|  |  |  |
| --- | --- | --- |
| **Name: SUNDEEP A** | **SRN: PES1UG20CS445** | **Section: H** |
| **Date: 13-12-21** | **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 |
| 1)  IN book\_collection.html  <!DOCTYPE html>  <head>      <title>Details of a Book</title>  </head>  <body>      <br/><br/><br/><br/>      <div align="center">          <h1>Enter the Book Info</h1>          <form action="/login" method="POST">              <label >Enter Book ID: </label>              <input type="text" name="book\_no"              placeholder="book\_id">              <br/><br/><br/><br/>              <label >Enter Book Name : </label>              <input type="text" name="book\_name" placeholder="book\_name">              <br/><br/><br/>              <label >Enter Book Price : </label>              <input type="text" name="book\_price" placeholder="book\_price">              <br/><br/><br/><br/>              <label >Enter the Author's Name: </label>              <input type="text" name="book\_auth" placeholder="book\_author">              <br/><br/><br/><br/>              <input type="submit">          </form>          <br/><br/><br/><br/>          <a href="/update"><strong>Click Me, if you want to update the book price</strong></a>      </div>  </body>  In Update\_books.html  <!DOCTYPE html>  <head>      <title>Details of a Book</title>  </head>  <body>      <br/><br/><br/><br/>      <div align="center">          <h1>Update the Book Price</h1>          <form action="/login/update" method="POST">              <label >Enter Book Number : </label>              <input type="text" name="book\_no" placeholder="book\_id">              <br/><br/><br/><br/>              <label >Enter Book Price : </label>              <input type="text" name="book\_price" placeholder="book\_price">              <br/><br/><br/><br/>              <input type="submit" value="Update">          </form>          <br/><br/><br/><br/>      </div>  </body>  In book\_backend.js  const express = require("express");  const app = express();  const path = require("path");  const { MongoClient } = require("mongodb");  app.use(express.json());  app.use(express.urlencoded({ extended: true }));  const uri = "mongodb://localhost:27017/book";  app.get("/", (req, res) => {      res.sendFile(path.join(\_\_dirname, "book\_collection.html"));  });  app.post("/login", (req, res) => {      const query1 = req.body;      const main\_1 = async () => {          MongoClient.connect(uri, async (err, client) => {              if (err) throw err;              const check = await client              .db("book\_collection")              .collection("details")              .findOne(query1);              if (check == null) {                  await                  client.db("book\_collection").collection("details").insertOne(query1);                  client.close();                  res.send(`                  <h1>Details of book inserted into the database : </h1>                  <h2> Book number : ${req.body.book\_no}</h2>                  <h2> Book Name : ${req.body.book\_name}</h2>                  <h2> Book Price : ${req.body.book\_price}</h2>                  <h2> Book author : ${req.body.book\_auth}</h2>                  `);              }              else {                  res.send(`<h1> Book Data is already stored in the database</h1>`);              }          });      };      main\_1();  });  app.post("/login/update", (req, res) => {      const query2 = {book\_no: req.body.book\_no};      const update = {book\_price: req.body.book\_price,};      const main2 = async () => {          MongoClient.connect(uri, async (err, client) => {              if (err) throw err;              const check = await              client.db("book\_collection").collection("details").findOne(query2);              if (check == null) {                  res.send(`<h1>Data is not present in the database</h1>`);              }              else {                  await                  client.db("book\_collection").collection("details").updateOne(query2, {$set: update,});                  client.close();                  res.send(`                  <h1>Details of book updated </h1>                  <h2> Book number : ${req.body.book\_no}</h2>                  <h2> Book Price : ${req.body.book\_price}</h2>                  `);              }          });      };      main2();  });  app.get("/update", (req, res) => {      res.sendFile(path.join(\_\_dirname, "Update\_books.html"));  });  app.listen(3001, () => {      console.log("Listening to port 3001");  });  2)  In Student.html  <!DOCTYPE html>  <head>      <title>Student details</title>  </head>  <body>        <div align="center">          <h1><u>Enter the Student details</u></h1>          <form action="/login" method="POST">              <label><h2>Enter student name </h2> </label>              <input type="text" name="name" class="inp" placeholder="Enter your name ">              <br /><br />              <label><h2>Enter date of birth </h2> </label>              <input type="datetime" name="dob" class="inp" placeholder="Enter your date of birth">              <br /><br />              <label><h2>Enter the Qualification </h2> </label>              <input type="text" name="qual" class="inp" placeholder="Enter Qualification">              <br /><br />              <label><h2>Enter the nationality </h2> </label>              <input type="text" name="nat" class="inp" placeholder="Enter the nationality">              <br /><br />              <label><h2>Student Photo </h2> </label>              <input type="file" id="image" accept="image/\*" /><br/><br />              <img id="profile" src="#" alt="Student photo" height="150px" width="150px"/>              <br /><br/><br />              <input type="submit" value="Upload" />            </form>      </div>      <script>          image.onchange = evt => {              const [file] = image.files              if (file) {                  profile.src = URL.createObjectURL(file)              }          }      </script>  </body>  In Std.js  const express = require("express");  const app = express();  const path = require("path");  const {MongoClient} = require("mongodb");  app.use(express.json());  app.use(express.urlencoded({extended:true}));  const uri = "mongodb://localhost:27017/student"  app.get("/",(req,res)=>{      res.sendFile(path.join(\_\_dirname,"Student.html"));  })  app.post("/login",(req,res)=>{      const query1 = req.body;      const main\_1 = async () => {          MongoClient.connect(uri, async (err, client) => {              if (err) throw err;              const check = await              client.db("student").collection("resume").findOne(query1);              if (check == null) {                  await                  client.db("student").collection("resume").insertOne(query1);                  client.close();                  res.send(`                  <h1>Details of the student </h1>                  <h2> student name : ${req.body.name} </h2>                  <h2> student date of birth : ${req.body.dob}</h2>                  <h2> student qualification : ${req.body.qual} </h2>                  <h2> student nationality : ${req.body.nat}</h2>                  `);              }              else {                  res.send(`<h1>Data is already present in the database</h1>`);              }          });      };      main\_1();  })  app.listen(4001,()=>{   console.log("Listening to port 4001");  }) |
| SCREENSHOT OF OUTPUT |
| After uploading into Database    If Same data already exists    Updating the Price    After Updating the price    2) |
|  |