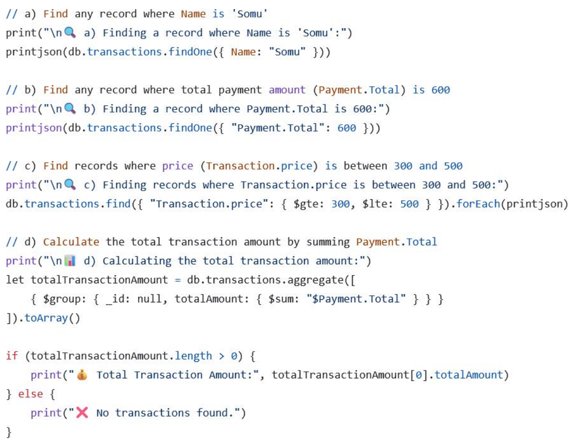
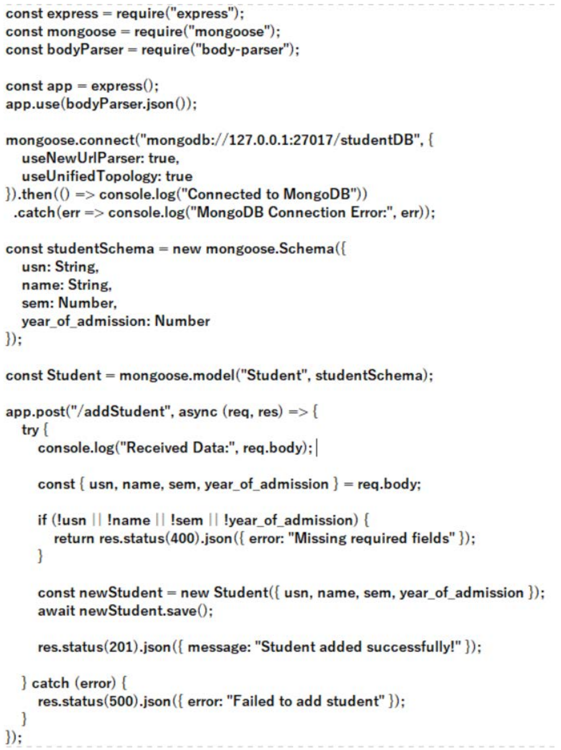
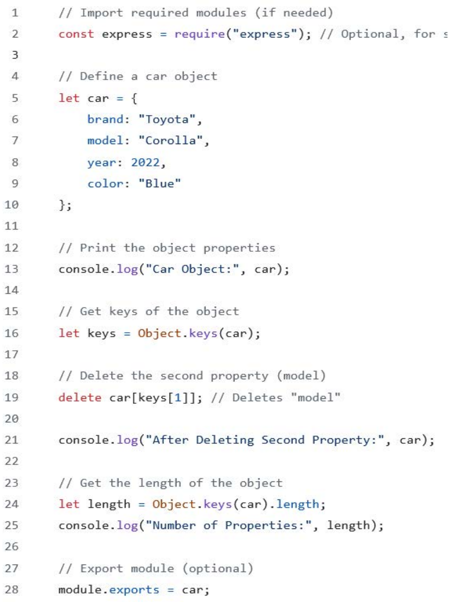


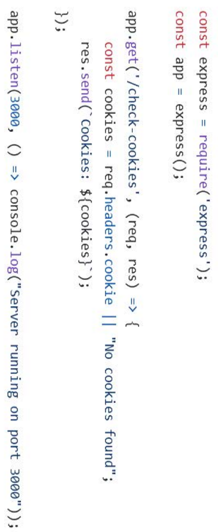
```
// a) Find any record where Name is 'Somu'
print("\n🔍 a) Finding a record where Name is 'Somu':")
printjson(db.transactions.findOne({ Name: "Somu" }))

// b) Find any record where total payment amount (Payment.Total) is 600
print("\n🔍 b) Finding a record where Payment.Total is 600:")
printjson(db.transactions.findOne({ "Payment.Total": 600 }))

// c) Find records where price (Transaction.price) is between 300 and 500
print("\n🔍 c) Finding records where Transaction.price is between 300 and 500:")
db.transactions.find({ "Transaction.price": { $gte: 300, $lte: 500 } }).forEach(printjson)

// d) Calculate the total transaction amount by summing Payment.Total
print("\n📊 d) Calculating the total transaction amount:")
let totalTransactionAmount = db.transactions.aggregate([
    { $group: { _id: null, totalAmount: { $sum: "$Payment.Total" } } }
]).toArray()

if (totalTransactionAmount.length > 0) {
    print("💰 Total Transaction Amount:", totalTransactionAmount[0].totalAmount)
} else {
    print("❌ No transactions found.")
}
```



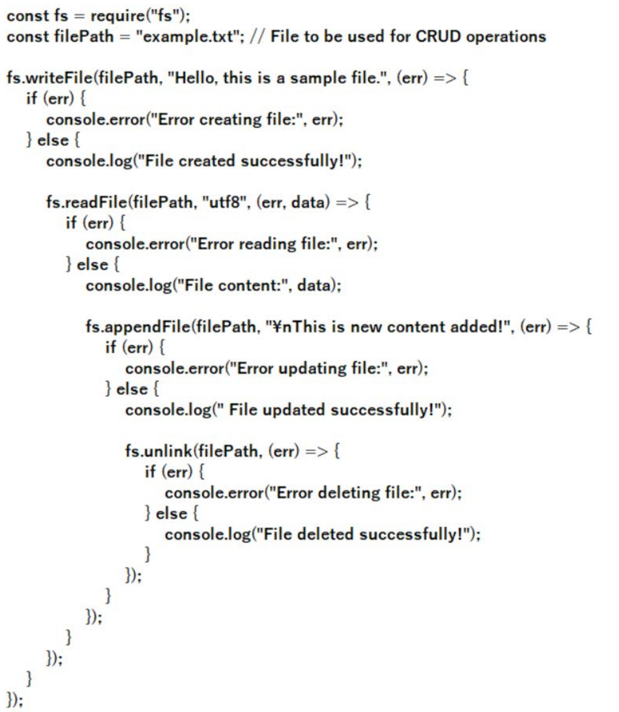
```
const express = require("express");
const mongoose = require("mongoose");
const bodyParser = require("body-parser");

const app = express();
app.use(bodyParser.json());

mongoose.connect("mongodb://127.0.0.1:27017/studentDB", {
  useNewUrlParser: true,
  useUnifiedTopology: true
}).then(() => console.log("Connected to MongoDB"))
 .catch(err => console.log("MongoDB Connection Error:", err));

const studentSchema = new mongoose.Schema({
  usn: String,
  name: String,
  sem: Number,
  year_of_admission: Number
});

const Student = mongoose.model("Student", studentSchema);

app.post("/addStudent", async (req, res) => {
  try {
    console.log("Received Data:", req.body);

    const { usn, name, sem, year_of_admission } = req.body;

    if (!usn || !name || !sem || !year_of_admission) {
      return res.status(400).json({ error: "Missing required fields" });
    }

    const newStudent = new Student({ usn, name, sem, year_of_admission });
    await newStudent.save();

    res.status(201).json({ message: "Student added successfully!" });

  } catch (error) {
    res.status(500).json({ error: "Failed to add student" });
  }
});
```



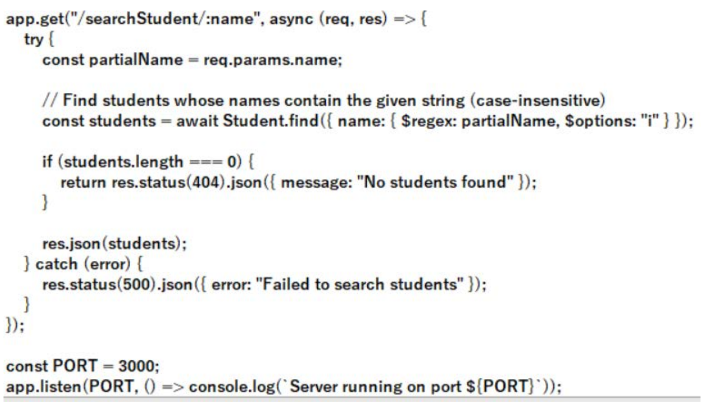
```
// Import required modules (if needed)
const express = require("express"); // Optional, for s

// Define a car object
let car = {
    brand: "Toyota",
    model: "Corolla",
    year: 2022,
    color: "Blue"
};

// Print the object properties
console.log("Car Object:", car);

// Get keys of the object
let keys = Object.keys(car);

// Delete the second property (model)
delete car[keys[1]]; // Deletes "model"

console.log("After Deleting Second Property:", car);

// Get the length of the object
let length = Object.keys(car).length;
console.log("Number of Properties:", length);

// Export module (optional)
module.exports = car;
```



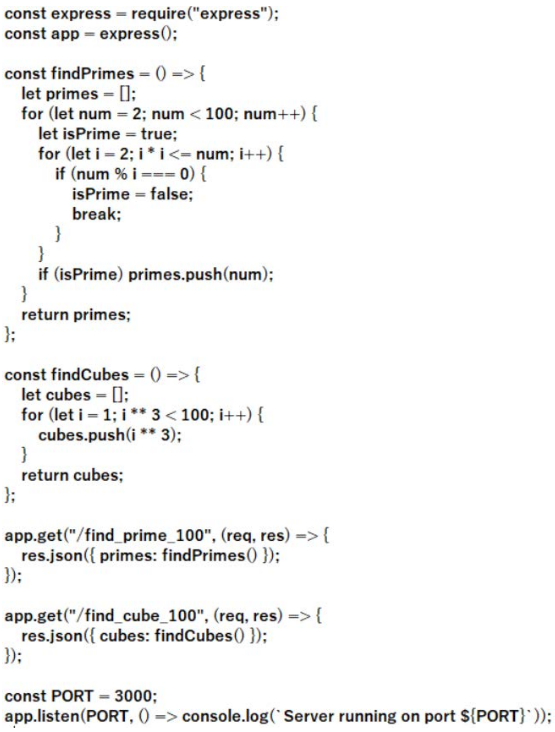
```
const express = require('express');
const app = express();

app.get('/check-cookies', (req, res) => {
    const cookies = req.headers.cookie || "No cookies found";
    res.send(`Cookies: ${cookies}`);
});

app.listen(3000, () => console.log("Server running on port 3000"));
```



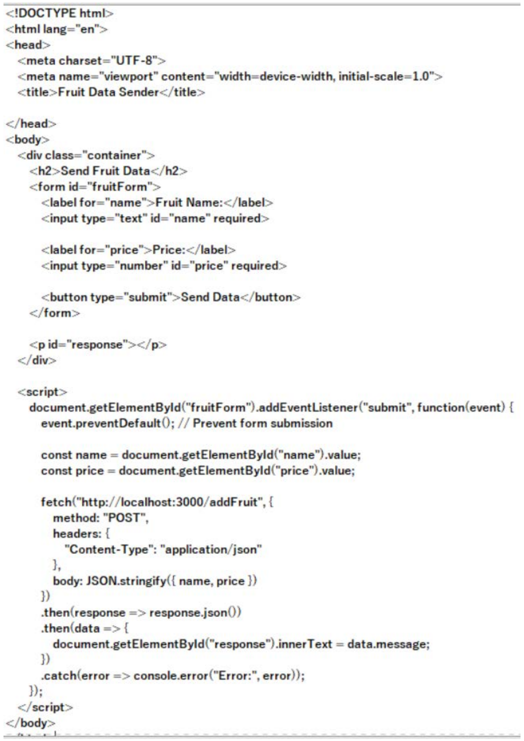
```
const fs = require("fs");
const filePath = "example.txt"; // File to be used for CRUD operations

fs.writeFile(filePath, "Hello, this is a sample file.", (err) => {
  if (err) {
    console.error("Error creating file:", err);
  } else {
    console.log("File created successfully!");

    fs.readFile(filePath, "utf8", (err, data) => {
      if (err) {
        console.error("Error reading file:", err);
      } else {
        console.log("File content:", data);

        fs.appendFile(filePath, "¥nThis is new content added!", (err) => {
          if (err) {
            console.error("Error updating file:", err);
          } else {
            console.log(" File updated successfully!");

            fs.unlink(filePath, (err) => {
              if (err) {
                console.error("Error deleting file:", err);
              } else {
                console.log("File deleted successfully!");
              }
            });
          }
        });
      }
    });
  }
});
```



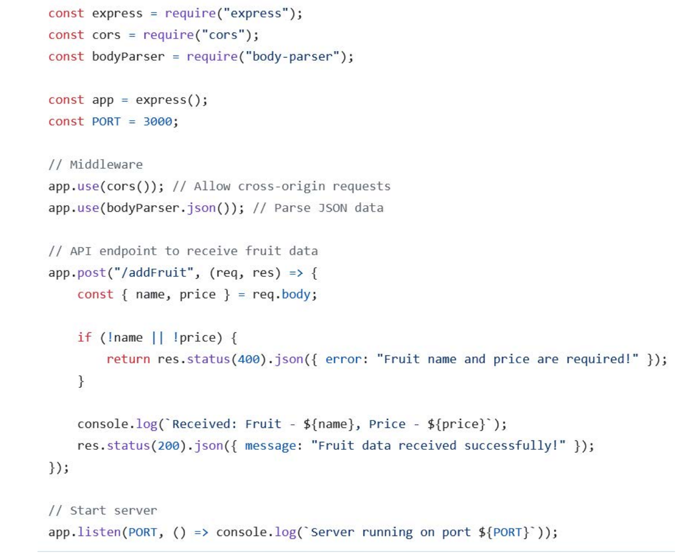
```
app.get("/searchStudent/:name", async (req, res) => {
  try {
    const partialName = req.params.name;

    // Find students whose names contain the given string (case-insensitive)
    const students = await Student.find({ name: { $regex: partialName, $options: "i" } });

    if (students.length === 0) {
      return res.status(404).json({ message: "No students found" });
    }

    res.json(students);
  } catch (error) {
    res.status(500).json({ error: "Failed to search students" });
  }
});

const PORT = 3000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```



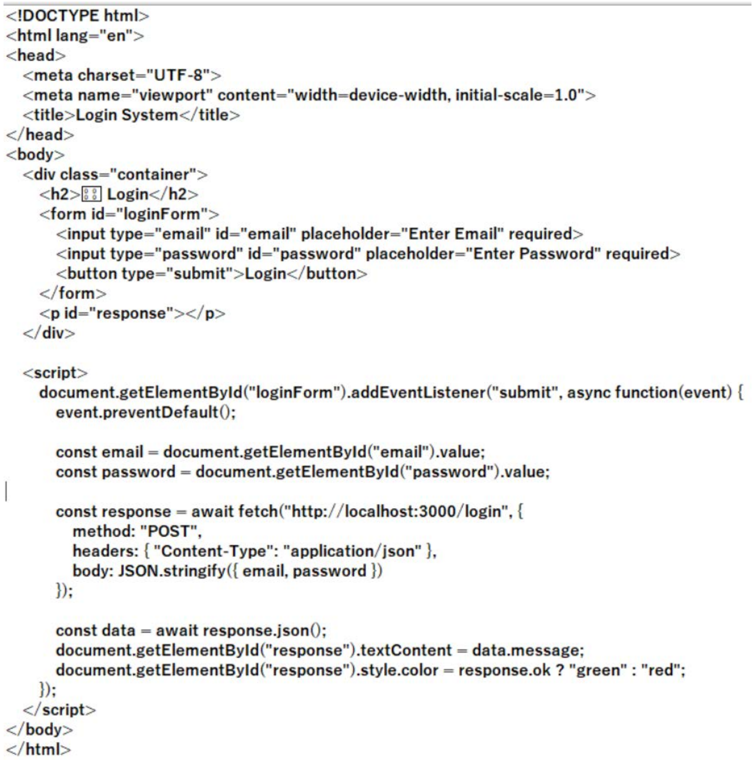
```
const express = require("express");
const app = express();

const findPrimes = () => {
  let primes = [];
  for (let num = 2; num < 100; num++) {
    let isPrime = true;
    for (let i = 2; i * i <= num; i++) {
      if (num % i === 0) {
        isPrime = false;
        break;
      }
    }
    if (isPrime) primes.push(num);
  }
  return primes;
};

const findCubes = () => {
  let cubes = [];
  for (let i = 1; i ** 3 < 100; i++) {
    cubes.push(i ** 3);
  }
  return cubes;
};

app.get("/find_prime_100", (req, res) => {
  res.json({ primes: findPrimes() });
});

app.get("/find_cube_100", (req, res) => {
  res.json({ cubes: findCubes() });
});

const PORT = 3000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fruit Data Sender</title>

</head>
<body>
  <div class="container">
    <h2>Send Fruit Data</h2>
    <form id="fruitForm">
      <label for="name">Fruit Name:</label>
      <input type="text" id="name" required>

      <label for="price">Price:</label>
      <input type="number" id="price" required>

      <button type="submit">Send Data</button>
    </form>

    <p id="response"></p>
  </div>

  <script>
    document.getElementById("fruitForm").addEventListener("submit", function(event) {
      event.preventDefault(); // Prevent form submission

      const name = document.getElementById("name").value;
      const price = document.getElementById("price").value;

      fetch("http://localhost:3000/addFruit", {
        method: "POST",
        headers: {
          "Content-Type": "application/json"
        },
        body: JSON.stringify({ name, price })
      })
      .then(response => response.json())
      .then(data => {
        document.getElementById("response").innerText = data.message;
      })
      .catch(error => console.error("Error:", error));
    });
  </script>
</body>
```



```
const express = require("express");
const cors = require("cors");
const bodyParser = require("body-parser");

const app = express();
const PORT = 3000;

// Middleware
app.use(cors()); // Allow cross-origin requests
app.use(bodyParser.json()); // Parse JSON data

// API endpoint to receive fruit data
app.post("/addFruit", (req, res) => {
    const { name, price } = req.body;

    if (!name || !price) {
        return res.status(400).json({ error: "Fruit name and price are required!" });
    }

    console.log(`Received: Fruit - ${name}, Price - ${price}`);
    res.status(200).json({ message: "Fruit data received successfully!" });
});

// Start server
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login System</title>
</head>
<body>
  <div class="container">
    <h2>□ Login</h2>
    <form id="loginForm">
      <input type="email" id="email" placeholder="Enter Email" required>
      <input type="password" id="password" placeholder="Enter Password" required>
      <button type="submit">Login</button>
    </form>
    <p id="response"></p>
  </div>

  <script>
    document.getElementById("loginForm").addEventListener("submit", async function(event) {
      event.preventDefault();

      const email = document.getElementById("email").value;
      const password = document.getElementById("password").value;

      const response = await fetch("http://localhost:3000/login", {
        method: "POST",
        headers: { "Content-Type": "application/json" },
        body: JSON.stringify({ email, password })
      });

      const data = await response.json();
      document.getElementById("response").textContent = data.message;
      document.getElementById("response").style.color = response.ok ? "green" : "red";
    });
  </script>
</body>
</html>
```