

Name: Sriram R	SRN: PES1UG20CS435	Section: H
	Date: 12-12-2021	Roll # : 16

### PROBLEM STATEMENT(EVEN SRN's)

1. Create an API that has a collection of books having different fields (such as book\_id, book\_name, book\_price,book\_authetc)for each book.  
Using HTTP methods GET method extract the data of book using book\_id,  
Using PUT method update the price, Using POST method insert a new data and display the same. (Use MongoDB database)
2. Create student resume with details (such as name, dob, qualification, nationality etc...) using formdata and upload the student photo to the form.

### OBJECTIVE

The objective of this exercise is to test the student on ExpressJS framework. It evaluates the student's knowledge of http request, response objects. Creating RestFul API and web services

### PREREQUISITE

In order to write this program, the student needs to understand the fundamentals of HTML and CSS. The student must be familiar with basic Javascript and express module.

### ALGORITHM

Create new document in MongoDB database -> upload data -> implement GET, POST & PUT requests -> link a directory for file upload (for Q2) -> execute

### PROGRAM

q1DB.js :

```
var MongoClient = require("mongodb").MongoClient;
var url = "mongodb://localhost:27017/bookDB";

MongoClient.connect(url, { useUnifiedTopology: true }, function
(err, db) {
```

**Unit 5: EXPRESS JS ,FORMS AND FILE UPLOAD**

```
if (err) throw err;
console.log("Database Created");
var dbo = db.db("bookDB");

// collection creation
dbo.createCollection("bookCollection", function (err, res) {
  if (err) throw err;
  console.log("Collection Created");
});

// inserting many documents
var myObjs = [{"book_id":1, "book_name":"qwerty",
"book_price":"100", "book_author":"uiop"},
              {"book_id":2, "book_name":"asdfg",
"book_price":"200", "book_author":"ghjkl"},
              {"book_id":3, "book_name":"zxcvb",
"book_price":"300", "book_author":"nm"},
              {"book_id":4, "book_name":"wdfvb",
"book_price":"400", "book_author":"oijhb"}]

dbo.collection("bookCollection").insertMany(myObjs, function
(err, res) {
  if (err) throw err;
  console.log("Number of documents inserted : ",
res.insertedCount);
  db.close();
});

});
```

**q1API.js :**

```
var express= require('express');
var MongoClient= require('mongodb').MongoClient;
var router= express.Router();

router.get("/", function(req,res) {
  MongoClient.connect("mongodb://localhost:27017",
{useUnifiedTopology:true}, function(err,client){
    if (err) throw err;
    const db = client.db('bookDB');

db.collection('bookCollection').find(req.query).toArray(function
```

## Unit 5: EXPRESS JS ,FORMS AND FILE UPLOAD

```
(err,objs) {
    res.send(objs);
  });
});
});

router.get("/:id",function(req,res){
  MongoClient.connect("mongodb://localhost:27017",
{useUnifiedTopology:true}, function(err,client) {
    if (err) throw err;
    const db = client.db('bookDB');

    db.collection('bookCollection').findOne({book_id:parseInt(req.pa
rams.id)}, function(err,objs) {
      res.send(objs);
    });
  });
});

router.post("/post",function(req,res){

MongoClient.connect("mongodb://localhost:27017",{useUnifiedTopol
ogy:true}, function(err,client){
  if (err) throw err;
  const db = client.db('bookDB');
  db.collection('bookCollection').insert(req.body,
function(err,objs){
    res.send("Save successful");
  });
});
});

router.put("/:price", function(req,res){
  MongoClient.connect("mongodb://localhost:27017",
{useUnifiedTopology:true}, function(err,client){
    if (err) throw err;
    const db= client.db('bookDB');

    db.collection('bookCollection').update({book_price:req.params.pr
ice}, {$set: req.body }, { new: true, upsert: true,
returnOriginal: false }, function(err,objs){
      res.send("Update successful");
    });
  });
});
```

**Unit 5: EXPRESS JS ,FORMS AND FILE UPLOAD**

```
});  
  
module.exports = router;  
  
q1Server.js :  
  
var express= require('express');  
var app = express();  
var https=require('https');  
var http = require('http');  
var bodyParser= require("body-parser");  
var studrouter = require("./q1API.js");  
var MongoClient=require("mongodb").MongoClient;  
const fetch = (...args) => import('node-fetch').then(({default:  
fetch}) => fetch(...args));  
  
var newd = {  
  "book_id":4,  
  "book_name":"helloWorld",  
  "book_price":"500",  
  "book_author":"Bjarne Stroustrup"  
}  
  
fetch("http://localhost:3000/book", {  
  method:"POST",  
  body: JSON.stringify(newd),  
  headers:{'content-type':'application/json'}  
}).then(res=>res.json).then(json=>console.log(json));  
  
var newd = {  
  "book_id":4,  
  "book_name":"helloWorld",  
  "book_price":"512",  
  "book_author":"Bjarne Stroustrup"  
}  
  
fetch("http://localhost:3000/book/512", {  
  method:"PUT",  
  body: JSON.stringify(newd),  
  headers:{'content-type':'application/json'}  
}).then(res=>res.json).then(json=>console.log(json));  
  
app.use(bodyParser.json());  
app.use("/book",studrouter);
```

**Unit 5: EXPRESS JS ,FORMS AND FILE UPLOAD**

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

**q2.js :**

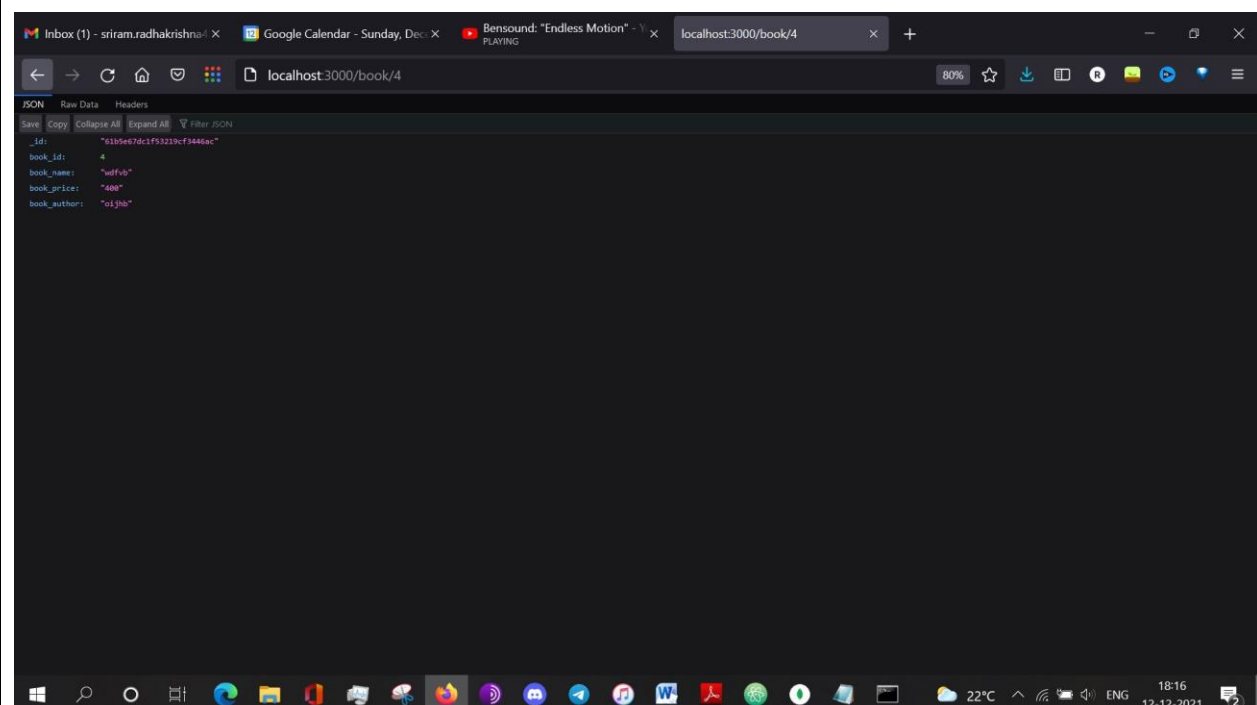
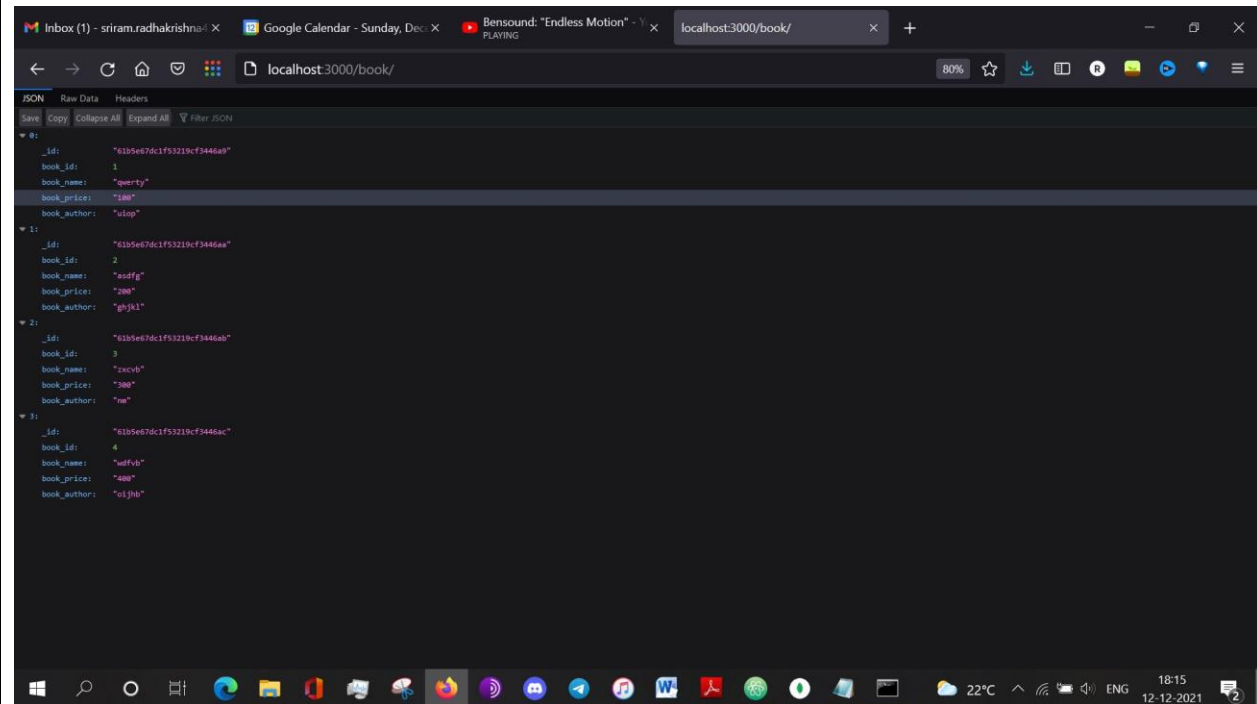
```
var express = require("express");  
var app = express();  
var fileupload = require("express-fileupload");  
  
app.use(fileupload());  
app.post('/upload',function(req,res){  
  
  if(!req.files || req.files.length==0)  
    return res.status(400).send("No file to upload");  
  
  var sampleFile = req.files.sampleFile;  
  
  sampleFile.mv("./files/" + sampleFile.name, function(err){  
  
    if(err) throw (err);  
    res.send("File" + sampleFile.name + " Uploaded");  
  });  
  
});  
  
app.get("/form",function(req,res){  
  var retform = "<form action='http://localhost:4000/upload'  
method='post' enctype='multipart/form-data'><input type='file'  
name='sampleFile'/> <input type='submit'  
value='upload'/></form>";  
  res.send(retform)  
});  
  
app.listen(4000,function(){  
  console.log("Server running ...")  
});
```

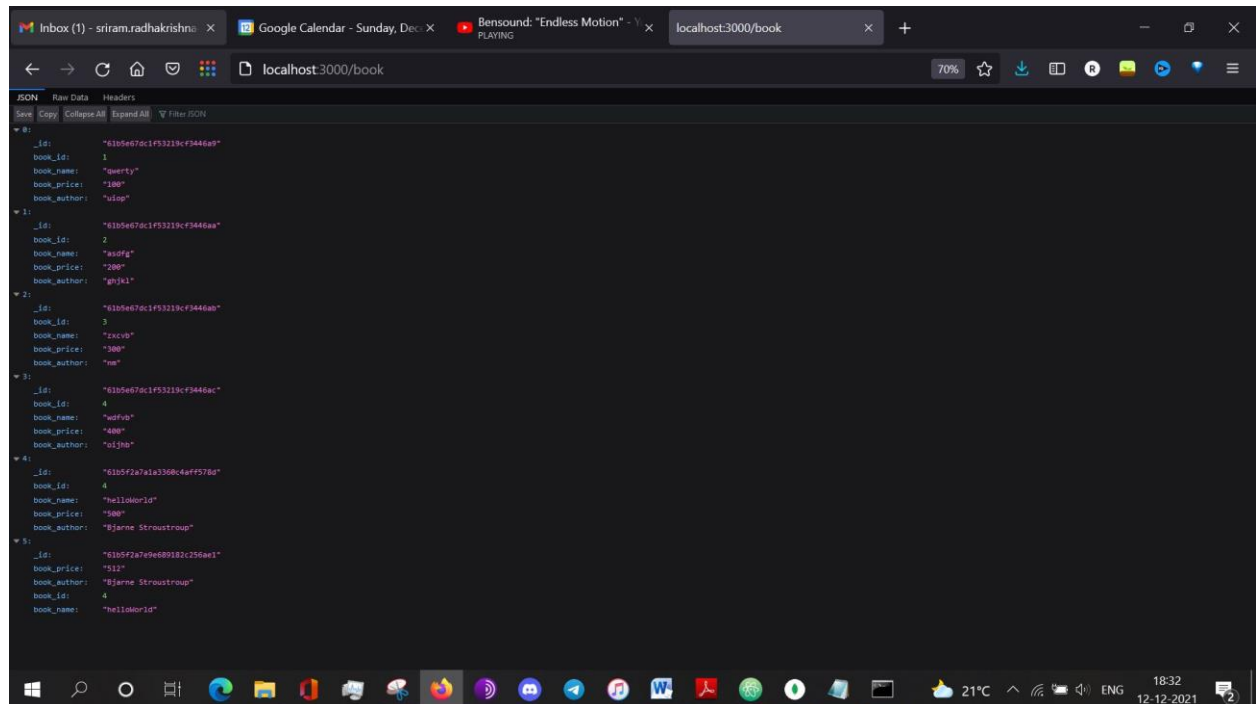
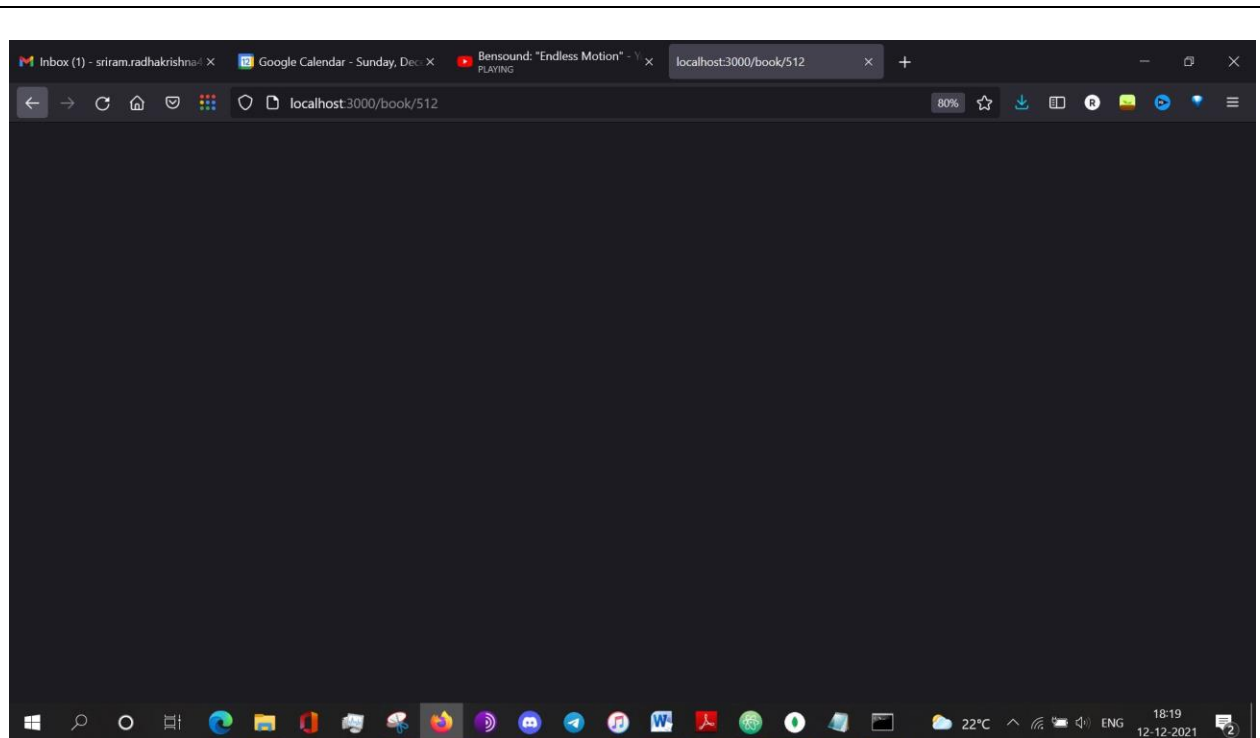
**TEST CASES**

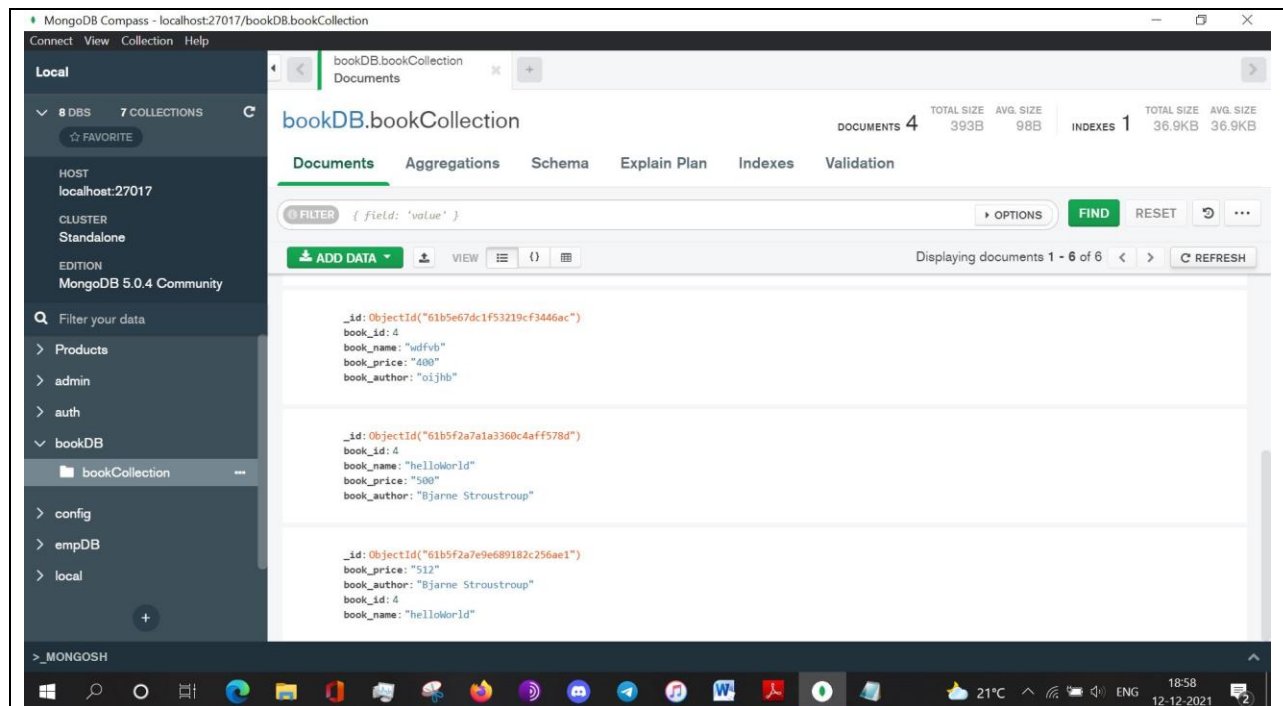
Q1 : Unmodified database, modified database, ID query & PUT request.  
Q2 : File upload

### SCREENSHOT OF OUTPUT

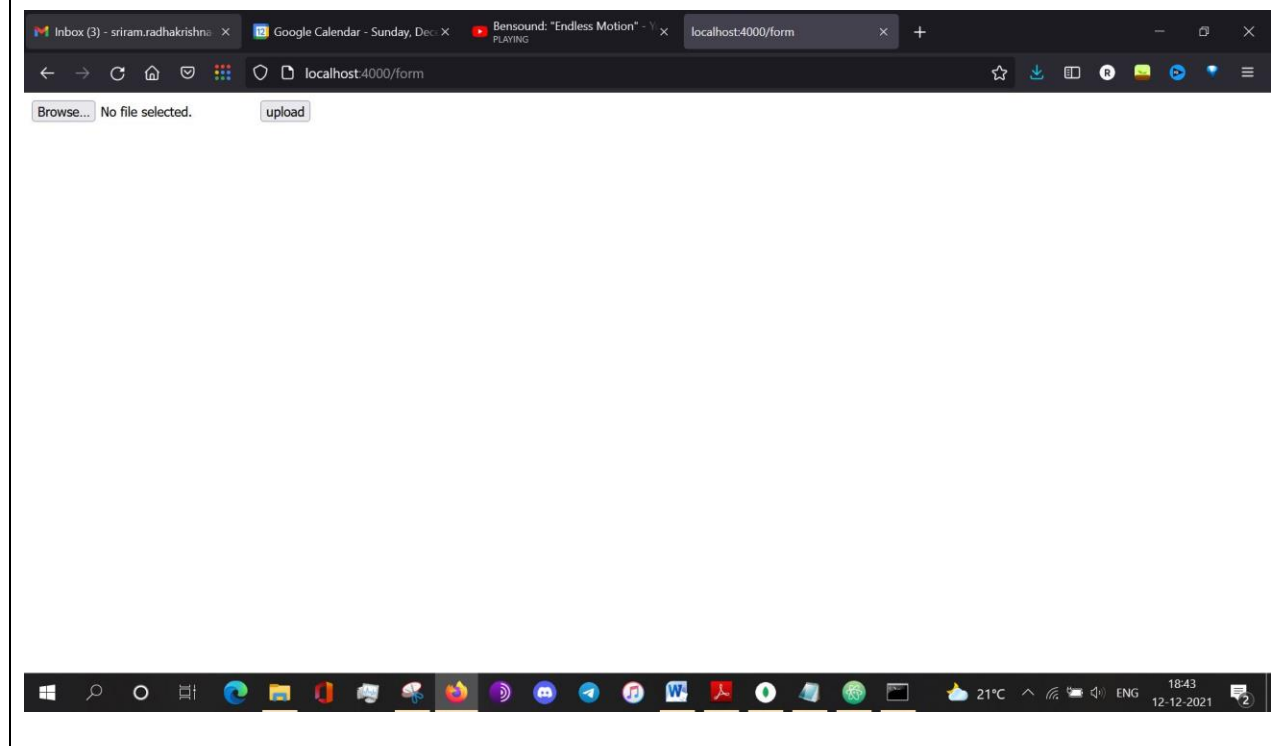
Q1 :



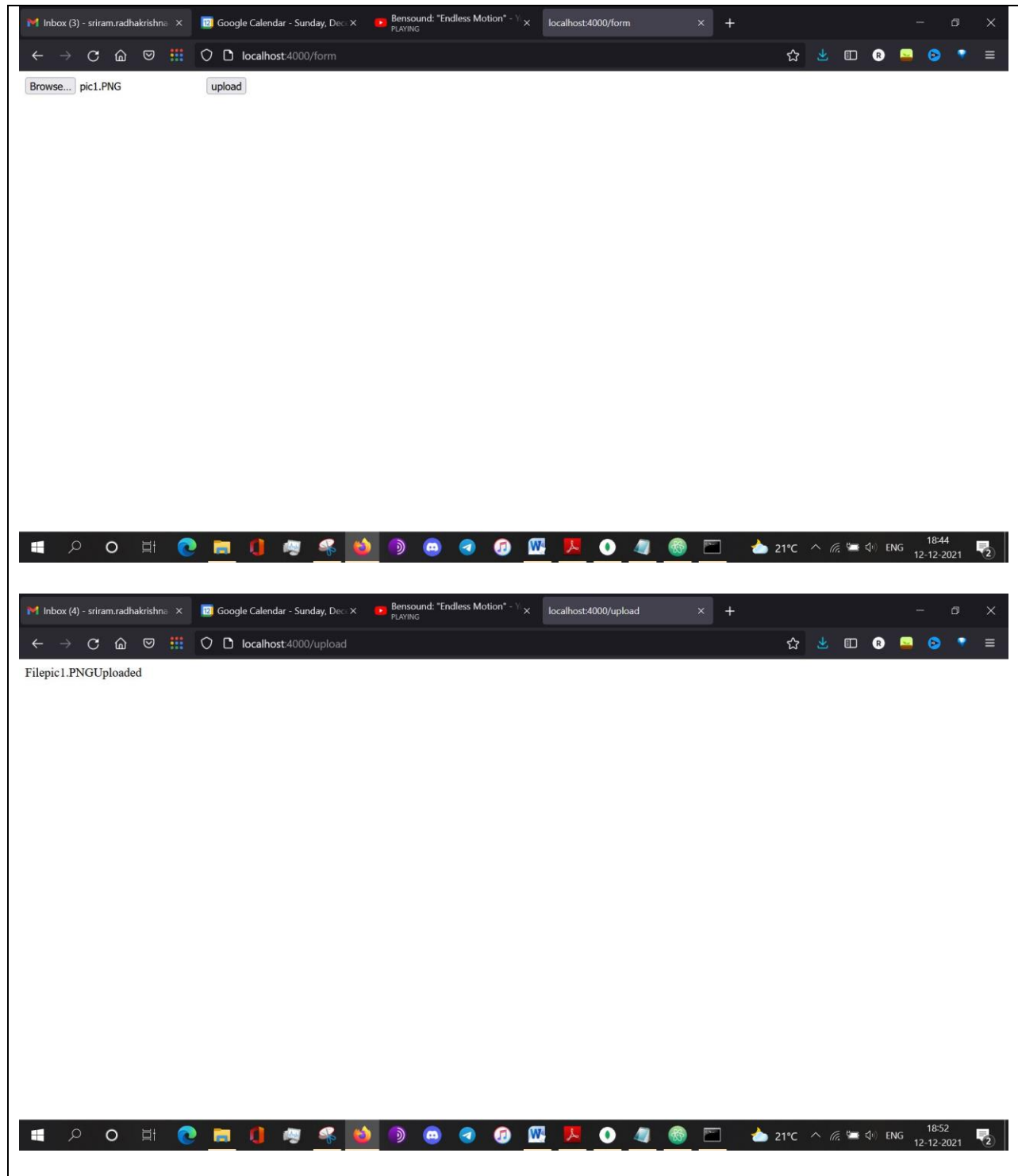




Q2 :







## Unit 5: EXPRESS JS ,FORMS AND FILE UPLOAD

