|  |  |  |
| --- | --- | --- |
| **Name: Adithya M** | **SRN: PES1UG20CS621** | **Section: K** |
| **Date: 13/12/2021** | **Exercise No: 5** |

|  |
| --- |
| 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)   1. 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, 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. |
| PROGRAM |
| const express = require("express");  const MongoClient = require("mongodb").MongoClient;  const app = express();  const dburl = "mongodb://localhost:27017/";  app.use(express.json());  app.use(express.urlencoded({ extended: true }));  app.get("/book", (req, res) => {  MongoClient.connect(dburl, (err, db) => {  if (err) {  res.send(err);  }  let dbo = db.db("book");  dbo  .collection("books")  .find({})  .toArray((err, result) => {  if (err) throw err;  res.send(result);  db.close();  });  });  });  app.get("/book/:id", (req, res) => {  bookId = req.params.id;  MongoClient.connect(dburl, (err, db) => {  if (err) {  res.send(err);  }  let dbo = db.db("book");  dbo.collection("books").findOne({ book\_id: bookId }, (err, result) => {  if (err) throw err;  res.send(result);  db.close();  });  });  });  app.post("/book", (req, res) => {  data = {  book\_id: req.body.book\_id,  book\_name: req.body.book\_name,  book\_price: req.body.book\_price,  book\_auth: req.body.book\_auth,  };  try {  MongoClient.connect(dburl, (err, db) => {  if (err) {  res.send(err);  }  let dbo = db.db("book");  dbo.collection("books").insertOne(data, (err, result) => {  if (err) throw err;  res.send({ message: "Data added successfully", data: data });  db.close();  });  });  } catch (error) {  res.send(error);  }  });  app.put("/book/:id", (req, res) => {  bookId = req.params.id;  MongoClient.connect(dburl, (err, db) => {  if (err) {  res.send(err);  }  let query = { book\_id: bookId };  let newValues = { $set: req.body };  let dbo = db.db("book");  dbo.collection("books").updateOne(query, newValues, { upsert: true }, (err, result) => {  if (err) throw err;  console.log("Updated document successfully");  res.send(result);  db.close();  });  });  });  app.listen(8081, () => {  console.log("Server started");  });  const express = require("express");  const bodyParser = require("body-parser");  const fileUpload = require("express-fileupload");  const fs = require("fs");  const app = express();  app.use(bodyParser.urlencoded({ extended: true }));  app.use(fileUpload());  app.get("/", (req, res) => {  res.sendFile(\_\_dirname + "/index.html");  });  app.post("/", (req, res) => {  if (req.files) {  console.log(req.files);  }  let fileObject = req.files.image;  let fileName = fileObject.name;  let fileSize = fileObject.size;  fileObject.mv("./uploads/" + fileName, (err) => {  if (err) {  console.log("Error : " + err);  } else {  res.send("FileName : " + fileName + " Uploaded Successfully to ./uploads Directory\n");  }  });  });  app.listen(8081, () => {  console.log("Server started");  }); |
| TEST CASES |
| 1. GET request      1. GET request to /book/yes      1. POST request      1. PUT request to /book/abcd |
| SCREENSHOT OF OUTPUT |
|  |