Basic Level

1. **Hello World Server**
   Create a simple HTTP server that responds with "Hello World".

2. **URL Routing Example**
   Handle different URLs (e.g., /home, /about) using http module.

3. **Serving HTML File**
   Read and serve an HTML file using fs and http modules.

4. **Command-Line Argument Reader**
   Access and display command-line arguments using process.argv.

5. **Simple Calculator API**
   Create a REST API that performs add, subtract, multiply, divide.

6. **File Creation and Reading**
   Create a text file and read its content using fs.writeFile and fs.readFile.

7. **Append Data to File**
   Append user input to a file using fs.appendFile.

8. **Delete File Example**
   Delete a file using fs.unlink.

9. **Synchronous vs Asynchronous File Read**
   Show the difference between blocking and non-blocking file reading.

10. **Custom Event Emitter**
    Use events module to create and emit custom events.

⚙ **Intermediate Level**

11. **Simple Express Server**
    Create a basic server using Express.js with a few routes.

12. **Route Parameters and Query Strings**
    Capture dynamic route parameters and query strings in Express.

13. **Serving Static Files (CSS, Images)**
    Use express.static() to serve static content.

14. **Form Data Handling (POST Request)**
    Accept form data using body-parser and handle it in Express.

15. **JSON Response API**
    Create a route that returns a JSON response (e.g., user profile).

16. **Middleware Example**
    Create a custom middleware to log request details.

17. **API with CRUD Operations (In-Memory)**
    Build Create, Read, Update, Delete routes for a fake user list.

18. **Using Nodemon for Live Reloading**
Install and use nodemon to auto-restart server on file changes.

19. **Reading Environment Variables (.env)**
Use dotenv package to read and use environment variables.

20. **Simple Logger Using Streams**
Log incoming requests to a file using streams.

💾 **Advanced Level**

21. **REST API with MongoDB (Mongoose)**
Create full CRUD API for a MongoDB collection using Mongoose.

22. **User Authentication with JWT**
Implement login and protect routes using JSON Web Tokens.

23. **File Upload Example**
Handle file upload using multer middleware.

24. **Input Validation Using Joi**
Validate incoming request data using Joi schema.

25. **Build and Consume a Weather API**
Use an external weather API (like OpenWeatherMap) to get real-time weather.