**Creating the EJS Form**

Here's the EJS form that collects the necessary user information:

Code snippet

<!DOCTYPE html>  
<html>  
<head>  
 <title>User Information Form</title>  
</head>  
<body>  
 <h1>User Information Form</h1>  
 <form action="/submit" method="POST">  
 <label for="username">Username:</label>  
 <input type="text" id="username" name="username" required><br><br>  
  
 <label for="sex">Sex:</label>  
 <select id="sex" name="sex" required>  
 <option value="male">Male</option>  
 <option value="female">Female</option>  
 </select><br><br>  
  
 <label for="height">Height (cm):</label>  
 <input type="range" id="height" name="height" min="120" max="250" required><br><br>  
  
 <label for="birthdate">Birthdate:</label>  
 <input type="date" id="birthdate" name="birthdate" required><br><br>  
  
 <input type="submit" value="Submit">  
 </form>  
</body>  
</html>

**Handling the Form Submission in Node.js/Express**

Here's the Node.js/Express code to handle the form submission, process the data, and send a response:

JavaScript

const express = require('express');  
const bodyParser = require('body-parser');  
const ejs = require('ejs');  
  
const app = express();  
const port = 3000;  
  
app.set('view engine', 'ejs');  
app.use(bodyParser.urlencoded({ extended: true }));  
  
app.get('/', (req, res) => {  
 res.render('index');  
});  
  
app.post('/submit', (req, res) => {  
 const { username, sex, height, birthdate } = req.body;  
  
 // Process the data here (e.g., store it in a database)  
  
 // Send a response back to the client  
 res.render('index', { username });  
});  
  
app.listen(port, () => {  
 console.log(`Server listening on port ${port}`);  
});

**Explanation:**

1. **EJS Form:**
   * The form collects the user's username, sex, height, and birthdate.
   * The required attribute ensures that all fields are filled.
   * The action="/submit" attribute specifies the route to handle the form submission.
   * The method="POST" attribute indicates that the form data will be sent using the POST HTTP method.
2. **Node.js/Express Server:**
   * The bodyParser.urlencoded() middleware is used to parse the form data.
   * The /submit route handles the POST request.
   * The form data is extracted from the req.body object.
   * You can process the data further here, such as storing it in a database or performing calculations.
   * The res.render('index', { username }) line renders the index.ejs template, passing the username variable to the template.

**Displaying the Username in the Response**

In your index.ejs template, you can display the username variable:

Code snippet

<!DOCTYPE html>  
<html>  
<head>  
 <title>User Information</title>  
</head>  
<body>  
 <h1>Hello, <%= username %>!</h1>  
 <p>Your information has been submitted.</p>  
</body>  
</html>

This will display a greeting message with the user's username on the page after form submission.

**Sources**

1. <https://github.com/lostwithoutnumbers/LearnCoding>

2. <https://github.com/Raju27-06/demo-html>

3. <https://github.com/JuMaxi/Movies-Catalogue>

4. <https://github.com/JoyceHsieh/Azure_Disney>

5. <https://github.com/sieunhantanh/QuanlidaythemPRJ301>

6. <https://github.com/Destravna/To-Do-List>

7. <https://github.com/julioolvr/pato-rank>