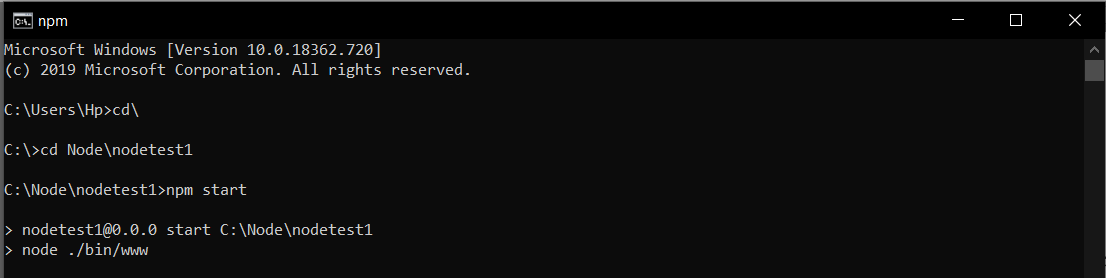
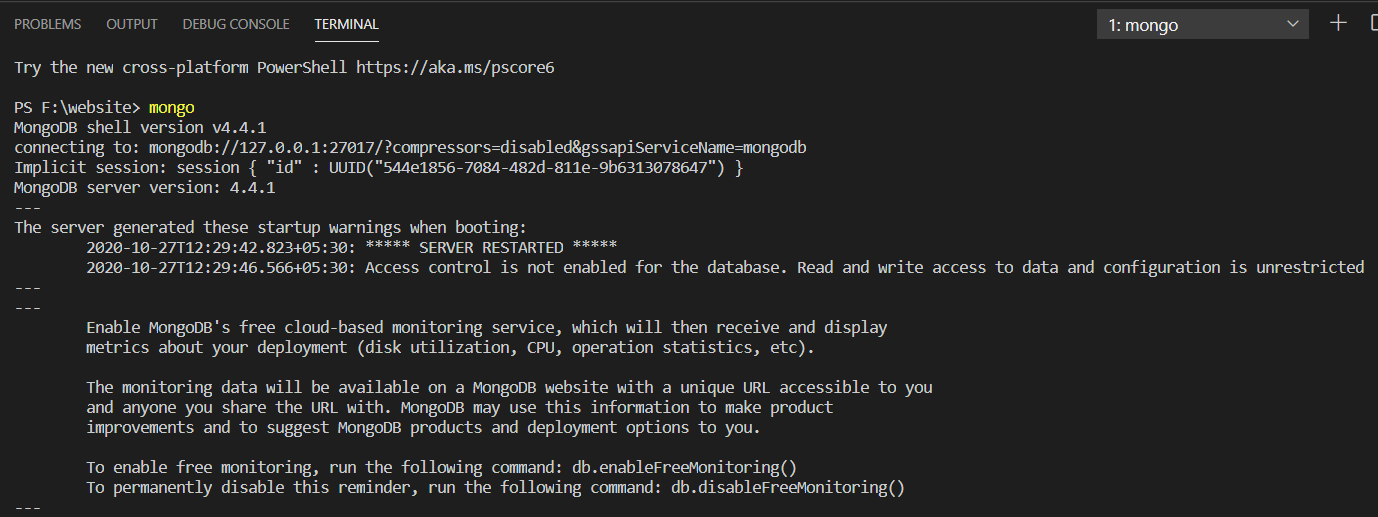
S**TART THE SERVER:-**

1. Start the Node.js server.



1. Start the Mongo DB server.



**SOURCE CODE:-**

**Mongo Db:**

In this , I create a database with the name of “nodetest1”, collection “ cust” and insert some documents in it.

[

  {

    "\_id": "5fa3a4be0150aca7e56a7040",

    "Creator": "admin@leap.com",

    "Tags": "Kids general Knowledge",

    "Title": "GK for Kids",

    "Owner": "admin@leap.com",

    "Description": "GK for kids course covers MCQ",

    "Category": "Competitive exam",

    "Sub-category": "General Knowledge",

    "Status": "Admin Approved"

  },

  {

    "\_id": "5fa3ad470150aca7e56a7041",

    "Creator": "admin1@leap.com",

    "Tags": "Python Basics",

    "Title": "Python for Beginners",

    "Owner": "admin1@leap.com",

    "Description": "Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.",

    "Category": "Programming",

    "Sub-category": "Learning",

    "Status": "Admin Approved"

  },

  {

    "\_id": "5fa3ade30150aca7e56a7042",

    "Creator": "admin2@leap.com",

    "Tags": "Java Basics",

    "Title": "Java for Beginners",

    "Owner": "admin2@leap.com",

    "Description": "Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications.",

    "Category": "Programming",

    "Sub-category": "Learning",

    "Status": "Admin Approved"

  },

  {

    "\_id": "5fa3ea44d82da320c25919c1",

    "Creator": "admin3@leap.com",

    "Tags": "C Basics",

    "Title": "C for Beginners",

    "Owner": "admin3@leap.com",

    "Description": "C is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs.",

    "Category": "Programming",

    "Sub-category": "Learning",

    "Status": "Admin Approved"

  },

  {

    "\_id": "5fa3ea9bd82da320c25919c2",

    "Creator": "admin4@leap.com",

    "Tags": "C++ Basics",

    "Title": "C++ for Beginners",

    "Owner": "admin4@leap.com",

    "Description": "C++ is a cross-platform language that can be used to create high-performance applications. C++ was developed by Bjarne Stroustrup, as an extension to the C language. ",

    "Category": "Programming",

    "Sub-category": "Learning",

    "Status": "Admin Approved"

  }

]

**App.js:**

The app.js file is like main file. In this file, I connect mongo db to it. And the source code given below:

var createError = require('http-errors');

var express = require('express');

var path = require('path');

var cookieParser = require('cookie-parser');

var logger = require('morgan');

//connecting mongodb

var monk = require('monk');

var db = monk('localhost:27017/nodetest1');

var indexRouter = require('./routes/index');

var usersRouter = require('./routes/users');

var app = express();

// view engine setup

app.set('views', path.join(\_\_dirname, 'views'));

app.set('view engine', 'ejs');

app.use(logger('dev'));

app.use(express.json());

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

app.use(cookieParser());

app.use(express.static(path.join(\_\_dirname, 'public')));

// Make our db accessible to our router

app.use(function(req,res,next){

  req.db = db;

  next();

});

app.use('/', indexRouter);

app.use('/users', usersRouter);

// catch 404 and forward to error handler

app.use(function(req, res, next) {

  next(createError(404));

});

// error handler

app.use(function(err, req, res, next) {

  // set locals, only providing error in development

  res.locals.message = err.message;

  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page

  res.status(err.status || 500);

  res.render('error');

});

module.exports = app;

**Lang.ejs :**

Lang.ejs is for front end and for implementing partial search

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

  h2 {text-align: center;}

\* {

  box-sizing: border-box;

}

#myInput {

  background-image: url('/css/searchicon.png');

  background-position: 10px 10px;

  background-repeat: no-repeat;

  width: 100%;

  font-size: 16px;

  padding: 12px 20px 12px 40px;

  border: 1px solid #ddd;

  margin-bottom: 12px;

}

#myTable {

  border-collapse: collapse;

  width: 100%;

  border: 1px solid #ddd;

  font-size: 18px;

}

#myTable th, #myTable td {

  text-align: left;

  padding: 12px;

}

#myTable tr {

  border-bottom: 1px solid #ddd;

}

#myTable tr.header, #myTable tr:hover {

  background-color: #f1f1f1;

}

</style>

</head>

<body>

<h2>CLEAdRFUNDs</h2>

<form style="margin:auto;max-width:300px">

  <input type="text" id="myInput" onkeyup="myFunction()" placeholder="Search for learning.." title="Type in a name" >

</form>

<table id="myTable" style="margin:auto;max-width:300px">

  <tr>

    <td>GK for Kids</td>

  </tr>

  <tr>

    <td>Python for Beginners</td>

  </tr>

  <tr>

    <td>Java for Beginners</td>

  </tr>

  <tr>

    <td>C for Beginners</td>

  </tr>

  <tr>

    <td>C++ for Beginners</td>

  </tr>

</table>

//implementing partial search

<script>

function myFunction() {

  var input, filter, table, tr, td, i, txtValue;

  input = document.getElementById("myInput");

  filter = input.value.toUpperCase();

  table = document.getElementById("myTable");

  tr = table.getElementsByTagName("tr");

  for (i = 0; i < tr.length; i++) {

    td = tr[i].getElementsByTagName("td")[0];

    if (td) {

      txtValue = td.textContent || td.innerText;

      if (txtValue.toUpperCase().indexOf(filter) > -1) {

        tr[i].style.display = "";

      } else {

        tr[i].style.display = "none";

      }

    }

  }

}

</script>

</body>

</html>

**Index.js:**

In this file, I link the lang.js (i.e., front end) to it.

var express = require('express');

var router = express.Router();

/\* GET home page. \*/

router.get('/', function(req, res) {

  res.render('index', { title: 'Express' });

});

/\* GET Lang page. \*/

router.get('/lang', function(req, res) {

  var db = req.db;

  var collection = db.get('cust');

  collection.find({},{},function(e,docs){

      res.render('lang', {

          "lang" : docs

      });

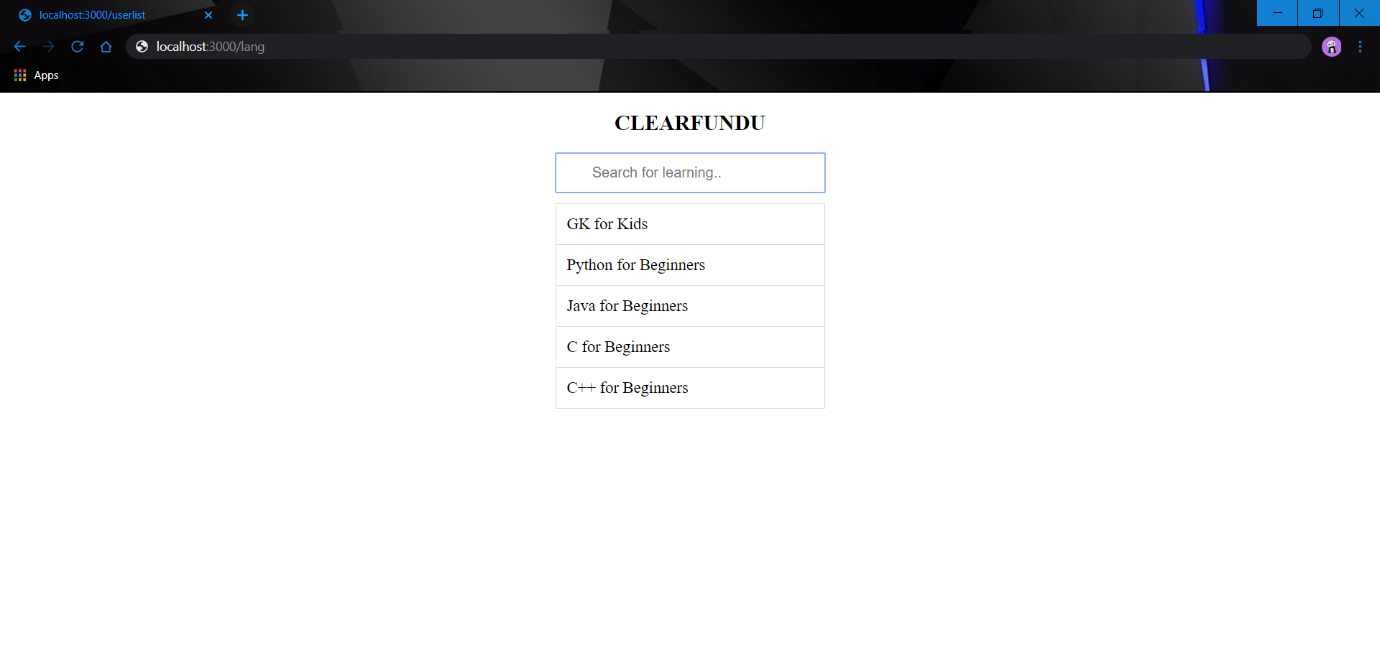
  });

})

module.exports = router;

**OUTPUT:-**

Now I run the above source code with the help of server [**https://localhost:3000/lang**](https://localhost:3000/lang)



Partial search output:-

