Name: Gaurav Kumar Singh

Registration Number:19BCE2119

Course: Web Technology

Lab FAT

Create a database which will store the Book information which includes the following:

1) Title of the book

2) Author Name

3) ISBN number

4) Publisher name

5) Edition

6) Price

Create a web page with all the above fields for inserting data into the database when a button is clicked. Also validate the fields wherever required. Use a simple styling.

CODE:

Index.js

```
var express = require("express");

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

const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/LabFAT', {

   useNewUrlParser: true

});

var app = express()

app.use(bodyParser.json());

app.use(express.static('public'));
app.use(bodyParser.urlencoded({
  extended: true
}));
const books = new mongoose.Schema({
  Title: String,
  Author: String,
  ISBN: Number,
  Publisher: String,
  Edition: String,
  Price: Number
});
const Books = mongoose.model("books", books);
app.get("/", function (req, res) {
  res.sendFile(__dirname + "/LabFAT.html")
})
app.get("/addbook", function (req, res) {
  var title = req.query.ti;
  var author = req.query.au;
  var isbn = req.query.is;
  var publisher = req.query.pu;
  var edition = req.query.ed;
  var price = req.query.pr;
  console.log(title)
  console.log(author)
  console.log(isbn)
  console.log(publisher)
  console.log(edition)
  console.log(price)
  stringreg = /\w+/;
  ISBNreg = /^(?=(?:\D*\d){10}(?:(?:\D*\d){3})?$)[\d-]+$/
  pricereg = /\d/
```

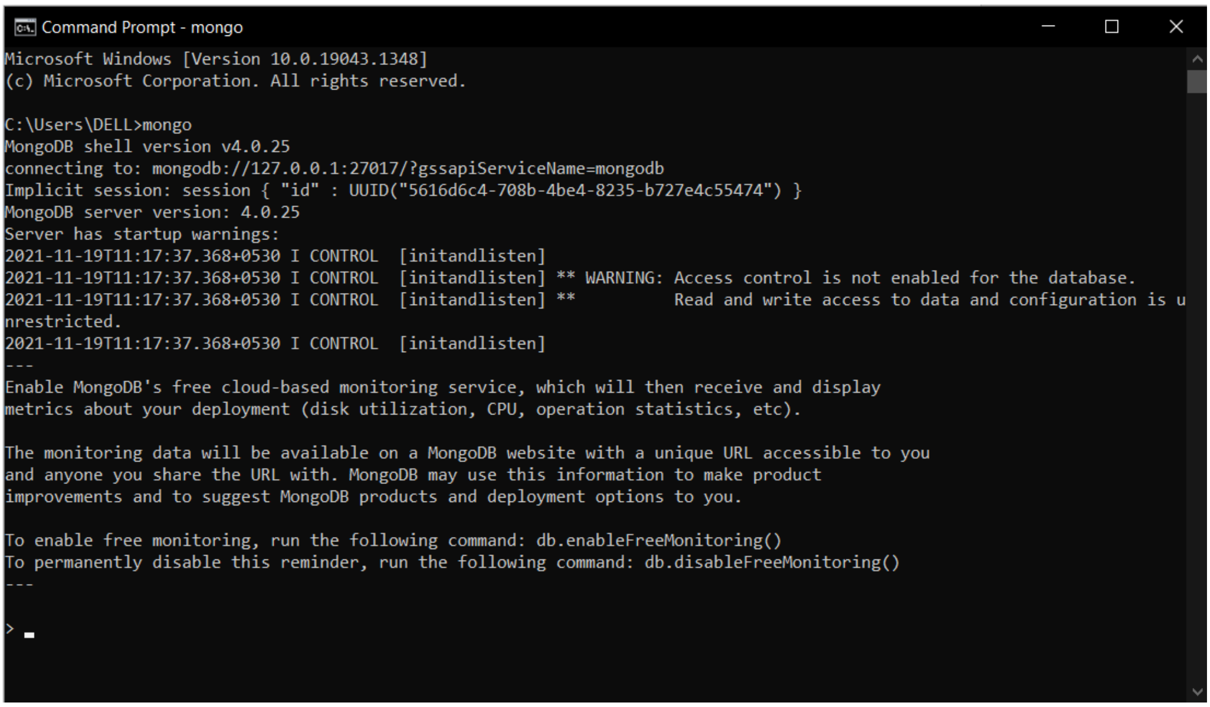
```
    var flag=0;
    if (stringreg.test(title) && stringreg.test(author) && stringreg.test(publisher) &&
stringreg.test(edition) && ISBNreg.test(isbn) && pricereg.test(price)) {
        Books.insertMany([{
            Title: title,
            Author: author,
            ISBN: isbn,
            Publisher: publisher,
            Edition: edition,
            Price: price
        }], function (err, result) {
            if (err) {
                res.end(err);
            } else {
                res.end(JSON.stringify(result));
            }
        })
    } else{
        res.end("Enter Valid Data");
    }

})
app.listen(3000, function () {
    console.log("Server Started");
})
```

LabFAT.html

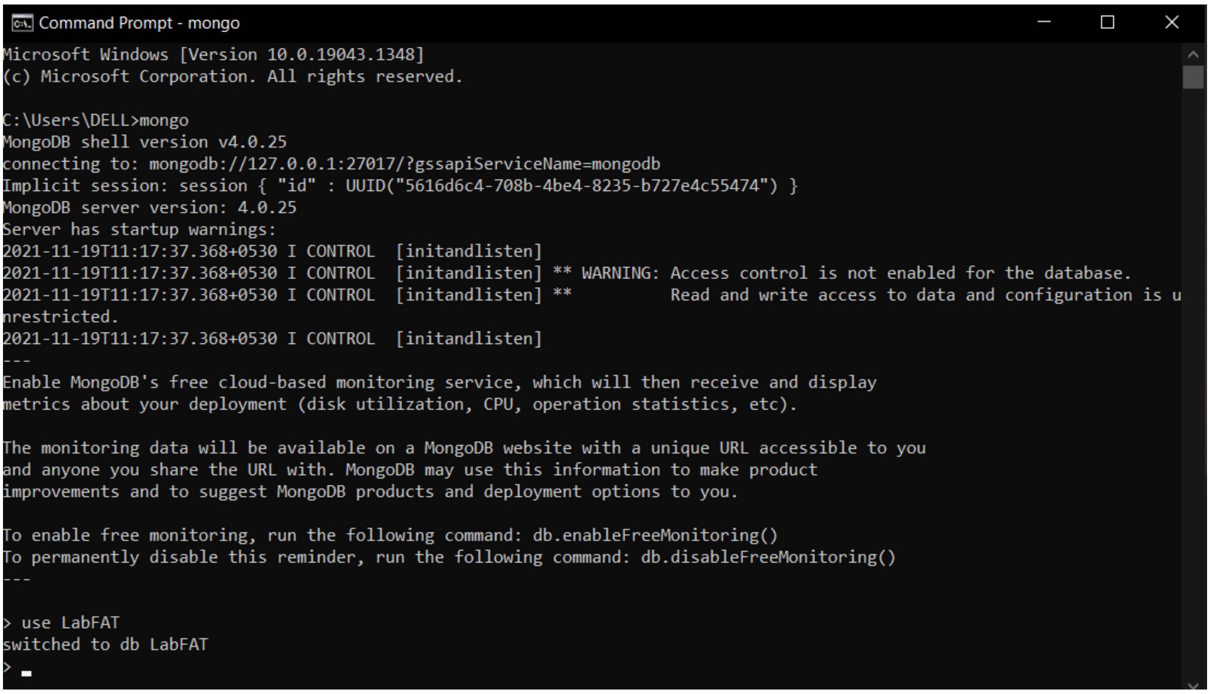
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        form{
            margin: 0 auto;
        }
        table tr td:nth-child(1){
            text-align:right
        }
        table tr{
            height:40px;
        }
    </style>
</head>
<body>
    <form action="/addbook" method=get style="width:fit-content">
        <fieldset>
            <legend><b>Add Book</b></legend>
            <table>
                <tr><td>Title:</td><td><input type=text name=ti required
placeholder=BookName></td></tr>
                <tr><td>Author:</td><td><input type=text name=au required
placeholder=Author></td></tr>
                <tr><td>ISBN no:</td><td><input type=number name=is required
placeholder=1234567890></td></tr>
                <tr><td>Publisher:</td><td><input type=text name=pu required
placeholder=Publisher></td></tr>
                <tr><td>Edition:</td><td><input type=text name=ed required
placeholder=Edition></td></tr>
                <tr><td>Price:</td><td><input  name=pr required ></td></tr>
            </table>
            <input type=submit>&nbsp;&nbsp;<input type=reset><br>
        </fieldset>
    </form>
</body>
</html>
```

SCREENSHOTS

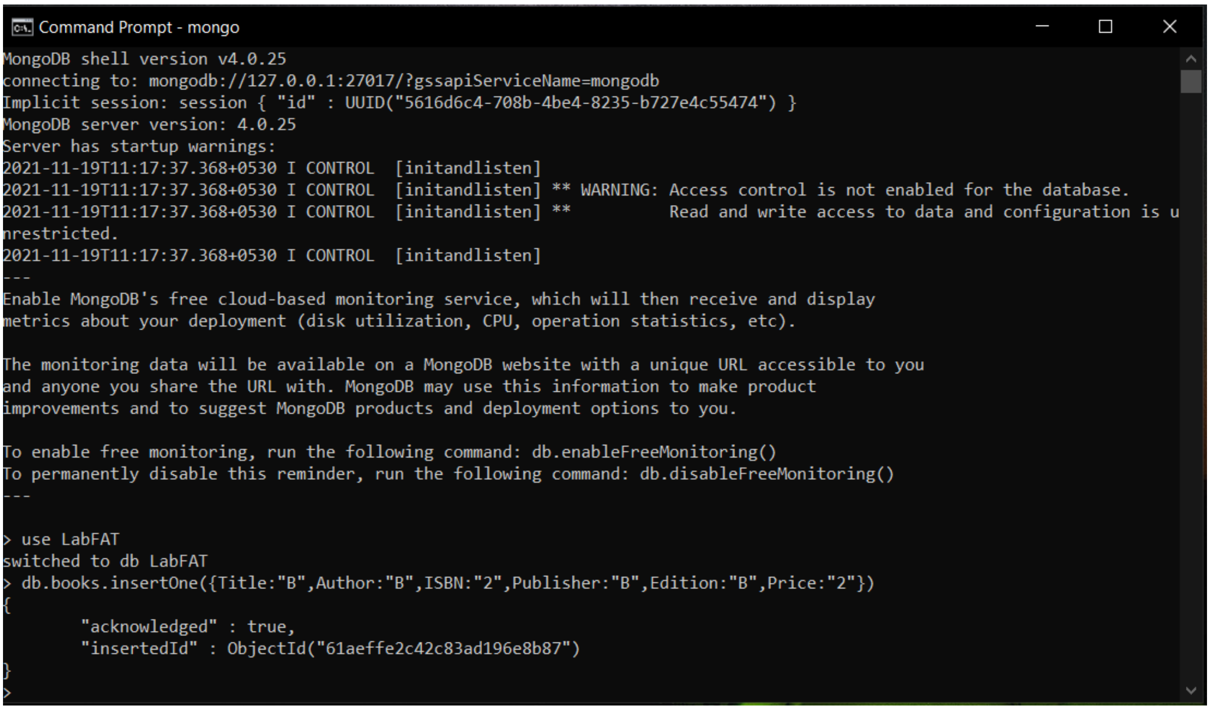
ADDING DATA THROUGH MONGO TERMINAL



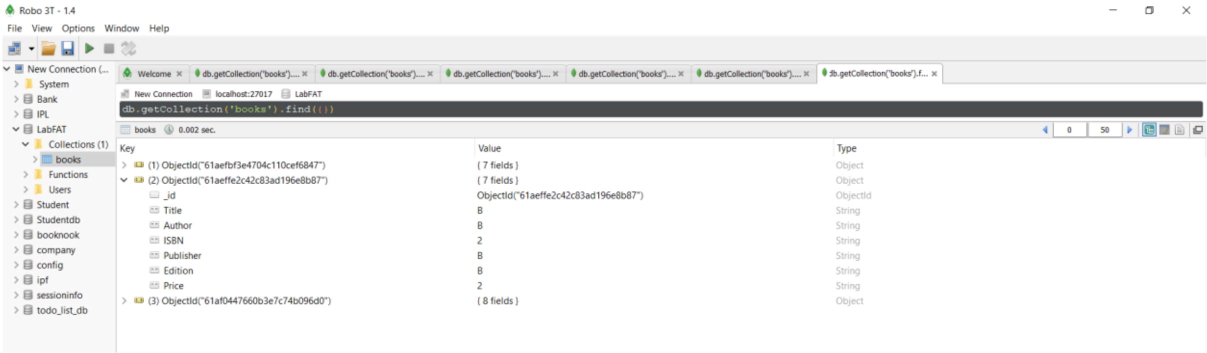
```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>mongo
MongoDB shell version v4.0.25
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("5616d6c4-708b-4be4-8235-b727e4c55474") }
MongoDB server version: 4.0.25
Server has startup warnings:
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten]
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is u
nrestricted.
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> 
```



```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>mongo
MongoDB shell version v4.0.25
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("5616d6c4-708b-4be4-8235-b727e4c55474") }
MongoDB server version: 4.0.25
Server has startup warnings:
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten]
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is u
nrestricted.
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> use LabFAT
switched to db LabFAT
> 
```



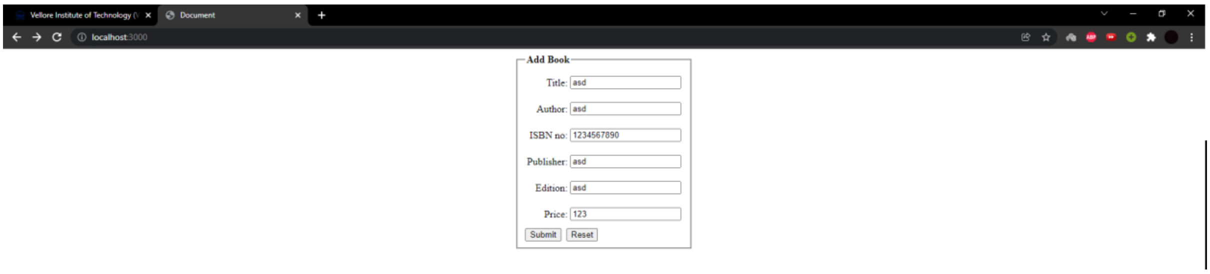
```
MongoDB shell version v4.0.25
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("5616d6c4-708b-4be4-8235-b727e4c55474") }
MongoDB server version: 4.0.25
Server has startup warnings:
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten]
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is u
nrestricted.
2021-11-19T11:17:37.368+0530 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> use LabFAT
switched to db LabFAT
> db.books.insertOne({Title:"B",Author:"B",ISBN:"2",Publisher:"B",Edition:"B",Price:"2"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("61aeffe2c42c83ad196e8b87")
}
>
```



```
db.getCollection('books').find({})
```

Key	Value	Type
(1) ObjectId("61aefbf3e4704c110cef6847")	{ 7 fields }	Object
(2) ObjectId("61aeffe2c42c83ad196e8b87")	{ 7 fields }	Object
_id	ObjectId("61aeffe2c42c83ad196e8b87")	ObjectId
Title	B	String
Author	B	String
ISBN	2	String
Publisher	B	String
Edition	B	String
Price	2	String
(3) ObjectId("61af0447660b3e7c74b096d0")	{ 8 fields }	Object

ADDING DATA THROUGH WEBPAGE USING EXPRESS NODE.JS AND MONGOOSE



Add Book

Title: asd

Author: asd

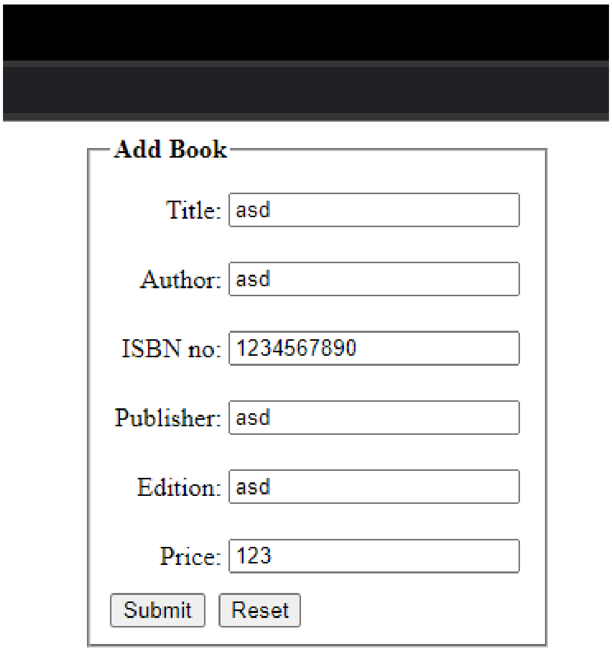
ISBN no: 1234567890

Publisher: asd

Edition: asd

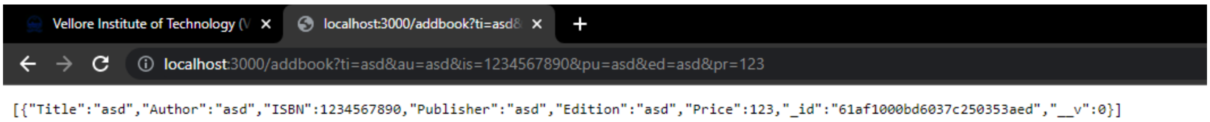
Price: 123

Submit   Reset



Add Book

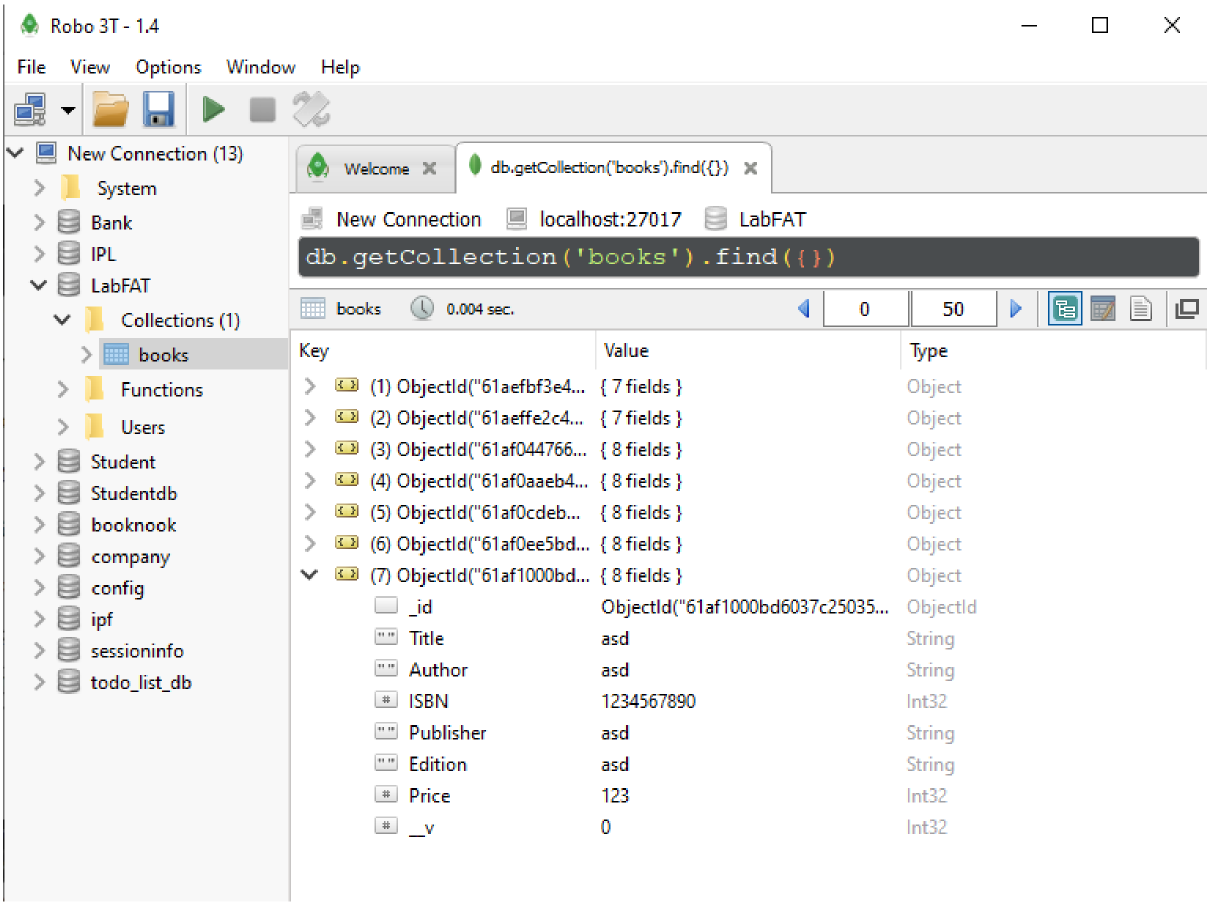
Title:	asd
Author:	asd
ISBN no:	1234567890
Publisher:	asd
Edition:	asd
Price:	123



Vellore Institute of Technology (V
localhost:3000/addbook?ti=asd&

localhost:3000/addbook?ti=asd&au=asd&is=1234567890&pu=asd&ed=asd&pr=123

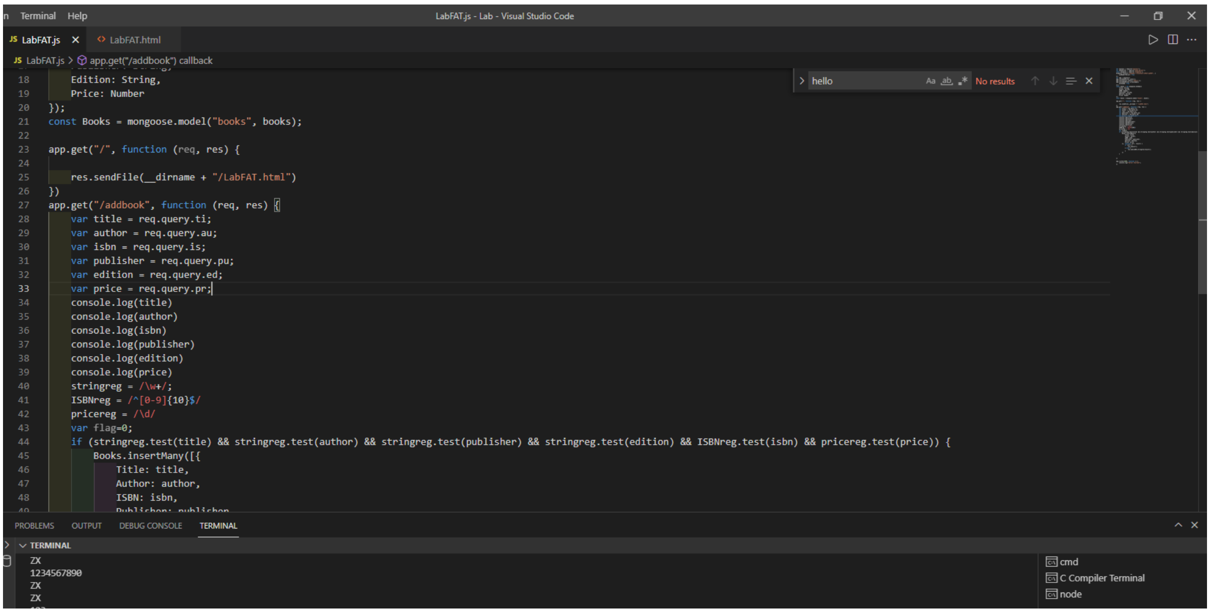
```
[{"Title":"asd","Author":"asd","ISBN":1234567890,"Publisher":"asd","Edition":"asd","Price":123,"_id":"61af1000bd6037c250353aed","__v":0}]
```



```
db.getCollection('books').find({})
```

Key	Value	Type
(1) ObjectId("61aefbf3e4...	{ 7 fields }	Object
(2) ObjectId("61aeffe2c4...	{ 7 fields }	Object
(3) ObjectId("61af044766...	{ 8 fields }	Object
(4) ObjectId("61af0aaeb4...	{ 8 fields }	Object
(5) ObjectId("61af0cdeb...	{ 8 fields }	Object
(6) ObjectId("61af0ee5bd...	{ 8 fields }	Object
(7) ObjectId("61af1000bd...	{ 8 fields }	Object
_id	ObjectId("61af1000bd6037c25035...	ObjectId
Title	asd	String
Author	asd	String
ISBN	1234567890	Int32
Publisher	asd	String
Edition	asd	String
Price	123	Int32
__v	0	Int32

CODE SCREENSHOT



```
    Edition: String,
    Price: Number
});
const Books = mongoose.model("books", books);

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

    res.sendFile(__dirname + "/LabFAT.html")
})
app.get("/addbook", function (req, res) {
    var title = req.query.ti;
    var author = req.query.au;
    var isbn = req.query.is;
    var publisher = req.query.pu;
    var edition = req.query.ed;
    var price = req.query.pr;
    console.log(title)
    console.log(author)
    console.log(isbn)
    console.log(publisher)
    console.log(edition)
    console.log(price)
    stringreg = /\w+/;
    ISBNreg = /^[0-9]{10}$/
    pricereg = /\d/
    var flag=0;
    if (stringreg.test(title) && stringreg.test(author) && stringreg.test(publisher) && stringreg.test(edition) && ISBNreg.test(isbn) && pricereg.test(price)) {
        Books.insertMany([{
            Title: title,
            Author: author,
            ISBN: isbn,
```

```
ZX
1234567890
ZX
ZX
```