Full Stack Semester Suggestions

1. Explain how to develop a basic web server using the HTTP module.
2. Define the url module in Node.js and describe how to parse and manipulate URL in example.
3. Demonstrate how to read the properties of the request object in the HTTP module.
4. Experiment with reading the header and response with res.write and res.end to produce response in a HTTP server.
5. Apply HTTP status code and identify some common ones used in Node.
6. Illustrate how to serve internal and external HTML file using the HTTP module.
7. Examine the technique used to handle different routes in an HTTP server.
8. Identify and list the properties of request object in Node.js
9. Analyze the use of core node debugger and explain how it can be accessed?
10. Categories an outline the essentials and debugging commands in the node debugger highlighting their key functionalities.
11. Develop and understanding of the event emitter class in node JS and its significance in event driven programming.
12. Write down the procedure for creating and triggering a custom event using the event emitter class in node JS.
13. Extend the functionalities of express and describe the factors that contributes to its popularity in web development.
14. Analyze the concept of middleware in express evaluates the different types available and explain the process of configuring routes in express JS.
15. Summarize the role of the express.router() function in express and how its helps in organizing and modularization of routing in web app.
16. Choose the correct code to apply when creating a basic web server using express.js : const app = express();
17. Choose the correct method for applying middleware globally in an express.js : app.use(middlewareFunc);
18. Choose the right way to handle a custom 404 page in express.js app by selecting the correct code: app.use((req, res) => {res.status(404).send('Not Found');})
19. Choose the right method to retrieve a URL query parameter from an express.js route : req.query();
20. Correct the method for retrieve a URL parameters from an express.js which option is accurate : req.params();
21. Identify the method used to pass control to the next middleware when the modifying the request object : next();
22. Identify the express.js middleware use for parsing incoming JSON request : express.json();

23. Choose the correct method to register an event listener that will only run once in NodeJS : event.once(event_name, callback);
24. Choose the correct method to apply middleware to a specific route in express.js : app.get('/', middleware_function, handler_function);
25. Select the correct method to handle get request for a route in express.js : app.get('/; callback);
26. Choose the correct method to implement a route that accept from data submission in express : app.post('/', callback);
27. Choose the correct method to apply middleware in express.js : app.use(req, res, next), callback});
28. Identify the correct the command that is used to inspect the value of a variable at the NodeJS debugger: use it to execute the next statement
29. Choose the method to implement a custom middleware function in express.js :
    app.use((req, res, next) => {
    next();
    })
30. Identify the correct command to execute an application in debug mode to trace function call: run = node inspect app.js
31. Identify the command that is used to inspect a value of a variable in the NodeJS debugger: use it to execute the next statement
32. Which debugging technique should you apply to monitor changes in a variable value: watch
33. Identify the correct description of middleware in express.js: a function that handle incoming http request
34. Select the correct method for defining for route handlers in express.js: app.get(), app.post(), app.put(), app.delete()
35. select the correct description of what this code does:
    app.post('/submit', (req, res)=> {
    res.status(201).send("Created");
    });

    : handle post request and send a response with status 201
36. Select the correct method provided by the event emitter class: on(), of(), emit(), addListener()
37. Select the correct description of an event loop in NodeJS: a loop that waits for events and handle them asynchronously
38. Select the correct order of middleware execution when a request is made to an express app: global middleware, route specific middleware
39. What is the purpose of response.render() in express.js: it renders html template and send then to the client as a response
40. Select a basic syntax of a middleware function in express.js:
    app.use(function (req, res, next) {

})

41. Select the error handling middleware signature in express.js: (error, req, res, next)
42. Which of the following statement is true about node.js debugging: it can be used to debug both synchronous and asynchronous code.