Sure, I can help you set up a simple Node.js project that connects to a MySQL database and performs basic login functionality using HTML and CSS for the front end. Here's a step-by-step guide:

### Step 1: Set Up Project Directory

1. Create a new directory for your project and navigate to it:

```sh

mkdir nodejs-login

cd nodejs-login

```

### Step 2: Initialize Node.js Project

2. Initialize a new Node.js project:

```sh

npm init -y

```

### Step 3: Install Dependencies

3. Install the required dependencies:

```sh

npm install express mysql body-parser

```

### Step 4: Project Structure

4. Create the following folder and file structure:

```

nodejs-login/

├── node\_modules/

├── public/

│ ├── styles.css

├── views/

│ ├── index.html

│ ├── welcome.html

├── app.js

├── package.json

```

### Step 5: Create `app.js` (Server Setup)

5. Create `app.js` and add the following code:

```javascript

const express = require('express');

const bodyParser = require('body-parser');

const mysql = require('mysql');

const path = require('path');

const app = express();

// MySQL connection

const db = mysql.createConnection({

host: 'localhost',

user: 'pranoti1',

password: '1234',

database: 'nodejs'

});

db.connect((err) => {

if (err) throw err;

console.log('Connected to database');

});

// Middleware

app.use(bodyParser.urlencoded({ extended: false }));

app.use(express.static(path.join(\_\_dirname, 'public')));

// Routes

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

res.sendFile(path.join(\_\_dirname, 'views', 'index.html'));

});

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

const { username, password } = req.body;

const query = 'SELECT \* FROM login WHERE user\_name = ? AND user\_pass = ?';

db.query(query, [username, password], (err, results) => {

if (err) throw err;

if (results.length > 0) {

res.sendFile(path.join(\_\_dirname, 'views', 'welcome.html'));

} else {

res.send('<h1>Incorrect username or password</h1>');

}

});

});

// Start server

app.listen(3000, () => {

console.log('Server running on port 3000');

});

```

### Step 6: Create HTML Files

6. Create `views/index.html` for the login page:

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

<link rel="stylesheet" href="/styles.css">

</head>

<body>

<div class="container">

<h2>Login</h2>

<form action="/login" method="post">

<div>

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

</div>

<div>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

</div>

<button type="submit">Login</button>

</form>

</div>

</body>

</html>

```

7. Create `views/welcome.html` for the welcome page:

```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="/styles.css">

</head>

<body>

<div class="container">

<h2>Welcome, User!</h2>

</div>

</body>

</html>

```

### Step 7: Create CSS File

8. Create `public/styles.css` for basic styling:

```css

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

}

.container {

background: white;

padding: 20px;

border-radius: 5px;

box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

}

form div {

margin-bottom: 15px;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"],

input[type="password"] {

width: 100%;

padding: 8px;

box-sizing: border-box;

}

button {

padding: 10px 15px;

background: #5cb85c;

border: none;

color: white;

cursor: pointer;

}

button:hover {

background: #4cae4c;

}

```

### Step 8: Create MySQL Database and Table

9. Ensure your MySQL server is running and execute the following SQL commands to create the database and table:

```sql

CREATE DATABASE nodejs;

USE nodejs;

CREATE TABLE login (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_name VARCHAR(50) NOT NULL,

user\_pass VARCHAR(50) NOT NULL

);

INSERT INTO login (user\_name, user\_pass) VALUES ('testuser', 'testpass');

```

### Step 9: Run the Project

10. Start your Node.js server:

```sh

node app.js

```

11. Open your browser and navigate to `http://localhost:3000` to see the login page.

This simple project should allow you to log in using the credentials you inserted into the database. If the login is successful, it will display the welcome page; otherwise, it will show an error message.