WEB TECHNOLOGIES

Express JS

Shruthi L

Department of Computer Science and Engineering.

WEB TECHNOLOGIES

Express JS
FORM DATA

Shruthi L

Department of Computer Science and Engineering

FORM DATA

Express JS-Form data

- Forms are an integral part of the web.
- Install
  - *body-parser*(for parsing JSON and url-encoded data) middleware

  npm install body-parser

  Create index.js file and add the following modules –

  var express = require('express');

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

MIDDLEWARE

Express JS

- The JSON parser middleware is created using ***bodyParser.json(),*** and is mounted at the application level using ***app.use().***

- The middleware ***bodyParse*** r looks at the Content-Type header and determines if and how the body can be parsed.

- For JSON, the default content type is application/json.

- In the absence of this header, ***bodyParser*** does not parse the request body, and the variable ***req.body*** ends up as an empty
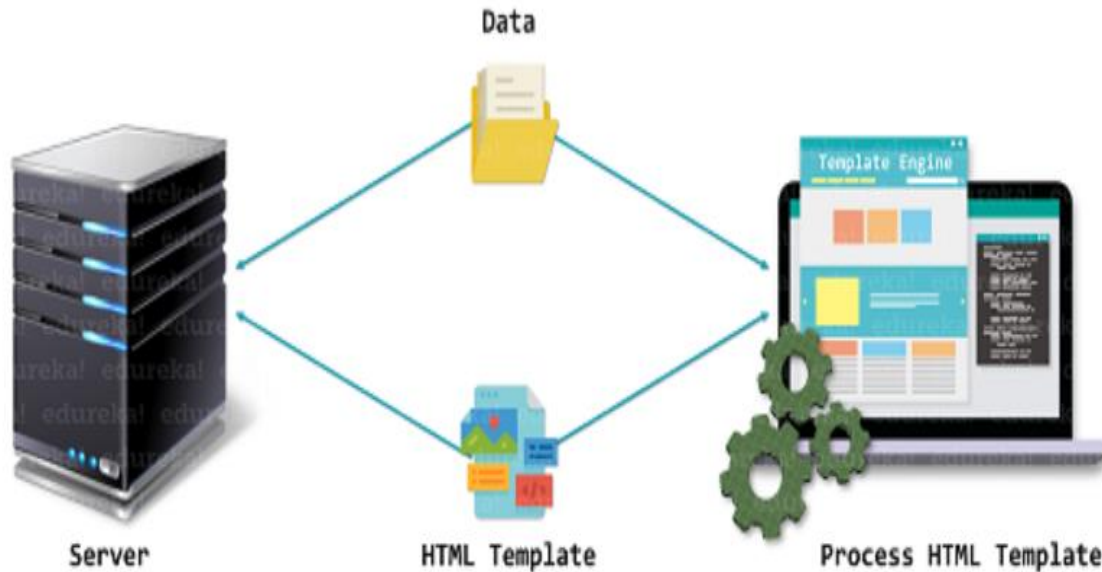- object.

WEB TECHNOLOGIES
EXPRESS TEMPLATE

- Template engine allows you to use static template files in your applications.
- At runtime, the template engine
  - replaces variables in a template file with actual values, and
  - transforms the template into an HTML file sent to the client.
- This approach makes it easier to design an HTML page.
- Express does not have a template engine built in, but it supports any template engine of your choice such as pug, mustache and EJS

WEB TECHNOLOGIES
EXPRESS TEMPLATE

- Pug is a template engine for express
- Earlier Pug was called Jade
- Npm command is used to install the package.
- Npm install –save pug

WEB TECHNOLOGIES
EXPRESS TEMPLATE

After installing Pug

- Set it as the templating engine for your app.
- You **don't** need to 'require' it.
- Add the following code to your **index.js (root)** file.

```
app.set('view engine', 'pug');
app.set('views','./views');
```

- Now create a new directory called views.
- Inside that create a file called form.pug.

WEB TECHNOLOGIES
FORM DATA

- Create a template html form using form.pug

```
HTTPserver.js          form.pug    ✕        JS  eventemitter.js  ●

: > Users > shrug > express >  form.pug
 1    html
 2    html
 3        head
 4
 5        body
 6            form(action = "/", method = "POST")
 7                div
 8                    label(for = "first") First name:
 9                    input(name = "first" value = "Web")
10                br
11                div
12                    label(for = "last") Last name:
13                    input(name = "last" value = "Technology")
14                br
15                button(type = "submit") Submit
```

WEB TECHNOLOGIES
FORM DATA

- To read form (form.pug)
    app.get()
- To create a view
        app.set()
- For parsing application/json
        app.use()
- To display
        app.post()
- Specify port
    `app.listen(3000);`

```javascript
var bodyparser=require('body-parser');

app.get('/',function(req,res)
{
    res.render('form');
});
//create view
app.set('view engine','pug');
app.set('views','./views');

//parsing
app.use(bodyparser.json());
app.use(bodyparser.urlencoded({extended:true}));

app.post('/',function(req,res)
{
    console.log(req.body);
    res.send("Received request");
})
```
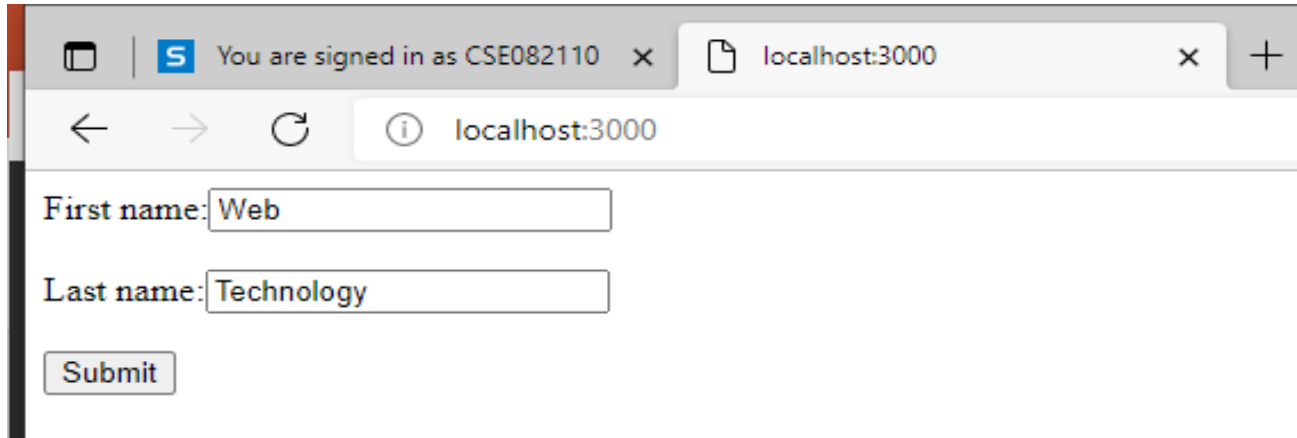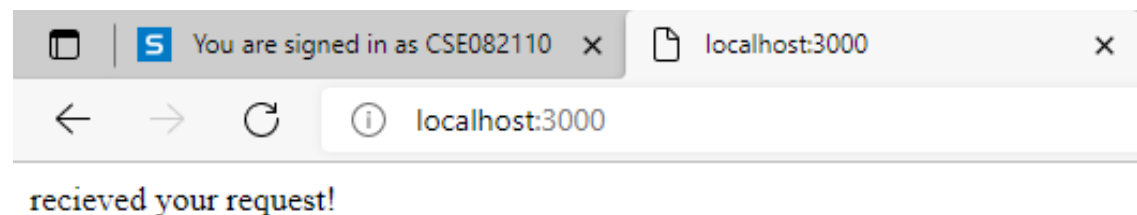
WEB TECHNOLOGIES
FORM DATA

- Run your server using the following.

  nodemon index.js

  Now go to localhost:3000/ following is displayed in browser---

- Response is generated after submit

WEB TECHNOLOGIES
FORM DATA

- In console; it will show you the body of your request as a JavaScript object as in the following screenshot –

```
PS C:\Users\shrug> cd .\express\
PS C:\Users\shrug\express> cd .\express_eg\
PS C:\Users\shrug\express\express_eg> node index.js
PS C:\Users\shrug\express\express_eg> node index.js
{ first: 'Web', last: 'Technology' }
```

THANK YOU

Shruthi L

Department of Computer Science Engineering

shruthil@pes.edu