**Project: Save and View User Data with Node.js**

**Goal:** Create a program where you can submit information through a form, save it, and view it later. For simplicity, only **you** will use this program.

**Steps:**

**1. Come up with a program idea**  
Think of something simple where you collect information from a user and save it. Example ideas:

* A list of favorite movies
* A diary entry system
* A list of friends’ birthdays

**2. Set up your project**

* Create a new **Node.js** project using **Express.js** and **EJS** for templates.

**3. Create your data file**

* Make a file called data.json with this inside:

{ "data": [] }

This will store all the entries from your form.

**4. Create the home page (/)**

* Make a GET endpoint for / that renders index.ejs.
* On this page, add **two links**:
  1. /add – to add a new entry
  2. /view – to see all entries

**5. Create the “Add Entry” page (/add)**

* Make a GET endpoint for /add that renders add.ejs.
* On this page, make a **form**:
  + id = any name you want
  + method="POST"
  + action="/add"
* Your form must include **at least two of these** (more is fine):
  + input type="checkbox"
  + input type="color"
  + input type="hidden"
  + input type="password"
  + input type="radio"
  + input type="range"
  + input type="text"
  + textarea
  + select with options
  + datalist
* Add a **submit button**

**6. Handle form submission (POST /add)**

* Create a POST endpoint for /add.

**Steps inside the POST endpoint:**

1. **Validate** the form data. If anything is missing or wrong, throw an error.
2. Convert the form data into a JavaScript object if needed.
3. Read the data.json file and parse it into a JS object using JSON.parse().
4. Add the new form data object to the data array.
5. Convert the updated object back to JSON with JSON.stringify() and save it back to data.json.
6. Redirect the user back to /.

**If there’s an error:**

* Render error.ejs and show the error message.
* Include a link back to /add.

**7. Create the “View Entries” page (/view)**

* Make a GET endpoint for /view.
* Read data.json and convert it to a JavaScript object.
* Render view.ejs and pass in the data.
* Use a **loop in EJS** to show each entry.
* Include a link back to / or /add.