identifier for the token
   - C) It specifies the expiration time
   - D) It indicates the audience

     Answer: B

Explanation: The 'jti' (JWT ID) claim is a unique identifier for the token, useful for preventing token replay.

328. **What is a common method for validating a JWT on the server?**
    - A) Decode the payload only
    - B) Verify the signature and expiration
    - C) Check the issuer only
    - D) Ignore the token

    Answer: B

    Explanation: Validation typically involves verifying the signature and checking the expiration time.

329. **What should you do if a JWT is signed with a weak algorithm?**
    - A) Continue using it
    - B) Change the algorithm and reissue tokens
    - C) Store it in a database
    - D) Notify users

    Answer: B

    Explanation: It's crucial to change the algorithm to a stronger one and reissue tokens to maintain security.

330. **What is a common use case for JWTs in a microservices architecture?**
    - A) Session management
    - B) Single sign-on (SSO)
    - C) API authorization
    - D) All of the above

    Answer: D

    Explanation: JWTs are versatile and can be used for session management, SSO, and API authorization in microservices.

331. **What is the typical method for refreshing a JWT?**
    - A) By re-authenticating the user
    - B) By using a refresh token
    - C) By changing the secret key
    - D) By deleting the old token

    Answer: B

    Explanation: A refresh token is used to obtain a new access token when the original expires.

332. **Which claim indicates the audience intended for the JWT?**
    - A) exp
    - B) aud
    - C) sub
    - D) nbf

    Answer: B

    Explanation: The 'aud' (audience) claim indicates the intended audience for the JWT.

333. **What is a typical security measure for a refresh token?**
    - A) Store it in local storage

- B) Use short expiration times
- C) Store it securely in HttpOnly cookies
- D) Expose it to the client

  Answer: C

  Explanation: Refresh tokens should be stored securely in HttpOnly cookies to protect them from XSS attacks.

334. **What is the effect of using an invalid signature in a JWT?**
   - A) The token will still be valid
   - B) The token will be rejected
   - C) The token will expire immediately
   - D) None of the above

     Answer: B

     Explanation: An invalid signature will cause the token to be rejected during validation.

335. **Which JWT claim is useful for ensuring a token is only used after a specific time?**
   - A) exp
   - B) nbf
   - C) sub
   - D) iat

     Answer: B

     Explanation: The 'nbf' (not before) claim ensures that the token is not accepted before a specified time.

336. **What is the primary risk of not implementing token expiration?**
   - A) Increased server load
   - B) Tokens can be used indefinitely if stolen
   - C) Users may forget their credentials
   - D) Reduced performance

     Answer: B

     Explanation: Not implementing token expiration allows stolen tokens to be used indefinitely, posing a significant security risk.

337. **How can you effectively manage user sessions in a JWT-based system?**
   - A) Use long-lived access tokens
   - B) Implement refresh tokens
   - C) Rely solely on the access token
   - D) Ignore session management

     Answer: B

     Explanation: Implementing refresh tokens allows for better management of user sessions in a JWT-based system.

338. **What is a good practice for logging sensitive actions in applications using JWTs?**
   - A) Log all JWT payloads
   - B) Log user IDs and action types only
   - C) Do not log anything

- D) Use plain text logs

  Answer: B

  Explanation: Logging user IDs and action types helps maintain security while not exposing sensitive data.

339. **What is a common approach to mitigate CSRF attacks when using JWTs?**
   - A) Use short-lived tokens
   - B) Use anti-CSRF tokens
   - C) Store tokens in local storage
   - D) Implement CORS

     Answer: B

     Explanation: Using anti-CSRF tokens alongside JWTs helps prevent CSRF attacks.

340. **What happens if you use a token with the wrong audience?**
   - A) The token will be accepted
   - B) The token will be rejected
   - C) The token will work partially
   - D) None of the above

     Answer: B

     Explanation: A token with a wrong audience will be rejected during validation as it does not match the expected audience.

341. **What is Client-Side Rendering (CSR)?**

   A) Rendering of HTML by the web server
   B) Rendering of HTML on the browser using JavaScript
   C) Rendering of HTML on the backend
   D) Rendering of HTML through an API call
   Answer: B) Rendering of HTML on the browser using JavaScript
   Explanation: Client-Side Rendering (CSR) refers to the process of rendering content in the user's browser via JavaScript, where the browser downloads an initial blank HTML file and fetches the content dynamically.

342. **What is Server-Side Rendering (SSR)?**

   A) Rendering of HTML by JavaScript in the browser
   B) Rendering of HTML on the server before it is sent to the browser
   C) Rendering of HTML by a CDN
   D) Rendering of HTML on a database server
   Answer: B) Rendering of HTML on the server before it is sent to the browser
   Explanation: Server-Side Rendering (SSR) is when HTML content is generated on the server and sent to the client fully rendered, typically before any JavaScript is executed on the browser.

343. **Which of the following is a benefit of Client-Side Rendering?**

   A) Faster initial page load
   B) Improved SEO
   C) Reduced server load
   D) Increased page size
   Answer: C) Reduced server load

Explanation: CSR offloads most of the rendering work to the client (browser), which can reduce the load on the server.

344. **Which of the following is a drawback of Server-Side Rendering?**
A) Poor SEO
B) Slow initial load time
C) Increased client-side rendering complexity
D) Greater server resource usage
Answer: D) Greater server resource usage
Explanation: SSR requires the server to generate the HTML for each request, which can consume more server resources.

345. **Which of the following is an advantage of Server-Side Rendering?**
A) Faster dynamic content updates
B) Better SEO performance
C) Smaller initial payload
D) Reduced time to interactive (TTI)
Answer: B) Better SEO performance
Explanation: SSR generates fully-rendered HTML on the server, making it more SEO-friendly since search engines can crawl the full content without executing JavaScript.

346. **What is the main difference between CSR and SSR?**
A) CSR renders on the server, and SSR renders on the client
B) CSR loads faster than SSR
C) CSR is more SEO-friendly than SSR
D) SSR renders on the server, and CSR renders on the client
Answer: D) SSR renders on the server, and CSR renders on the client
Explanation: The key difference lies in where the rendering occurs—SSR renders content on the server, while CSR relies on JavaScript in the client (browser).

347. **Which of the following rendering techniques improves SEO the most?**
A) CSR
B) SSR
C) Static Site Generation (SSG)
D) Progressive Web Apps (PWA)
Answer: B) SSR
Explanation: SSR improves SEO because search engines can crawl fully-rendered HTML content, while CSR often requires extra effort to be SEO-friendly (like using prerendering tools).

348. **Which of the following technologies supports Client-Side Rendering?**
A) Node.js
B) React
C) Express.js
D) Django
Answer: B) React
Explanation: React is a JavaScript library that enables client-side rendering by dynamically updating the HTML content in the browser.

349. **Which of the following best describes the “Time to Interactive” (TTI) metric?**
A) The time it takes for the first byte to arrive from the server
B) The time it takes for the page to load completely
C) The time it takes for the page to become fully interactive
D) The time it takes for the browser to download all assets
Answer: C) The time it takes for the page to become fully interactive
Explanation: TTI is a critical performance metric that measures when a web page becomes usable (interactive) for users, including handling user inputs.

350. **Which of the following rendering techniques can lead to slower perceived performance for the user?**
A) Client-Side Rendering
B) Server-Side Rendering
C) Static Site Generation
D) Incremental Static Regeneration
Answer: A) Client-Side Rendering
Explanation: CSR can lead to slower perceived performance because the browser needs to download and execute JavaScript, leading to a longer wait time before the page becomes interactive.

351. **Which of the following best describes “Server-Side Rendering”?**
A) HTML is generated dynamically by JavaScript running in the browser
B) HTML is pre-rendered and delivered to the browser from the server
C) HTML is generated through client-side frameworks like Angular or React
D) HTML is served as static files directly from a CDN
Answer: B) HTML is pre-rendered and delivered to the browser from the server
Explanation: SSR involves generating the HTML content on the server before sending it to the client.

352. **Which of the following is a drawback of Client-Side Rendering?**
A) Poor SEO performance
B) Higher server load
C) Slower interaction speed
D) Difficulty in dynamic content rendering
Answer: A) Poor SEO performance
Explanation: CSR can lead to poor SEO performance because search engines might not fully execute JavaScript, meaning they can't index dynamically loaded content effectively.

353. **What does the term “hydration” refer to in the context of SSR?**
A) The process of generating HTML content on the server
B) The process of JavaScript taking over server-rendered HTML to make it interactive
C) The process of caching static content on the client-side
D) The process of compressing HTML files for faster delivery
Answer: B) The process of JavaScript taking over server-rendered HTML to make it interactive
Explanation: Hydration refers to the process in which the browser takes over server-rendered HTML and attaches JavaScript event handlers to make it interactive.

354. **Which of the following can significantly improve SEO in Client-Side Rendering (CSR)?**
A) Prerendering
B) Lazy loading
C) Service workers
D) CDN caching
Answer: A) Prerendering
Explanation: Prerendering involves generating static HTML content for CSR pages, which can be served to search engines for better SEO performance.

355. **Which framework is known for implementing Server-Side Rendering (SSR) out-of-the-box?**
A) Vue.js
B) React
C) Next.js
D) Angular
Answer: C) Next.js
Explanation: Next.js is a popular React framework that offers built-in SSR capabilities, allowing for pages to be rendered on the server.

356. **Which of the following performance improvements is associated with Server-Side Rendering?**
A) Faster initial page load
B) Less dependency on JavaScript for interactivity
C) Reduced bandwidth usage
D) Reduced server-side cache time
Answer: A) Faster initial page load
Explanation: SSR can lead to faster initial page loads because the HTML is already generated on the server and sent to the browser, reducing the time to first meaningful paint.

357. **Which of the following is NOT typically a drawback of Server-Side Rendering?**
A) High server load
B) Increased time to first byte (TTFB)
C) Poor interactivity on initial load
D) SEO limitations
Answer: D) SEO limitations
Explanation: SSR actually helps with SEO because search engines can crawl pre-rendered HTML content easily, unlike CSR which requires additional techniques to be SEO-friendly.

358. **In a typical Client-Side Rendering workflow, what happens after the HTML file is loaded in the browser?**
A) The browser requests the HTML content from the server
B) The browser downloads JavaScript files to render the page
C) The browser generates HTML content on the server
D) The browser sends an API request to load static content
Answer: B) The browser downloads JavaScript files to render the page

Explanation: In CSR, after the initial HTML file is loaded, the browser downloads JavaScript files that dynamically generate the content.

359. **Which of the following is a key feature of frameworks like Next.js and Nuxt.js?**
A) Full static site generation (SSG) only
B) Hybrid rendering with both SSR and CSR
C) Only support client-side JavaScript rendering
D) Limited SEO capabilities
Answer: B) Hybrid rendering with both SSR and CSR
Explanation: Next.js and Nuxt.js support hybrid rendering, allowing pages to be either server-side rendered (SSR) or client-side rendered (CSR) depending on the use case.

360. **Which of the following strategies can help mitigate the SEO drawbacks of Client-Side Rendering (CSR)?**
A) Using static site generation (SSG)
B) Server-side data fetching
C) Server-side rendering with hydration
D) Implementing a Service Worker
Answer: A) Using static site generation (SSG)
Explanation: SSG pre-renders the HTML at build time, which can be served to search engines, improving SEO in a similar way to SSR without the drawbacks of CSR.

361. **Which of the following is true about Static Site Generation (SSG)?**
A) Content is generated dynamically at runtime
B) Content is pre-built during the build process
C) SSG is suitable only for e-commerce websites
D) SSG does not improve performance
Answer: B) Content is pre-built during the build process
Explanation: In SSG, pages are generated at build time and are served as static files, making them fast to load and SEO-friendly.

362. **What is a common use case for Server-Side Rendering (SSR)?**
A) Single-page applications (SPAs)
B) Dynamic content-heavy websites with high SEO needs
C) Applications requiring minimal JavaScript
D) Websites with limited SEO requirements
Answer: B) Dynamic content-heavy websites with high SEO needs
Explanation: SSR is ideal for dynamic websites that need good SEO performance, as it allows content to be rendered and indexed by search engines.

363. **Which of the following techniques can help reduce the initial load time in Client-Side Rendering (CSR)?**
A) Preloading fonts and images
B) Server-side caching
C) Lazy loading of JavaScript and images
D) All of the above
Answer: D) All of the above
Explanation: Preloading essential resources, server-side caching, and lazy loading are all techniques that help optimize CSR performance by reducing initial load time.

364. **Which of the following is typically used for Server-Side Rendering in React applications?**
A) Express.js
B) Webpack
C) ReactDOMServer
D) Angular
Answer: C) ReactDOMServer
Explanation: ReactDOMServer is a Node.js package that allows React components to be rendered on the server for SSR.

365. **Which of the following methods reduces the time-to-first-byte (TTFB) in Server-Side Rendering?**
A) Using a CDN to cache static files
B) Lazy loading JavaScript
C) Using a service worker for caching
D) Rendering content asynchronously in the client
Answer: A) Using a CDN to cache static files
Explanation: A Content Delivery Network (CDN) caches static files, reducing the time it takes for the server to respond to the client request and improving TTFB.

366. **What is a template engine in web development?**
A) A tool for optimizing images
B) A tool that generates dynamic HTML content
C) A server for serving static files
D) A front-end JavaScript library
Answer: B) A tool that generates dynamic HTML content
Explanation: A template engine helps generate dynamic HTML pages by inserting dynamic data into static templates. It separates the logic from the view, making development more modular.

367. **Which of the following is an example of a template engine in Node.js?**
A) AngularJS
B) EJS
C) React
D) jQuery
Answer: B) EJS
Explanation: EJS (Embedded JavaScript) is a popular template engine for Node.js that allows you to generate HTML dynamically by embedding JavaScript logic in HTML templates.

368. **How do you install EJS in a Node.js application?**
A) `npm install ejs`
B) `npm install template-ejs`
C) `npm install ejs-template`
D) `npm install ejs-node`
Answer: A) `npm install ejs`

Explanation: To use EJS in your Node.js application, you can install it using `npm install ejs`.

369. **Which of the following is a feature of template engines like EJS?**
A) Allows embedding server-side logic into HTML
B) Automatically compiles JavaScript files
C) Manages database connections
D) Transforms CSS into HTML
Answer: A) Allows embedding server-side logic into HTML
Explanation: EJS allows embedding JavaScript logic (like loops and conditions) inside HTML templates, making it easy to generate dynamic content.

370. **Which of the following is the correct syntax to include a partial view in EJS?**
A) `<%= include('partial') %>`
B) `<%- partial('partial') %>`
C) `<%= render('partial') %>`
D) `<%- include('partial') %>`
Answer: D) `<%- include('partial') %>`
Explanation: The correct syntax for including partials in EJS templates is `<%- include('partial') %>`. The `%-` ensures that the content is rendered without escaping any HTML tags.

371. **What is the purpose of EJS partials?**
A) To store reusable pieces of data
B) To split large templates into smaller, reusable chunks
C) To compile templates on the server
D) To handle database queries in views
Answer: B) To split large templates into smaller, reusable chunks
Explanation: EJS partials allow developers to break down larger templates into smaller, reusable pieces, making the code cleaner and easier to maintain.

372. **Which of the following is used to pass data to an EJS template?**
A) `res.send(data)`
B) `res.render('template', data)`
C) `res.write(data)`
D) `res.json(data)`
Answer: B) `res.render('template', data)`
Explanation: The `res.render('template', data)` function is used to render an EJS template and pass the dynamic data to it in a Node.js Express application.

373. **Which of the following is the correct way to access data passed to an EJS template?**
A) `<%= data.property %>`
B) `<%= property.data %>`
C) `<% data.property %>`
D) `<%- data.property %>`

Answer: A) `<%= data.property %>`

Explanation: In EJS, to access a property passed from the server, you use the syntax `<%= data.property %>`. This will render the value of the property.

374. **How can you pass data from a route to an EJS view in Express.js?**

A) `app.get('/', (req, res) => res.render('index', { data: 'Hello World' }))`

B) `app.get('/', (req, res) => res.send('index', { data: 'Hello World' }))`

C) `app.get('/', (req, res) => res.json({ data: 'Hello World' }))`

D) `app.get('/', (req, res) => res.render('index', 'data': 'Hello World'))`

Answer: A) `app.get('/', (req, res) => res.render('index', { data: 'Hello World' }))`

Explanation: The correct way to pass data to a view is by using `res.render('index', { data: 'Hello World' })` in your route handler.

375. **Which of the following is the correct way to render a template with a specific view engine in Express.js?**

A) `app.set('view engine', 'ejs')`

B) `app.use('view engine', 'ejs')`

C) `app.view('ejs')`

D) `app.set('template engine', 'ejs')`

Answer: A) `app.set('view engine', 'ejs')`

Explanation: In Express.js, you use `app.set('view engine', 'ejs')` to set EJS as the view engine for rendering templates.

376. **What is the purpose of the `include` function in EJS?**

A) To include a file from the file system

B) To add dynamic data to a template

C) To handle asynchronous operations

D) To add external CSS files to the page

Answer: A) To include a file from the file system

Explanation: The `include` function in EJS allows you to include other templates (partials) in the current template.

377. **How do you pass an object to an EJS view in Node.js?**

A) `{ name: 'John', age: 25 }`

B) `name = 'John'; age = 25;`

C) `res.render('view', { name: 'John', age: 25 })`

D) `res.view('view', { name: 'John', age: 25 })`

Answer: C) `res.render('view', { name: 'John', age: 25 })`

Explanation: In Express, to pass an object to an EJS view, you use the `res.render()` method, passing the view name and the object containing the data.

378. **Which of the following is used to escape special HTML characters in EJS?**

A) `<%= %>`

B) `<%- %>`

C) `<%= <%- %>`

D) `<% == %>`

Answer: A) `<%= %>`

Explanation: The `<%= %>` syntax in EJS escapes special characters to prevent XSS attacks, ensuring that HTML is properly rendered.

379. **What does the `<%- %>` syntax do in EJS?**

A) It escapes HTML characters

B) It directly injects unescaped HTML into the view

C) It outputs raw JavaScript

D) It calls a function inside the view

Answer: B) It directly injects unescaped HTML into the view

Explanation: The `<%- %>` syntax in EJS is used for injecting raw HTML without escaping it, useful for embedding HTML markup.

380. **Which EJS feature is used to pass reusable components between multiple views?**

A) Partials

B) Templates

C) Controllers

D) Views

Answer: A) Partials

Explanation: Partials are reusable EJS templates that can be included in other views, making the code more modular and reducing redundancy.

381. **Which of the following commands is used to start a Node.js application after creating it?**

A) `node app.js`

B) `npm start`

C) `npm run app`

D) `node start.js`

Answer: A) `node app.js`

Explanation: Once your Node.js application is created, you can start it using `node app.js`, where `app.js` is the entry file for the application.

382. **How do you handle form data in an Express.js app when using EJS?**

A) Use `req.body` to access form data

B) Use `req.query` to access form data

C) Use `res.body` to access form data

D) Use `res.query` to access form data

Answer: A) Use `req.body` to access form data

Explanation: In Express.js, form data sent via POST requests can be accessed using `req.body`, assuming the body parser middleware is set up.

383. **Which EJS feature allows the insertion of JavaScript code in templates?**

A) `forEach` loop

B) `<% %>`

C) `{{ }}`

D) `<%% %>`

Answer: B) `<% %>`

Explanation: The `<% %>` syntax allows you to embed JavaScript code (like loops, conditionals, etc.) directly into EJS templates.

384. **How do you render a view in Express using EJS with dynamic data?**

A) `res.render('view', { name: 'John', age: 25 })`

B) `res.view('view', { name: 'John', age: 25 })`

C) `res.html('view', { name: 'John', age: 25 })`

D) `res.response('view', { name: 'John', age: 25 })`

Answer: A) `res.render('view', { name: 'John', age: 25 })`

Explanation: To render a view with dynamic data, you use `res.render('view', { name: 'John', age: 25 })`.

385. **Which EJS tag is used for conditional statements like `if` and `else`?**

A) `<% if %>`

B) `<% if condition %>`

C) `<%= if condition %>`

D) `<%: if condition %>`

Answer: B) `<% if condition %>`

Explanation: In EJS, you use `<% if condition %>` to write conditional statements in your templates.

386. **How do you pass dynamic data to partials in EJS?**

A) `<%- include('partial', { data: data }) %>`

B) `<%= include('partial', { data: data }) %>`

C) `<% include('partial', { data: data }) %>`

D) `<%- partial('partial', { data: data }) %>`

Answer: A) `<%- include('partial', { data: data }) %>`

Explanation: To pass dynamic data to a partial in EJS, you use `<%- include('partial', { data: data }) %>`, where `data` is the object being passed.

387. **Which of the following is true about passing data to views in Express.js?**

A) Data can be passed to views through the route handler

B) Data can only be passed through query strings

C) Data cannot be passed to views in Express

D) Data must be stored in global variables

Answer: A) Data can be passed to views through the route handler

Explanation: Data is passed to views through the route handler using `res.render()` in Express.js.

388. **What is the primary advantage of using partials in EJS?**

A) It improves security by escaping HTML

B) It allows for dynamic content generation

C) It helps in reusing HTML code across different templates

D) It compiles JavaScript faster

Answer: C) It helps in reusing HTML code across different templates

Explanation: Partials allow you to reuse common components (like headers, footers) across multiple templates, making the code more modular and maintainable.

389. **Which of the following is used to embed unescaped HTML in an EJS view?**

A) `<%= %>`

B) `<%- %>`

C) `<% %>`

D) `<%% %>`

Answer: B) `<%- %>`

Explanation: The `<%- %>` syntax is used to inject raw, unescaped HTML into an EJS template.

390. **What is the purpose of the `app.set('view engine', 'ejs')` in Express.js?**

A) It configures the template engine to use EJS for rendering views

B) It sets the view directory for the application

C) It compiles JavaScript files

D) It configures the database connection

Answer: A) It configures the template engine to use EJS for rendering views

Explanation: The `app.set('view engine', 'ejs')` line in Express.js configures EJS as the view engine for the application.

391. **How do you include a JavaScript file in an EJS template?**

A) `<script src="file.js"></script>`

B) `<script include="file.js"></script>`

C) `<%- include('file.js') %>`

D) `<%= include('file.js') %>`

Answer: A) `<script src="file.js"></script>`

Explanation: To include a JavaScript file in an EJS template, you use the regular HTML `<script>` tag with the `src` attribute pointing to the file location.

392. **Which of the following is used to ensure that the data passed into EJS is not escaped (used for raw HTML)?**

A) `<%= %>`

B) `<%- %>`

C) `<%# %>`

D) `<%% %>`

Answer: B) `<%- %>`

Explanation: The `<%- %>` tag in EJS is used to inject raw, unescaped HTML into the output.

393. **Which of the following is true about the EJS `res.render()` function?**

A) It directly sends a response to the browser without rendering a template
B) It sends the rendered HTML from a template to the browser
C) It renders a template and sends a JSON response
D) It fetches and sends data from a database

Answer: B) It sends the rendered HTML from a template to the browser

Explanation: `res.render()` is used in Express to render an EJS template with data and send the rendered HTML as a response.

394. **What is the purpose of the `res.send()` function in Express.js?**

A) To send an HTML response to the client
B) To send a static file to the client
C) To send data to a database
D) To send a JSON response to the client

Answer: A) To send an HTML response to the client

Explanation: `res.send()` is used in Express to send an HTML response to the client.

395. **Which of the following is a common use case for EJS partials?**

A) Reusing HTML templates like headers and footers across multiple views
B) Caching the response data for faster page loading
C) Handling authentication logic
D) Managing API requests

Answer: A) Reusing HTML templates like headers and footers across multiple views

Explanation: Partials are commonly used to reuse components like headers, footers, and navigation across multiple views, ensuring consistency and reducing code duplication.

396. **How do you link a CSS file in an EJS template?**

A) `<link src="styles.css" rel="stylesheet">`
B) `<link href="styles.css" rel="stylesheet">`
C) `<%- include('styles.css') %>`
D) `<%= include('styles.css') %>`

Answer: B) `<link href="styles.css" rel="stylesheet">`

Explanation: The correct way to link a CSS file in an EJS template is by using the `<link>` tag with the `href` attribute pointing to the CSS file.

397. **Which of the following is the proper syntax to define a loop in EJS?**

A) `<% for (let i = 0; i < 5; i++) { %> ... <% } %>`
B) `<% loop (i = 0; i < 5; i++) { %> ... <% endloop %>`
C) `<% each i in range(5) { %> ... <% } %>`
D) `<% repeat (i = 0; i < 5; i++) { %> ... <% %>`

Answer: A) `<% for (let i = 0; i < 5; i++) { %> ... <% } %>`

Explanation: EJS allows embedding JavaScript code, including loops. The correct

syntax for a `for` loop in EJS is `<% for (let i = 0; i < 5; i++) { %> ... <% } %>`.

398. **How can you pass an array to an EJS template?**

A) `res.render('template', { array: [1, 2, 3] })`

B) `res.array('template', [1, 2, 3])`

C) `res.render('template', array: [1, 2, 3])`

D) `res.send('template', [1, 2, 3])`

Answer: A) `res.render('template', { array: [1, 2, 3] })`

Explanation: You pass an array to an EJS template through the `res.render()` method by including it in an object.

399. **What is the correct way to use a conditional statement in EJS?**

A) `<%= if (condition) %> ... <% endif %>`

B) `<% if (condition) { %> ... <% } %>`

C) `<%= if (condition) then %> ... <% %>`

D) `<% if condition then %> ... <% %>`

Answer: B) `<% if (condition) { %> ... <% } %>`

Explanation: EJS allows you to embed JavaScript code. The correct syntax for an `if` statement is `<% if (condition) { %> ... <% } %>`.

400. **Which method in Express.js is used to serve static files like CSS and images?**

A) `app.static()`

B) `app.serve()`

C) `app.use(express.static())`

D) `app.serveFiles()`

Answer: C) `app.use(express.static())`

Explanation: The `app.use(express.static())` method in Express is used to serve static files like images, CSS, and JavaScript from a specific directory.

401. **Which of the following is a valid way to set up an EJS view engine in Express?**

A) `app.set('view engine', 'html')`

B) `app.set('view engine', 'pug')`

C) `app.set('view engine', 'ejs')`

D) `app.set('template engine', 'ejs')`

Answer: C) `app.set('view engine', 'ejs')`

Explanation: In Express, you set EJS as the view engine by using `app.set('view engine', 'ejs')`.

402. **Which of the following is the correct way to render a template with a layout in EJS?**

A) `res.render('layout', { body: 'template' })`

B) `res.render('layout', { content: 'template' })`

C) `res.render('template', { layout: 'layout' })`

D) `res.render('template', { view: 'layout' })`

Answer: A) `res.render('layout', { body: 'template' })`

Explanation: A common way to render a layout with EJS is to render the layout and pass the content as a variable (e.g., `body`) to be injected into the layout.

403. **What is the primary role of the `app.set()` function in Express.js?**

A) To set configurations such as view engine or port

B) To define middleware for handling requests

C) To handle database connections

D) To manage static files

Answer: A) To set configurations such as view engine or port

Explanation: The `app.set()` method in Express is used for setting various application-level settings, such as the view engine and the port number.

404. **Which EJS function is used to include the contents of another file into the current template?**

A) `include()`

B) `render()`

C) `load()`

D) `import()`

Answer: A) `include()`

Explanation: The `include()` function in EJS is used to embed the contents of one template into another, such as for reusable partials.

405. **Which of the following is the correct way to include a partial in an EJS template?**

A) `<%- include('partial') %>`

B) `<%- partial('partial') %>`

C) `<%= include('partial') %>`

D) `<%= partial('partial') %>`

Answer: A) `<%- include('partial') %>`

Explanation: The correct syntax for including a partial in EJS is `<%- include('partial') %>`, where `'partial'` refers to the file name of the partial.

406. **How do you handle form submission data in EJS templates?**

A) By accessing `req.body` in the route handler

B) By using the `form-data` attribute in EJS

C) By including a `submit()` function in EJS

D) By embedding HTML form elements inside EJS code

Answer: A) By accessing `req.body` in the route handler

Explanation: Form submission data is typically accessed in Express.js via `req.body`, which contains the data sent through a POST request. You would use this data in your route handler.

407. **What is the recommended way to avoid JavaScript injection vulnerabilities in EJS?**

A) Always use `<%- %>` for outputting data

B) Escape user-generated content with `<%= %>`

C) Avoid using `<%= %>` tags in any templates

D) Only use server-side rendered data

Answer: B) Escape user-generated content with `<%= %>`

Explanation: Using `<%= %>` ensures that any special characters in the data (like < or >) are escaped, preventing JavaScript injection attacks.

408. **Which of the following is correct for using EJS as the view engine with Express.js?**

A) `express().set('ejs')`

B) `express().set('view engine', 'ejs')`

C) `express().use('ejs')`

D) `express().set('template', 'ejs')`

Answer: B) `express().set('view engine', 'ejs')`

Explanation: In Express, you set EJS as the view engine using `express().set('view engine', 'ejs')`.

409. **How do you configure the layout of EJS templates?**

A) By using a layout middleware like `express-ejs-layouts`

B) By manually including the header and footer in each view

C) By setting a global layout file in `app.set()`

D) By passing the layout option to `res.render()`

Answer: A) By using a layout middleware like `express-ejs-layouts`

Explanation: The `express-ejs-layouts` middleware helps manage layouts in EJS, allowing you to define a single layout template that can be reused across multiple views.

410. **Which of the following commands installs the `express-ejs-layouts` package?**

A) `npm install express-ejs-layouts`

B) `npm install ejs-layouts`

C) `npm install express-layouts`

D) `npm install layout-engine`

Answer: A) `npm install express-ejs-layouts`

Explanation: The `express-ejs-layouts` package is installed using `npm install express-ejs-layouts` to support layouts in your EJS templates.

411. **How do you use a partial to render a header in EJS?**

A) `<%- include('header') %>`

B) `<%= include('header') %>`

C) `<%- render('header') %>`

D) `<%= partial('header') %>`

Answer: A) `<%- include('header') %>`

Explanation: The correct syntax for including a header partial in EJS is `<%- include('header') %>`. The `<%-` ensures that unescaped HTML is injected.

412. **What is the default file extension for EJS template files?**

A) `.html`

B) `.ejs`

C) `.js`

D) `.tpl`

Answer: B) `.ejs`

Explanation: The default file extension for EJS templates is `.ejs`.

413. **Which of the following does not require a layout file to be passed in EJS?**

A) `express-ejs-layouts`

B) Using the `app.set()` method

C) `res.render()`

D) `app.use(express.static())`

Answer: D) `app.use(express.static())`

Explanation: `app.use(express.static())` serves static files (like CSS, JS, images) and does not require a layout to be passed to EJS templates.

414. **In EJS, which symbol is used to output raw HTML without escaping it?**

A) `<%= %>`

B) `<%- %>`

C) `<%# %>`

D) `<% %>`

Answer: B) `<%- %>`

Explanation: The `<%- %>` syntax is used in EJS to inject raw HTML content, without escaping any special characters.

415. **How do you configure the view engine to use EJS in an Express.js application?**

A) `app.set('view', 'ejs')`

B) `app.set('view engine', 'ejs')`

C) `app.use('view engine', 'ejs')`

D) `app.view('ejs')`

Answer: B) `app.set('view engine', 'ejs')`

Explanation: The correct way to configure EJS as the view engine in Express is `app.set('view engine', 'ejs')`.

416. **What is a layout file used for in EJS?**

A) To render all content dynamically

B) To manage repetitive content like headers, footers, and sidebars

C) To manage static files

D) To process user input forms

Answer: B) To manage repetitive content like headers, footers, and sidebars

Explanation: Layouts in EJS allow you to manage common content like headers, footers, and navigation bars across multiple views.

417. **Which Express middleware is commonly used to handle static assets (CSS, JS, images) in an EJS application?**

A) `express.static()`

B) `express.staticMiddleware()`

C) `express.serveStatic()`

D) `express.asset()`

Answer: A) `express.static()`

Explanation: `express.static()` is used in Express to serve static files such as CSS, JavaScript, and images from a designated folder.

418. **How do you pass an array of items to an EJS template?**

A) `res.render('template', { items: ['item1', 'item2'] })`

B) `res.render('template', items: ['item1', 'item2'])`

C) `res.send('template', { items: ['item1', 'item2'] })`

D) `res.view('template', { items: ['item1', 'item2'] })`

Answer: A) `res.render('template', { items: ['item1', 'item2'] })`

Explanation: To pass an array to an EJS template, you use `res.render()` and provide an object with the array.

419. **Which EJS syntax would you use to render a loop?**

A) `<% forEach(data) { %> ... <% } %>`

B) `<% for (i = 0; i < data.length; i++) { %> ... <% } %>`

C) `<%= loop(data) %>`

D) `<%- for (i in data) { %> ... <% } %>`

Answer: B) `<% for (i = 0; i < data.length; i++) { %> ... <% } %>`

Explanation: In EJS, a loop can be written using regular JavaScript syntax, such as `<% for (i = 0; i < data.length; i++) { %> ... <% } %>`.

420. **What does the `<%= %>` EJS syntax do?**

A) Executes a block of JavaScript code

B) Outputs a JavaScript expression value into HTML, escaping HTML characters

C) Renders raw HTML into the view

D) Outputs an HTML string directly

Answer: B) Outputs a JavaScript expression value into HTML, escaping HTML characters

Explanation: The `<%= %>` syntax is used to output the value of a JavaScript expression into the view, escaping any special HTML characters to avoid security vulnerabilities.

421. **How do you pass dynamic data to an EJS partial?**

A) `<%- include('partial', { name: 'John' }) %>`

B) `<%- include('partial') { name: 'John' } %>`

C) `<%= include('partial', { name: 'John' }) %>`

D) `<%- partial('partial', { name: 'John' }) %>`

Answer: A) `<%- include('partial', { name: 'John' }) %>`

Explanation: The correct way to pass dynamic data to an EJS partial is to use `<%- include('partial', { name: 'John' }) %>`, where the data is passed as an object.

422. **What type of data can be passed to an EJS template?**

A) Strings and numbers

B) Arrays and objects

C) Boolean values

D) All of the above

Answer: D) All of the above

Explanation: EJS templates can accept all types of data—strings, numbers, arrays, objects, and booleans—as long as they are passed as part of an object.

423. **How can you dynamically load a different EJS template within a layout?**

A) By using `res.render()` in the layout

B) By manually including a template tag in the layout

C) By including a `yield` statement in the layout

D) By including a `content` variable in the layout

Answer: D) By including a `content` variable in the layout

Explanation: You can pass a variable (e.g., `content`) to the layout and render it dynamically based on which view is being requested.

424. **Which method is used to load the EJS view engine in Express.js?**

A) `app.set('view', 'ejs')`

B) `app.set('view engine', 'ejs')`

C) `app.render('ejs')`

D) `app.use('view engine', 'ejs')`

Answer: B) `app.set('view engine', 'ejs')`

Explanation: To load EJS as the view engine in Express, you use `app.set('view engine', 'ejs')`.

425. **What is the purpose of using the `express-ejs-layouts` middleware in Express?**

A) To manage database layouts

B) To use EJS with an MVC architecture

C) To handle routing in the application

D) To allow the use of layouts across multiple EJS views

Answer: D) To allow the use of layouts across multiple EJS views

Explanation: The `express-ejs-layouts` middleware allows you to define a layout template and apply it to multiple EJS views, ensuring consistent structure and reducing redundancy.

426. **How do you configure the `express-ejs-layouts` middleware in Express?**

A) `app.use(express.ejsLayouts())`

B) `app.use('express-ejs-layouts')`

C) `app.set('view engine', 'ejs-layouts')`

D) `app.use(require('express-ejs-layouts'))`

Answer: D) `app.use(require('express-ejs-layouts'))`

Explanation: To use `express-ejs-layouts` middleware, you need to first install it using `npm install express-ejs-layouts` and then add it to your Express application with `app.use(require('express-ejs-layouts'))`.

427. **What is the purpose of `res.render()` in an Express.js route?**

A) To execute JavaScript in the browser

B) To send a response directly without rendering a template

C) To render a view template and send it as a response

D) To render a static HTML file

Answer: C) To render a view template and send it as a response

Explanation: The `res.render()` method in Express is used to render an EJS template with dynamic data and send the rendered HTML as a response to the client.

428. **Which of the following methods is used to serve static files in Express?**

A) `app.serveStatic()`

B) `app.use(express.static())`

C) `app.serve()`

D) `app.static()`

Answer: B) `app.use(express.static())`

Explanation: `app.use(express.static())` is the method used in Express to serve static assets such as CSS, JavaScript, and images.

429. **Which EJS syntax is used to include an external JavaScript file inside a template?**

A) `<%- include('script.js') %>`

B) `<script src="script.js"></script>`

C) `<%= include('script.js') %>`

D) `<%- src('script.js') %>`

Answer: B) `<script src="script.js"></script>`

Explanation: You include a JavaScript file in an EJS template by using the regular HTML `<script>` tag with the `src` attribute.

430. **How do you pass multiple pieces of dynamic data to an EJS template?**

A) `res.render('template', { name: 'John', age: 30 })`

B) `res.send('template', { name: 'John', age: 30 })`

C) `res.render('template', name: 'John', age: 30)`

D) `res.view('template', { name: 'John', age: 30 })`

Answer: A) `res.render('template', { name: 'John', age: 30 })`

Explanation: Multiple pieces of dynamic data are passed to an EJS template by including them as properties in an object, e.g., `{ name: 'John', age: 30 }`.

431. **Which method is used to handle form submissions in Express?**

A) `app.submit()`

B) `app.get()`

C) `app.post()`

D) `app.form()`

Answer: C) `app.post()`

Explanation: In Express, the `app.post()` method is used to handle form submissions made with the POST method.

432. **What does the EJS `<%- %>` tag do?**

A) It escapes HTML output

B) It outputs a JavaScript value into HTML

C) It injects raw HTML without escaping

D) It renders a partial template

Answer: C) It injects raw HTML without escaping

Explanation: The `<%- %>` tag is used to inject unescaped raw HTML into an EJS template, useful for rendering HTML from user input.

433. **What is the output when you use the EJS tag `<%= name %>` with `name = "<script>alert('hello');</script>"`?**

A) `<script>alert('hello');</script>`

B) `&lt;script&gt;alert('hello');&lt;/script&gt;`

C) `&lt;script&gt;alert('hello');</script>`

D) `alert('hello')`

Answer: B) `&lt;script&gt;alert('hello');&lt;/script&gt;`

Explanation: The `<%= %>` tag escapes any special HTML characters, so the output will be the escaped version of the script tag (`&lt;` and `&gt;`).

434. **How do you access a variable in an EJS template passed from the Express route?**

A) By referencing the variable directly with its name, e.g., `<%= name %>`

B) By using `{{ name }}` syntax

C) By using `{{ object.name }}` syntax

D) By using `[[ name ]]` syntax

Answer: A) By referencing the variable directly with its name, e.g., `<%= name %>`

Explanation: In EJS, variables passed from the route handler can be accessed directly by their name using the `<%= %>` syntax.

435. **How do you include a CSS file in an EJS template?**

A) `<%- include('style.css') %>`

B) `<link href="style.css" rel="stylesheet">`

C) `<style>include('style.css')</style>`

D) `<%= include('style.css') %>`

Answer: B) `<link href="style.css" rel="stylesheet">`

Explanation: To include a CSS file in an EJS template, use the standard HTML `<link>` tag with the `href` attribute pointing to the CSS file location.

436. **Which of the following is the correct way to include an external JavaScript file inside an EJS template?**

A) `<%- include('script.js') %>`

B) `<script src="script.js"></script>`

C) `<%= include('script.js') %>`

D) `<%- src('script.js') %>`

Answer: B) `<script src="script.js"></script>`

Explanation: To include an external JavaScript file in an EJS template, you would use the standard HTML `<script>` tag with the `src` attribute pointing to the JavaScript file.

437. **What is the correct syntax for passing data to a partial in EJS?**

A) `<%- include('partial', { title: 'Home' }) %>`

B) `<%= include('partial', title: 'Home') %>`

C) `<%- include('partial') %>`

D) `<%- include('partial', 'title: Home') %>`

Answer: A) `<%- include('partial', { title: 'Home' }) %>`

Explanation: The correct way to pass data to a partial is by including the second argument with an object, such as `{ title: 'Home' }`.

438. **Which of the following can be used to display a variable inside an EJS template without escaping HTML characters?**

A) `<%= variable %>`

B) `<%- variable %>`

C) `<% variable %>`

D) `<%= raw(variable) %>`

Answer: B) `<%- variable %>`

Explanation: The `<%- %>` tag is used to inject raw HTML without escaping the content, which can be useful for rendering HTML elements from variables.

439. **Which of the following options is used to manage layouts with EJS in an Express application?**

A) `express-layouts`

B) `ejs-layouts`

C) `express-ejs-layouts`

D) `ejs-manager`

Answer: C) `express-ejs-layouts`

Explanation: `express-ejs-layouts` is the middleware used in Express to manage layouts across multiple EJS templates, enabling you to reuse headers, footers, and other sections.

440. **How do you specify a layout template in an EJS file when using `express-ejs-layouts`?**

A) `res.render('page', { layout: 'main' })`
B) `res.layout('page', { layout: 'main' })`
C) `res.render('page')` with the layout defined in the `app.set()`
D) `res.set('layout', 'main')`
Answer: C) `res.render('page')` with the layout defined in the `app.set()`
Explanation: In `express-ejs-layouts`, you define the layout globally in Express using `app.set('layout', 'layout-name')`, and then render views with `res.render('page')`.

441. **What is the default layout file name in `express-ejs-layouts` if not specified?**
A) `layout.ejs`
B) `default.ejs`
C) `main.ejs`
D) `index.ejs`
Answer: A) `layout.ejs`
Explanation: By default, `express-ejs-layouts` uses a file named `layout.ejs` as the layout file, unless you configure it otherwise.

442. **How do you render a page with a specific layout in `express-ejs-layouts`?**
A) `res.render('page', { layout: 'specificLayout' })`
B) `res.layout('page', { layout: 'specificLayout' })`
C) `res.render('page', { useLayout: 'specificLayout' })`
D) `res.render('page')` and specify the layout in the `app.set()` method
Answer: A) `res.render('page', { layout: 'specificLayout' })`
Explanation: You can render a page with a specific layout by passing the layout name in the `res.render()` method, like `{ layout: 'specificLayout' }`.

443. **In Express, what does the `app.use()` method do?**
A) It is used to set global variables for templates
B) It allows you to define middleware functions for request handling
C) It serves static files to the client
D) It defines routes for a web server
Answer: B) It allows you to define middleware functions for request handling
Explanation: The `app.use()` method is used to define middleware functions that are executed during the request-response cycle.

444. **What happens if you try to render an EJS view without setting a view engine in Express?**
A) It will default to rendering a Pug view
B) It will render the view as plain HTML
C) It will throw an error
D) It will render the view as a static file
Answer: C) It will throw an error
Explanation: If you attempt to render an EJS view without setting the view engine in

Express, it will throw an error because Express doesn't know which engine to use for rendering.

445. **How do you pass a dynamic JavaScript object to an EJS template?**

A) `res.render('template', { object })`

B) `res.send('template', { object: object })`

C) `res.render('template', object)`

D) `res.view('template', { object })`

Answer: A) `res.render('template', { object })`

Explanation: In Express, dynamic JavaScript objects are passed to EJS templates using `res.render()` with an object literal, e.g., `{ object }`.

446. **In EJS, how would you conditionally display content based on a variable?**

A) `<%- if (condition) { %> ... <% } %>`

B) `<% if (condition) { %> ... <% else %> ... <% } %>`

C) `<%= if (condition) { %> ... <% } %>`

D) `<%- condition ? "..." : "..." %>`

Answer: B) `<% if (condition) { %> ... <% else %> ... <% } %>`

Explanation: EJS uses the standard JavaScript syntax for conditionals, and the correct syntax to conditionally display content is `<% if (condition) { %> ... <% else %> ... <% } %>`.

447. **What should you use to render dynamic content inside an HTML element in EJS?**

A) `<%- content %>`

B) `<%= content %>`

C) `<%= raw(content) %>`

D) `<%- content %>` for raw HTML

Answer: B) `<%= content %>`

Explanation: The `<%= %>` tag is used to insert dynamic content inside HTML elements. It automatically escapes HTML special characters to prevent injection attacks.

448. **What is the advantage of using partials in EJS templates?**

A) They allow reusing JavaScript code across multiple templates

B) They allow reusing HTML structures, like headers and footers, across multiple views

C) They allow you to conditionally render templates

D) They can help manage static file dependencies

Answer: B) They allow reusing HTML structures, like headers and footers, across multiple views

Explanation: Partials help to avoid code duplication by allowing HTML structures like headers and footers to be reused across multiple EJS templates.

449. **How do you prevent a JavaScript variable from being HTML-encoded in EJS?**

A) Use `<%= %>` instead of `<%- %>`

B) Use `<%- %>` instead of `<%= %>`

C) Use `<%= raw() %>`

D) Use {{ }} for JavaScript injection

Answer: B) Use <%- %> instead of <%= %>

Explanation: The <%- %> tag in EJS outputs raw HTML, meaning it doesn't escape characters like < and >, which is useful when you want to inject raw JavaScript or HTML code.

450. **How can you render a specific EJS template inside a layout in Express?**

A) Use `res.render('template', { layout: 'layout.ejs' })`

B) Define the layout globally using `app.set()`

C) Use `res.include('template', { layout: 'layout.ejs' })`

D) Specify the layout inside the template itself

Answer: B) Define the layout globally using `app.set()`

Explanation: In Express with `express-ejs-layouts`, layouts are defined globally using `app.set('layout', 'layoutName')`, and the specific templates are rendered using `res.render()`.