**Assignment 1: Enhanced Node.js and EJS Project**

**Objective: Create a Node.js server project with Express, EJS, and express-ejs-layouts modules. The project will involve passing data between EJS files and the server and include dynamic routing, partials, and layouts.**

**Requirements**

1. **Setup the Node.js Server (20 points)**
   * **Create a new Node.js project.**
   * **Install the following modules: express, ejs, express-ejs-layouts, and nodemon.**
   * **Use port 3000 to run the project on localhost:3000.**
   * **Create a server.js file to start the server. Use nodemon to auto-restart the server on save.**
   * **The server should:**
     + **Use express-ejs-layouts for a consistent layout across pages.**
     + **Serve static files from a public folder for stylesheets or images if needed.**
2. **home.ejs - Home Page (15 points)**
   * **Route: /**
   * **Server-Sent Data: Pass your pawprint from the server as a variable.**
   * **EJS-Sent Data: The user will input their learning goals for this class (via a form). When submitted, the server should save the data and display it below the form. The server should retain and append multiple submissions in a list format.**

**Example form fields:**

* + **Your Learning Goal: <input type="text" name="goal" required />**

**Display the list of learning goals dynamically on the same page.**

1. **about.ejs - About Page (15 points)**
   * **Route: /about**
   * **Server-Sent Data: Pass your name as a variable from the server.**
   * **EJS-Sent Data: Include a form where users can input their name and a short bio. Once submitted, the data should be sent to the server and displayed on the page below the form.**

**Example form fields:**

* + **Name: <input type="text" name="userName" required />**
  + **Bio: <textarea name="userBio" required></textarea>**

**Display a dynamic list of all users and their bios submitted so far.**

1. **movie.ejs - Movie Page (20 points)**
   * **Route: /movies**
   * **Server-Sent Data: Pass an array of your top 10 favorite movies to the EJS file. Each movie entry in the array should include the title, leading stars, and director.**
   * **EJS-Sent Data: Provide a form where users can suggest additional movies with the same fields (title, stars, director). The server should append new entries to the movie list and display them dynamically on the page.**

**Example form fields:**

* + **Movie Title: <input type="text" name="title" required />**
  + **Leading Stars: <input type="text" name="stars" required />**
  + **Director: <input type="text" name="director" required />**

1. **layout.ejs - Layout File (15 points)**
   * **This file should serve as the boilerplate for all EJS pages.**
   * **Include a header.ejs partial for navigation and any footer content directly within layout.ejs.**
2. **header.ejs - Navigation Partial (15 points)**
   * **Create a Bootstrap-based navbar.**
   * **The navbar should include links to:**
     + **Home (/)**
     + **About (/about)**
     + **Movies (/movies)**

**Submission Guidelines**

* **Do not include node\_modules.**
* **Submit only the following:**
  + **server.js**
  + **Views folder with all .ejs files**
  + **Any static assets in a public folder**
  + **A README.md file if additional packages are used.**

**Grading Criteria**

* **20 points: Functional Node.js server setup with Express, EJS, and express-ejs-layouts.**
* **15 points: Proper implementation of the home page, including form handling and displaying user-submitted goals.**
* **15 points: Proper implementation of the about page, including form handling and displaying user-submitted bios.**
* **20 points: Proper implementation of the movie page, including dynamic list rendering and handling user-suggested movies.**
* **15 points: Correct use of the layout file and the navigation partial.**
* **15 points: Clean code, proper folder structure, and adherence to requirements.**