NODEJS AND MONGODB ABOUT FILTERING TUTORIAL

Ayşegül KURT

In this tutorial, we will see how we can use MongoDB database from within Node js application.

What is the ‘Node.js’?

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js’ package ecosystem, npm, is the largest ecosystem of open source libraries in the world. That is one of the reasons why this application be uses.

Lots of major and popular industry leaders chose to rely on Node.js such as Netflix, Uber, Paypal...

What is the ‘MongoDB’?

The MongoDB Node.js driver is the officially supported node.js driver for MongoDB.As a definition, MongoDB is an open-source database that uses a document-oriented data model and a non-structured query language. It is one of the most powerful NoSQL systems and databases around, today.

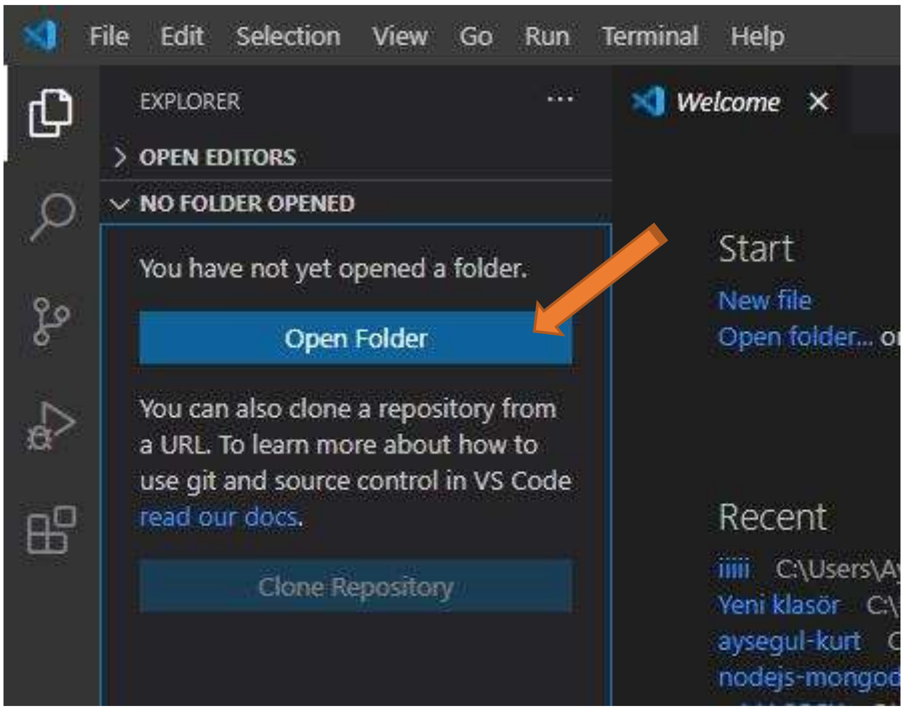
We use the “Visual Studio Code” text editor for this project.

These are the packages that must be installed to create the project;

- Express ➡ Express.js is a Node.js web application server framework, which is specifically designed for building single-page, multi-page, and hybrid web applications. It has become the standard server framework for node. Web applications are web apps that you can run on a web browser.

- Body-parser ➡ Body-parser parses your request and converts it into a format from which you can easily extract relevant information that you may need.

1)CREATING ENVIRONMENT

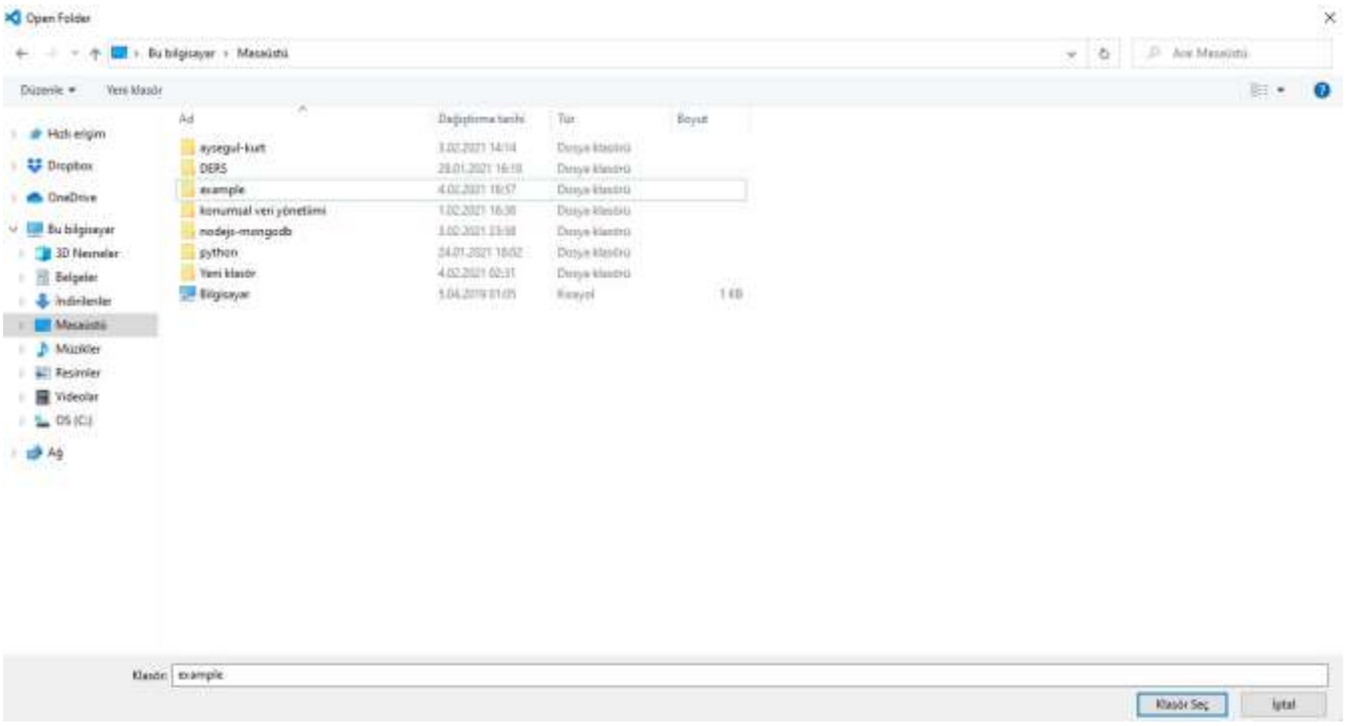


You have not yet opened a folder.

You can also clone a repository from a URL. To learn more about how to use git and source control in VS Code read our docs.

iiiii C:\Users\A
Yeni klasör C:\
aysegul-kurt C
nodejs-mongod

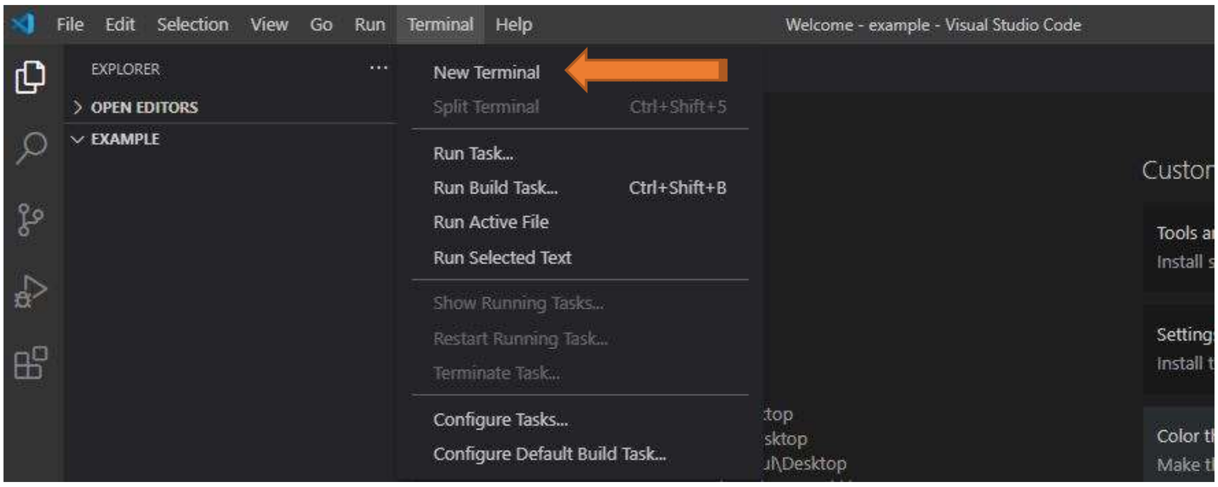
We open Visiual Studio ➡ Open Folder



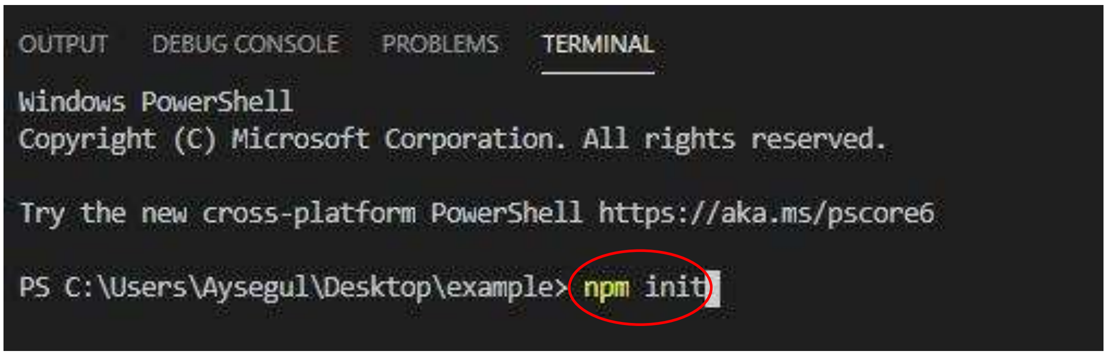
Ad	Değiştirme tarihi	Tür	Boyut
aysegul-kurt	3.02.2021 14:14	Dosya klasörü	
DERS	28.01.2021 16:18	Dosya klasörü	
example	4.02.2021 18:57	Dosya klasörü	
konumsal veri yönetimi	1.02.2021 16:38	Dosya klasörü	
nodejs-mongodb	3.02.2021 23:58	Dosya klasörü	
python	24.01.2021 18:02	Dosya klasörü	
Yeni klasör	4.02.2021 02:31	Dosya klasörü	
Bilgisayar	5.04.2019 01:05	Kısayol	1 KB

Klasör: example

We create a new file and open this file.



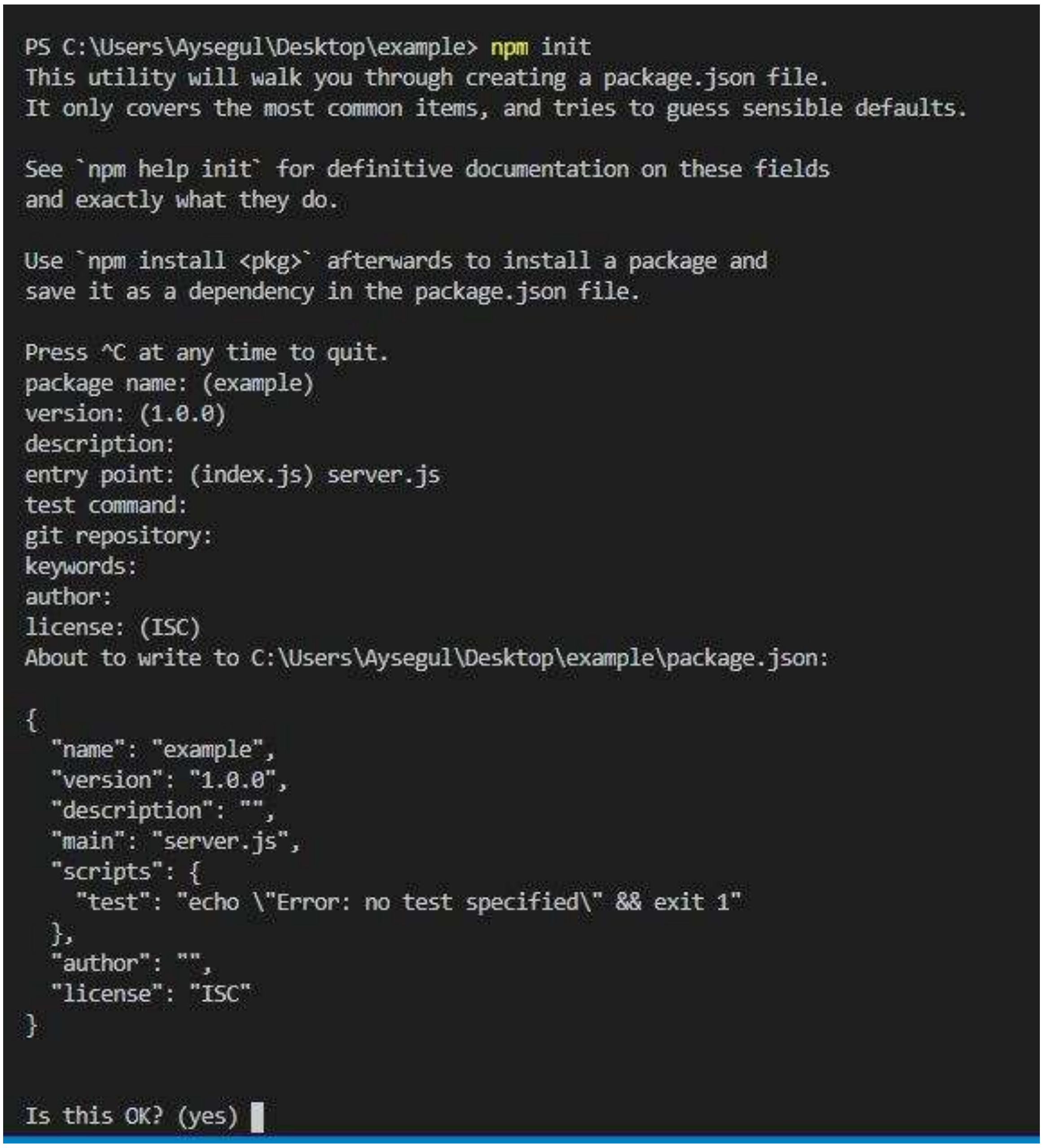
We open new terminal. Write **npm init** in terminal.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Aysegul\Desktop\example> npm init
```

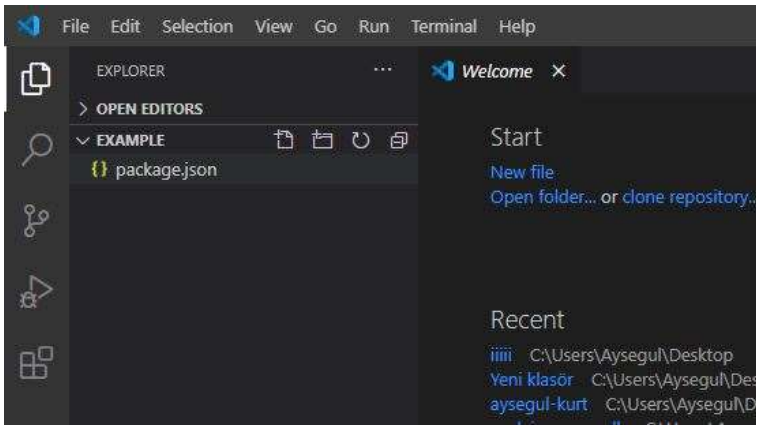
npm init is a convenient way of scaffolding our package.

npm install, however, installs your dependencies in node_modules folder.

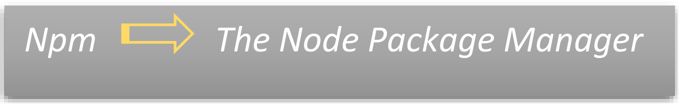


```
PS C:\Users\Aysegul\Desktop\example> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (example)
version: (1.0.0)
description:
entry point: (index.js) server.js
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\Aysegul\Desktop\example\package.json:

{
  "name": "example",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes)
```

We have done with the configuration we have created this packaged or json file.


File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
EXAMPLE
package.json
Welcome
Start
New file
Open folder... or clone repository...
Recent
iiiii C:\Users\Aysegul\Desktop
Yeni klasör C:\Users\Aysegul\De
aysegul-kurt C:\Users\Aysegul\D

How we install the packages?

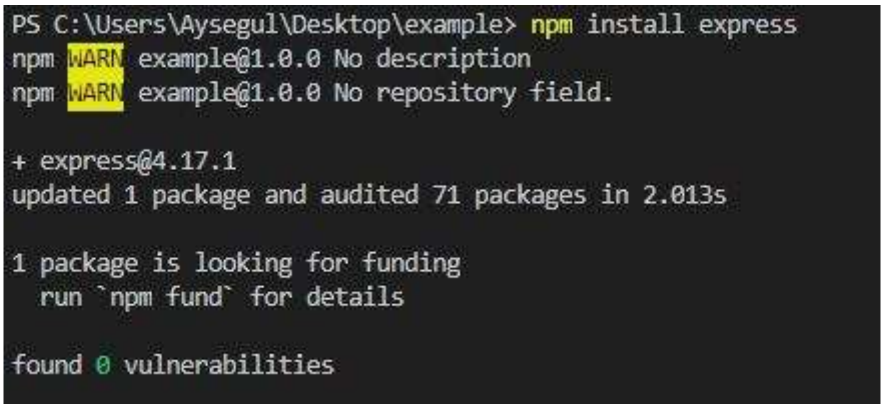


npm init express

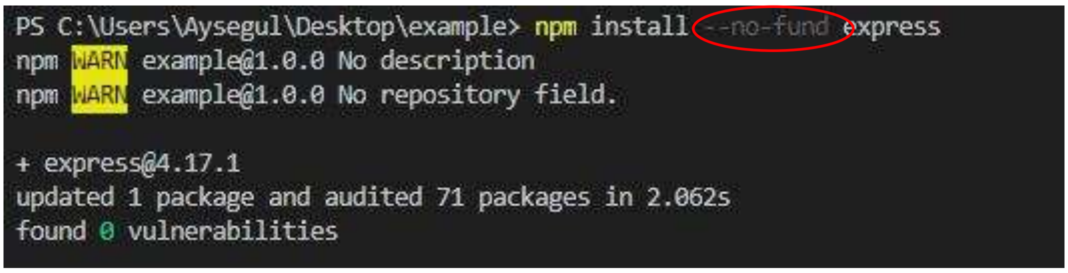
npm init bodyparser

npm init mongodb

For express:

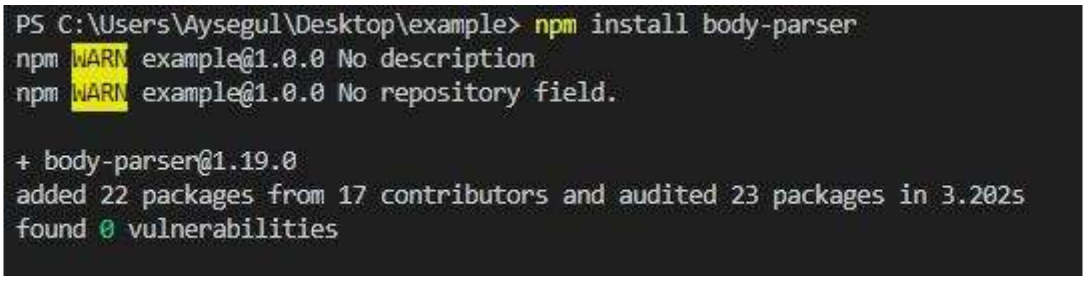


```
PS C:\Users\Aysegul\Desktop\example> npm install express
npm WARN example@1.0.0 No description
npm WARN example@1.0.0 No repository field.

+ express@4.17.1
updated 1 package and audited 71 packages in 2.013s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



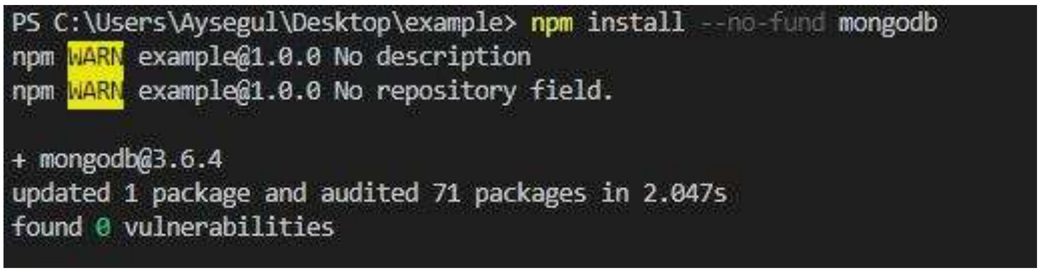
```
PS C:\Users\Aysegul\Desktop\example> npm install --no-fund express
npm WARN example@1.0.0 No description
npm WARN example@1.0.0 No repository field.

+ express@4.17.1
updated 1 package and audited 71 packages in 2.062s
found 0 vulnerabilities
```

For body-parser:



```
PS C:\Users\Aysegul\Desktop\example> npm install body-parser
npm WARN example@1.0.0 No description
npm WARN example@1.0.0 No repository field.

+ body-parser@1.19.0
added 22 packages from 17 contributors and audited 23 packages in 3.202s
found 0 vulnerabilities
```

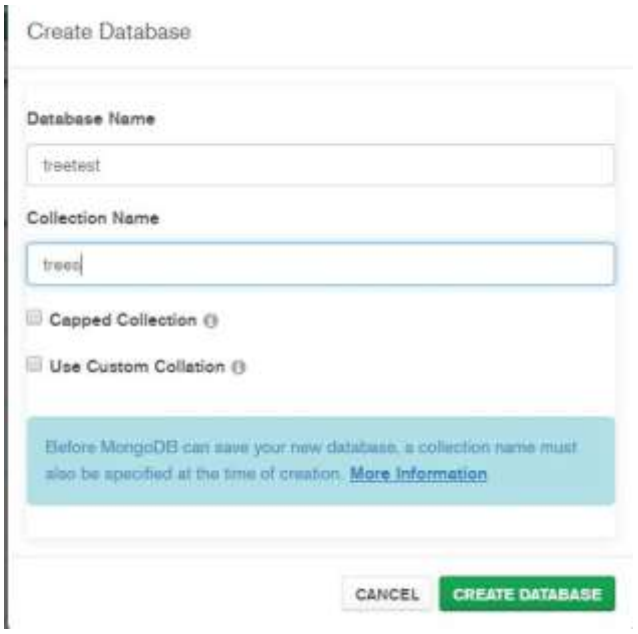
For mongodb:



```
PS C:\Users\Aysegul\Desktop\example> npm install --no-fund mongodb
npm WARN example@1.0.0 No description
npm WARN example@1.0.0 No repository field.

+ mongodb@3.6.4
updated 1 package and audited 71 packages in 2.047s
found 0 vulnerabilities
```

2)CREATING DATABASE

We open the MongoDB Compass and create our database.


Create Database

Database Name

treetest

Collection Name

trees

☐ Capped Collection

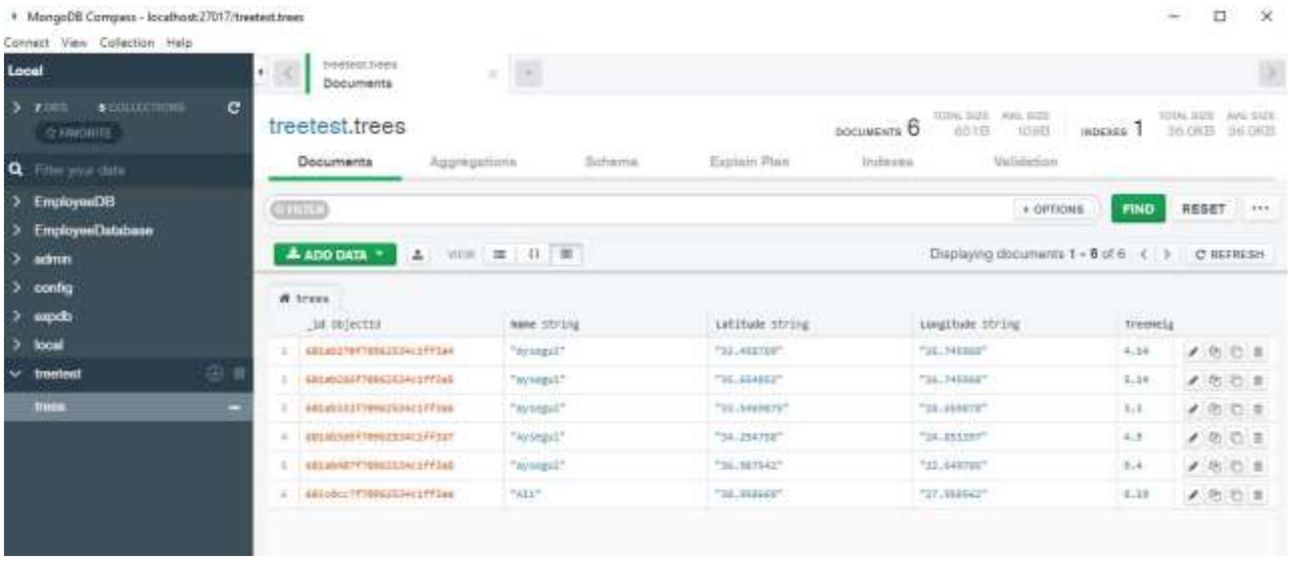
☐ Use Custom Collation

Before MongoDB can save your new database, a collection name must also be specified at the time of creation. More Information

Database name: “treetest”

Collection name: “tree”

- ✓ ID
- ✓ Name
- ✓ Latitude
- ✓ Longitude
- ✓ Tree Height


MongoDB Compass - localhost:27017/treetest.trees

treetest.trees

DOCUMENTS 6 INDEXES 1

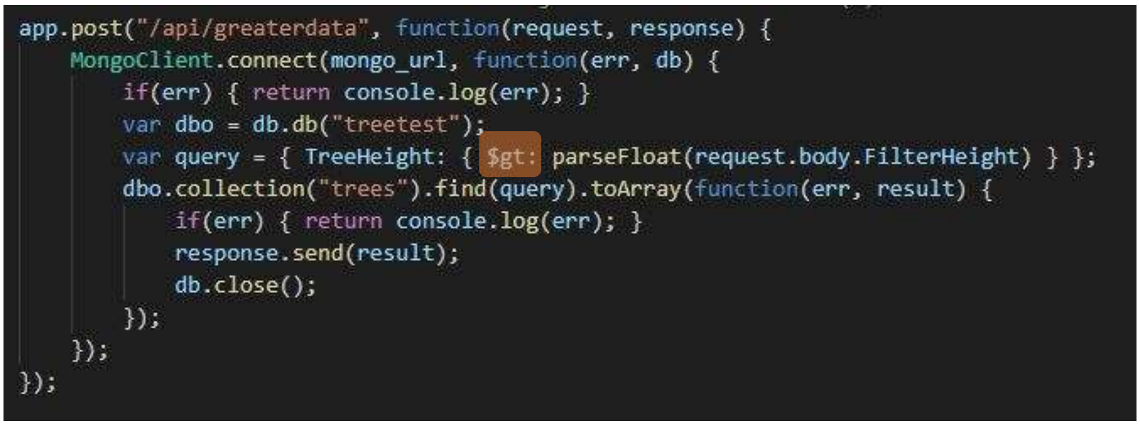
Displaying documents 1 - 6 of 6

	_id ObjectId	Name String	Latitude String	Longitude String	TreeHeig
1	[illegible]	"Aysegul"	[illegible]	[illegible]	4.14
2	[illegible]	"Aysegul"	[illegible]	[illegible]	5.14
3	[illegible]	"Aysegul"	[illegible]	[illegible]	3.3
4	[illegible]	"Aysegul"	[illegible]	[illegible]	4.3
5	[illegible]	"Aysegul"	[illegible]	[illegible]	5.4
6	[illegible]	"Ali"	[illegible]	[illegible]	[illegible]

After we write the code, occur this table.

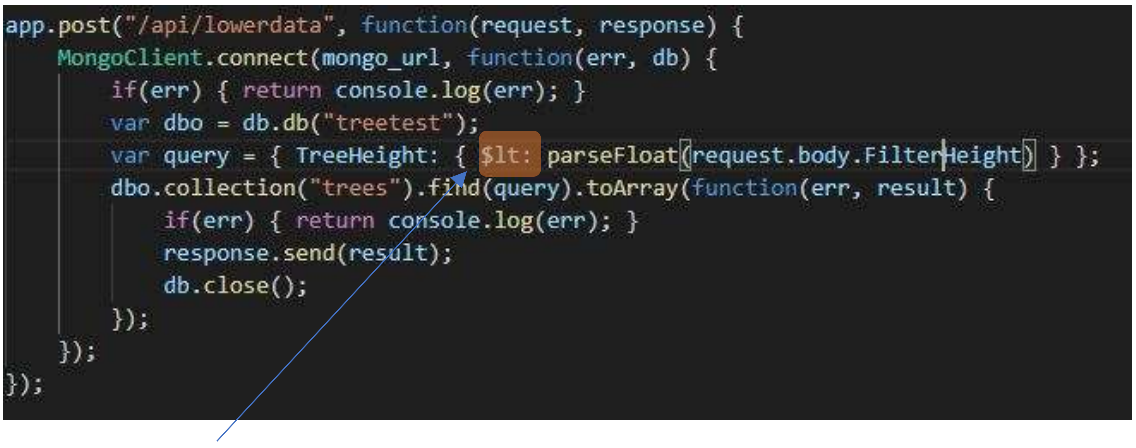
We can edit or delete each data here.

3)FILTERING



```
app.post("/api/greaterdata", function(request, response) {
    MongoClient.connect(mongo_url, function(err, db) {
        if(err) { return console.log(err); }
        var dbo = db.db("treetest");
        var query = { TreeHeight: { $gt: parseFloat(request.body.FilterHeight) } };
        dbo.collection("trees").find(query).toArray(function(err, result) {
            if(err) { return console.log(err); }
            response.send(result);
            db.close();
        });
    });
});
```

Since it is desired to display the ones that are bigger than the incoming data here, in the TreeHeight index, $ gt, the query was created by converting the incoming data to float with the parseFloat function and assigned to the query variable.

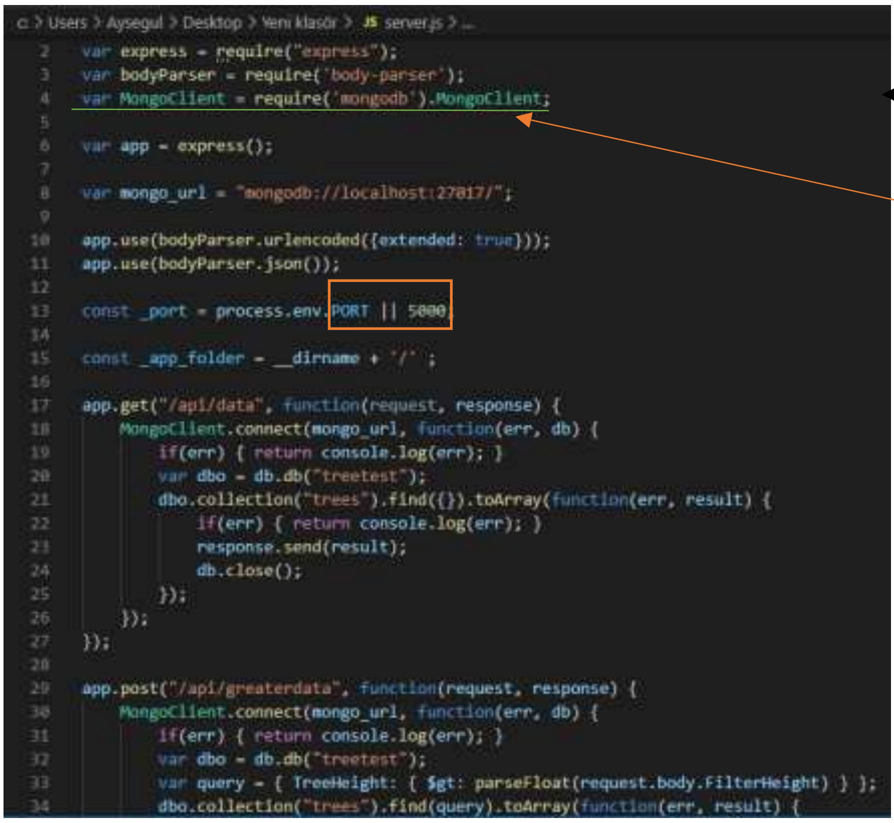


```
app.post("/api/lowerdata", function(request, response) {
    MongoClient.connect(mongo_url, function(err, db) {
        if(err) { return console.log(err); }
        var dbo = db.db("treetest");
        var query = { TreeHeight: { $lt: parseFloat(request.body.FilterHeight) } };
        dbo.collection("trees").find(query).toArray(function(err, result) {
            if(err) { return console.log(err); }
            response.send(result);
            db.close();
        });
    });
});
```

Here the same operations of greater then were applied, using $ lt to get only the minors.

4)CODE PART

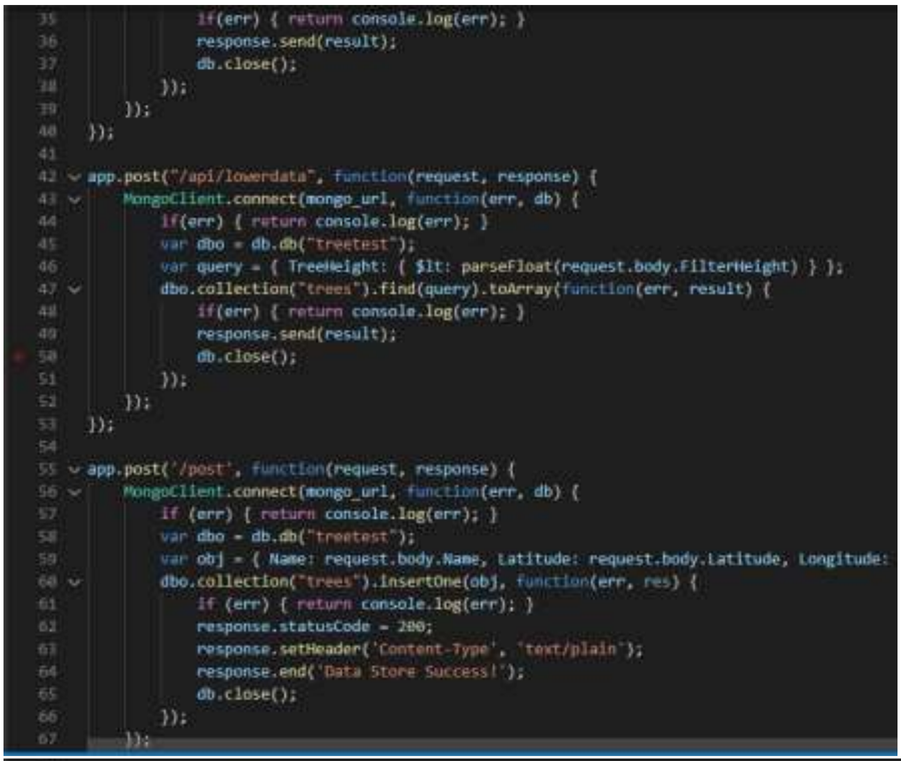
server.js



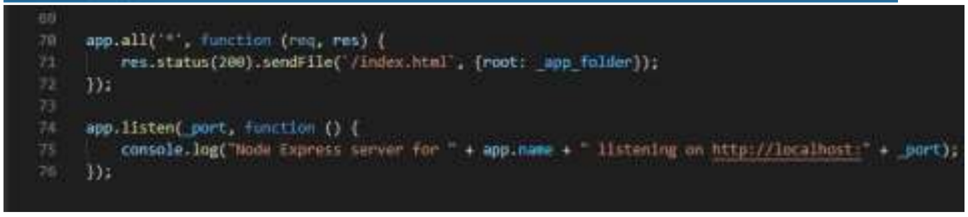
c: > Users > Aysegul > Desktop > Yeni klasör > JS server.js > ...

```
var express = require("express");
var bodyParser = require('body-parser');
var MongoClient = require('mongodb').MongoClient;

var app = express();

var mongo_url = "mongodb://localhost:27017/";

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

const _port = process.env.PORT || 5000;

const _app_folder = __dirname + '/' ;

app.get("/api/data", function(request, response) {
    MongoClient.connect(mongo_url, function(err, db) {
        if(err) { return console.log(err); }
        var dbo = db.db("treetest");
        dbo.collection("trees").find({}).toArray(function(err, result) {
            if(err) { return console.log(err); }
            response.send(result);
            db.close();
        });
    });
});

app.post("/api/greaterdata", function(request, response) {
    MongoClient.connect(mongo_url, function(err, db) {
        if(err) { return console.log(err); }
        var dbo = db.db("treetest");
        var query = { TreeHeight: { $gt: parseFloat(request.body.FilterHeight) } };
        dbo.collection("trees").find(query).toArray(function(err, result) {
```

MongoDB database connection string has been created and assigned to mongo_url variable to be used for future database connection.

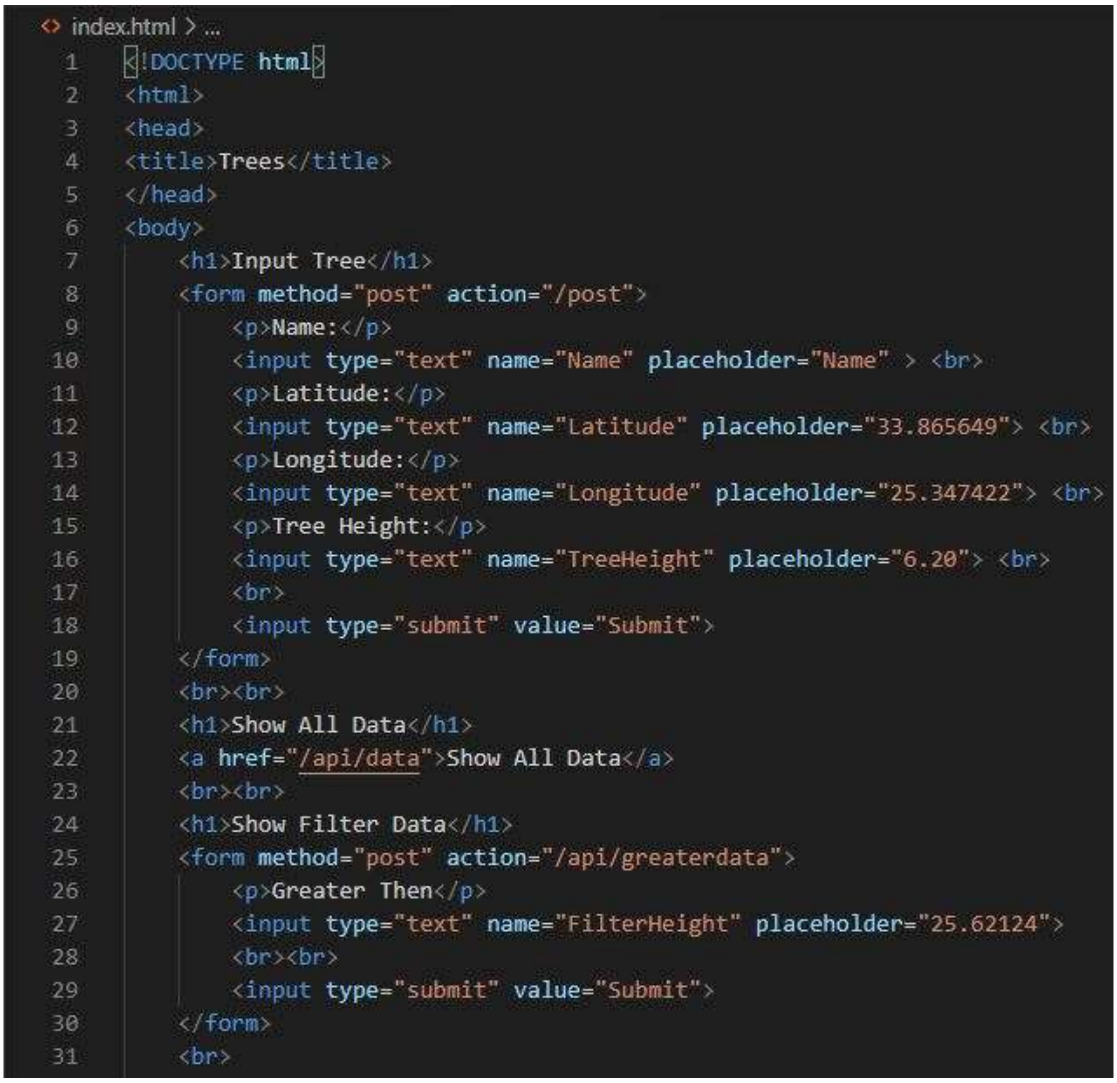


```
            if(err) { return console.log(err); }
            response.send(result);
            db.close();
        });
    });
});

app.post("/api/lowerdata", function(request, response) {
    MongoClient.connect(mongo_url, function(err, db) {
        if(err) { return console.log(err); }
        var dbo = db.db("treetest");
        var query = { TreeHeight: { $lt: parseFloat(request.body.FilterHeight) } };
        dbo.collection("trees").find(query).toArray(function(err, result) {
            if(err) { return console.log(err); }
            response.send(result);
            db.close();
        });
    });
});

app.post('/post', function(request, response) {
    MongoClient.connect(mongo_url, function(err, db) {
        if (err) { return console.log(err); }
        var dbo = db.db("treetest");
        var obj = { Name: request.body.Name, Latitude: request.body.Latitude, Longitude:
        dbo.collection("trees").insertOne(obj, function(err, res) {
            if (err) { return console.log(err); }
            response.statusCode = 200;
            response.setHeader('Content-Type', 'text/plain');
            response.end('Data Store Success!');
            db.close();
        });
    });
```

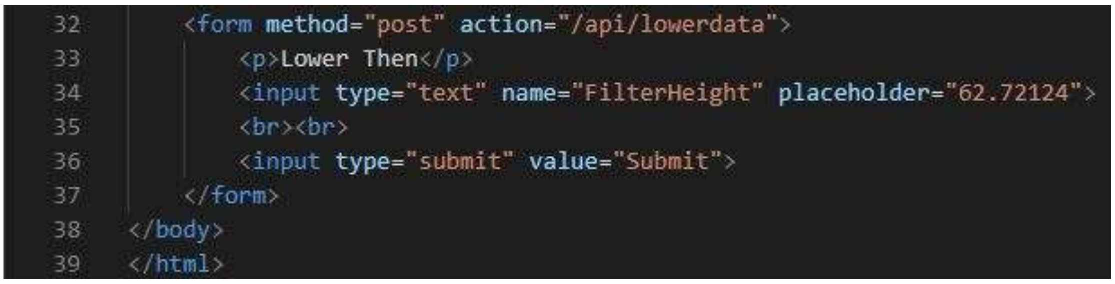


```

app.all('*', function (req, res) {
    res.status(200).sendFile('/index.html', {root: _app_folder});
});

app.listen(_port, function () {
    console.log("Node Express server for " + app.name + " listening on http://localhost:" + _port);
});
```

Index.html



```html
<!DOCTYPE html>
<html>
<head>
<title>Trees</title>
</head>
<body>
    <h1>Input Tree</h1>
    <form method="post" action="/post">
        <p>Name:</p>
        <input type="text" name="Name" placeholder="Name" > <br>
        <p>Latitude:</p>
        <input type="text" name="Latitude" placeholder="33.865649"> <br>
        <p>Longitude:</p>
        <input type="text" name="Longitude" placeholder="25.347422"> <br>
        <p>Tree Height:</p>
        <input type="text" name="TreeHeight" placeholder="6.20"> <br>
        <br>
        <input type="submit" value="Submit">
    </form>
    <br><br>
    <h1>Show All Data</h1>
    <a href="/api/data">Show All Data</a>
    <br><br>
    <h1>Show Filter Data</h1>
    <form method="post" action="/api/greaterdata">
        <p>Greater Then</p>
        <input type="text" name="FilterHeight" placeholder="25.62124">
        <br><br>
        <input type="submit" value="Submit">
    </form>
    <br>
```

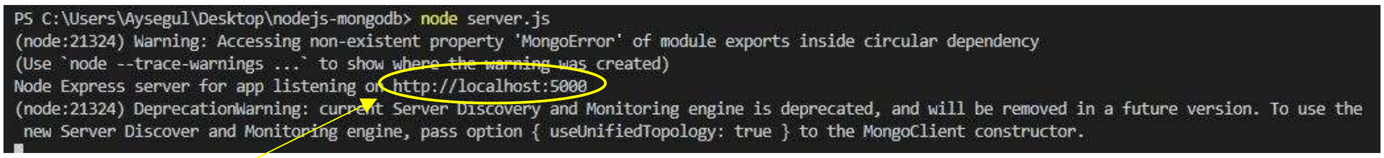


```html
    <form method="post" action="/api/lowerdata">
        <p>Lower Then</p>
        <input type="text" name="FilterHeight" placeholder="62.72124">
        <br><br>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

https://github.com/aysegulkurt/Mongodb-Nodejs-Tutorial/blob/master/nodejs-

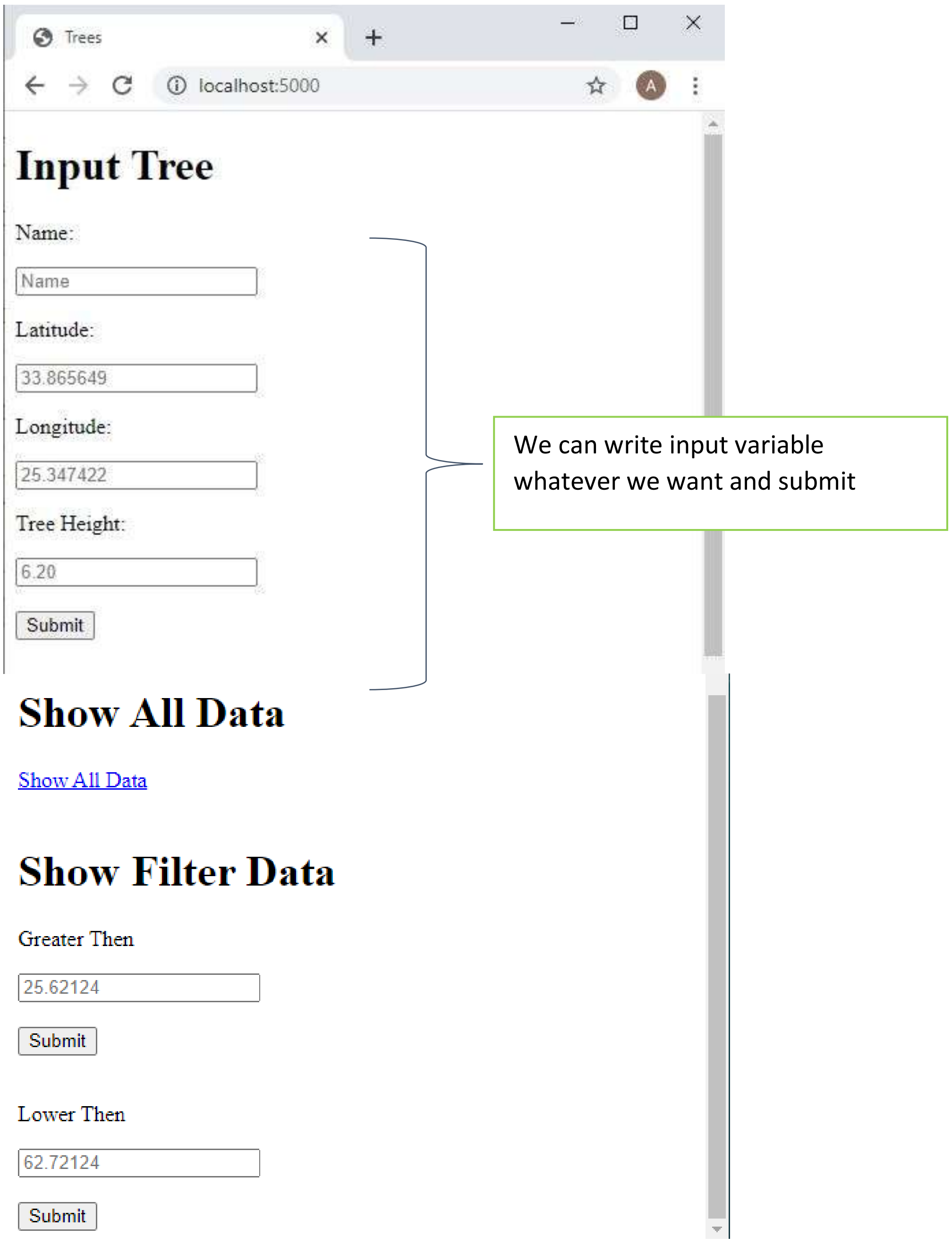
https://github.com/aysegulkurt/Mongodb-Nodejs-Tutorial/blob/master/nodejs-mongodb/index.html

We run our code by writing **node server.js** to the terminal**.**



```
PS C:\Users\Aysegul\Desktop\nodejs-mongodb> node server.js
(node:21324) Warning: Accessing non-existent property 'MongoError' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
Node Express server for app listening on http://localhost:5000
(node:21324) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the
 new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
```

We follow this link http://localhost:5000/ and we see this page:



Trees

localhost:5000

Input Tree

Name:

Name

Latitude:

33.865649

Longitude:

25.347422

Tree Height:

6.20

Submit

We can write input variable whatever we want and submit

Show All Data

Show All Data

Show Filter Data

Greater Then

25.62124

Submit

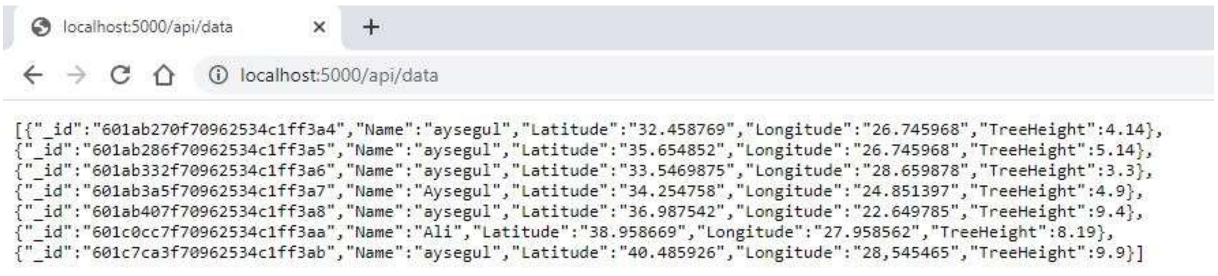
Lower Then

62.72124

Submit

When we click on show all data, we will see all submitted.

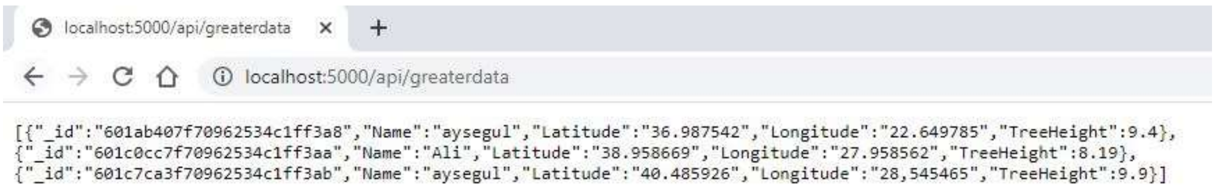
http://localhost:5000/api/data



```
[{"_id":"601ab270f70962534c1ff3a4","Name":"aysegul","Latitude":"32.458769","Longitude":"26.745968","TreeHeight":4.14},
{"_id":"601ab286f70962534c1ff3a5","Name":"aysegul","Latitude":"35.654852","Longitude":"26.745968","TreeHeight":5.14},
{"_id":"601ab332f70962534c1ff3a6","Name":"aysegul","Latitude":"33.5469875","Longitude":"28.659878","TreeHeight":3.3},
{"_id":"601ab3a5f70962534c1ff3a7","Name":"Aysegul","Latitude":"34.254758","Longitude":"24.851397","TreeHeight":4.9},
{"_id":"601ab407f70962534c1ff3a8","Name":"aysegul","Latitude":"36.987542","Longitude":"22.649785","TreeHeight":9.4},
{"_id":"601c0cc7f70962534c1ff3aa","Name":"Ali","Latitude":"38.958669","Longitude":"27.958562","TreeHeight":8.19},
{"_id":"601c7ca3f70962534c1ff3ab","Name":"aysegul","Latitude":"40.485926","Longitude":"28,545465","TreeHeight":9.9}]
```

If we want to apply filtering;

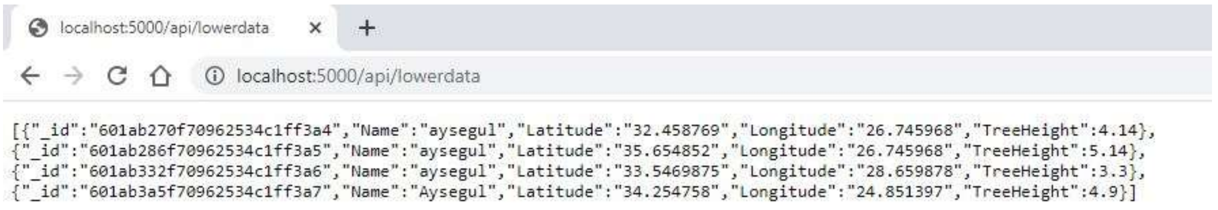
For example, we can see those with tree heights greater than 5.2 by writing the number and clicking the submit.


localhost:5000/api/greaterdata
localhost:5000/api/greaterdata

```
[{"_id":"601ab407f70962534c1ff3a8","Name":"aysegul","Latitude":"36.987542","Longitude":"22.649785","TreeHeight":9.4},
{"_id":"601c0cc7f70962534c1ff3aa","Name":"Ali","Latitude":"38.958669","Longitude":"27.958562","TreeHeight":8.19},
{"_id":"601c7ca3f70962534c1ff3ab","Name":"aysegul","Latitude":"40.485926","Longitude":"28,545465","TreeHeight":9.9}]
```

http://localhost:5000/api/greaterdata

For lower then filtering to 5.2:



```
[{"_id":"601ab270f70962534c1ff3a4","Name":"aysegul","Latitude":"32.458769","Longitude":"26.745968","TreeHeight":4.14},
{"_id":"601ab286f70962534c1ff3a5","Name":"aysegul","Latitude":"35.654852","Longitude":"26.745968","TreeHeight":5.14},
{"_id":"601ab332f70962534c1ff3a6","Name":"aysegul","Latitude":"33.5469875","Longitude":"28.659878","TreeHeight":3.3},
{"_id":"601ab3a5f70962534c1ff3a7","Name":"Aysegul","Latitude":"34.254758","Longitude":"24.851397","TreeHeight":4.9}]
```

http://localhost:5000/api/lowerdata

That is all about information this project.