PES UNIVERSITY

Unit 5: EXPRESS JS ,FORMS AND FILE

AUG-DEC

Name: SANMAT SANJAYAKUMAR PAYAGOUDAR	SRN: PES1UG20CS385	Section:G
	Date:11-12-2021	Exercise No:5

PROBLEM STATEMENT(ODD SRN's)

1. Create an API that has a collection of watches having different fields (such as model_no, model_name, model_price,model_competc)for each watch. Using HTTP methods GET method extract the data of watch using model_no, Using PUT method update the price, Using POST method insert a new data and display the same. (Use MongoDb database)
2. Create employee resume with details (such as name, dob, qualification,
job_expetc…) using formdata and upload the employee photo.

OBJECTIVE

The objective of this exercise is to test the student on ExpressJS framework.
It evaluates the student's knowledge of http request, respose 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

[redacted]

PROGRAM

PROGRAM 1:

IN ind.js

```
express = require("express")


st_router = express.Router()


mongoclient = require("mongodb").MongoClient
```

```
mongoclient.connect("mongodb://localhost:27017/?readPreference=primary&appname=MongoDB%20Compass&directConnection=true&ssl=false",{useUnifiedTopology:true},function(err,client){

    if(err) throw err

    console.log("connected successfully to mongo")

    db = client.db("Newwatches")

})

st_router.get("/",(req,res)=>{


db.collection("watch").find({"model_no":"XX01"}).toArray(function(err,docs){

        if(err) throw err

        res.send(docs)

        console.log(docs)

    }

    )
})

st_router.get("/post-details",(req,res)=>{
```

```
db.collection("watch").insertOne({"nodel_no":"G003","model_name":"Gshock",
"model_price":"10000"},function(err,docs){

        if(err) throw err

        res.send("inserted succesfully to db")
        res.send(res)




    }

    )




})

st_router.get("/update-details",(req,res)=>{


db.collection("watch").update({"model_no":"G003"},{$set:{"model_price":"10
000"}},function(err,docs){

        if(err) throw err

        res.send("updated successfully")

    })



})
```

```
module.exports = st_router
```

IN ser.js

```
express = require("express")
app = express()

std_router = require("./ind.js")

app.get("/",std_router)

app.get("/post-details",std_router)
app.get("/update-details",std_router)

port = 3000
app.listen(port,()=>{
    console.log("server started at port ",port)
})
```

PROGRAM 2:

IN 2.js

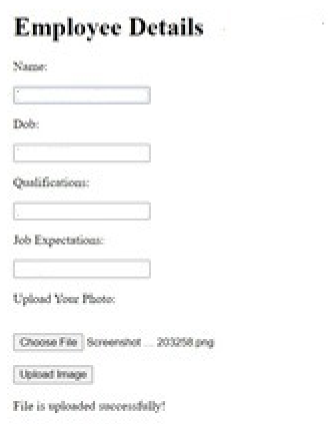
```
var express =   require("express");
var multer  =   require('multer');
var app =   express();
var storage =   multer.diskStorage({
  destination: function (req, file, callback) {
    callback(null, './uploads');
  },
  filename: function (req, file, callback) {
    callback(null, file.originalname);
  }
});
var upload = multer({ storage : storage}).single('myfile');

app.get('/',function(req,res){
      res.sendFile(_dirname + "/2.html");
});
```

```
app.post('/uploadjavatpoint',function(req,res){
    upload(req,res,function(err) {
        if(err) {
            return res.end("Error uploading file.");
        }
        res.end("File is uploaded successfully!");
    });
});

app.listen(2000,function(){
    console.log("Server is running on port 2000");
});


IN 2.html
<html>
  <head>
    <title>File upload in Node.js by Javatpoint</title>
 <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></sc
ript>
  <script
src="http://cdnjs.cloudflare.com/ajax/libs/jquery.form/3.51/jquery.form.mi
n.js"></script>
  <script>
  $(document).ready(function() {
     $('#uploadForm').submit(function() {
         $("#status").empty().text("File is uploading...");

        $(this).ajaxSubmit({

            error: function(xhr) {
                    status('Error: ' + xhr.status);
            },
            success: function(response) {
                    console.log(response)
                    $("#status").empty().text(response);
            }
     });

     return false;
     });
});
  </script>
  </head>
  <body>
      <h1>Employee Details i.e Resume</h1>
      <form id="uploadForm" enctype="multipart/form-data"
action="/uploadjavatpoint" method="post">
       <p>Name:</p> <input type="text" name="name"/> <br/>
       <p>Dob:</p> <input type="text" name="dob"/> <br/>
       <p>Qualifications:</p> <input type="text" name="quali"/> <br/>
       <p>Job Expectations:</p> <input type="text" name="job"/> <br/>
       <p>Upload Your Photo:</p> <br/>
       <input type="file" name="myfile" /><br/><br/>
       <input type="submit" value="Upload Image" name="submit"><br/><br/>
       <span id="status"></span>
      </form>
  </body>
</html>
```

TEST CASES

SCREENSHOT OF OUTPUT

PROGRAM 1:



Employee Details

Name:

Dob:

Qualifications:

Job Expectations:

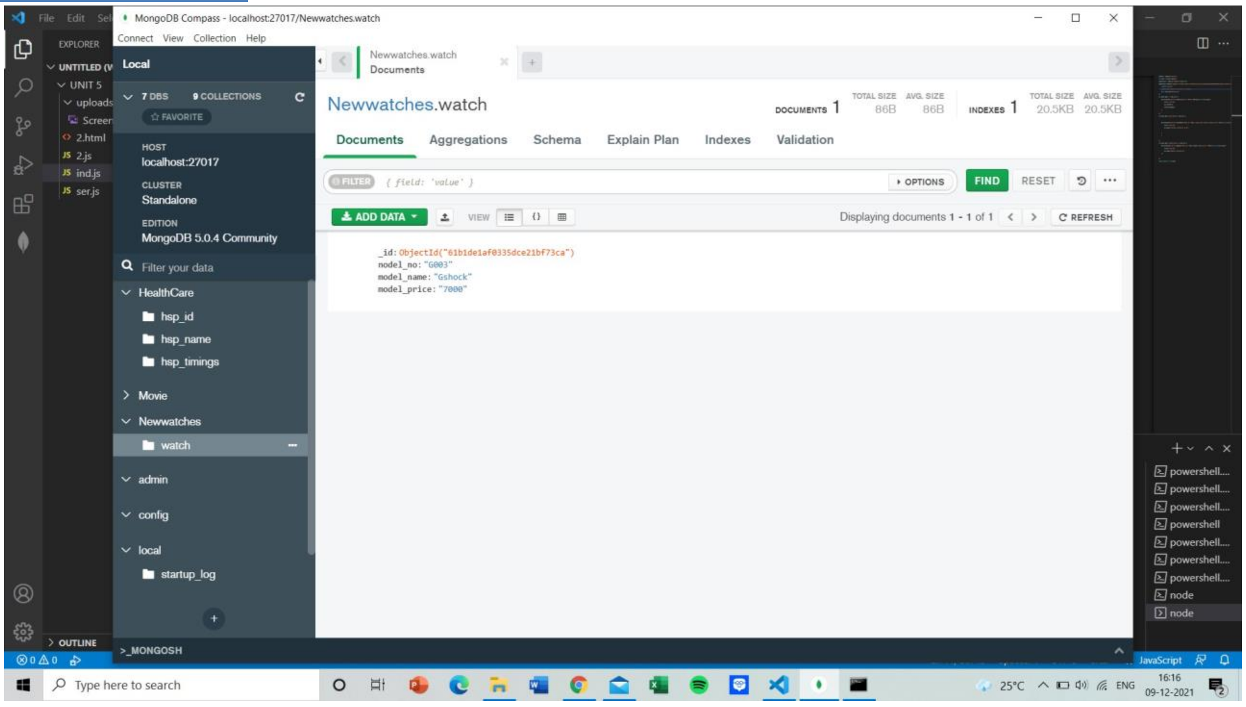
Upload Your Photo:

Choose File Screenshot ... 203258.png

Upload Image

File is uploaded successfully!

PROGRAM 2:



MongoDB Compass - localhost:27017/Newwatches.watch

Local

7 DBS 9 COLLECTIONS

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.4 Community

HealthCare: hsp_id, hsp_name, hsp_timings

Movie

Newwatches: watch

admin

config

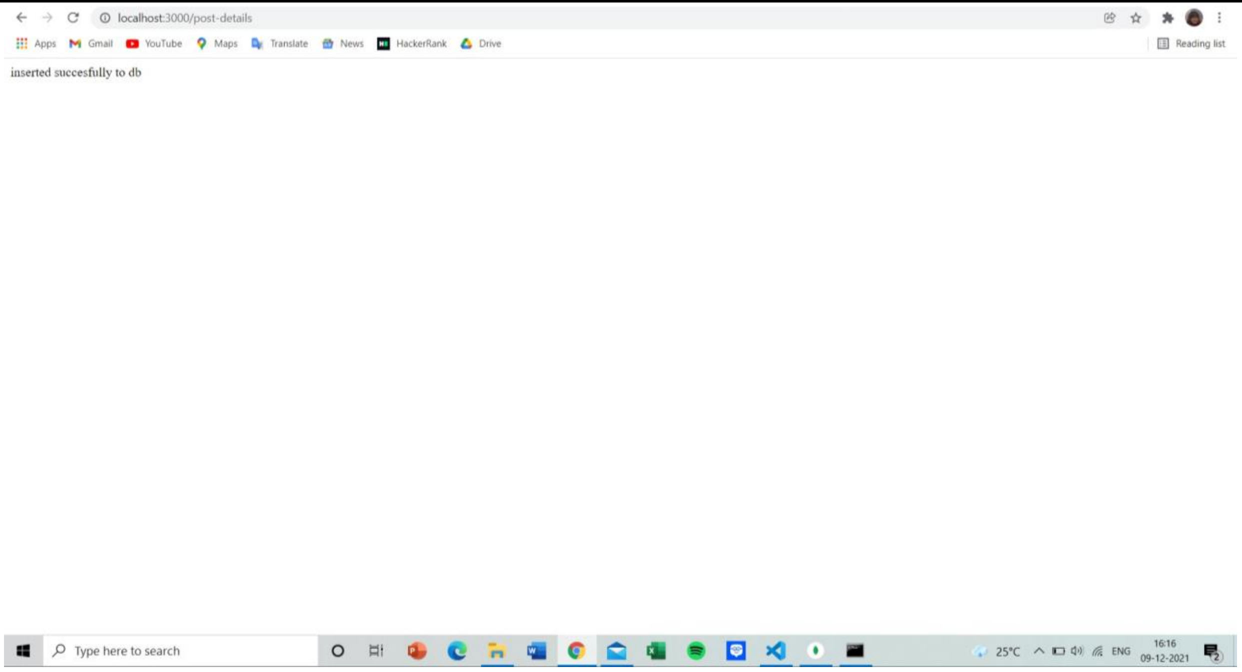
local: startup_log

Newwatches.watch

DOCUMENTS 1 TOTAL SIZE 86B AVG. SIZE 86B INDEXES 1 TOTAL SIZE 20.5KB AVG. SIZE 20.5KB

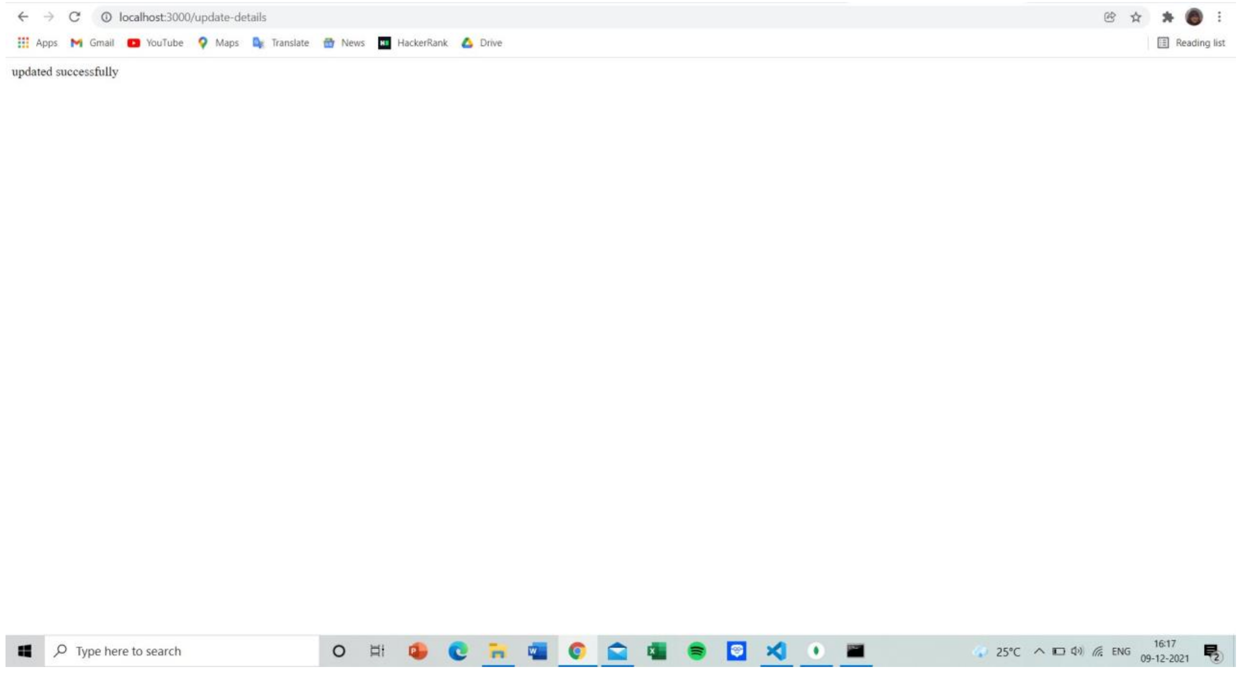
Displaying documents 1 - 1 of 1

```
_id: ObjectId("61b1de1af0335dce21bf73ca")
model_no: "G003"
model_name: "Gshock"
model_price: "7000"
```



localhost:3000/post-details

```
inserted succesfully to db
```



localhost:3000/update-details

```
updated successfully
```