"Hello" app in Node.js using Express.js that displays a welcome message and a logo.

Steps to Create the Hello App

1. Set Up the Project

2. Install Required Packages

3. Create an Entry Point File

4. Set Up Static Files

5. Run the Server

Step 1: Set Up the Project

First, create a new directory for your project and navigate to it:

```
mkdir hello-app

cd hello-app

npm init –y      ( Initialize project with Node.js project )
```

Step 2: Install Express Package

Install Express.js to handle routing and serving static files:

```
npm install express
```

Step 3: Create an Entry Point File

Create a file named `app.js` and add the following code:

```
const express = require('express');

// express: This imports the Express framework, which is used to create and manage the server.

const path = require('path');

//  path: This imports the path module, provides utilities for working with file and directory paths.

const app = express();

//  This line creates an instance of an Express application. The app variable is used to define routes, middleware, and other functionalities.

const port = 3000;

// This line sets the port number to 3000. The server will listen for incoming requests on this port.

// Middleware to serve static files

app.use(express.static(path.join(__dirname, 'public')));

// This line sets up middleware to serve static files from the public directory. Any files in the public directory, such as CSS, images, and JavaScript files, will be accessible from the root URL.

app.get('/', (req, res) => {

   res.sendFile(path.join(__dirname, 'public/index.html'));

});

//  app.get('/'): Defines a route for the root URL (/).
// (req, res): Represents the request and response objects.

// res.sendFile: Sends the index.html file from the public directory as the response.

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

// app.listen(port): Starts the server and listens for incoming requests on the specified port.

// Callback function: Logs a message to the console indicating that the server is running and provides the URL.
```

Step 4: Set Up Static Files

Create a `public` directory at project root. Inside the `public` directory, create an `index.html` file and a `style.css` file.

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <img src="logo.png" alt="Logo" class="logo">
    <h1>Welcome to My Hello App!</h1>
  </div>
</body>
</html>
```

style.css:

```
body {
  font-family: Arial, sans-serif;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}
.container {
  text-align: center;
}
.logo {
  width: 100px;
  height: auto;
}
```

Add a Logo

Place an image file named `logo.png` inside the `public` directory.

Step 5: Run the Server

Run the server using the following command:

```
node app.js
```

Open browser and open `http://localhost:3000` - welcome message and logo displayed.