



---

# CAR MANGEMENT

---

CSD 3103 Full Stack JavaScript



SUBMITTED BY:

<b>Dhariya Vinod Vayas:</b>	<b>C0840249</b>
<b>Hemani Patel:</b>	<b>C0853622</b>
<b>Anik Hasan:</b>	<b>C0847377</b>
<b>Md Kamrul Islam Antar:</b>	<b>C0826256</b>
<b>Ishank Agarwal:</b>	<b>C0850072</b>

## Details about car website:

### I. Introduction

- The purpose of the website is to provide a comprehensive list of cars for car enthusiasts to browse and search through, as well as a platform for users to add their own car listings, reviews, and participate in community forums.
- The project car list website is designed to be user-friendly and accessible to all car enthusiasts, regardless of their level of expertise.

### II. Website Features

- The website features a search function that allows users to search for cars by make, model, year, and other specifications.
- Sorting and filtering options are available, allowing users to refine their search results based on criteria such as price, mileage, and location.
- Car profiles include photos, specifications, and reviews, as well as user ratings and comments.
- The community forum and discussion boards allow users to interact with each other, share information, and ask questions.

### III. User Interface Design

- The website design is clean and modern, with a minimalist color scheme and intuitive navigation.
- The homepage features a prominent search bar, as well as a featured car of the week and links to popular car categories.
- Car profiles are displayed in a visually appealing format, with photos and specifications prominently displayed.

Create Car

Id:

Make:

Model:

Year:

Color:

Kms:

Vin:

Price:

Images:

Choose Files

No file chosen

Submit

Cancel

Car Management System

Final Team Project

Submission By: Dhariya Vinod Vayas, Hemani Patel, Anik Hasan, Md Kamrul Islam Antar, Ishank Agarwal

Car List

Add a new Car

Sort By Filter

Total Cars: 20

Audi TT

STARTING AT

\$7304

Fuel Efficiency

L/100KM\*

9.9/7.9

(city/hwy)

Model Year

2010

Mileage

8451\* Kms

VW Tiguan

STARTING AT

\$9610

Fuel Efficiency

L/100KM\*

8.4/9.8

(city/hwy)

Model Year

2010

Mileage

9010\* Kms

Skoda Fabia

STARTING AT

\$8118

Fuel Efficiency

L/100KM\*

7.7/9.8

(city/hwy)

Model Year

2010

Mileage

7542\* Kms

Toyota Avenis

STARTING AT

\$9526

Fuel Efficiency

L/100KM\*

9.1/9.4

(city/hwy)

Model Year

2008

Mileage

7779\* Kms

Car Management System

Final Team Project

Submission By: Dhariya Vinod Vayas, Hemani Patel, Anik Hasan, Md Kamrul Islam Antar, Ishank Agarwal

Car List

Add a new Car

Price low to high

Total Cars: 20

Kia Forte

STARTING AT

\$7045

Fuel Efficiency

L/100KM\*

9.3/8.3

(city/hwy)

Model Year

2007

Mileage

8306\* Kms

Audi TT

STARTING AT

\$7304

Fuel Efficiency

L/100KM\*

7.9/8.8

(city/hwy)

Model Year

2010

Mileage

8451\* Kms

VW Fox

STARTING AT

\$7603

Fuel Efficiency

L/100KM\*

8.0/8.1

(city/hwy)

Model Year

2011

Mileage

7650\* Kms

Jeep Renegade

STARTING AT

\$7728

Fuel Efficiency

L/100KM\*

7.5/9.3

(city/hwy)

Model Year

2004

Mileage

8177\* Kms

©2022 LAMBTON COLLEGE IN TORONTO

# Full Stack JavaScript CSD 3103

## Cars List

Add a new Car Price high to low | Total Cars: 20



**Audi R8**

STARTING AT **\$9655**

Fuel Efficiency  
L/100km\* **8.3/9.9**  
(city/hwy)

Model Year **2007** Mileage **7685\* Kms**



**VW Tiguan**

STARTING AT **\$9610**

Fuel Efficiency  
L/100km\* **8.6/8.3**  
(city/hwy)

Model Year **2010** Mileage **9016\* Kms**



**Toyota Avensis**

STARTING AT **\$9526**

Fuel Efficiency  
L/100km\* **9.7/7.7**  
(city/hwy)

Model Year **2008** Mileage **7779\* Kms**



**Chevrolet Silverado**

STARTING AT **\$9467**

Fuel Efficiency  
L/100km\* **9.2/8.3**  
(city/hwy)

Model Year **2005** Mileage **7354\* Kms**

Car Management System

Car List

Final Team Project

Submission By: Dhariya Vinod Vayas, Hemani Patel, Anik Hasan, Md Kamrul Islam Antar, Ishank Agarwal

## Cars List

Add a new Car Kms low to high | Total Cars: 20



**Mini Countryman**

STARTING AT **\$8887**

Fuel Efficiency  
L/100km\* **9.3/8.0**  
(city/hwy)

Model Year **2006** Mileage **7310\* Kms**



**Chevrolet Silverado**

STARTING AT **\$9467**

Fuel Efficiency  
L/100km\* **9.8/7.8**  
(city/hwy)

Model Year **2005** Mileage **7354\* Kms**



**Skoda Fabia**

STARTING AT **\$8118**

Fuel Efficiency  
L/100km\* **8.2/8.7**  
(city/hwy)

Model Year **2010** Mileage **7542\* Kms**



**VW Fox**

STARTING AT **\$7603**

Fuel Efficiency  
L/100km\* **8.3/9.2**  
(city/hwy)

Model Year **2011** Mileage **7650\* Kms**

Car Management System

Car List

Final Team Project

Submission By: Dhariya Vinod Vayas, Hemani Patel, Anik Hasan, Md Kamrul Islam Antar, Ishank Agarwal

## Cars List

Add a new Car Kms high to low | Total Cars: 20



**Jeep Wrangler**

STARTING AT **\$8601**

Fuel Efficiency  
L/100km\* **7.9/9.0**  
(city/hwy)

Model Year **2003** Mileage **9724\* Kms**



**Peugeot 208**

STARTING AT **\$7841**

Fuel Efficiency  
L/100km\* **8.6/9.0**  
(city/hwy)

Model Year **2007** Mileage **9652\* Kms**



**Fiat Panda**

STARTING AT **\$8547**

Fuel Efficiency  
L/100km\* **8.1/7.5**  
(city/hwy)

Model Year **2012** Mileage **9242\* Kms**



**VW Tiguan**

STARTING AT **\$9610**

Fuel Efficiency  
L/100km\* **7.8/8.5**  
(city/hwy)

Model Year **2010** Mileage **9016\* Kms**

Car Management System

Car List

Final Team Project


Submission By: Dhariya Vinod Vayas, Hemani Patel, Anik Hasan, Md Kamrul Islam Antar, Ishank Agarwal

Cars List

Add a new Car

Price high to low

Total Cars: 20




Audi R8

STARTING AT \$9655

Fuel Efficiency L/100km\* 9.6/8.3 (city/hwy)

Model Year 2007 Mileage 7685\* Kms




VW Tiguan

STARTING AT \$9610

Fuel Efficiency L/100km\* 8.9/9.6 (city/hwy)

Model Year 2010 Mileage 9016\* Kms




Toyota Avenis

STARTING AT \$9526

Fuel Efficiency L/100km\* 8.6/7.6 (city/hwy)

Model Year 2008 Mileage 7779\* Kms



Chevrolet Silverado

STARTING AT \$9467

Fuel Efficiency L/100km\* 8.1/9.4 (city/hwy)

Model Year 2005 Mileage 7354\* Kms


Car Management System

Car List

Final Team Project

Submission By: Dhariya Vinod Vayas, Hemani Patel, Anik Hasan, Md Kamrul Islam Antar, Ishank Agarwal

Car for Sale



Car for Sale

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce aliquet nunc elit, id blandit tellus volutpat id. Fusce euismod turpis a enim pretium fringilla. Nam dapibus, quam non aliquam vestibulum, justo arcu feugiat lorem, a auctor mauris lacus vel mi.

- Id: 00113
- Make: Audi
- Model: R8
- Year: 2007
- Color: Black
- Kms: 7685
- Vin: 1258609434
- Price: 9655

Car Management System

Car List

Final Team Project

Submission By: Dhariya Vinod Vayas, Hemani Patel, Anik Hasan, Md Kamrul Islam Antar, Ishank Agarwal

## IV. Technical Details

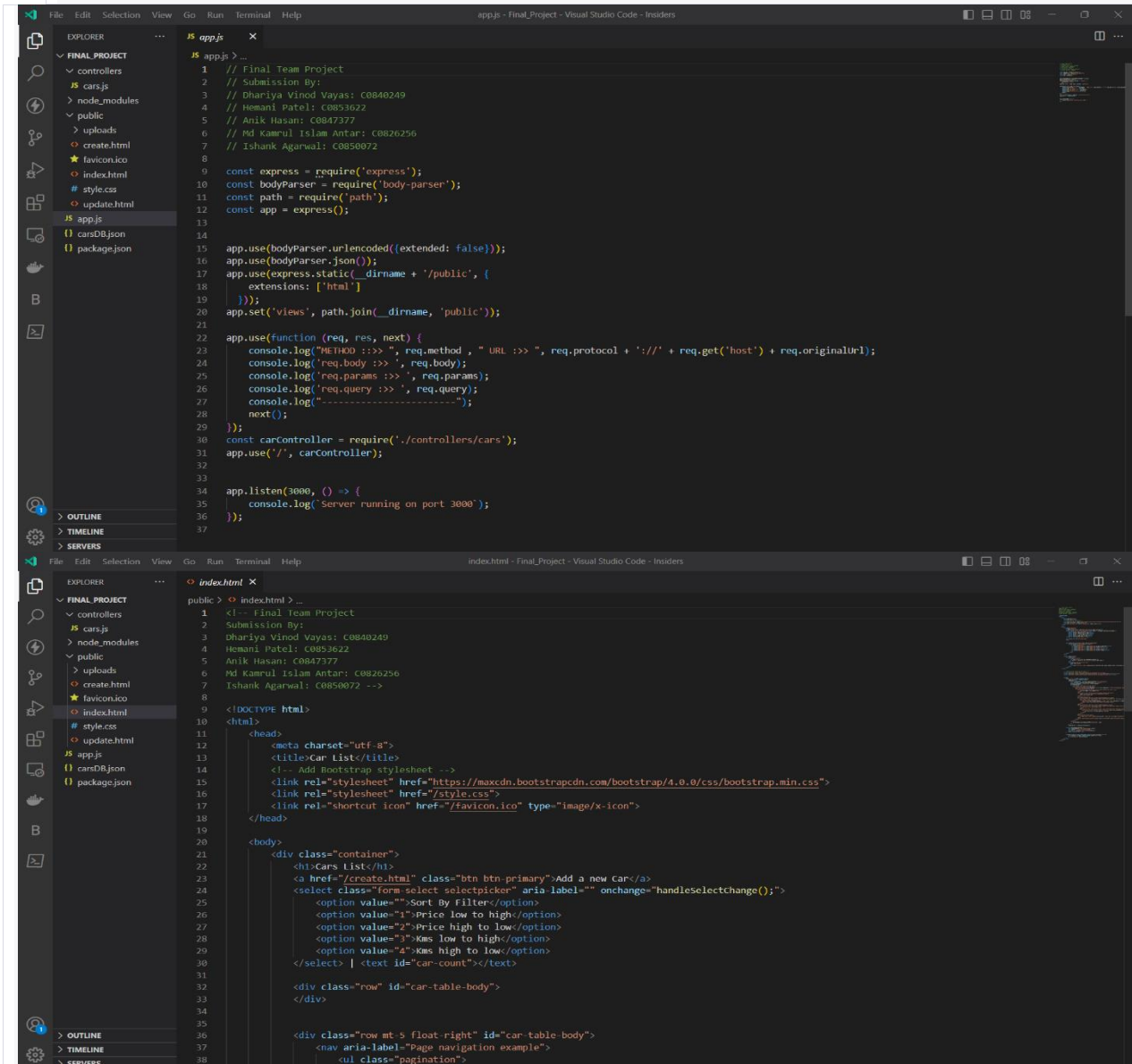
- The website is built using HTML, CSS, JavaScript, and with a JSON database for storing car listings and user information.
- The website uses the Specific information for each car model to display the specification and features.
- The website also uses the Disqus commenting system for community forums and discussion boards.

©2022 LAMBTON COLLEGE IN TORONTO



# Full Stack JavaScript

## CSD 3103

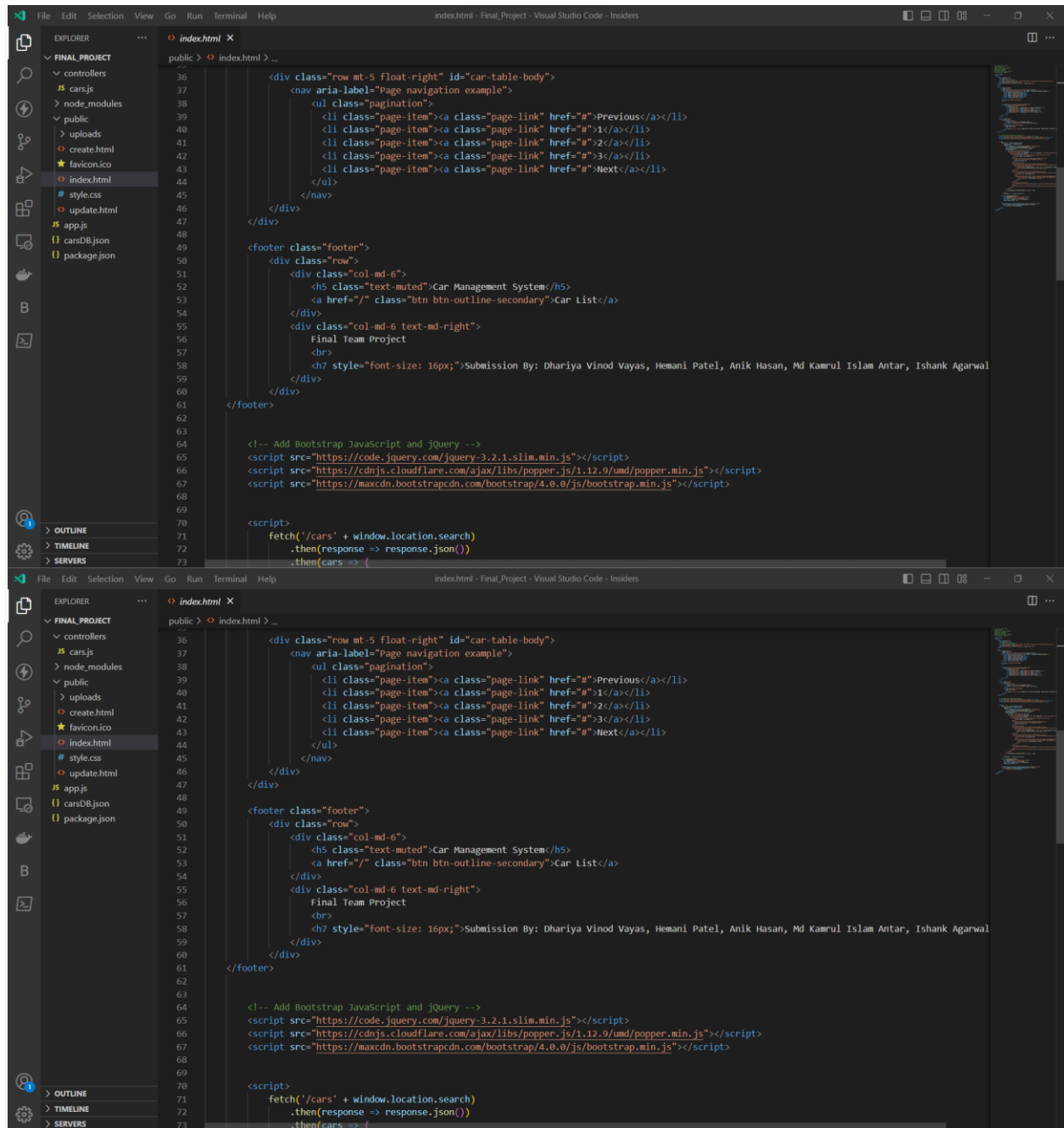


The screenshot displays two files in a Visual Studio Code editor. The top file, `app.js`, is a Node.js application using Express.js. It sets up a server on port 3000, uses body-parser for JSON and URL-encoded data, and serves static files from the `public` directory. A `carController` is imported from `./controllers/cars.js`. The bottom file, `index.html`, is an HTML page that uses Bootstrap 4.0.0 CSS and includes a form for adding a new car. The form has a text input for the car name and a select dropdown for filtering by price or kms. The page also includes a table body for displaying car listings and a pagination example.

```
1 // Final Team Project
2 // Submission By:
3 // Dhariya Vinod Vayya: C0840249
4 // Hemani Patel: C0836022
5 // Anik Hasan: C0847377
6 // Md Kamrul Islam Antar: C0826256
7 // Ishank Agarwal: C0850072
8
9 const express = require('express');
10 const bodyParser = require('body-parser');
11 const path = require('path');
12 const app = express();
13
14
15 app.use(bodyParser.urlencoded({extended: false}));
16 app.use(bodyParser.json());
17 app.use(express.static(__dirname + '/public', {
18   extensions: ['html']
19 }));
20 app.set('views', path.join(__dirname, 'public'));
21
22 app.use(function (req, res, next) {
23   console.log('Method: ' + req.method + ' URL: ' + req.protocol + '://' + req.get('host') + req.originalUrl);
24   console.log('req.body: ' + req.body);
25   console.log('req.params: ' + req.params);
26   console.log('req.query: ' + req.query);
27   console.log('-----');
28   next();
29 });
30 const carController = require('./controllers/cars');
31 app.use('/', carController);
32
33
34 app.listen(3000, () => {
35   console.log('Server running on port 3000');
36 });
37
```

```
1 <!-- Final Team Project
2 Submission By:
3 Dhariya Vinod Vayya: C0840249
4 Hemani Patel: C0836022
5 Anik Hasan: C0847377
6 Md Kamrul Islam Antar: C0826256
7 Ishank Agarwal: C0850072 -->
8
9 <!DOCTYPE html>
10 <html>
11 <head>
12   <meta charset="utf-8">
13   <title>Car List</title>
14   <!-- Add Bootstrap stylesheet -->
15   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
16   <link rel="stylesheet" href="/style.css">
17   <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon">
18 </head>
19
20 <body>
21   <div class="container">
22     <h1>Cars List</h1>
23     <a href="/create.html" class="btn btn-primary">Add a new car</a>
24     <select class="form-select selectpicker" aria-label="" onchange="handleselectchange();">
25       <option value="">Sort By Filter</option>
26       <option value="1">Price low to high</option>
27       <option value="2">Price high to low</option>
28       <option value="3">Kms low to high</option>
29       <option value="4">Kms high to low</option>
30     </select> | <text id="car-count"></text>
31
32     <div class="row" id="car-table-body">
33     </div>
34
35     <div class="row mt-5 float-right id="car-table-body">
36       <nav aria-label="Page navigation example">
37         <ul class="pagination">
38
```

## Full Stack JavaScript CSD 3103



```
public > index.html > ...
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73

<div class="row mt-5 float-right" id="car-table-body">
  <nav aria-label="Page navigation example">
    <ul class="pagination">
      <li class="page-item"><a class="page-link" href="#">Previous</a></li>
      <li class="page-item"><a class="page-link" href="#">1</a></li>
      <li class="page-item"><a class="page-link" href="#">2</a></li>
      <li class="page-item"><a class="page-link" href="#">3</a></li>
      <li class="page-item"><a class="page-link" href="#">Next</a></li>
    </ul>
  </nav>
</div>
</div>

<footer class="footer">
  <div class="row">
    <div class="col-md-6">
      <h5 class="text-muted">Car Management System</h5>
      <a href="/" class="btn btn-outline-secondary">Car List</a>
    </div>
    <div class="col-md-6 text-md-right">
      Final Team Project
      <br>
      <h7 style="font-size: 16px;">Submission By: Dhariya Vinod Vayas, Hemani Patel, Anik Hasan, Md Kamrul Islam Antar, Ishank Agarwal
    </div>
  </div>
</div>
</div>

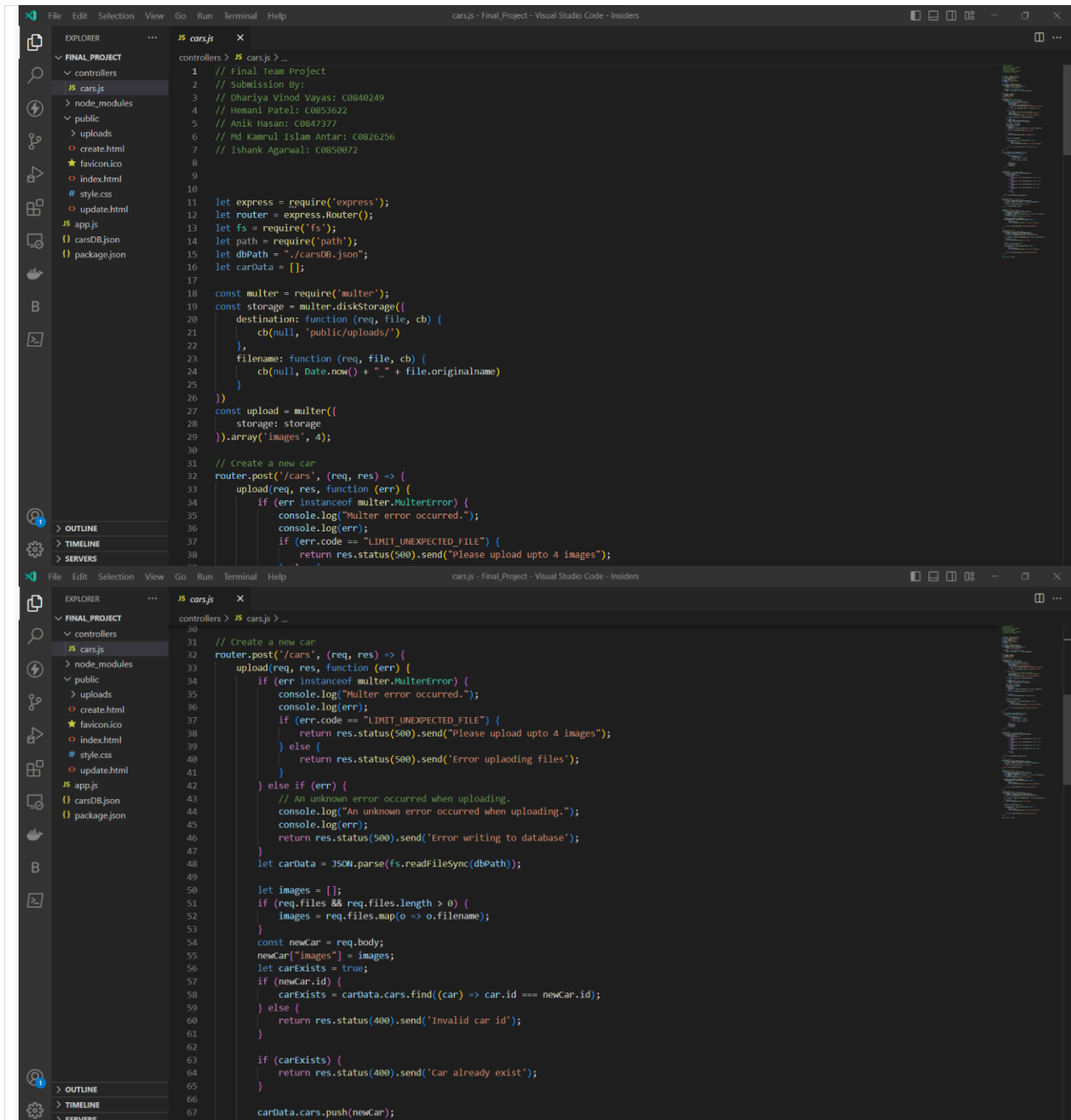
<!-- Add Bootstrap Javascript and jQuery -->
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<script>
  fetch('/cars' + window.location.search)
    .then(response => response.json())
    .then(cars => {
```

# CSD 3103



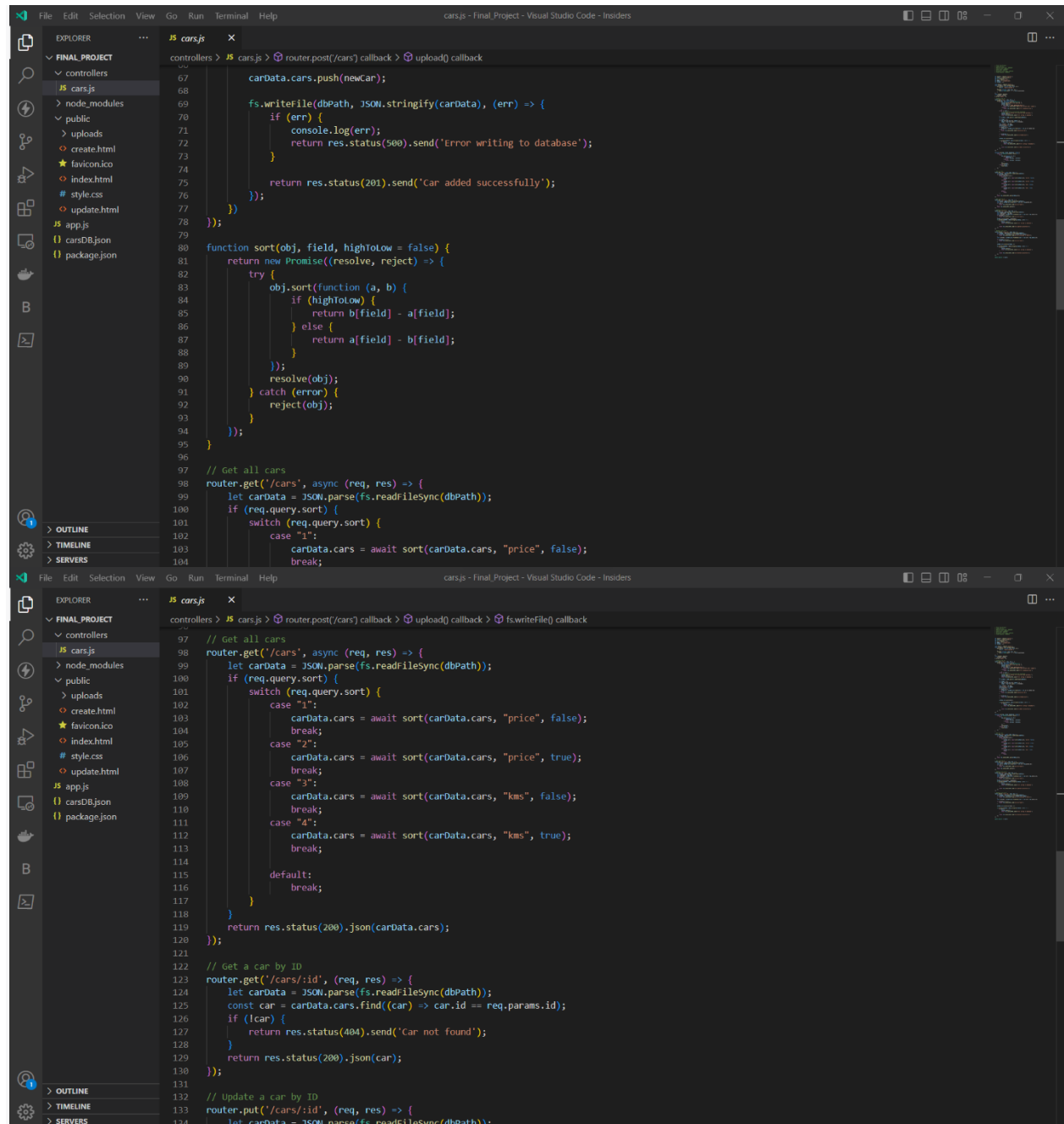
## Full Stack JavaScript CSD 3103



```
1 // final team Project
2 // Submission By:
3 // Dhariya Vinod Vayas: C0840249
4 // Hemani Patel: C0853622
5 // Anik Hasan: C0847377
6 // Md Kamrul Islam Antar: C0826256
7 // Ishank Agarwal: C0850072
8
9
10
11 let express = require('express');
12 let router = express.Router();
13 let fs = require('fs');
14 let path = require('path');
15 let dbPath = './carsDB.json';
16 let carData = [];
17
18 const multer = require('multer');
19 const storage = multer.diskStorage({
20   destination: function (req, file, cb) {
21     cb(null, 'public/uploads/');
22   },
23   filename: function (req, file, cb) {
24     cb(null, Date.now() + "_" + file.originalname);
25   }
26 });
27 const upload = multer({
28   storage: storage
29 }).array('images', 4);
30
31 // Create a new car
32 router.post('/cars', (req, res) => {
33   upload(req, res, function (err) {
34     if (err instanceof multer.MulterError) {
35       console.log("Multer error occurred.");
36       console.log(err);
37       if (err.code === "LIMIT_UNEXPECTED_FILE") {
38         return res.status(500).send("Please upload upto 4 images");
39       }
40     }
41     // An unknown error occurred when uploading.
42     console.log("An unknown error occurred when uploading.");
43     console.log(err);
44     return res.status(500).send("Error uploading files");
45   }
46   // An unknown error occurred when uploading.
47   console.log("An unknown error occurred when uploading.");
48   console.log(err);
49   return res.status(500).send("Error writing to database");
50 }
51 let carData = JSON.parse(fs.readFileSync(dbPath));
52
53 let images = [];
54 if (req.files && req.files.length > 0) {
55   images = req.files.map(o => o.filename);
56 }
57 const newCar = req.body;
58 newCar["images"] = images;
59 let carExists = true;
60 if (newCar.id) {
61   carExists = carData.cars.find(car => car.id === newCar.id);
62 } else {
63   return res.status(400).send('Invalid car id');
64 }
65
66 if (carExists) {
67   return res.status(400).send('Car already exist');
68 }
69
70 carData.cars.push(newCar);
```

# Full Stack JavaScript

## CSD 3103



The image shows two screenshots of a Visual Studio Code editor window. The top screenshot displays the 'cars.js' file with the following code:

```
controllers > JS cars.js > router.post('/cars') callback > upload() callback
carData.cars.push(newCar);

fs.writeFile(dbPath, JSON.stringify(carData), (err) => {
  if (err) {
    console.log(err);
    return res.status(500).send('Error writing to database');
  }
  return res.status(201).send('Car added successfully');
});
});

function sort(obj, field, highToLow = false) {
  return new Promise((resolve, reject) => {
    try {
      obj.sort(function (a, b) {
        if (highToLow) {
          return b[field] - a[field];
        } else {
          return a[field] - b[field];
        }
      });
      resolve(obj);
    } catch (error) {
      reject(error);
    }
  });
}

// Get all cars
router.get('/cars', async (req, res) => {
  let carData = JSON.parse(fs.readFileSync(dbPath));
  if (req.query.sort) {
    switch (req.query.sort) {
      case "1":
        carData.cars = await sort(carData.cars, "price", false);
        break;
    }
  }
});
```

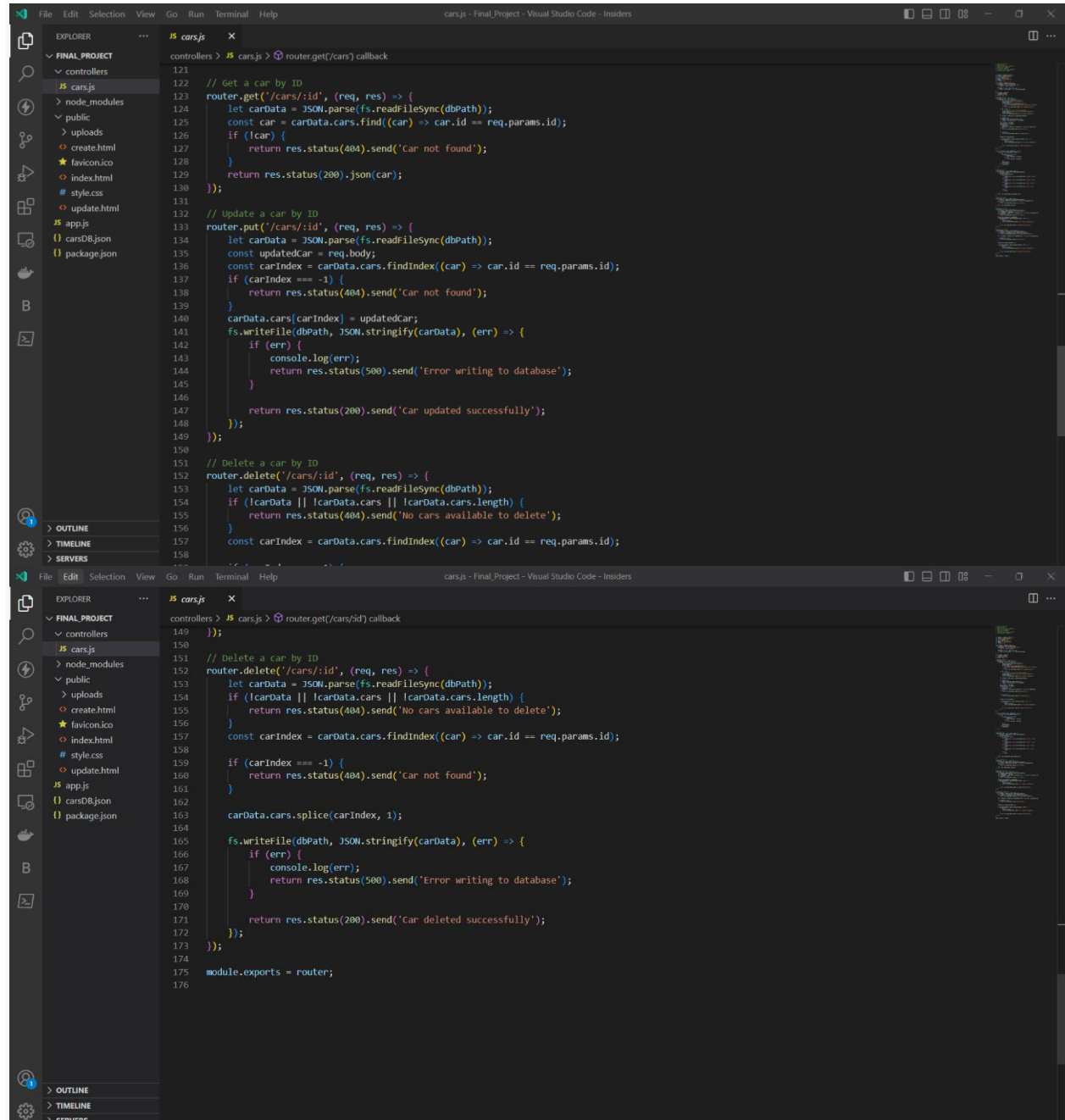
The bottom screenshot displays the 'cars.js' file with the following code:

```
controllers > JS cars.js > router.post('/cars') callback > upload() callback > fs.writeFile() callback
// Get all cars
router.get('/cars', async (req, res) => {
  let carData = JSON.parse(fs.readFileSync(dbPath));
  if (req.query.sort) {
    switch (req.query.sort) {
      case "1":
        carData.cars = await sort(carData.cars, "price", false);
        break;
      case "2":
        carData.cars = await sort(carData.cars, "price", true);
        break;
      case "3":
        carData.cars = await sort(carData.cars, "kms", false);
        break;
      case "4":
        carData.cars = await sort(carData.cars, "kms", true);
        break;
      default:
        break;
    }
  }
  return res.status(200).json(carData.cars);
});

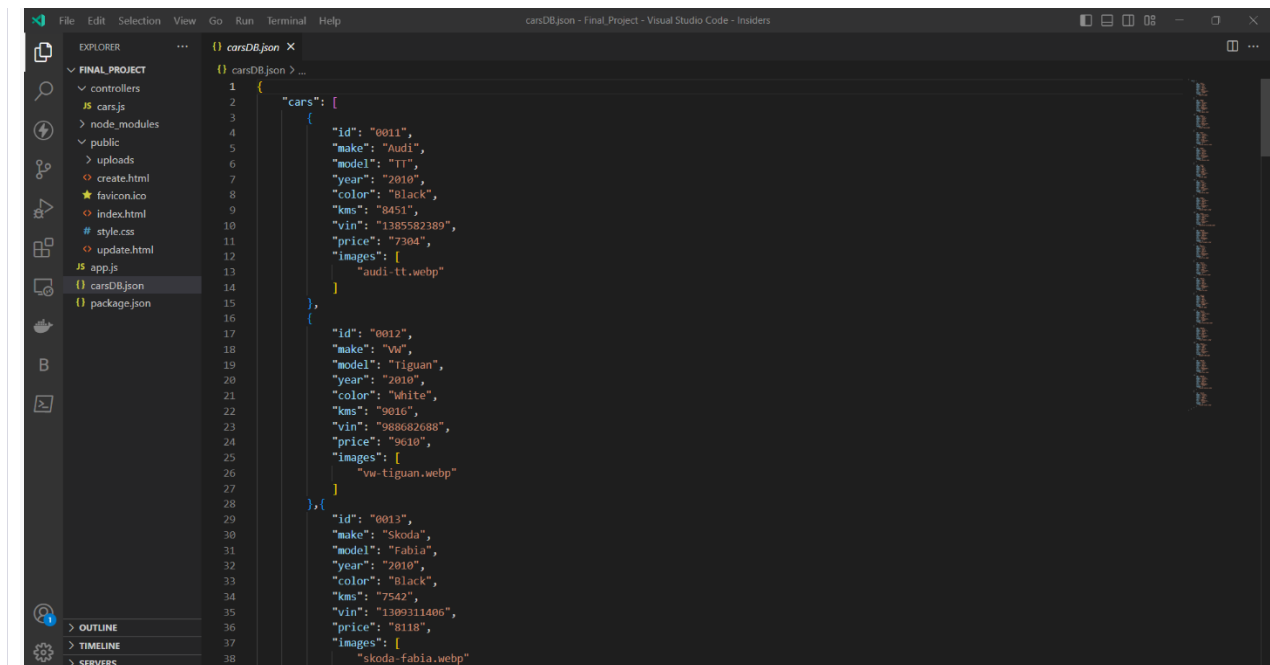
// Get a car by ID
router.get('/cars/:id', (req, res) => {
  let carData = JSON.parse(fs.readFileSync(dbPath));
  const car = carData.cars.find(car => car.id == req.params.id);
  if (!car) {
    return res.status(404).send('Car not found');
  }
  return res.status(200).json(car);
});

// Update a car by ID
router.put('/cars/:id', (req, res) => {
  let carData = JSON.parse(fs.readFileSync(dbPath));
  const car = carData.cars.find(car => car.id == req.params.id);
  if (!car) {
    return res.status(404).send('Car not found');
  }
  carData.cars[carData.cars.indexOf(car)] = req.body;
  fs.writeFile(dbPath, JSON.stringify(carData), (err) => {
    if (err) {
      console.log(err);
      return res.status(500).send('Error writing to database');
    }
    return res.status(200).send('Car updated successfully');
  });
});
```

## Full Stack JavaScript CSD 3103



```
controllers > JS cars.js > router.get('/cars') callback
121
122 // Get a car by ID
123 router.get('/cars/:id', (req, res) => {
124   let carData = JSON.parse(fs.readFileSync(dbPath));
125   const car = carData.cars.find((car) => car.id == req.params.id);
126   if (!car) {
127     return res.status(404).send('Car not found');
128   }
129   return res.status(200).json(car);
130 });
131
132 // Update a car by ID
133 router.put('/cars/:id', (req, res) => {
134   let carData = JSON.parse(fs.readFileSync(dbPath));
135   const updatedCar = req.body;
136   const carIndex = carData.cars.findIndex((car) => car.id == req.params.id);
137   if (carIndex === -1) {
138     return res.status(404).send('Car not found');
139   }
140   carData.cars[carIndex] = updatedCar;
141   fs.writeFile(dbPath, JSON.stringify(carData), (err) => {
142     if (err) {
143       console.log(err);
144       return res.status(500).send('Error writing to database');
145     }
146     return res.status(200).send('Car updated successfully');
147   });
148 });
149
150 // Delete a car by ID
151 router.delete('/cars/:id', (req, res) => {
152   let carData = JSON.parse(fs.readFileSync(dbPath));
153   if (!carData || !carData.cars || !carData.cars.length) {
154     return res.status(404).send('No cars available to delete');
155   }
156   const carIndex = carData.cars.findIndex((car) => car.id == req.params.id);
157   if (carIndex === -1) {
158     return res.status(404).send('Car not found');
159   }
160   carData.cars.splice(carIndex, 1);
161   fs.writeFile(dbPath, JSON.stringify(carData), (err) => {
162     if (err) {
163       console.log(err);
164       return res.status(500).send('Error writing to database');
165     }
166     return res.status(200).send('Car deleted successfully');
167   });
168 });
169
170 module.exports = router;
171
172
173
174
175
176
```



```
1 {
2   "cars": [
3     {
4       "id": "0011",
5       "make": "Audi",
6       "model": "TT",
7       "year": "2010",
8       "color": "Black",
9       "kms": "8451",
10      "vin": "1385582389",
11      "price": "7304",
12      "images": [
13        "audi-tt.webp"
14      ]
15    },
16    {
17      "id": "0012",
18      "make": "Vw",
19      "model": "Tiguan",
20      "year": "2010",
21      "color": "White",
22      "kms": "9016",
23      "vin": "988682688",
24      "price": "9610",
25      "images": [
26        "vw-tiguan.webp"
27      ]
28    },
29    {
30      "id": "0013",
31      "make": "Skoda",
32      "model": "Fabia",
33      "year": "2010",
34      "color": "Black",
35      "kms": "7542",
36      "vin": "1309311406",
37      "price": "8118",
38      "images": [
39        "skoda-fabia.webp"
40      ]
41    }
42  ]
43 }
```

## V. Project Management

- The development process of the website followed the Agile methodology, with regular sprints and meetings to ensure progress was being made.
- Challenges faced during development included ensuring the website was scalable and could handle large amounts of traffic, as well as ensuring user privacy and data security.
- These challenges were overcome through careful planning and testing, as well as the implementation of robust security protocols and server infrastructure.

## VI. Future Development

- Future features for the website include a mobile app version, additional search criteria such as fuel efficiency and safety ratings, and integration with popular car sales platforms.
- The roadmap for future development includes regular updates and feature additions based on user feedback and market trends.

## VII. Conclusion

## Full Stack JavaScript CSD 3103

- The project car list website is a valuable resource for car enthusiasts, providing a comprehensive list of cars for browsing and searching, as well as a platform for users to share their own knowledge and opinions.
- The website's intuitive user interface and comprehensive feature set make it an essential tool for car enthusiasts, and its robust technical infrastructure ensures it will continue to be a valuable resource for years to come.