# Department of Computing

**CS-213: Advanced Programming**

**Class: BSCS 7AB**

# Lab 07: Express JS

**Date: 17 October, 2019**

**Time: 10:00-01:00pm & 02:00-05:00pm**

# Instructor: Dr. Sidra Sultana

**Lab Engineer: Ms. Ayesha Asif**

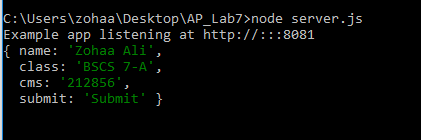
# 

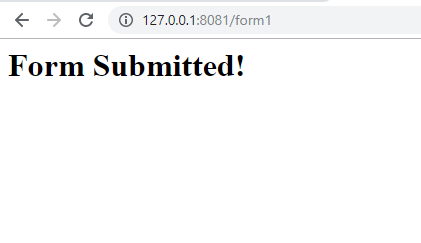
# Lab 07: Express JS

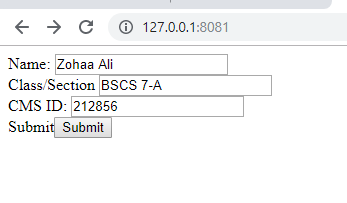
**Lab Tasks**

**Task 1:** Create Index.html file in the root folder of your application and write the HTML FORM POST method code in it. Modify server.js to handle home page requests as well as the input sent by the HTML form.

**Hint:** To handle HTTP POST request in Express.js version 4 and above, you need to install middleware module called body-parser. The middleware was a part of Express.js earlier but now you have to install it separately. This body-parser module parses the JSON, buffer, string and url encoded data submitted using HTTP POST request. Install body-parser using NPM as shown below.







**Index.html:**

<!DOCTYPE html>

<html>

<head>

<title>Task1</title>

</head>

<body>

<form action="form1" method="POST">

Name: <input type="text" name="name"> <br>

Class/Section <input type="text" name="class"> <br>

CMS ID: <input type="number" name="cms"> <br>

Submit<input type="submit" name="submit">

</form>

</body>

</html>

**Server.js:**

var express = require('express');

var app = express();

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

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.get('/', function (req, res) {

res.sendFile('/Users/zohaa/Desktop/AP\_Lab7/index.html');

});

app.post('/form1', function(req, res){

res.send("<h1>Form Submitted!");

console.log(req.body);

});

var server = app.listen(8081, function () {

var host = server.address().address

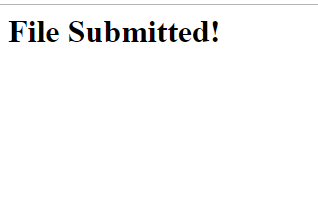
var port = server.address().port

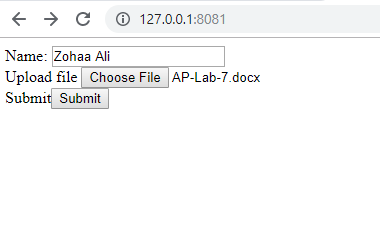
console.log("Example app listening at http://%s:%s", host, port)

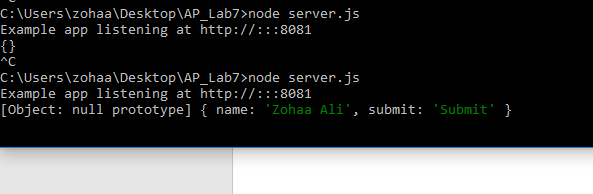
})

;

**Task 2:** Create a file uploader form in an html file. This form has method attribute set to POST and enctype attribute is set to multipart/form-data. Modify server.js to handle home page requests as well as file upload.







**Index.html:**

<!DOCTYPE html>

<html>

<head>

<title>Task1</title>

</head>

<body>

<form action="form2" method="POST" enctype="multipart/form-data">

Name: <input type="text" name="name"> <br>

Upload file <input type="file" name="file"> <br>

Submit<input type="submit" name="submit">

</form>

</body>

</html>

**Server.js:**

var express = require('express');

var app = express();

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

var multer = require('multer');

var mul = multer();

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.get('/', function (req, res) {

res.sendFile('/Users/zohaa/Desktop/AP\_Lab7/index.html');

});

app.post('/form2', mul.single('file'), function(req, res){

if(req.file)

{

res.send("<h1>File Submitted!");

console.log(req.body);

}

else{

res.send("<h1> Error uploading file!");

}

});

var server = app.listen(8081, function () {

var host = server.address().address

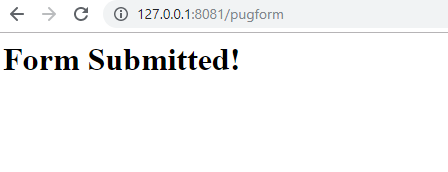
var port = server.address().port

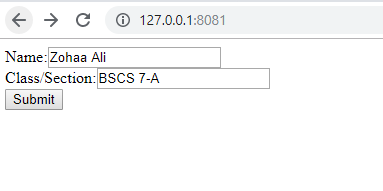
console.log("Example app listening at http://%s:%s", host, port)

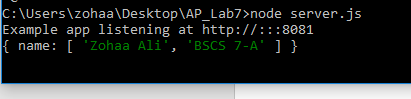
})

;

**Task 3:** By using the Pug templating engine create an HTML registration form.







**Server.js:**

var express = require('express');

var app = express();

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

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.set("view engine", "pug");

app.set("views", "/Users/zohaa/Desktop/AP\_Lab7");

app.get('/', function (req, res) {

res.render('form');

});

app.post('/pugform', function(req, res){

res.send("<h1>Form Submitted!");

console.log(req.body);

});

var server = app.listen(8081, function () {

var host = server.address().address

var port = server.address().port

console.log("Example app listening at http://%s:%s", host, port)

})

;

**Form.pug:**

html

head

title Pug Form

body

form(action='/pugform',method='POST')

label(for="name") Name:

input(type="text",name="name",title="df")

br

label(for="class") Class/Section:

input(type="text",name="name",title="df")

br

input(type="submit")

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva/quiz related to the tasks. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems with submissions on LMS, submit your Lab assignments by emailing it to Ms. Ayesha Asif: [ayesha.asif@seecs.edu.pk](mailto:ayesha.asif@seecs.edu.pk).