Programs to practice in the Lab

Topics Covered: Node JS Modules (http, fs), express

Question 1: Create a Node.js HTTP server that returns the appropriate HTTP status code:

- 404 for invalid routes.
- 200 for valid GET requests.
- 500 for internal server errors.

Question 2: Create a simple routing system within the Node.js http module that handles different paths (e.g., /home, /about, /contact).For each route, send a different response, such as "Welcome to Home", "About Us", etc.

Question 3: Write a Node.js server that parses query parameters from a GET request. For example, if the request is /greet?name=John, the server should respond with "Hello, John!".

Question 4: Implement a file server using the http module that serves a static HTML file when a client makes a GET request to /index.html.

Question 5: Set up a basic Express server that listens on port 3000 and responds with Hello, World! when you visit /.

Question 6: Build a simple Express API with the following CRUD routes:

- GET /items: Returns a list of items.
- POST /items: Accepts data to add a new item.
- PUT /items/:id: Updates an item based on the id.
- DELETE /items/:id: Deletes an item based on the id.

Question 7: Create an Express endpoint that accepts JSON data via POST, parses it, and responds with the data in the response body. Make sure to use express.json() middleware to parse incoming JSON.

Question 8: Create a custom middleware in Express that logs each incoming request's method and URL. Use the middleware globally and for specific routes.

@@@###@@@