# Experiment 6

**(Express)**

**Name- Tanishka Kothari**

**Sap ID- 60009200052**

**Div/Batch- K/K2**

**Branch - Data Science**

**Subject- Web Engineering**

**Date :12/05/2022**

npm init

npm install express

var express=require('express');

var app=express();

//Creating a callback function. This function will be called whenever anybody browses to the root of our web application which

//is http:/ back function will be used to send the string ‘Hello World’ to the web page.

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

{

res.send('Hello World!');

});

var server=app.listen(3000,function() {});

/localhost:3000.

//The call

npm start

**http://localhost:3000**

npm install body-parser –save

/\* var express=require('express');

var app=express();

//Creating a callback function. This function will be called whenever anybody browses to the root of our web application which

//is http://localhost:3000.

//The callback function will be used to send the string ‘Hello World’ to the web page.

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

{

res.send('Hello World!');

});

var server=app.listen(3000,function() {});  \*/

/\* var express = require('express');

var app = express();

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

    res.send('<html><body><h1>Hello World</h1></body></html>');

});

app.post('/submit-data', function (req, res) {

    res.send('POST Request');

});

app.put('/update-data', function (req, res) {

    res.send('PUT Request');

});

app.delete('/delete-data', function (req, res) {

    res.send('DELETE Request');

});

var server = app.listen(3000, function () {

    console.log('Node server is running..');

});  \*/

/\* var express = require('express');

var app = express();

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

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

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

    res.sendFile(\_\_dirname + '/index.html');

});

app.post('/submit-student-data', function (req, res) {

    var name = req.body.firstName + ' ' + req.body.lastName;

    res.send(name + ' Submitted Successfully!');

});

var server = app.listen(3000, function () {

    console.log('Node server is running..');

}); \*/

//URL Building

/\* var express = require('express');

var app = express();

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

   res.send('The id you specified is ' + req.params.id);

});

app.listen(3000); \*/

var express = require('express');

var app = express();

app.get('/things/:name/:id', function(req, res) {

   res.send('id: ' + req.params.id + ' and name: ' + req.params.name);

});

app.listen(5000);

**MongoDB:**

1. <https://www.mongodb.com/try/download/community> link and download it for our windows operating system.
2. This cloud solution name is MongoDB Atlas, and it is a cloud-hosted MongoDB database that we can conveniently use.
3. Scroll down the page and click on the **Create a cluster** button to create a cluster.
4. Create a project
5. Click on the **Database Access** option of the **SECURITY** and create a user by clicking on the **Add New Database User**.
6. Scroll down the page and fill the username and password.
7. Go to the **Network Access** section of the page. Here, we will click on the **Add IP Address** button to add an IP.
8. Choose our current [IP](https://www.javatpoint.com/ip) address.
9. Click on the **Confirm** button for the confirmation of it.
10. Go back to project.

**Connect to our database:**

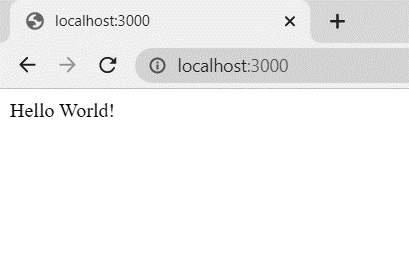
A third-party package mongoose builds upon the official MongoDB driver, but it makes accessing [MongoDB](https://www.javatpoint.com/mongodb-tutorial) much easier and more convenient. Mongoose use schemas that mongodb does not really use, so we can define how our data should look like and that allows us to conveniently store and fetch data.

npm install mongodb

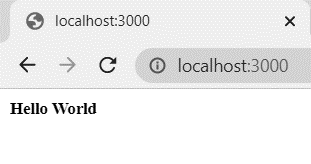
npm install mongoose

**output ss:**

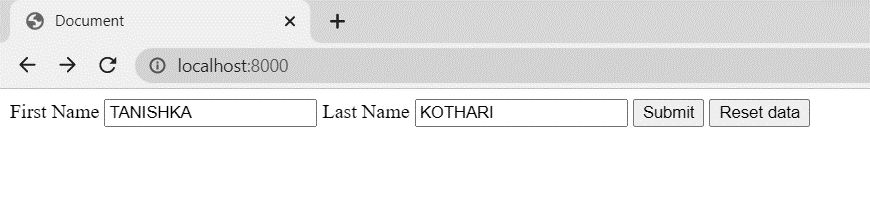
Step 1:

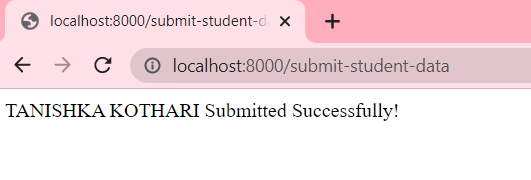


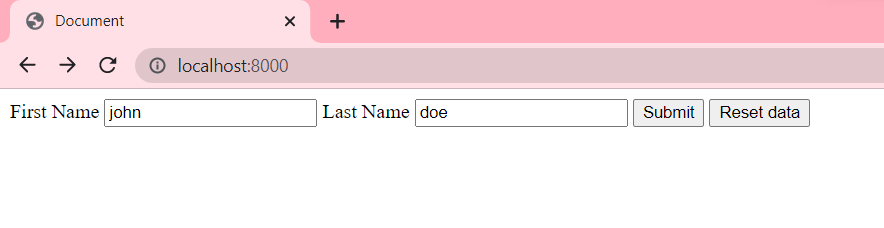
Step 2 :

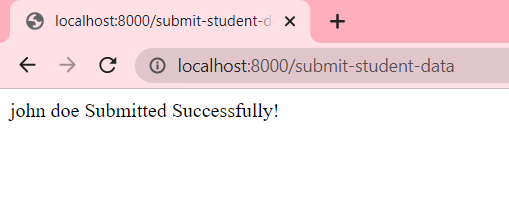


Step 3 :

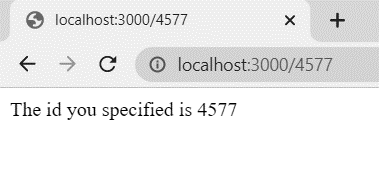




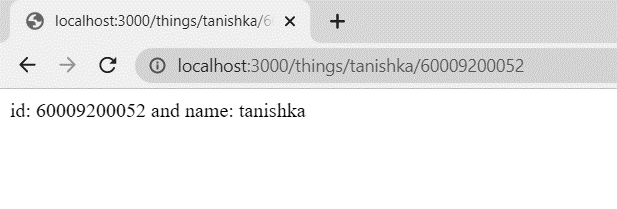




Step 3:



Step 4 :



Step 5:

