PROGRAM – 4
ExpressJS – Templating, Form Data

a. AIM: Write a program using templating engine.

What is a Template Engine?

A **Template Engine** is a tool that helps you **build HTML pages dynamically** using your data. Instead of writing static HTML files manually, a template engine lets you:

- Insert variables (like user names, messages)
- Use logic (like if, for loops)
- Separate your **HTML layout** from your **JavaScript code**

👉 This makes your web app easier to build and maintain.

What is EJS?

EJS stands for **Embedded JavaScript**.

It is a **popular template engine** for Node.js and Express that lets you:

- Embed JavaScript code **inside HTML**
- Generate dynamic web pages on the server

✅ **Features of EJS:**

- Easy to use
- Supports loops and conditionals
- Clean syntax like <%= %> for inserting data

How it Works in Express:

1. You set EJS as the view engine:

   app.set('view engine', 'ejs');

2. Place .ejs files in the /views folder.
3. Use res.render('filename', data) to render the template with dynamic data.

   In web development, `render` means:

   ▨ **To generate and send a final HTML page** to the client (browser), often using a template and dynamic data.

   Example: index.ejs

Step 1: Check Your Views Directory Structure

Make sure you have this folder and file:
C:\Users\ksbam\my-express-app\views\index.ejs

- ☑ views folder **must exist**
- ☑ index.ejs file **must be inside** the views folder

If you're using .ejs, the filename should be: **index.ejs**

```
<!DOCTYPE html>
<html>
<head>
  <title>Home</title>
</head>
<body>
  <h1>Hello from EJS!</h1>
</body>
</html>
```

Step 2: Check That EJS Is Installed
Run this command in your project directory:

```
npm init -y
npm install ejs
```

◇ **Step 3: Set the View Engine in 4A.js**
Make sure you have the correct configuration at the top of your 4A.js:

☑ **full code for the program --- 4a.js**

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

app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

app.get('/', (req, res) => {
  res.render('index');
});

app.listen(3000, () => {
  console.log('Server running at http://localhost:3000');
});
```

Run the app:
node 4a.js
Open in your browser :
http://localhost:3000

OUTPUT: **"Hello from EJS"**

B) AIM: Write a program to work with form data.

Step 1: Create a views folder

Make sure the directory C:\Users\ksbam\my-express-app\views exists.
If it doesn't:

1. Open your project folder.
2. Create a folder named views.

☑ **Step 2: Create the form view file**
Inside the views folder, add a file named:

- If you're using **EJS**: form.ejs

views/form.ejs:

```
<!DOCTYPE html>
<html>
<head>
 <title>Form</title>
</head>
<body>
 <h1>Sample Form</h1>
 <form method="post">
   <input type="text" name="username" placeholder="Enter name" />
   <button type="submit">Submit</button>
 </form>

 <% if (username) { %>
   <h2>You entered: <%= username %></h2>
 <% } %>
</body>
</html>
```

☑ **Step 3: Check your app settings**
Ensure your Express app sets the views and view engine correctly in 4b.js:

4b.js

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();

app.use(bodyParser.urlencoded({ extended: true }));
app.set('view engine', 'ejs');

app.get('/', (req, res) => res.render('form', { username: null }));

app.post('/', (req, res) => res.render('form', { username: req.body.username }));

app.listen(3000, () => console.log('Server running at http://localhost:3000'));
```

☑ **Step 4: Install EJS if not installed**

Type the following commands in command prompt

```
npm init -y
npm install ejs
npm install express ejs body-parser
```

☑ **Step 5: Run the app**

Start your server:

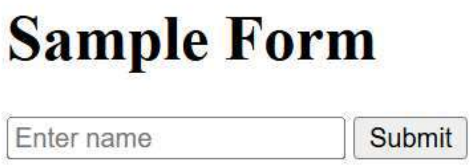
node 4b.js

Then visit:

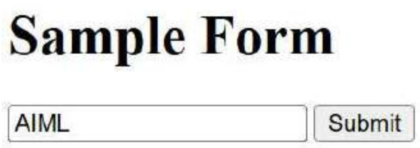
☞ **http://localhost:3000**

You should see your form.

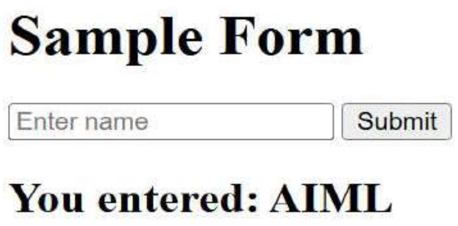
OUTPUT:



Sample Form



Sample Form



Sample Form

You entered: AIML