**1.Create a Node.js server to implement routes for /home, /about, and /contact that respond with different messages for each route.**

**Explanation:**

**1. const express = require(express);**

This line is attempting to import the Express library, but there's a small error. The require function should take a string as an argument, so it should be require('express') with quotes.

Express is a web application framework for Node.js, which helps in building web applications and APIs.

**2. const app = express();**

Here, you are creating an instance of an Express application by calling express().

The app variable represents your Express application and will be used to define routes and middleware.

**3. app.use(express.json());**

This middleware allows the Express application to parse incoming requests with JSON payloads.

It helps in handling JSON data sent in HTTP request bodies.

**4. app.get('/', (req, res) => { res.send(This is main page); });**

Defines a GET route for the root URL (/).

When someone accesses the root URL, the server will respond with the text This is main page.

There’s a syntax error here as well: The string This is main page should be wrapped in quotes like 'This is main page'.

**5. app.get('/home', (req, res) => { res.send(I am from home page); });**

Sets up a GET route at the /home URL.

When someone visits /home, the server responds with Iam from home page.

Again, Iam from home page should be wrapped in quotes.

**6. app.get('/about', (req, res) => { res.send(I am a computer science and engineering student); });**

Sets up a GET route at the /about URL.

When someone visits /about, the server responds with Iam a computer science and engineering student.

The response text should also be in quotes here.

**7. app.get('/contact', (req, res) => { res.send('contact me on roogle.com'); });**

Sets up a GET route at the /contact URL.

When /contact is accessed, the server sends back the text contact me on roogle.com.

**8. app.get('/greet', (req, res) => { const name = req.query.name || guest; res.send(Hello, ${name}!); });**

Defines a GET route at the /greet URL.

Retrieves a name parameter from the query string (e.g., /greet?name=John), with a default value of guest if no name is provided.

Sends back Hello, ${name}!, with the name inserted into the response.

The variable guest and the string Hello, ${name}! should be in quotes ('guest' and `Hello, ${name}!` respectively).

**9. port = process.env.PORT || 3069;**

Sets the port variable to either the value from process.env.PORT (an environment variable) or defaults to 3069 if process.env.PORT is not set.

**10. app.listen(port, () => console.log(Port is running on ${port}));**

Starts the server and listens on the specified port.

Logs a message to the console with the port number when the server is running.

The string Port is running on ${port} should be wrapped in backticks (`Port is running on ${port}`).

**2.Build a server with a route /greet that accepts a query parameter name and responds with a personalized greeting message, e.g., "Hello, [name]!"**

**1. const express = require('express');**

Imports the Express module, which is a web framework for Node.js that simplifies the process of creating web servers.

**2. const app = express();**

Creates an instance of an Express application. This app object will be used to define routes, middleware, and other aspects of the server.

**3. app.use(express.urlencoded({ extended: true }));**

This line adds middleware to parse URL-encoded data, which is commonly used for HTML form submissions.

The extended: true option allows the parsing of nested objects, typically generated by form data.

**4. app.get('/', (req, res) => { ... });**

Defines a GET route for the root URL (/).

When a client visits the root URL, it responds with an HTML form.

**5-11. Inside the app.get('/', (req, res) => { ... }); block:**

```
res.send(`
  <form action="/greet" method="POST">
    <label for="name">Enter your name:</label>
    <input type="text" id="name" name="name">
    <button type="submit">Greet me!</button>
  </form>
`);
```

The res.send() function sends an HTML form back to the client.

This form has a text input field (<input type="text" id="name" name="name">) where users can enter their name.

The form submits a POST request to the /greet endpoint when the "Greet me!" button is clicked, using the method="POST" attribute.

**12. app.post('/greet', (req, res) => { ... });**

Defines a POST route for the /greet URL. This route will handle the form submission.

When the form is submitted, the server will receive the request and run this function.

**13-14. Inside the app.post('/greet', (req, res) => { ... }); block:**

```
const name = req.body.name || 'Guest';
res.send(`Hello, ${name}!`);
```

Extracts the name value from the form data (available in req.body thanks to the express.urlencoded middleware).

If no name was provided in the form, it defaults to 'Guest'.

Sends a greeting back to the client using res.send() with the user's name injected into the response using template literals.

**15. const PORT = 3000;**

Defines the port number (3000) on which the server will listen for incoming connections.

**16-19. app.listen(PORT, () => { ... });**

```
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

Starts the server and tells it to listen for incoming requests on port 3000.

Once the server is running, it logs a message to the console with a link to the local server URL.