Stage 1

I have chosen two open datasets from Kaggle (JACKYWANG, 2019) regarding One-Star Michelin restaurants and Three-Stars Michelin restaurants respectively. Both datasets have the same columns, “name”, “year”, “latitude”, “longitude”, “city”, “region”, “zipCode”, “cuisine”, “url”, “price”.

In terms of quality, the data is reliable as the “url” column contains credible URLs to the information pages of the Michelin Restaurants. The columns also made sense as there is no made-up city in the “city” column.

For details, the information provided are sufficient and helpful for people to decide on which Michelin restaurant they would go to using columns such as “cuisine” to infer what type of food the restaurant serves and “price” to determine if it is affordable.

Regarding the documentation of the data, it is clear what the objective of the dataset is, which is to display the meta information of Michelin restaurants such as the name, city it is in and the general pricing of the food. The dataset was easy to find since it is located on Kaggle, which is a popular website for people to find datasets in.

As for the dataset’s capability for interrelation, it can be easy to connect to other datasets such as different stars Michelin restaurants or even other tourist attractions, since food is also a factor in tourism, if they are in the same city as there is a “city” column. For example, as we can see from this project, I plan to connect One-Star Michelin restaurants to Three-Star Michelin restaurants as both are Michelin restaurants, it is just that they have different number of Michelin stars.

I would use the dataset as a reference to see what Michelin restaurants are available in the area that I could be in, for example, if I am in Wien and I am looking for a One-Star Michelin restaurant, there are 8 options for me to choose from and from there I can filter them by checking on the “cuisine” and “price” columns. One question that I would like to ask the dataset but could not is how the restaurant looks like, such as the interior design and the furniture used, if possible, I would like to see some pictures of the restaurant so I can find it more efficiently and know the environment I would be dining in. Another question is what the customers’ feedbacks and reviews are, which can be about the food or the service. Both questions are not answered by the dataset, but I would like to know the answers to both questions.

Lastly, in terms of discoverability, as I have mentioned it before, it is easy to find both datasets as they are from Kaggle. There are many other alternatives such as Google Dataset Search and Amazon’s AWS Public Datasets.

This is an interesting dataset to me because I am interested in good food. When a restaurant receives a Michelin Star, the food must be delicious. However, as mentioned above, the two questions that I would like to ask the datasets are how the restaurants look like, preferably with pictures and customers’ reviews on the food and service.

Stage 2

I have come up with the following E/R diagrams. The first one consists of only one entity being Restaurants and the attributes being One-Star Michelin Restaurant and Three-Stars Michelin Restaurant. The relationship between the entity and both attributes are many to zero or many. This is because there must exist restaurants in the world for humanity to exist and Michelin stars can be taken away, thus, there is a possibility that there might not be any One-Star or Three-Stars Michelin restaurants in the world.

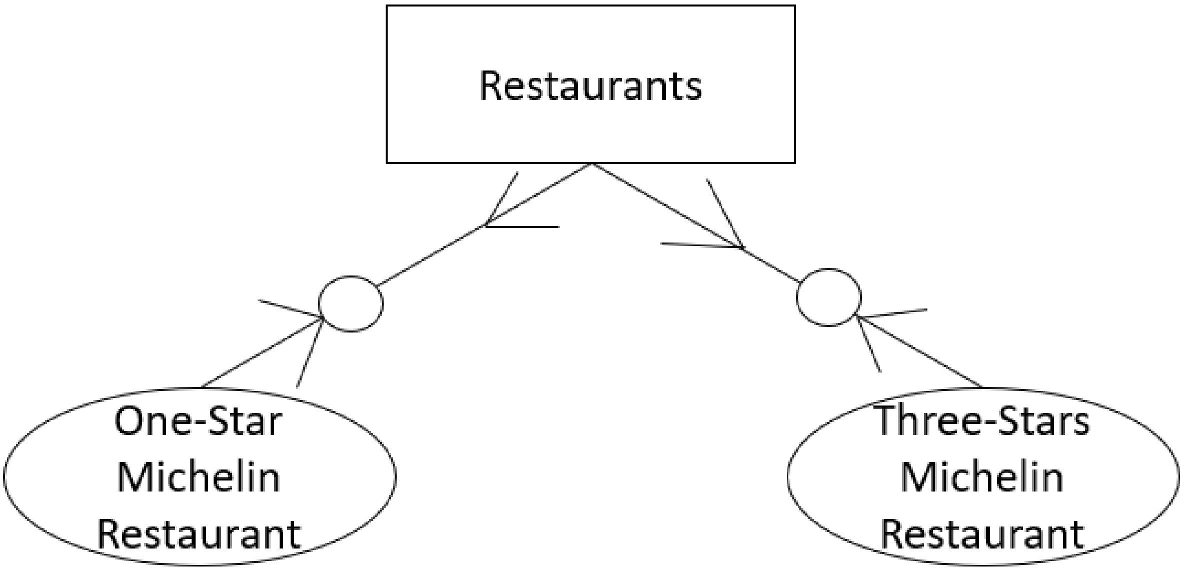


Image 1.1: E/R Diagram 1

The entities in the second and third E/R diagrams are One-Star Michelin Restaurant and Three-Stars Michelin Restaurant respectively. There are ten attributes in each E/R diagram, which are the columns from the datasets, namely name, year, latitude, longitude, city, region, zipCode, cuisine, price and url. The entities and the attributes city and region have a one or more to one or more relationship as there is a possibility to exist two or more One-Star or Three-Stars Michelin restaurants in a single city or region or two or more cities and regions having the same Michelin restaurant. As the price, cuisine and year attributes are a general grouping of expenses, types of food and year, there exist a one to one or more relationship from the entity to the attribute as well.

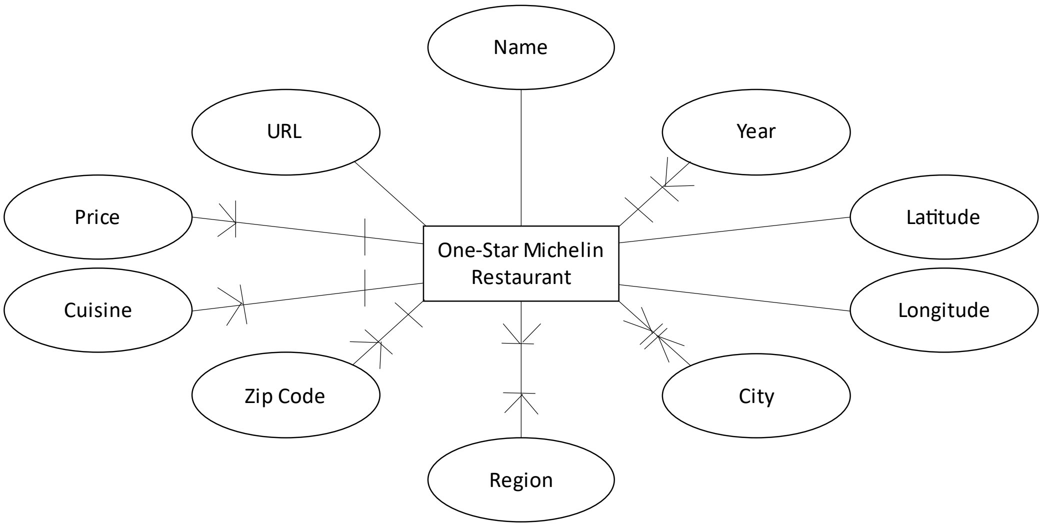


Image 1.2: E/R Diagram 2

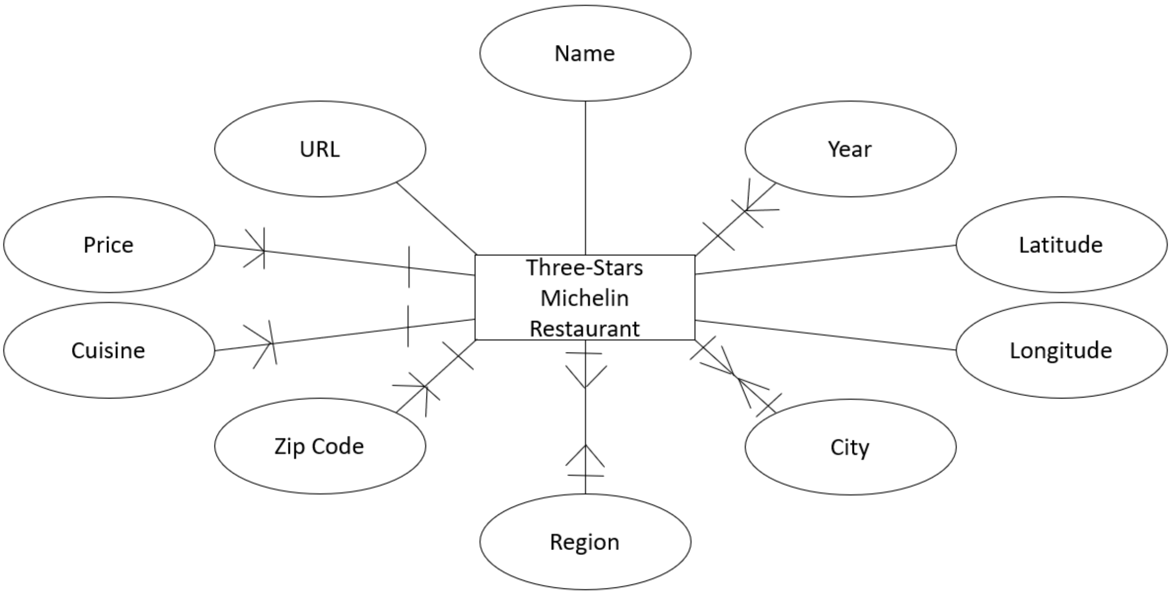


Image 1.3: E/R Diagram 3

The last E/R diagram has two entities namely One-Star Michelin Restaurant and Three-Star Michelin Restaurant. The relationship between both entities is a weak relationship as both entities can exist without the relationship which is the competition between them in the Food and Beverage Industry. The attributes of the entities as seen before in Diagram 2 and 3 are the columns of the datasets.

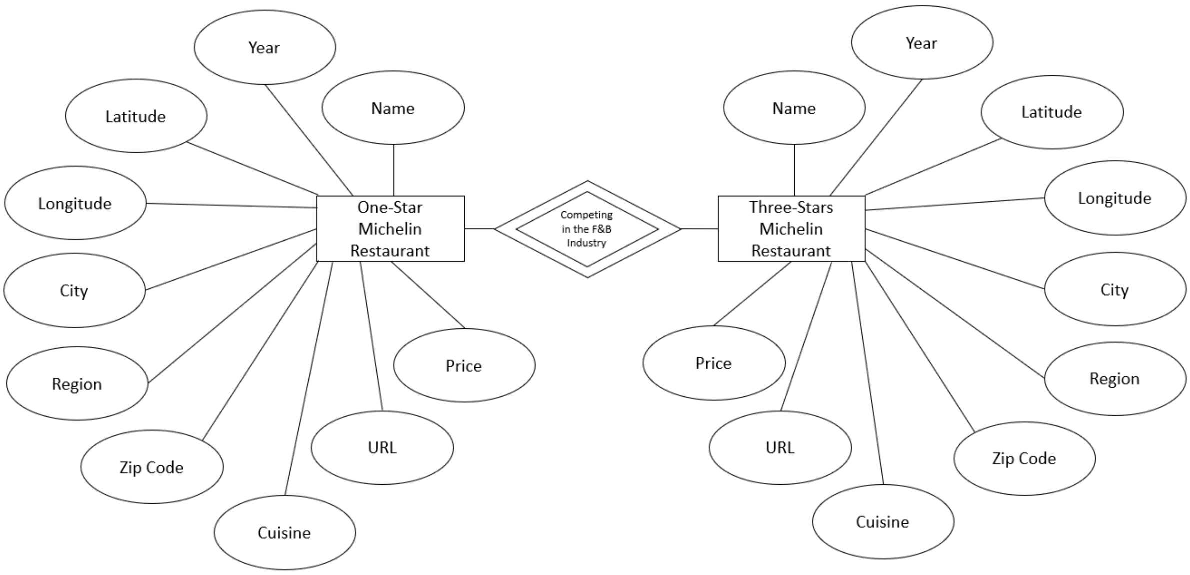


Image 1.4: E/R Diagram 4

The database that I will be creating is called Restaurants. The tables will be OneStarMichelin and ThreeStarsMichelin. The fields of both tables that I intend to create are similar which are name, year, latitude, longitude, city, region, zipCode, cuisine, price and url. However, to ensure the database is in at least 3NF, I need to make sure it is in 2NF first which means there needs to be a primary key and a unique attribute in the table. However, the same Michelin Restaurant can be in two or more different cities and regions, in this case, The Kitchen is one such example that exists in Sacramento and Macau. Therefore, I will have to create a new column in the datasets called "id" and make it the primary key and the unique constraint. Furthermore, to make sure it is in 3NF, I must delete two columns which are the region and the zipCode because they are dependent on the city column. For example, if the city is changed from Wien to Singapore, the region and the zipCode column will be different. Therefore, I must remove both columns. They will be saved into files ending with "revised" such as one-star-michelin-restaurants-revised.csv. I will put both the original CSV files and the revised CSV files into the lab environment for reference. As I have made changes to the datasets, I will attach the new E/R diagrams 2, 3 and 4 below. There is no change to the first E/R diagram.

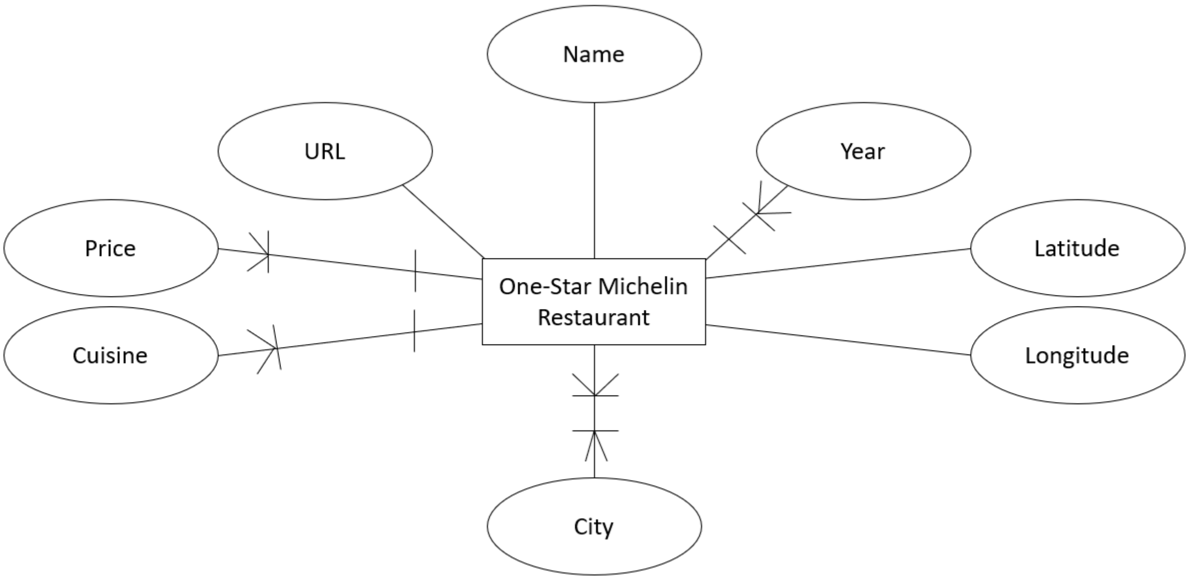


Image 2.1: New E/R Diagram 2

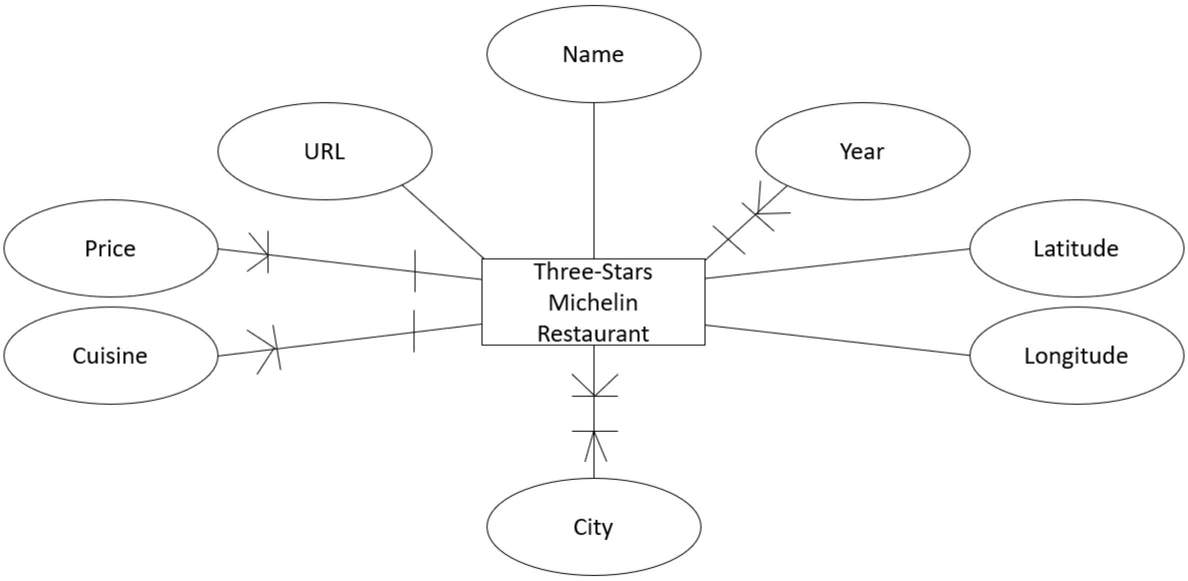


Image 2.2: New E/R Diagram 3

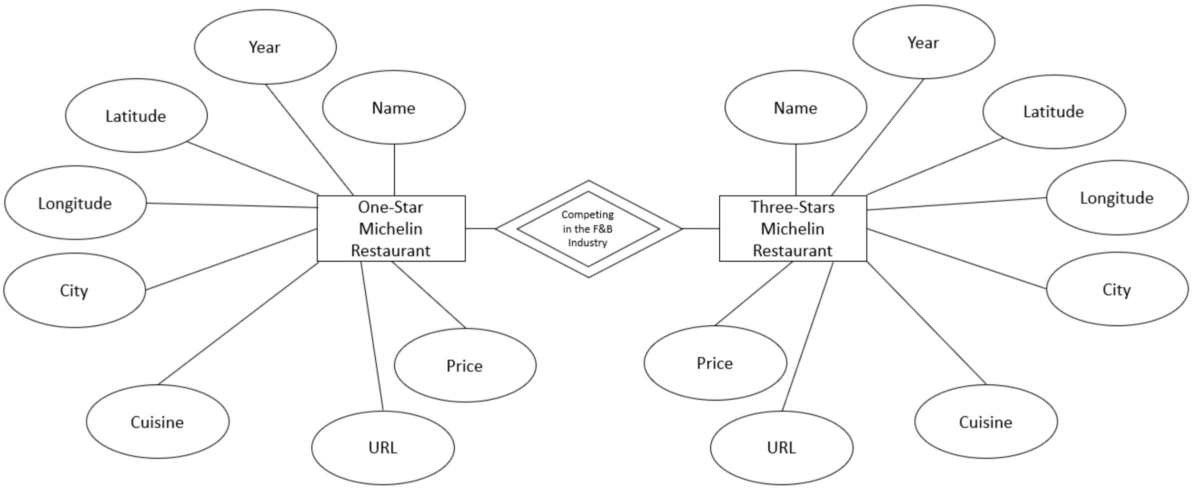
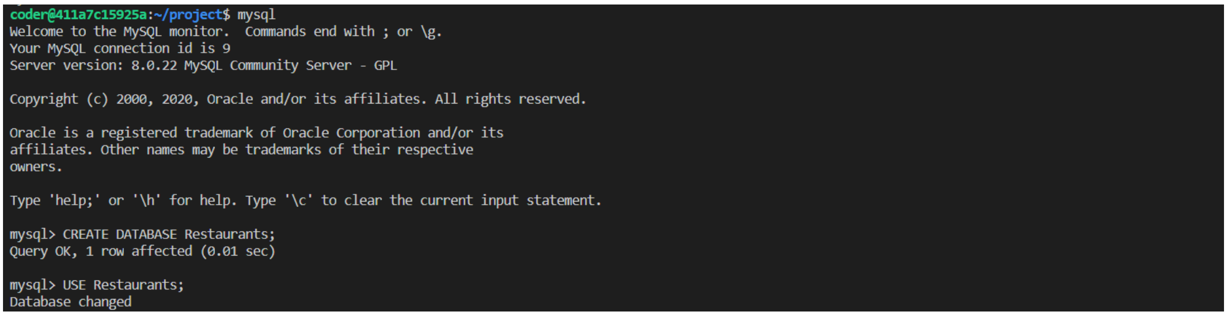

Year
Latitude
Name
Longitude
One-Star Michelin Restaurant
City
Price
Cuisine
URL
Competing in the F&B Industry
Year
Name
Latitude
Three-Stars Michelin Restaurant
Longitude
City
Price
Cuisine
URL

Image 2.3: New E/R Diagram 4

Stage 3

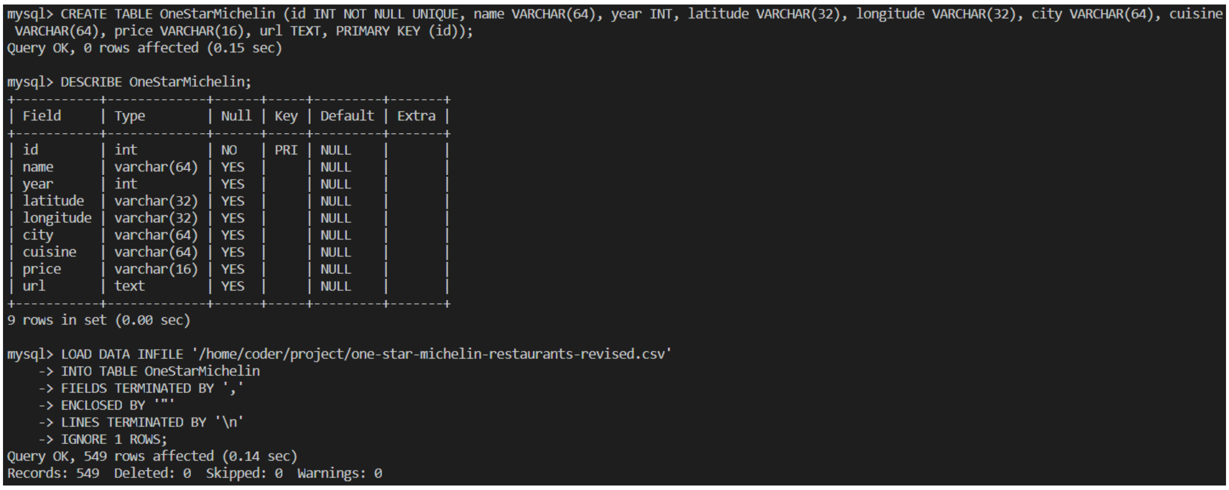
In Image 3.1, I created a database called Restaurants in the SQL server and connected to it.



```
coder@411a7c15925a:~/project$ mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.22 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE Restaurants;
Query OK, 1 row affected (0.01 sec)

mysql> USE Restaurants;
Database changed
```

Image 3.1: MySQL

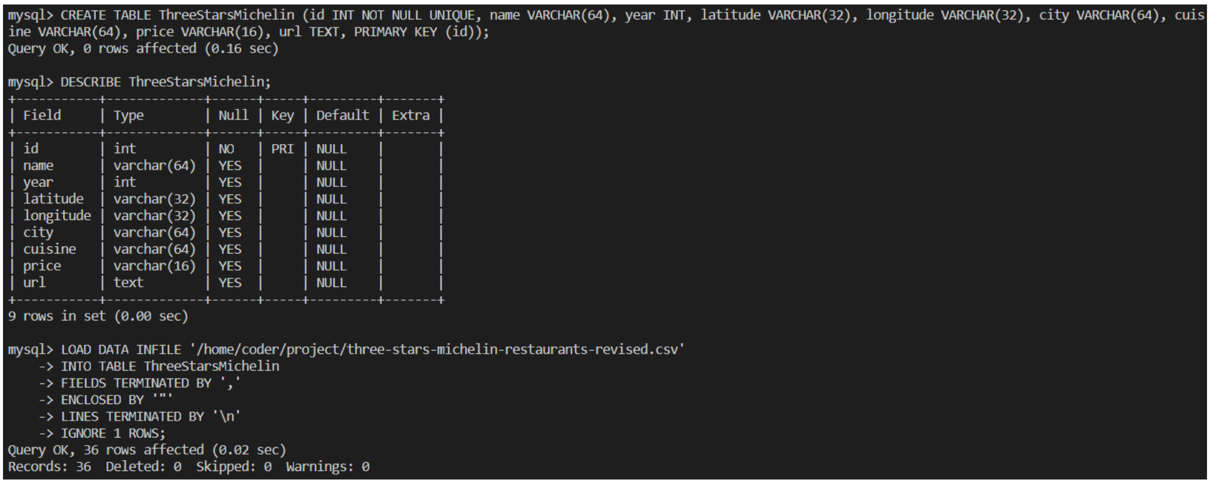
In Image 3.2, I created a table called OneStarMichelin which will consist of all the attributes from the CSV file 'one-star-michelin-restaurants-revised.csv'. I used suitable datatypes and lengths and set the primary key and unique constraint to the id column. Afterwards, I imported the entire data from the CSV file into the table, with a record of 549 rows. There is a 'IGNORE 1 ROWS' to exclude the header row in the CSV file.



```
mysql> CREATE TABLE OneStarMichelin (id INT NOT NULL UNIQUE, name VARCHAR(64), year INT, latitude VARCHAR(32), longitude VARCHAR(32), city VARCHAR(64), cuisine
 VARCHAR(64), price VARCHAR(16), url TEXT, PRIMARY KEY (id));
Query OK, 0 rows affected (0.15 sec)

mysql> DESCRIBE OneStarMichelin;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | int         | NO   | PRI | NULL    |       |
| name      | varchar(64) | YES  |     | NULL    |       |
| year      | int         | YES  |     | NULL    |       |
| latitude  | varchar(32) | YES  |     | NULL    |       |
| longitude | varchar(32) | YES  |     | NULL    |       |
| city      | varchar(64) | YES  |     | NULL    |       |
| cuisine   | varchar(64) | YES  |     | NULL    |       |
| price     | varchar(16) | YES  |     | NULL    |       |
| url       | text        | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)

mysql> LOAD DATA INFILE '/home/coder/project/one-star-michelin-restaurants-revised.csv'
    -> INTO TABLE OneStarMichelin
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 ROWS;
Query OK, 549 rows affected (0.14 sec)
Records: 549  Deleted: 0  Skipped: 0  Warnings: 0
```

Image 3.2: MySQL

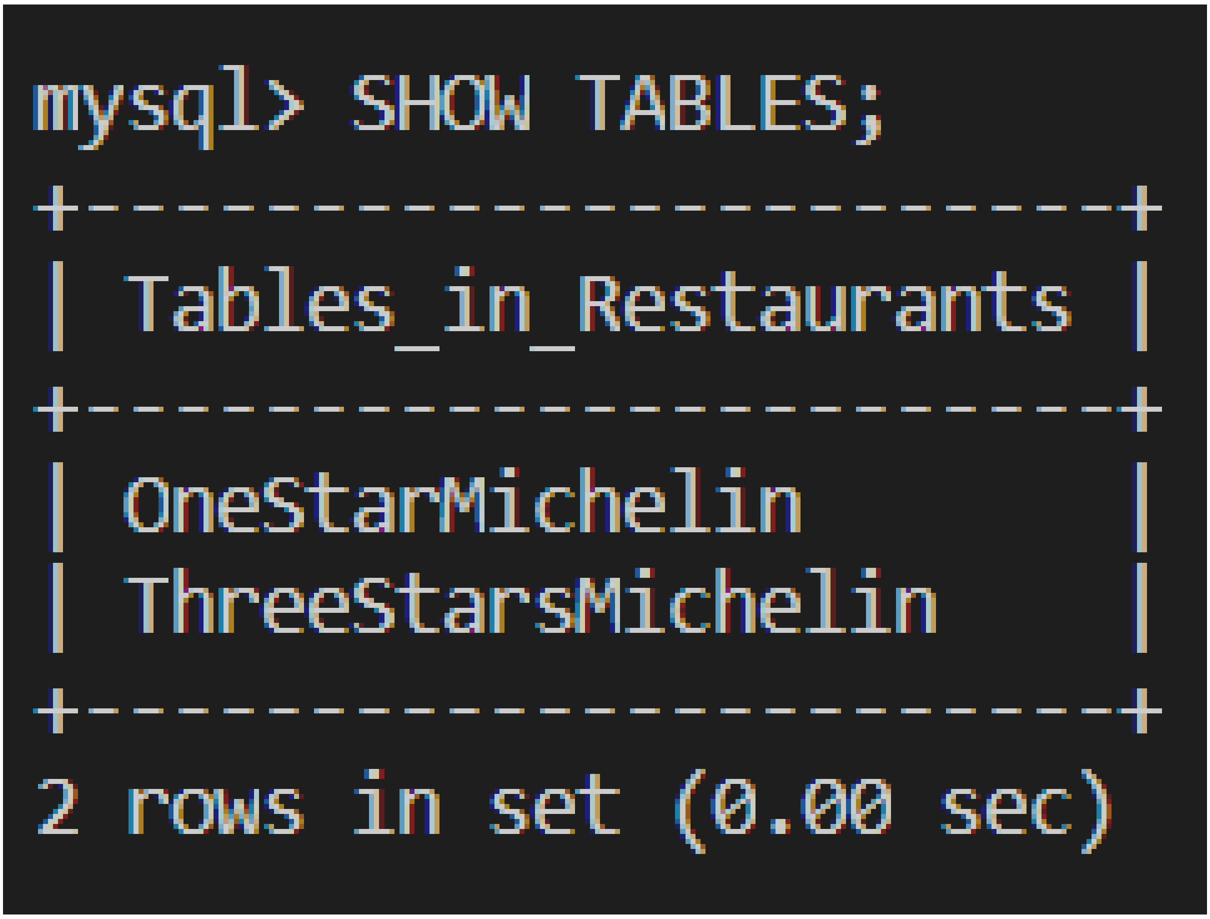
In Image 3.3, it is the same as in Image 3.2, but this time I created the table for ‘three-stars-michelin-restaurants-revised.csv’ called ThreeStarsMichelin. The attributes have similar datatypes and lengths to the previous table and the record is 36 rows.



```
mysql> CREATE TABLE ThreeStarsMichelin (id INT NOT NULL UNIQUE, name VARCHAR(64), year INT, latitude VARCHAR(32), longitude VARCHAR(32), city VARCHAR(64), cuis
ine VARCHAR(64), price VARCHAR(16), url TEXT, PRIMARY KEY (id));
Query OK, 0 rows affected (0.16 sec)

mysql> DESCRIBE ThreeStarsMichelin;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | int         | NO   | PRI | NULL    |       |
| name      | varchar(64) | YES  |     | NULL    |       |
| year      | int         | YES  |     | NULL    |       |
| latitude  | varchar(32) | YES  |     | NULL    |       |
| longitude | varchar(32) | YES  |     | NULL    |       |
| city      | varchar(64) | YES  |     | NULL    |       |
| cuisine   | varchar(64) | YES  |     | NULL    |       |
| price     | varchar(16) | YES  |     | NULL    |       |
| url       | text        | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)

mysql> LOAD DATA INFILE '/home/coder/project/three-stars-michelin-restaurants-revised.csv'
    -> INTO TABLE ThreeStarsMichelin
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 ROWS;
Query OK, 36 rows affected (0.02 sec)
Records: 36  Deleted: 0  Skipped: 0  Warnings: 0
```

Image 3.3: MySQL

In Image 3.4, I simply double checked my database and ensure that there are two tables, OneStarMichelin and ThreeStarsMichelin in Restaurants.



```
mysql> SHOW TABLES;
+-----------------------+
| Tables_in_Restaurants |
+-----------------------+
| OneStarMichelin       |
| ThreeStarsMichelin    |
+-----------------------+
2 rows in set (0.00 sec)
```

Image 3.4: MySQL

Reflecting on how well the database reflects the data, it already has strong normalization because it is in 3NF. This means that each row can be identified with the primary key and unique constraint which is the id, and the database has reduced redundancy and dependency between the attributes, thus, it becomes easier if one wants to insert, update, or delete any data. The data types for each attribute are also suitable, for example, id is classified as an integer and since we can import the data from the CSV files into the database.

Regarding my question to see pictures of the restaurants, the steps are to firstly create a new attribute for images in the table and then import the image with its path. The SQL commands are as follows:

```
ALTER TABLE OneStarMichelin ADD image BLOB;

UPDATE OneStarMichelin

SET image = LOAD_FILE(‘path to image file’)

WHERE id = 1;
```

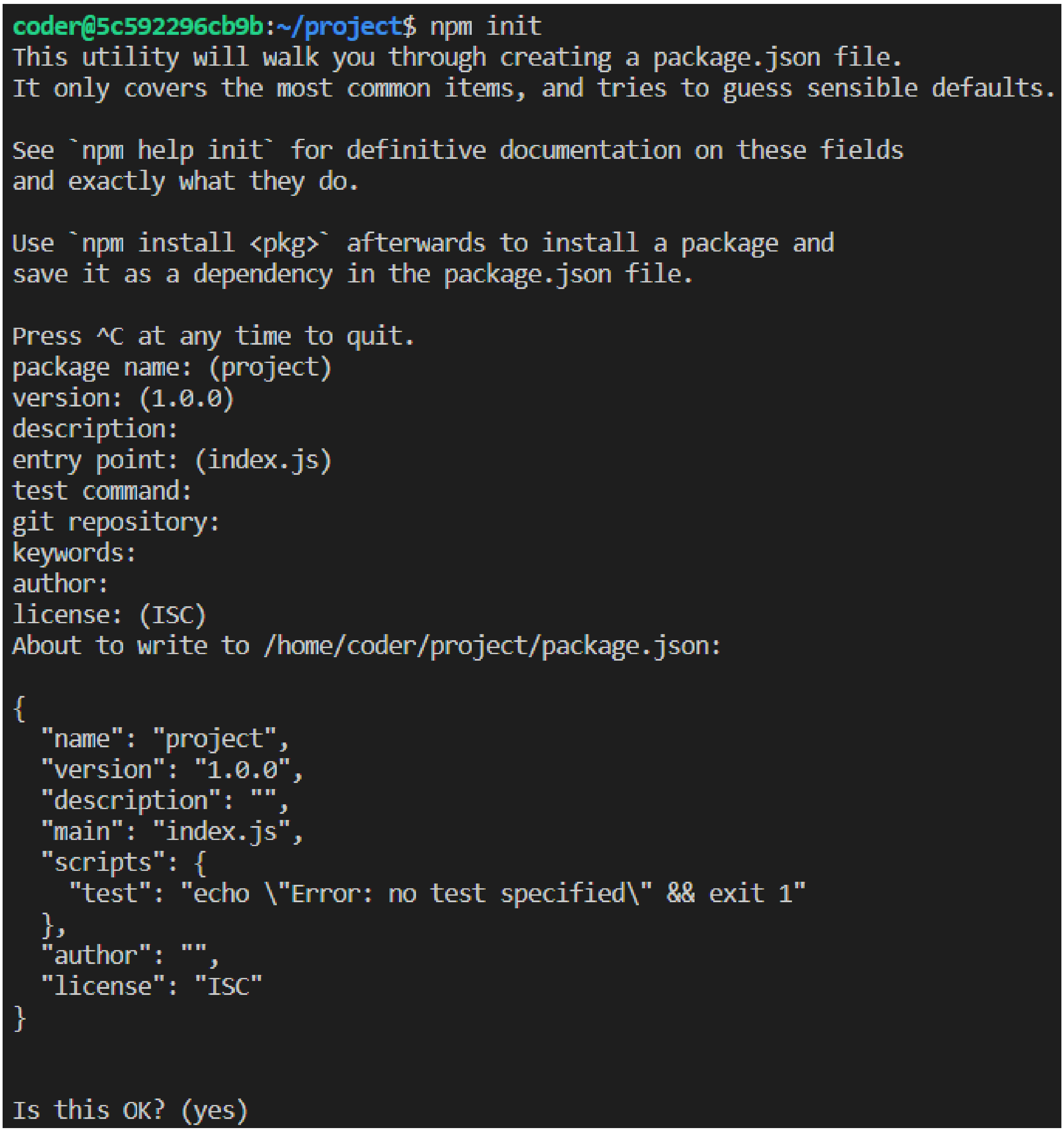
The second command will insert an image of the restaurant as a Binary Large Object into the image column where the id column is 1. However, it is usually not recommended to insert images into the database as it takes up a lot of space and this method will be very tedious as we can only insert one image at a time.

The second question to check customers’ reviews has about the same solution as the previous question, we will have to create a new column in the table and then allow users of the web application to input their reviews. It will not be wise for many users to access the SQL Server and so we will use the web application instead. The SQL command to add a new column called review is as follow:

```
ALTER TABLE OneStarMichelin ADD review TEXT;
```

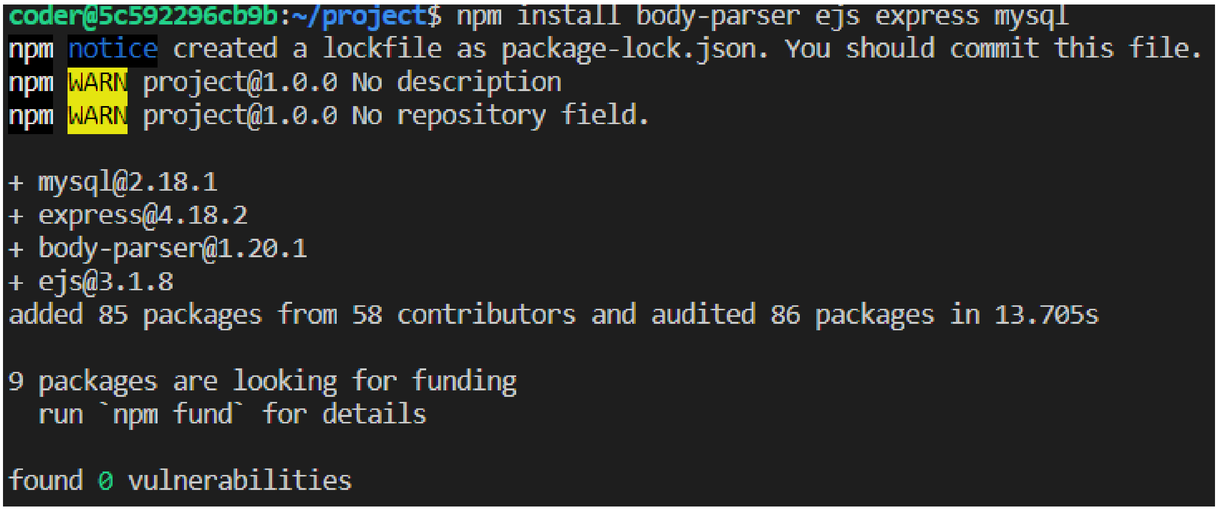
Stage 4

In order to create a Node.js application, I will need to run the commands shown in Image 4.1 and 4.2 to install the necessary packages and files.



```
coder@5c592296cb9b:~/project$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (project)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/coder/project/package.json:

{
  "name": "project",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes)
```

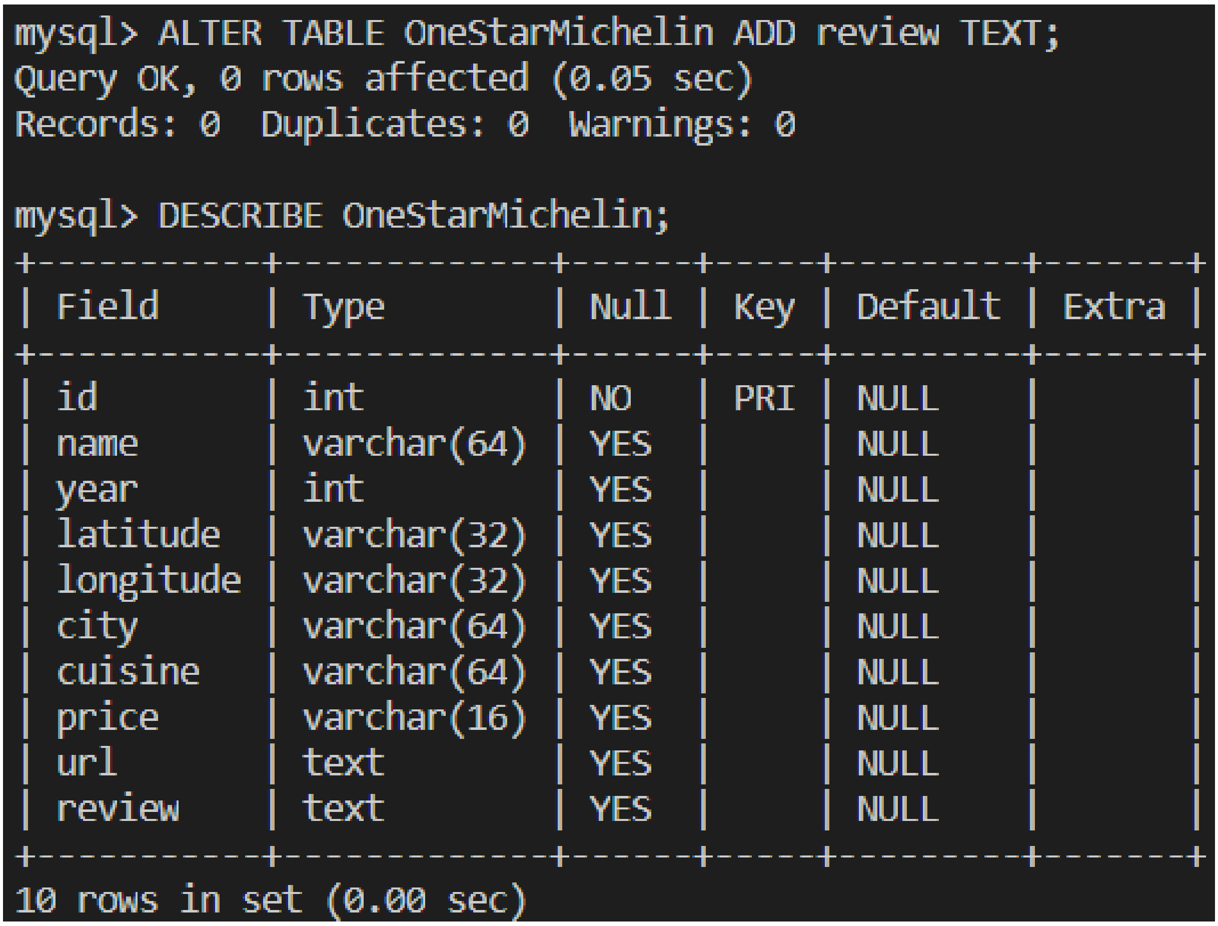
Image 4.1: Node.js



```
coder@5c592296cb9b:~/project$ npm install body-parser ejs express mysql
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN project@1.0.0 No description
npm WARN project@1.0.0 No repository field.

+ mysql@2.18.1
+ express@4.18.2
+ body-parser@1.20.1
+ ejs@3.1.8
added 85 packages from 58 contributors and audited 86 packages in 13.705s

9 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

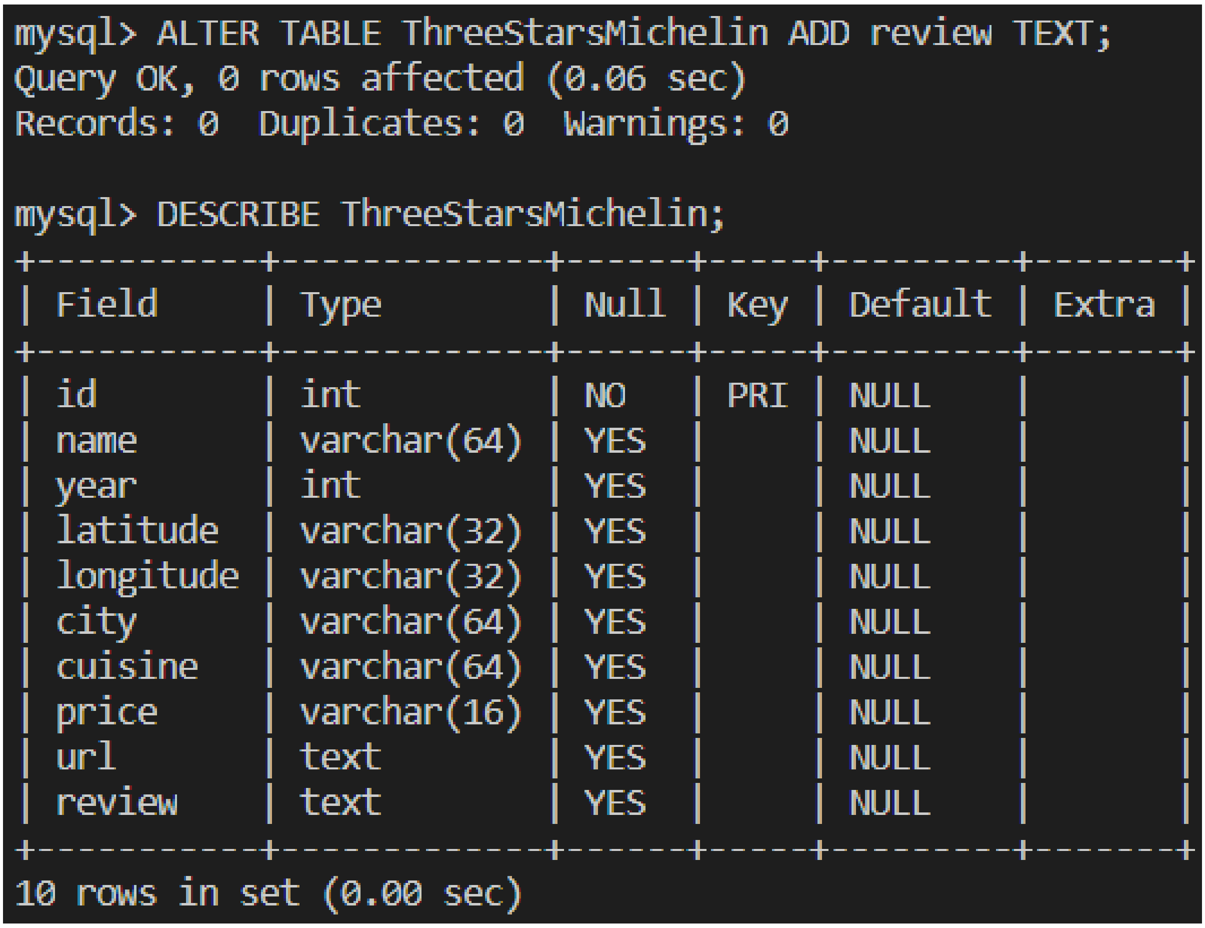
Image 4.2: Node.js

Next, to address the query of having customers' reviews for the restaurants, I altered the tables in the database and added in an additional attribute called review. This can be seen from Image 4.3 and 4.4.



```
mysql> ALTER TABLE OneStarMichelin ADD review TEXT;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE OneStarMichelin;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | int         | NO   | PRI | NULL    |       |
| name      | varchar(64) | YES  |     | NULL    |       |
| year      | int         | YES  |     | NULL    |       |
| latitude  | varchar(32) | YES  |     | NULL    |       |
| longitude | varchar(32) | YES  |     | NULL    |       |
| city      | varchar(64) | YES  |     | NULL    |       |
| cuisine   | varchar(64) | YES  |     | NULL    |       |
| price     | varchar(16) | YES  |     | NULL    |       |
| url       | text        | YES  |     | NULL    |       |
| review    | text        | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

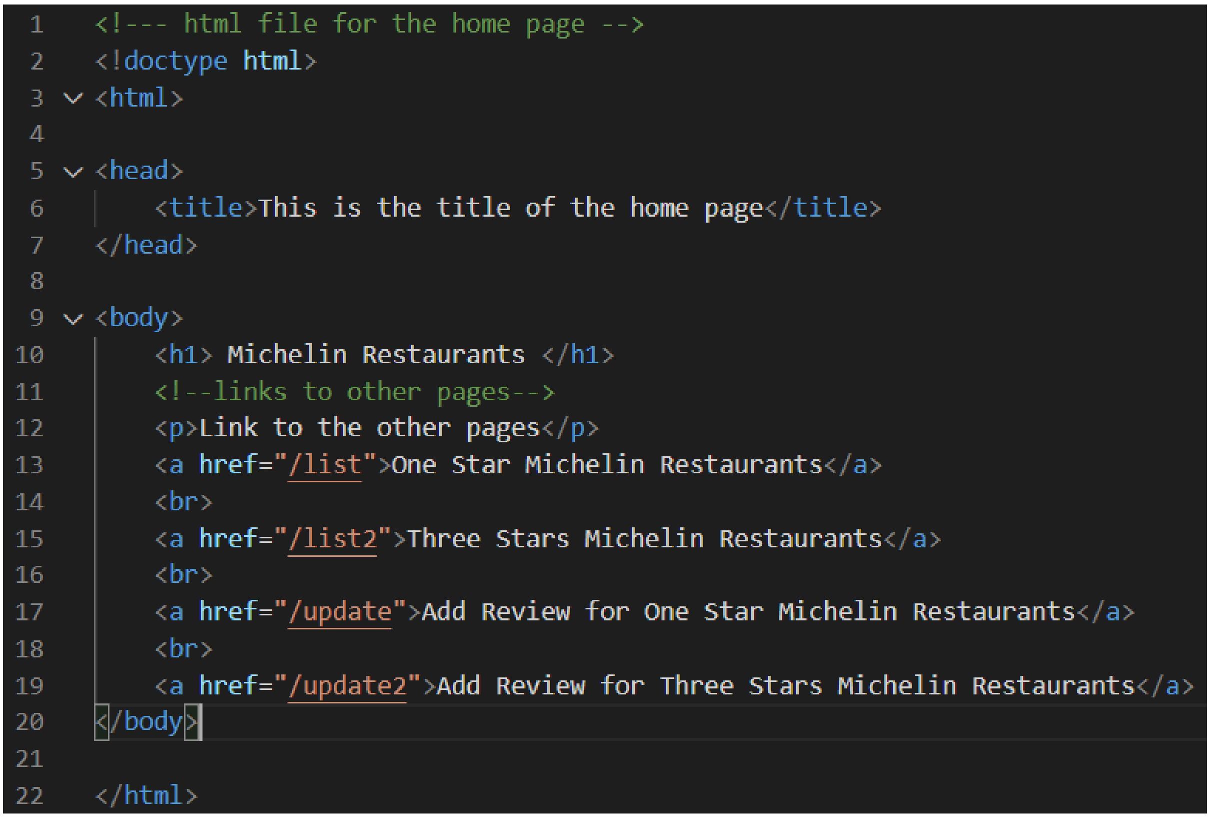
Image 4.3: MySQL



```
mysql> ALTER TABLE ThreeStarsMichelin ADD review TEXT;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE ThreeStarsMichelin;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | int         | NO   | PRI | NULL    |       |
| name      | varchar(64) | YES  |     | NULL    |       |
| year      | int         | YES  |     | NULL    |       |
| latitude  | varchar(32) | YES  |     | NULL    |       |
| longitude | varchar(32) | YES  |     | NULL    |       |
| city      | varchar(64) | YES  |     | NULL    |       |
| cuisine   | varchar(64) | YES  |     | NULL    |       |
| price     | varchar(16) | YES  |     | NULL    |       |
| url       | text        | YES  |     | NULL    |       |
| review    | text        | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

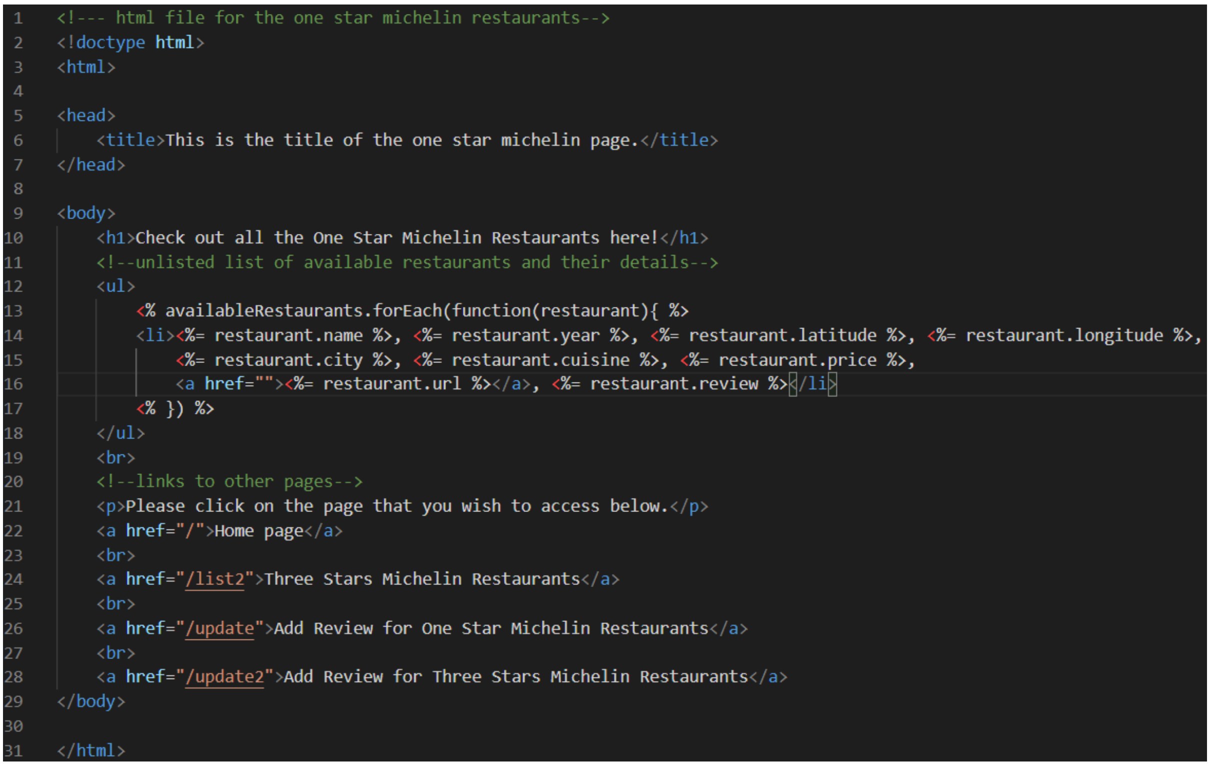
Image 4.4: MySQL

The following images from Image 4.5 to 4.11 are the html files that display the web application. index.html displays the home page, list.html and list2.html display the entire list of One Star Michelin and Three Stars Michelin restaurants respectively, addreview.html and addreview2.html display the webpages to select a One Star Michelin and Three Stars Michelin restaurant respectively to add a review, and the reviewdetails.html and reviewdetails2.html display the webpage to fill up a form for the review. The main.js file, from Image 4.12 to 4.15, consists of code for the scripts and functions to work. Many of the code are also split into One Star Michelin and Three Stars Michelin restaurants.



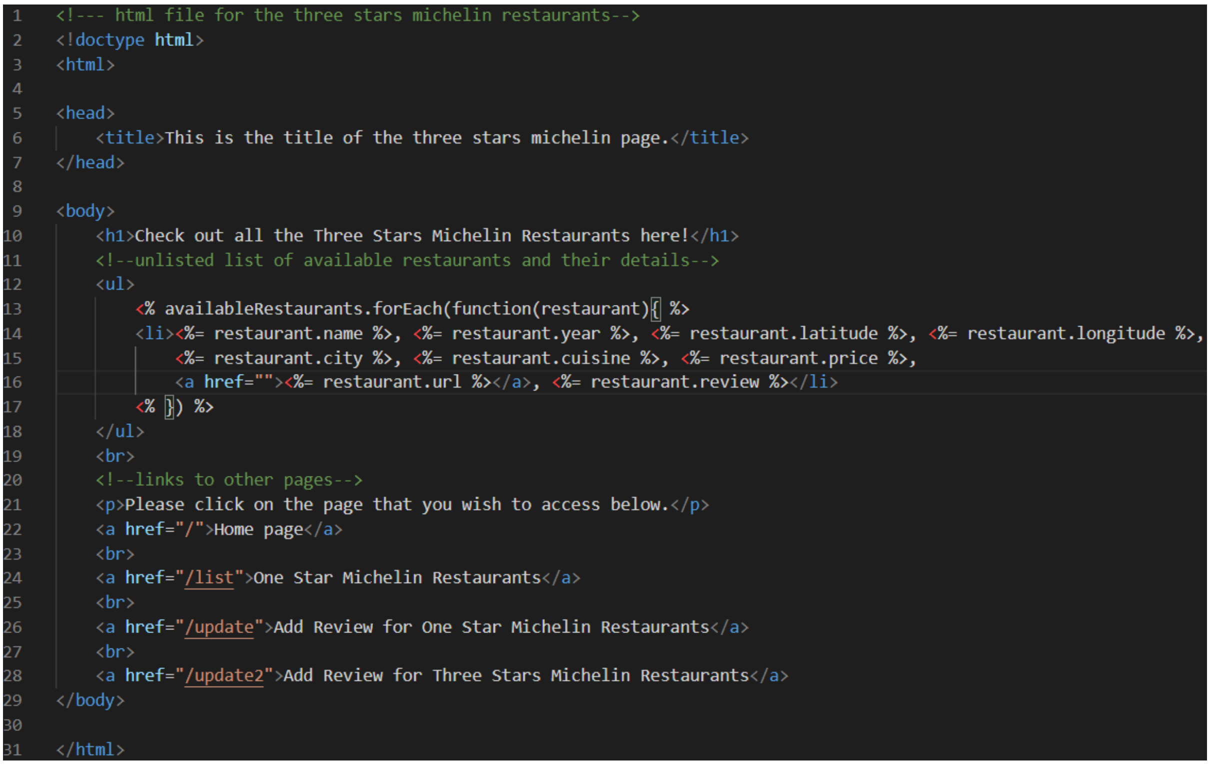
```html
<!--- html file for the home page -->
<!doctype html>
<html>

<head>
    <title>This is the title of the home page</title>
</head>

<body>
    <h1> Michelin Restaurants </h1>
    <!--links to other pages-->
    <p>Link to the other pages</p>
    <a href="/list">One Star Michelin Restaurants</a>
    <br>
    <a href="/list2">Three Stars Michelin Restaurants</a>
    <br>
    <a href="/update">Add Review for One Star Michelin Restaurants</a>
    <br>
    <a href="/update2">Add Review for Three Stars Michelin Restaurants</a>
</body>

</html>
```

Image 4.5: index.html



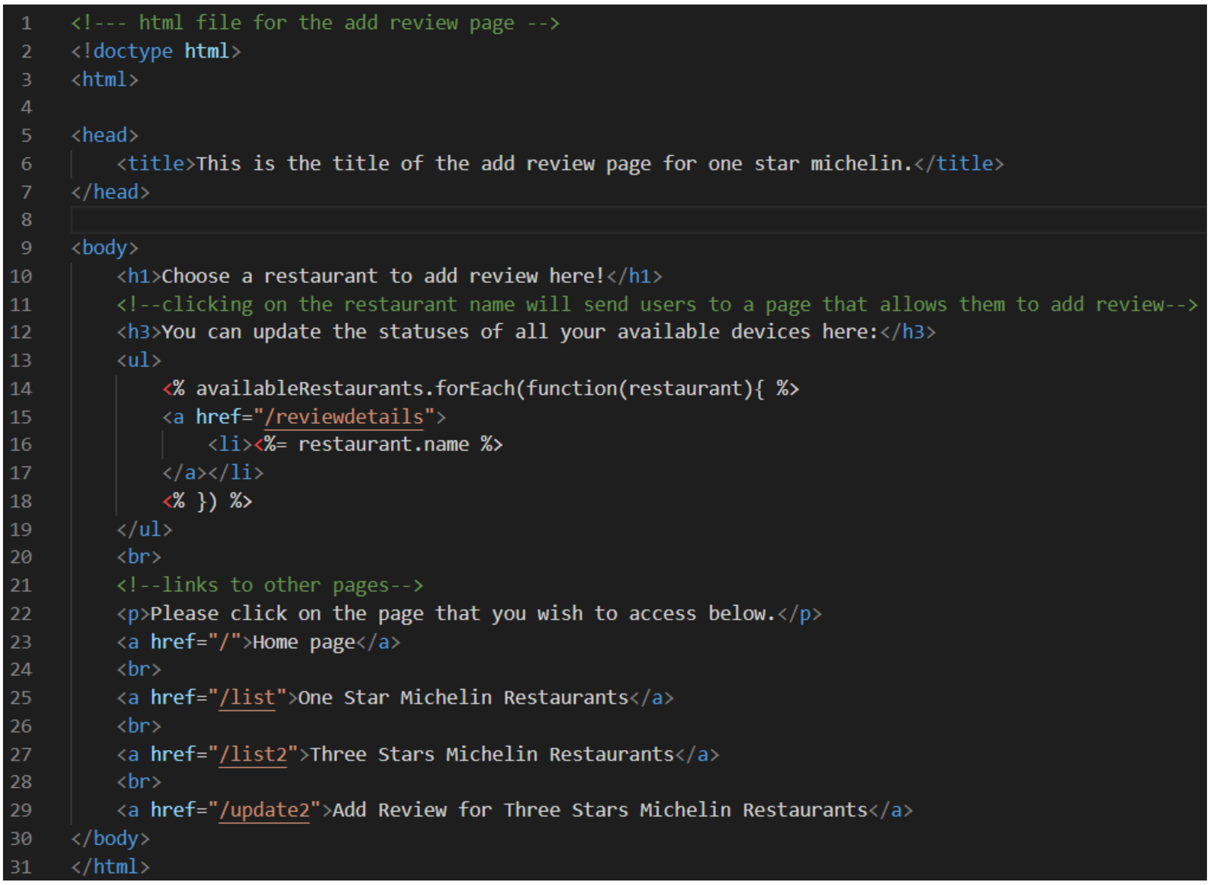
```html
<!--- html file for the one star michelin restaurants-->
<!doctype html>
<html>

<head>
    <title>This is the title of the one star michelin page.</title>
</head>

<body>
    <h1>Check out all the One Star Michelin Restaurants here!</h1>
    <!--unlisted list of available restaurants and their details-->
    <ul>
        <% availableRestaurants.forEach(function(restaurant){ %>
        <li><%= restaurant.name %>, <%= restaurant.year %>, <%= restaurant.latitude %>, <%= restaurant.longitude %>,
            <%= restaurant.city %>, <%= restaurant.cuisine %>, <%= restaurant.price %>,
            <a href=""><%= restaurant.url %></a>, <%= restaurant.review %></li>
        <% }) %>
    </ul>
    <br>
    <!--links to other pages-->
    <p>Please click on the page that you wish to access below.</p>
    <a href="/">Home page</a>
    <br>
    <a href="/list2">Three Stars Michelin Restaurants</a>
    <br>
    <a href="/update">Add Review for One Star Michelin Restaurants</a>
    <br>
    <a href="/update2">Add Review for Three Stars Michelin Restaurants</a>
</body>

</html>
```

Image 4.6: list.html



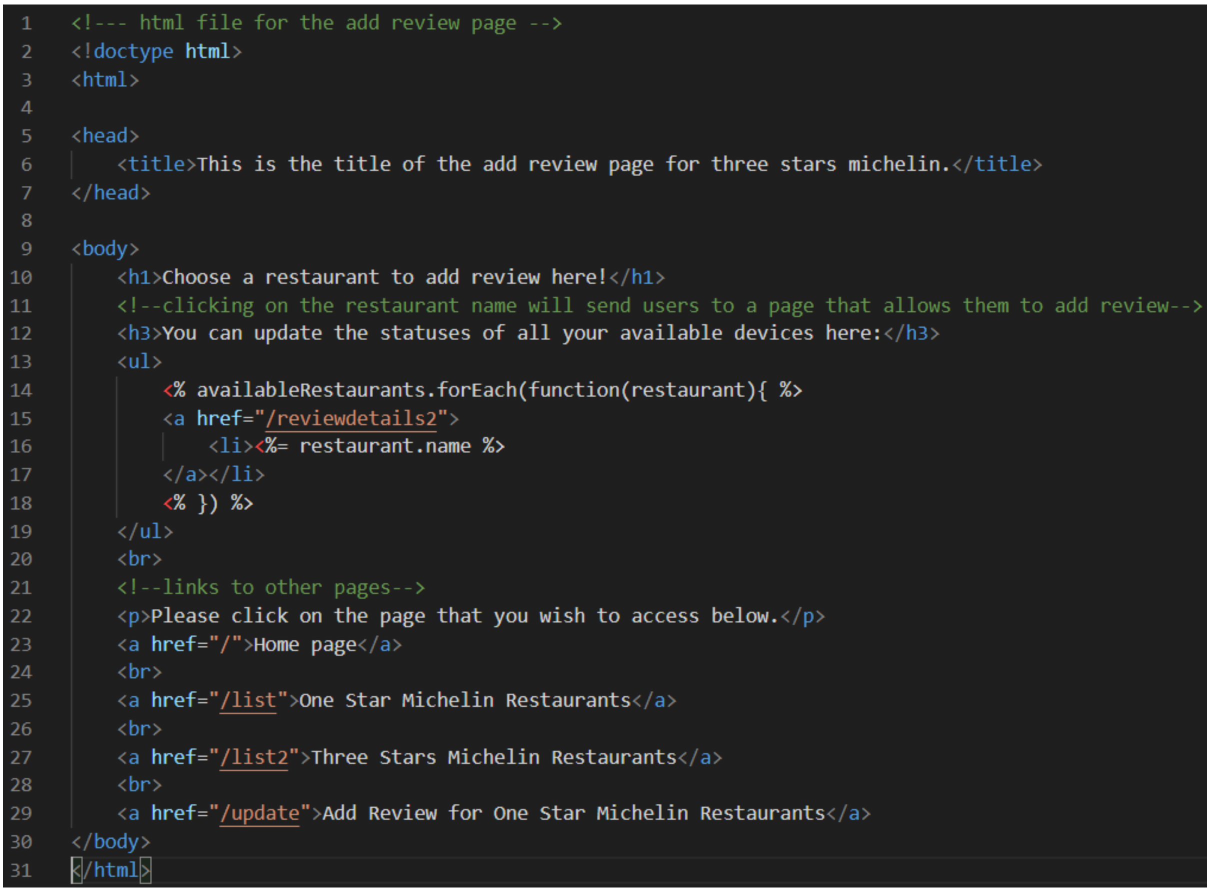
```html
<!--- html file for the three stars michelin restaurants-->
<!doctype html>
<html>

<head>
    <title>This is the title of the three stars michelin page.</title>
</head>

<body>
    <h1>Check out all the Three Stars Michelin Restaurants here!</h1>
    <!--unlisted list of available restaurants and their details-->
    <ul>
        <% availableRestaurants.forEach(function(restaurant){ %>
        <li><%= restaurant.name %>, <%= restaurant.year %>, <%= restaurant.latitude %>, <%= restaurant.longitude %>,
            <%= restaurant.city %>, <%= restaurant.cuisine %>, <%= restaurant.price %>,
            <a href=""><%= restaurant.url %></a>, <%= restaurant.review %></li>
        <% }) %>
    </ul>
    <br>
    <!--links to other pages-->
    <p>Please click on the page that you wish to access below.</p>
    <a href="/">Home page</a>
    <br>
    <a href="/list">One Star Michelin Restaurants</a>
    <br>
    <a href="/update">Add Review for One Star Michelin Restaurants</a>
    <br>
    <a href="/update2">Add Review for Three Stars Michelin Restaurants</a>
</body>

</html>
```

Image 4.7: list2.html



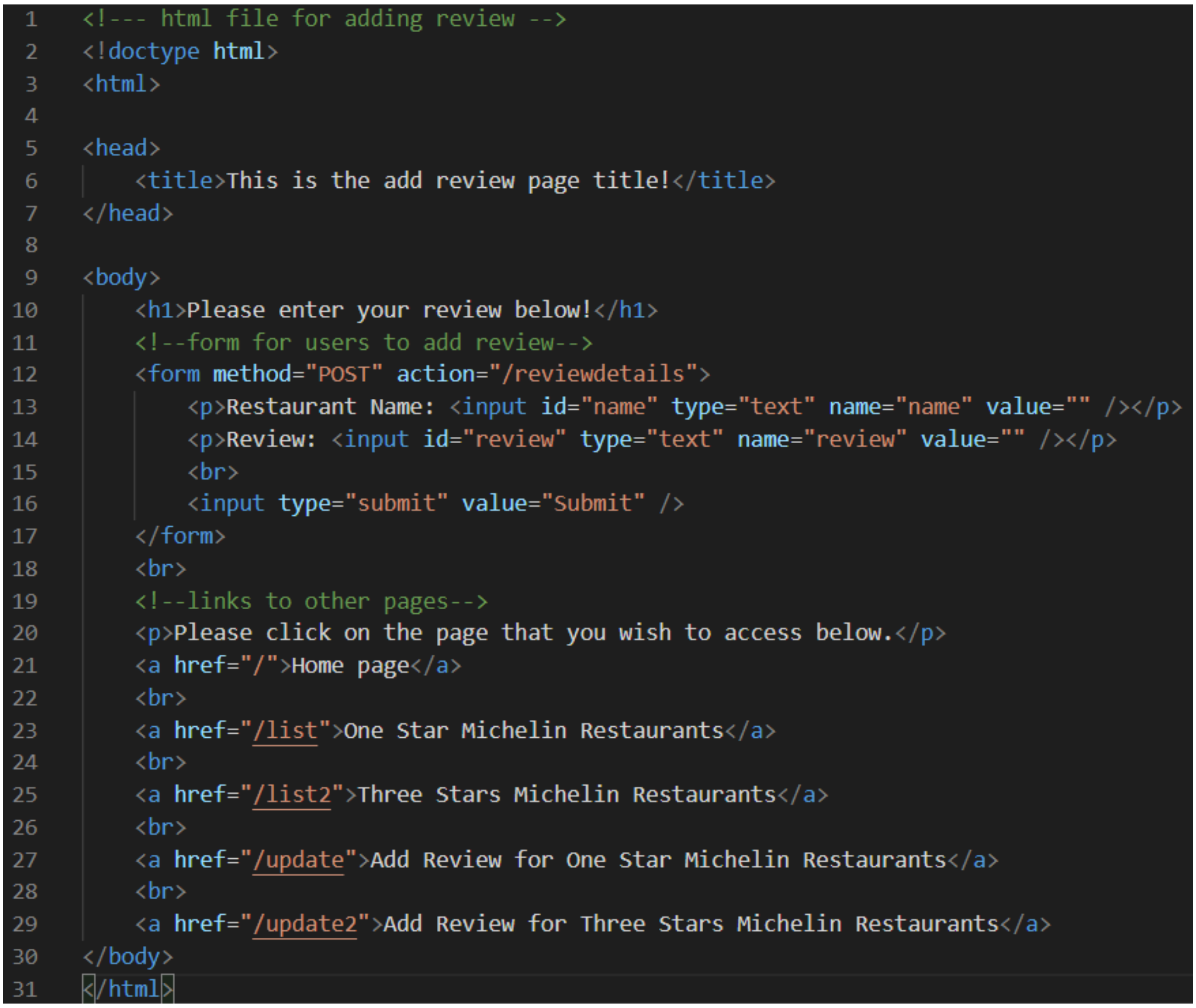
```html
<!--- html file for the add review page -->
<!doctype html>
<html>

<head>
    <title>This is the title of the add review page for one star michelin.</title>
</head>

<body>
    <h1>Choose a restaurant to add review here!</h1>
    <!--clicking on the restaurant name will send users to a page that allows them to add review-->
    <h3>You can update the statuses of all your available devices here:</h3>
    <ul>
        <% availableRestaurants.forEach(function(restaurant){ %>
        <a href="/reviewdetails">
            <li><%= restaurant.name %>
        </a></li>
        <% }) %>
    </ul>
    <br>
    <!--links to other pages-->
    <p>Please click on the page that you wish to access below.</p>
    <a href="/">Home page</a>
    <br>
    <a href="/list">One Star Michelin Restaurants</a>
    <br>
    <a href="/list2">Three Stars Michelin Restaurants</a>
    <br>
    <a href="/update2">Add Review for Three Stars Michelin Restaurants</a>
</body>
</html>
```

Image 4.8: addreview.html



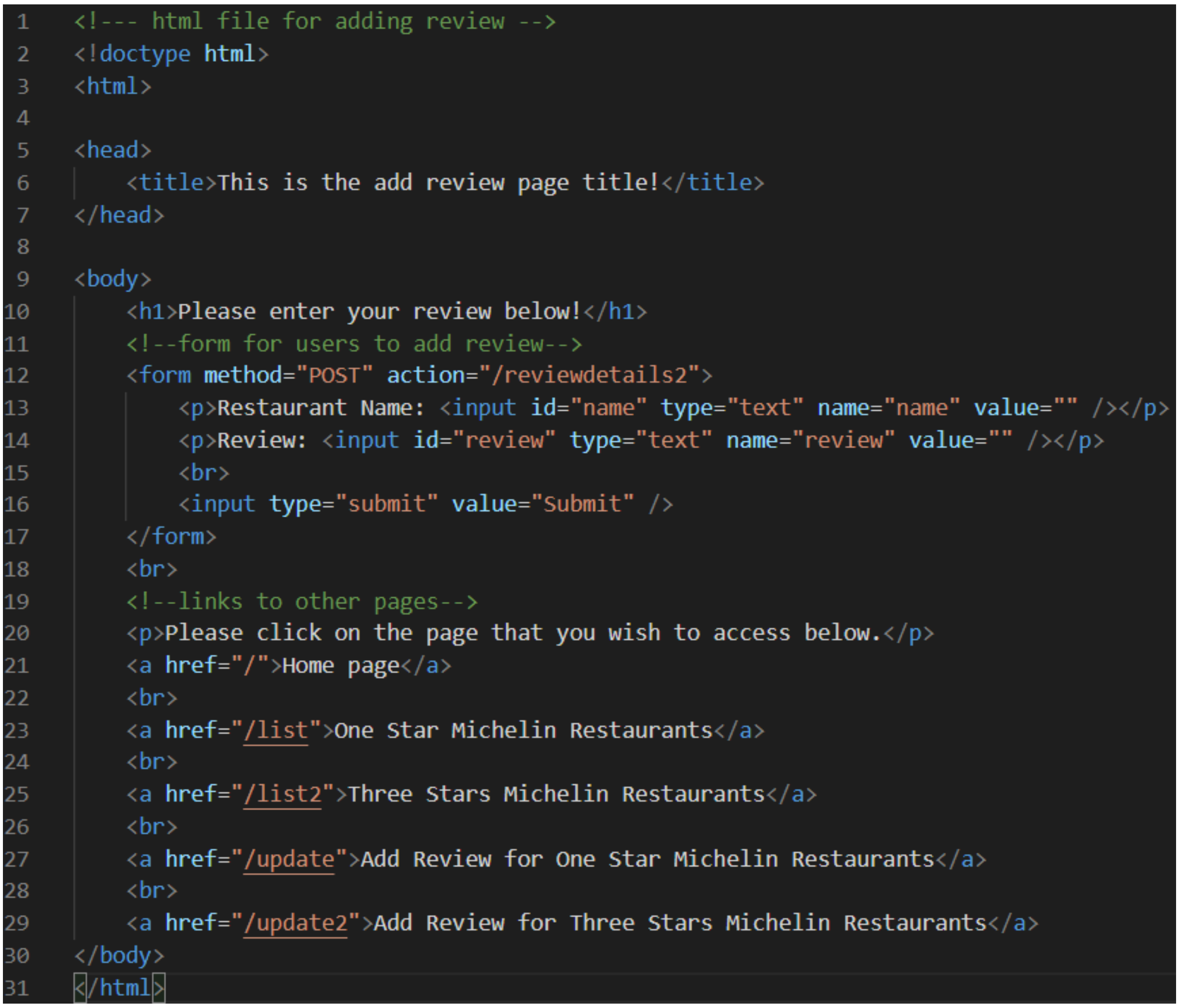
```html
<!--- html file for the add review page -->
<!doctype html>
<html>

<head>
    <title>This is the title of the add review page for three stars michelin.</title>
</head>

<body>
    <h1>Choose a restaurant to add review here!</h1>
    <!--clicking on the restaurant name will send users to a page that allows them to add review-->
    <h3>You can update the statuses of all your available devices here:</h3>
    <ul>
        <% availableRestaurants.forEach(function(restaurant){ %>
        <a href="/reviewdetails2">
            <li><%= restaurant.name %>
        </a></li>
        <% }) %>
    </ul>
    <br>
    <!--links to other pages-->
    <p>Please click on the page that you wish to access below.</p>
    <a href="/">Home page</a>
    <br>
    <a href="/list">One Star Michelin Restaurants</a>
    <br>
    <a href="/list2">Three Stars Michelin Restaurants</a>
    <br>
    <a href="/update">Add Review for One Star Michelin Restaurants</a>
</body>
</html>
```

Image 4.9: addreview2.html



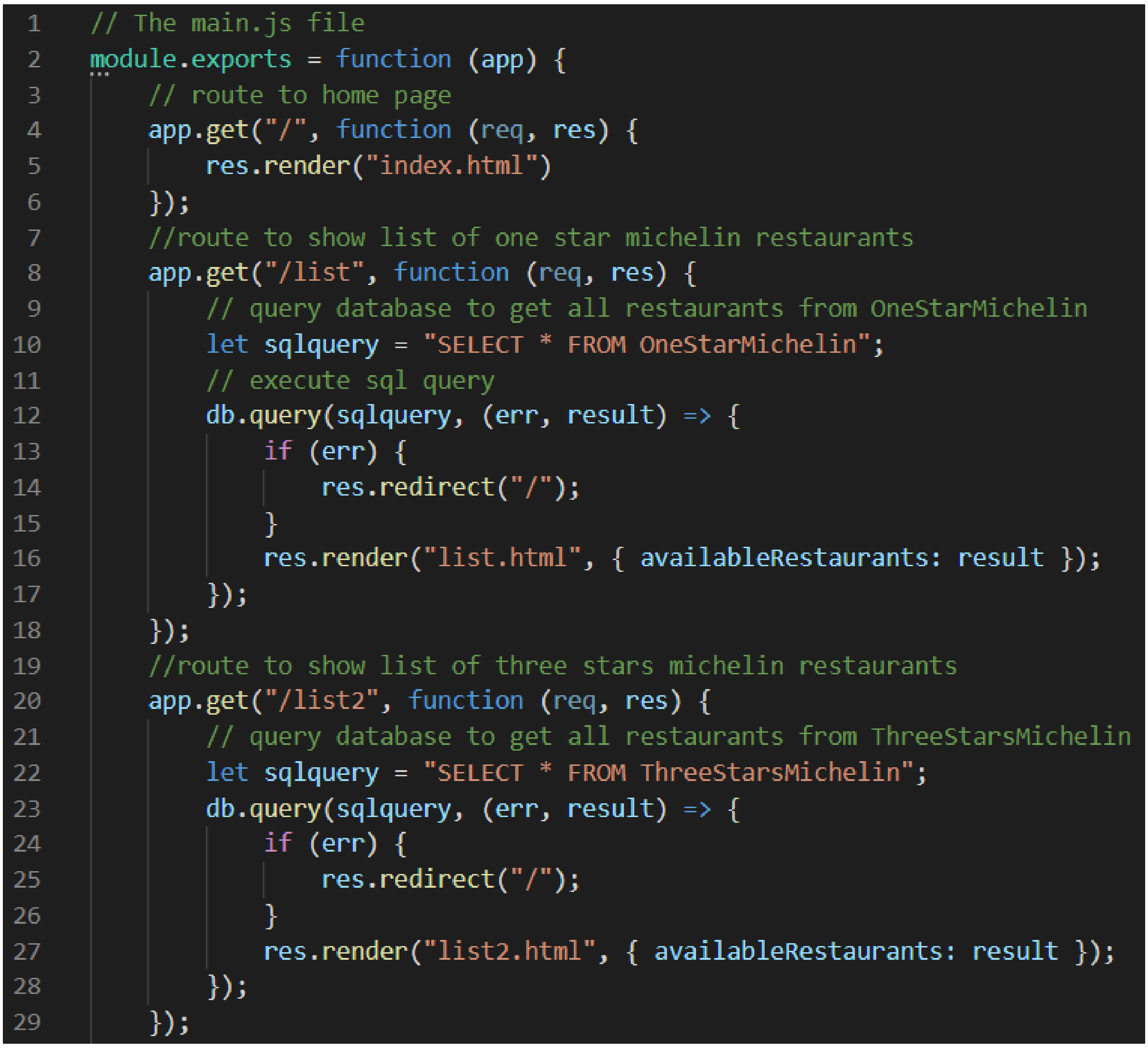
```html
<!--- html file for adding review -->
<!doctype html>
<html>

<head>
    <title>This is the add review page title!</title>
</head>

<body>
    <h1>Please enter your review below!</h1>
    <!--form for users to add review-->
    <form method="POST" action="/reviewdetails">
        <p>Restaurant Name: <input id="name" type="text" name="name" value="" /></p>
        <p>Review: <input id="review" type="text" name="review" value="" /></p>
        <br>
        <input type="submit" value="Submit" />
    </form>
    <br>
    <!--links to other pages-->
    <p>Please click on the page that you wish to access below.</p>
    <a href="/">Home page</a>
    <br>
    <a href="/list">One Star Michelin Restaurants</a>
    <br>
    <a href="/list2">Three Stars Michelin Restaurants</a>
    <br>
    <a href="/update">Add Review for One Star Michelin Restaurants</a>
    <br>
    <a href="/update2">Add Review for Three Stars Michelin Restaurants</a>
</body>
</html>
```

Image 4.10: reviewdetails.html



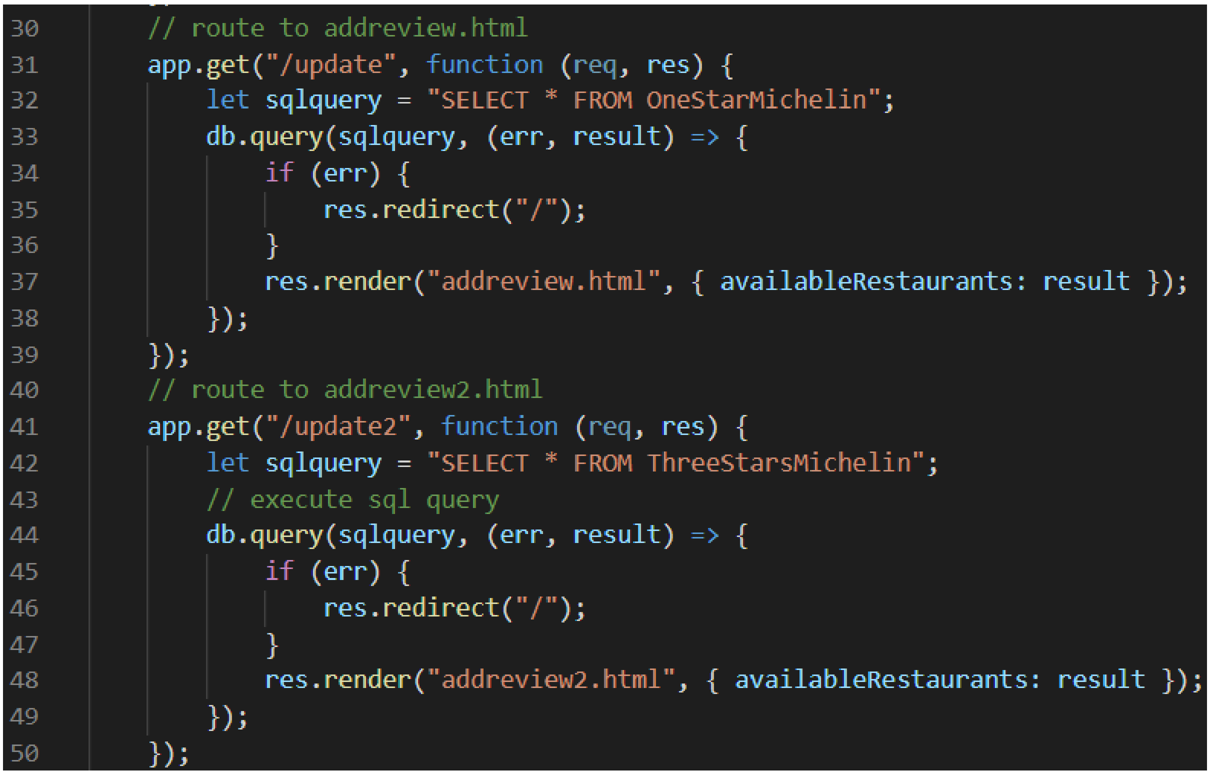
```html
<!--- html file for adding review -->
<!doctype html>
<html>

<head>
    <title>This is the add review page title!</title>
</head>

<body>
    <h1>Please enter your review below!</h1>
    <!--form for users to add review-->
    <form method="POST" action="/reviewdetails2">
        <p>Restaurant Name: <input id="name" type="text" name="name" value="" /></p>
        <p>Review: <input id="review" type="text" name="review" value="" /></p>
        <br>
        <input type="submit" value="Submit" />
    </form>
    <br>
    <!--links to other pages-->
    <p>Please click on the page that you wish to access below.</p>
    <a href="/">Home page</a>
    <br>
    <a href="/list">One Star Michelin Restaurants</a>
    <br>
    <a href="/list2">Three Stars Michelin Restaurants</a>
    <br>
    <a href="/update">Add Review for One Star Michelin Restaurants</a>
    <br>
    <a href="/update2">Add Review for Three Stars Michelin Restaurants</a>
</body>
</html>
```

Image 4.11: reviewdetails2.html



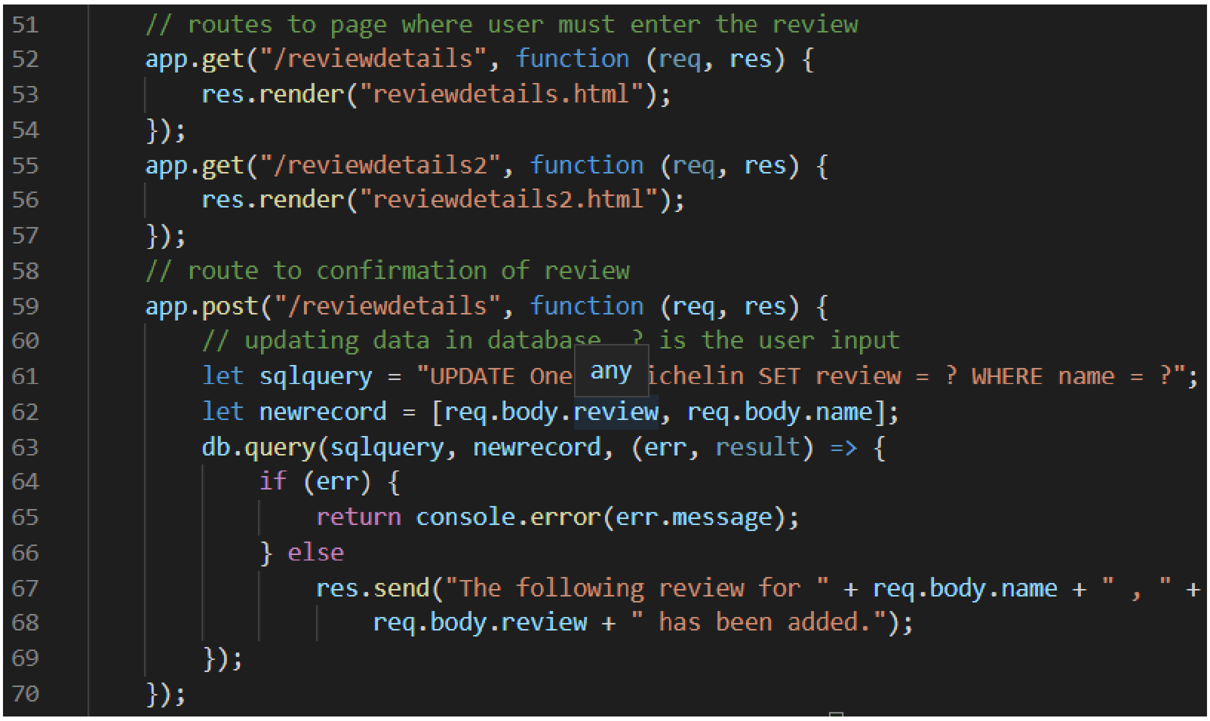
```javascript
// The main.js file
module.exports = function (app) {
    // route to home page
    app.get("/", function (req, res) {
        res.render("index.html")
    });
    //route to show list of one star michelin restaurants
    app.get("/list", function (req, res) {
        // query database to get all restaurants from OneStarMichelin
        let sqlquery = "SELECT * FROM OneStarMichelin";
        // execute sql query
        db.query(sqlquery, (err, result) => {
            if (err) {
                res.redirect("/");
            }
            res.render("list.html", { availableRestaurants: result });
        });
    });
    //route to show list of three stars michelin restaurants
    app.get("/list2", function (req, res) {
        // query database to get all restaurants from ThreeStarsMichelin
        let sqlquery = "SELECT * FROM ThreeStarsMichelin";
        db.query(sqlquery, (err, result) => {
            if (err) {
                res.redirect("/");
            }
            res.render("list2.html", { availableRestaurants: result });
        });
    });
```

Image 4.12: main.js



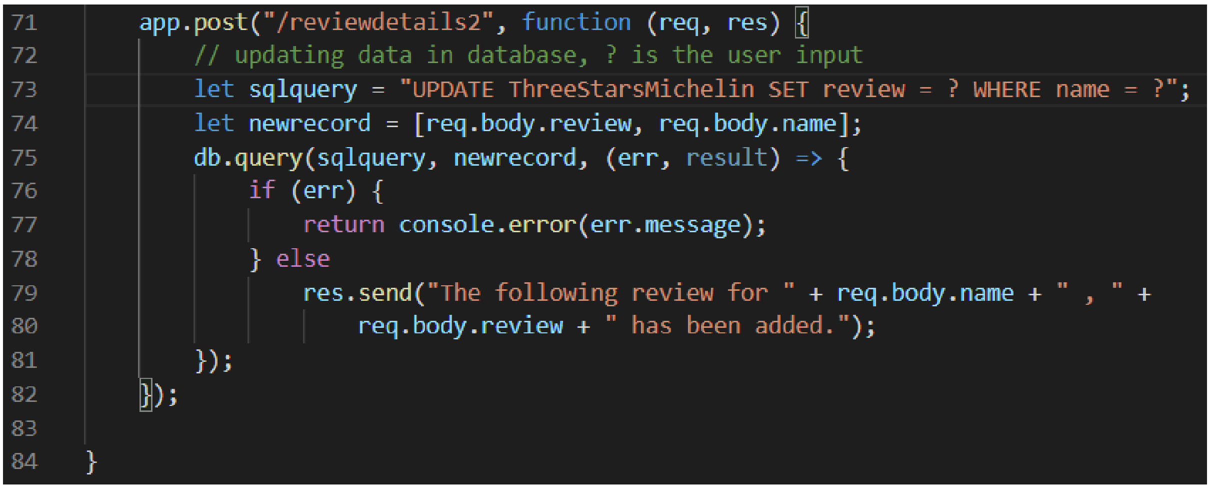
```javascript
// route to addreview.html
app.get("/update", function (req, res) {
    let sqlquery = "SELECT * FROM OneStarMichelin";
    db.query(sqlquery, (err, result) => {
        if (err) {
            res.redirect("/");
        }
        res.render("addreview.html", { availableRestaurants: result });
    });
});
// route to addreview2.html
app.get("/update2", function (req, res) {
    let sqlquery = "SELECT * FROM ThreeStarsMichelin";
    // execute sql query
    db.query(sqlquery, (err, result) => {
        if (err) {
            res.redirect("/");
        }
        res.render("addreview2.html", { availableRestaurants: result });
    });
});
```

Image 4.13: main.js



```javascript
// routes to page where user must enter the review
app.get("/reviewdetails", function (req, res) {
    res.render("reviewdetails.html");
});
app.get("/reviewdetails2", function (req, res) {
    res.render("reviewdetails2.html");
});
// route to confirmation of review
app.post("/reviewdetails", function (req, res) {
    // updating data in database ? is the user input
    let sqlquery = "UPDATE One any ichelin SET review = ? WHERE name = ?";
    let newrecord = [req.body.review, req.body.name];
    db.query(sqlquery, newrecord, (err, result) => {
        if (err) {
            return console.error(err.message);
        } else
            res.send("The following review for " + req.body.name + " , " +
                req.body.review + " has been added.");
    });
});
```

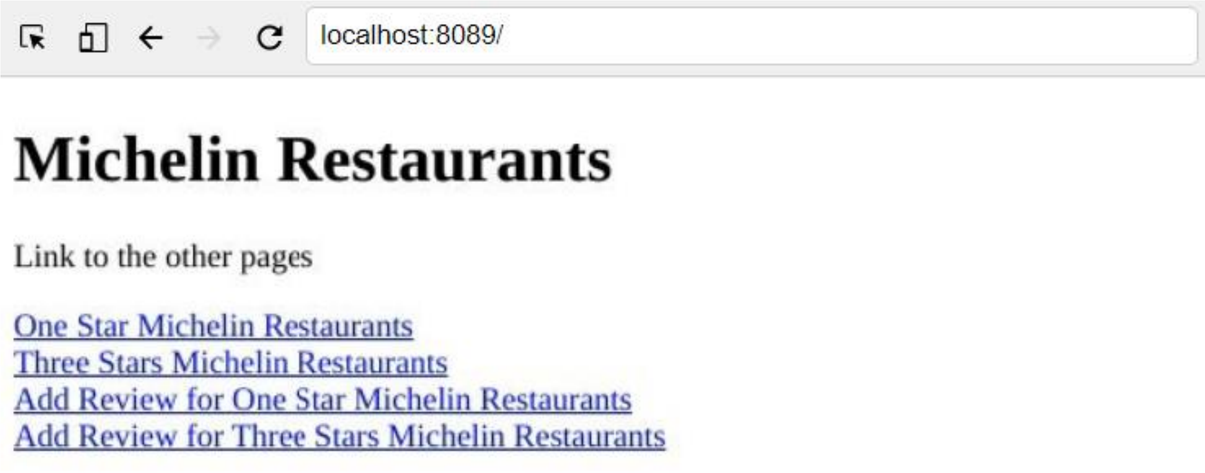
Image 4.14: main.js



```
    app.post("/reviewdetails2", function (req, res) {
        // updating data in database, ? is the user input
        let sqlquery = "UPDATE ThreeStarsMichelin SET review = ? WHERE name = ?";
        let newrecord = [req.body.review, req.body.name];
        db.query(sqlquery, newrecord, (err, result) => {
            if (err) {
                return console.error(err.message);
            } else
                res.send("The following review for " + req.body.name + " , " +
                    req.body.review + " has been added.");
        });
    });

}
```

Image 4.15: main.js

From Image 4.16 to 4.23, are the displays of the web application. As we can see from Image 4.17 and 4.18, there are not yet any reviews for the restaurants. However, in Image 4.21, I have filled up the form to enter a review and how this works is that it uses the name of the restaurant to update the review attribute of the respective table in the database. Once the user has clicked the submit button, there will be a message to show that it is successful in Image 4.22 and the review will be showed at the back of the line as shown in Image 4.23.



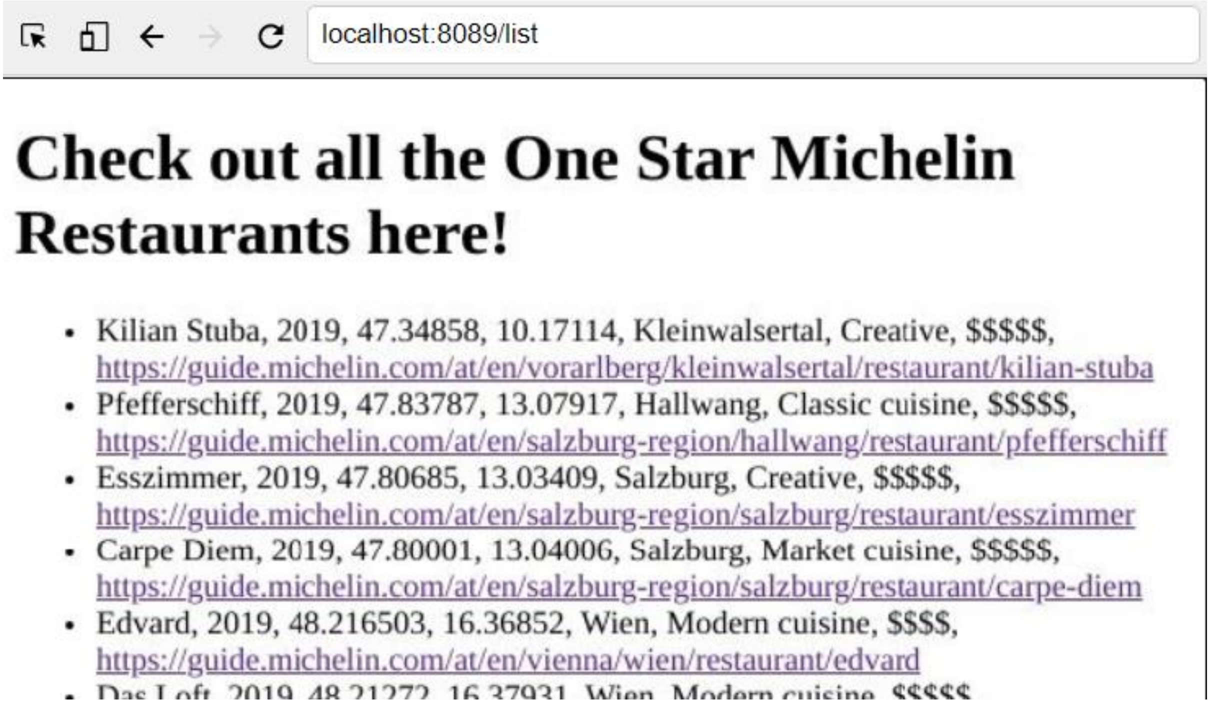
localhost:8089/

Michelin Restaurants

Link to the other pages

One Star Michelin Restaurants
Three Stars Michelin Restaurants
Add Review for One Star Michelin Restaurants
Add Review for Three Stars Michelin Restaurants

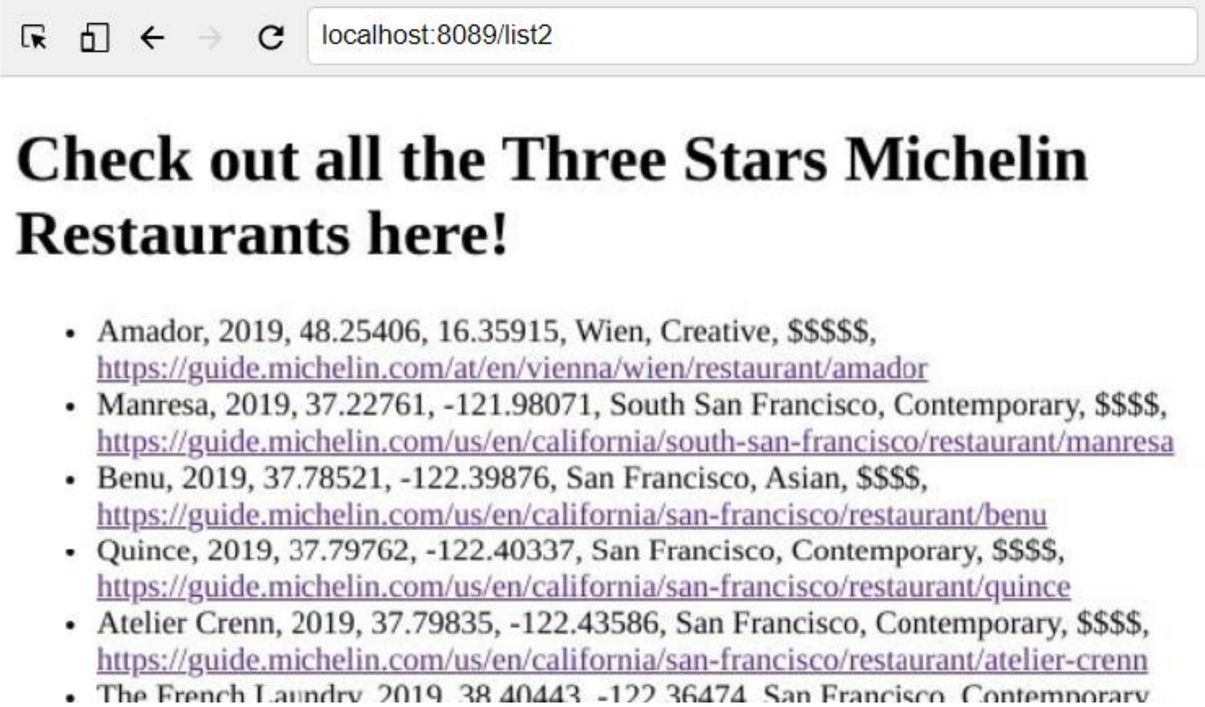
Image 4.16: Web Application



Check out all the One Star Michelin Restaurants here!

- Kilian Stuba, 2019, 47.34858, 10.17114, Kleinwalsertal, Creative, $$$$$, https://guide.michelin.com/at/en/vorarlberg/kleinwalsertal/restaurant/kilian-stuba
- Pfefferschiff, 2019, 47.83787, 13.07917, Hallwang, Classic cuisine, $$$$$, https://guide.michelin.com/at/en/salzburg-region/hallwang/restaurant/pfefferschiff
- Esszimmer, 2019, 47.80685, 13.03409, Salzburg, Creative, $$$$$, https://guide.michelin.com/at/en/salzburg-region/salzburg/restaurant/esszimmer
- Carpe Diem, 2019, 47.80001, 13.04006, Salzburg, Market cuisine, $$$$$, https://guide.michelin.com/at/en/salzburg-region/salzburg/restaurant/carpe-diem
- Edvard, 2019, 48.216503, 16.36852, Wien, Modern cuisine, $$$$, https://guide.michelin.com/at/en/vienna/wien/restaurant/edvard
- Das Loft, 2019, 48.21272, 16.37931, Wien, Modern cuisine, $$$$$,

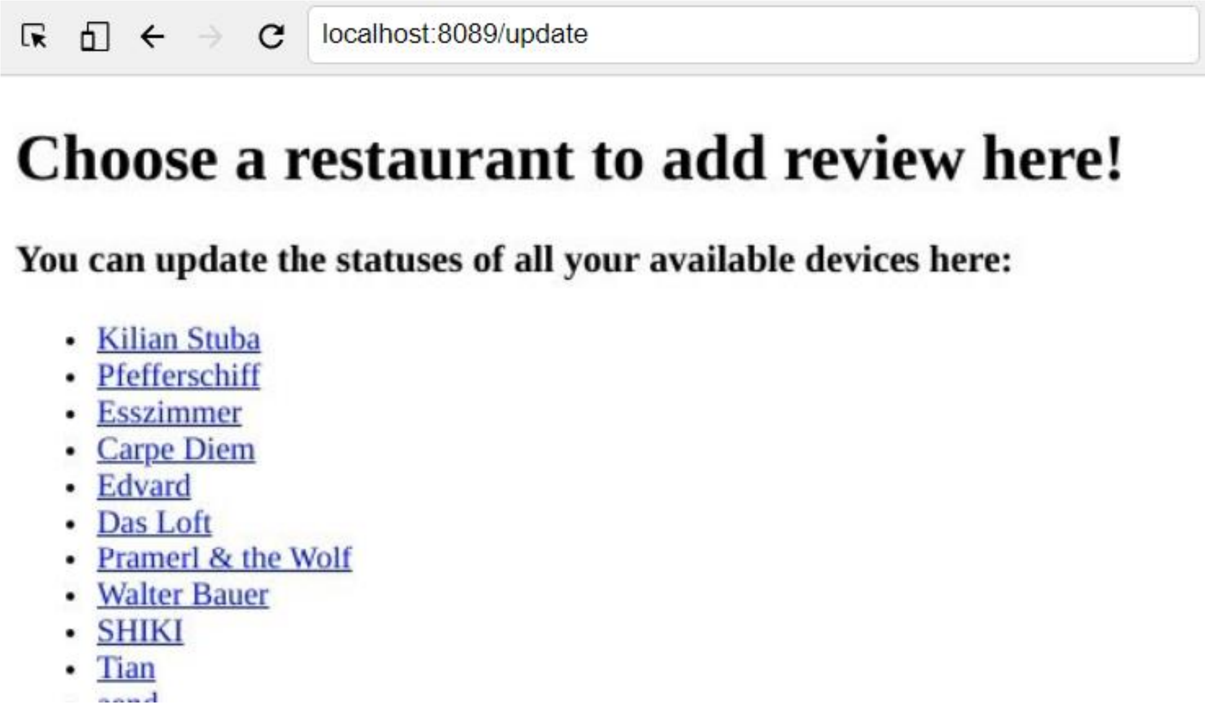
Image 4.17: Web Application



Check out all the Three Stars Michelin Restaurants here!

- Amador, 2019, 48.25406, 16.35915, Wien, Creative, $$$$$, https://guide.michelin.com/at/en/vienna/wien/restaurant/amador
- Manresa, 2019, 37.22761, -121.98071, South San Francisco, Contemporary, $$$$, https://guide.michelin.com/us/en/california/south-san-francisco/restaurant/manresa
- Benu, 2019, 37.78521, -122.39876, San Francisco, Asian, $$$$, https://guide.michelin.com/us/en/california/san-francisco/restaurant/benu
- Quince, 2019, 37.79762, -122.40337, San Francisco, Contemporary, $$$$, https://guide.michelin.com/us/en/california/san-francisco/restaurant/quince
- Atelier Crenn, 2019, 37.79835, -122.43586, San Francisco, Contemporary, $$$$, https://guide.michelin.com/us/en/california/san-francisco/restaurant/atelier-crenn
- The French Laundry, 2019, 38.40443, -122.36474, San Francisco, Contemporary,

Image 4.18: Web Application



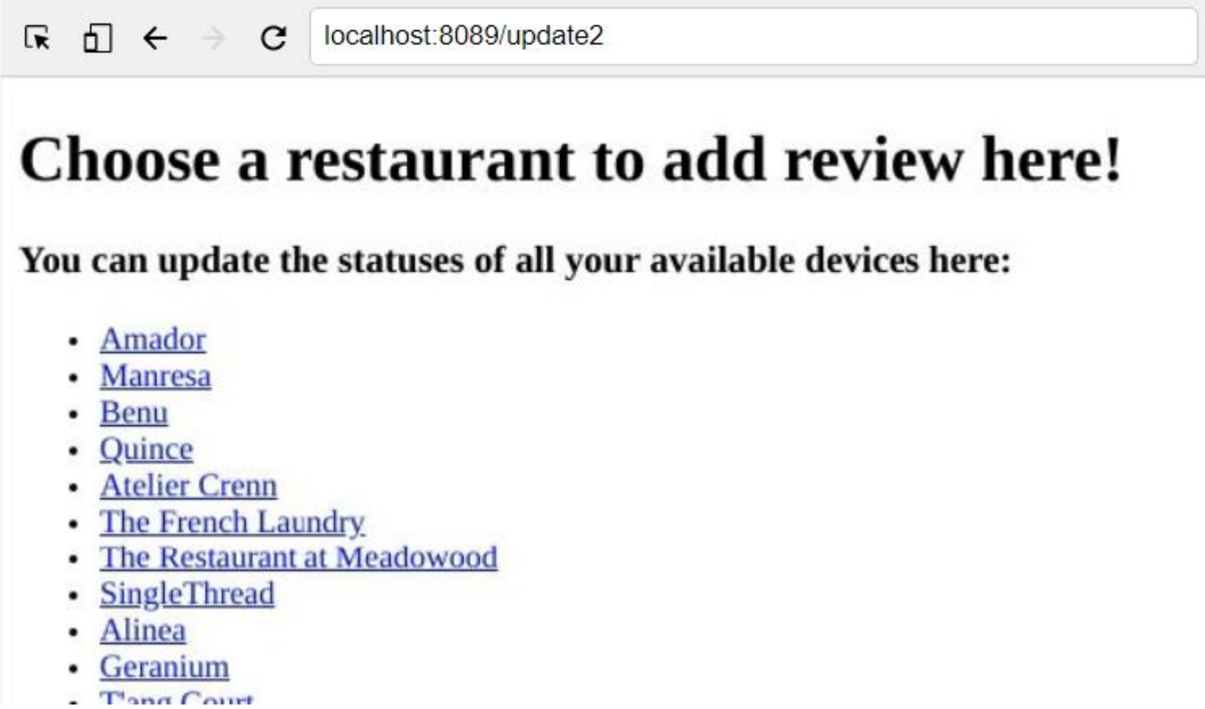
localhost:8089/update

Choose a restaurant to add review here!

You can update the statuses of all your available devices here:

- Kilian Stuba
- Pfefferschiff
- Esszimmer
- Carpe Diem
- Edvard
- Das Loft
- Pramerl & the Wolf
- Walter Bauer
- SHIKI
- Tian

Image 4.19: Web Application



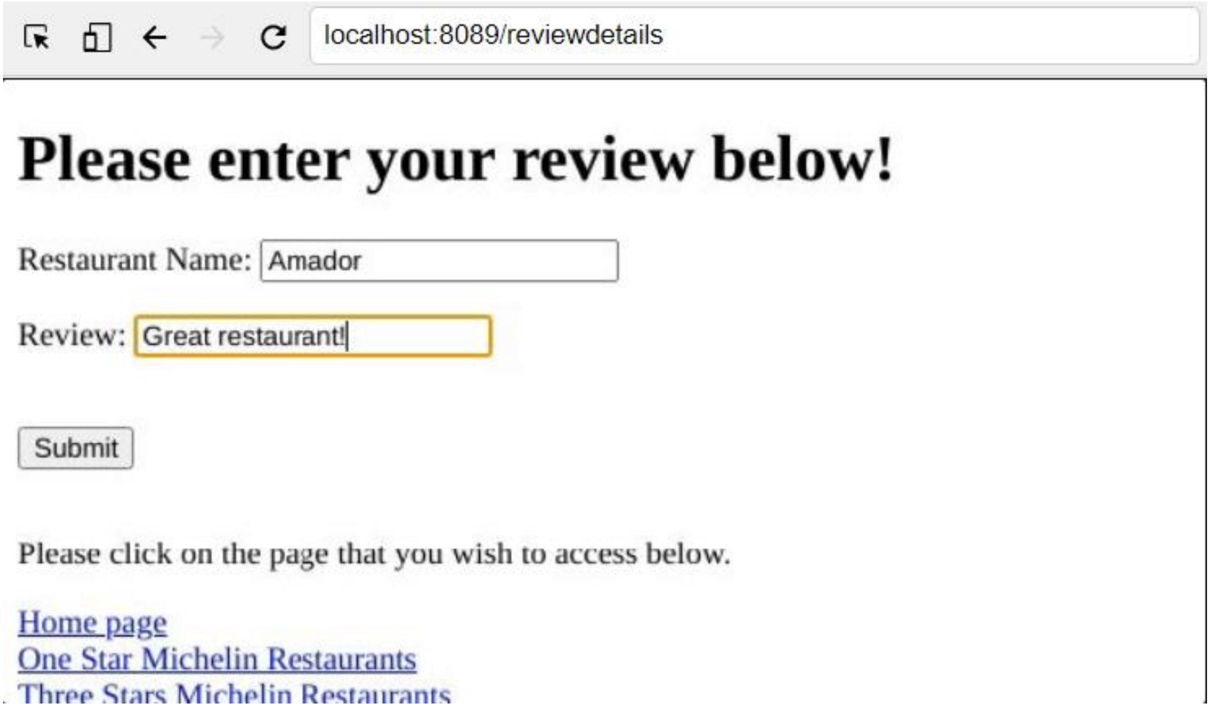
localhost:8089/update2

Choose a restaurant to add review here!

You can update the statuses of all your available devices here:

- Amador
- Manresa
- Benu
- Quince
- Atelier Crenn
- The French Laundry
- The Restaurant at Meadowood
- SingleThread
- Alinea
- Geranium
- Tang Court

Image 4.20: Web Application


localhost:8089/reviewdetails

Please enter your review below!

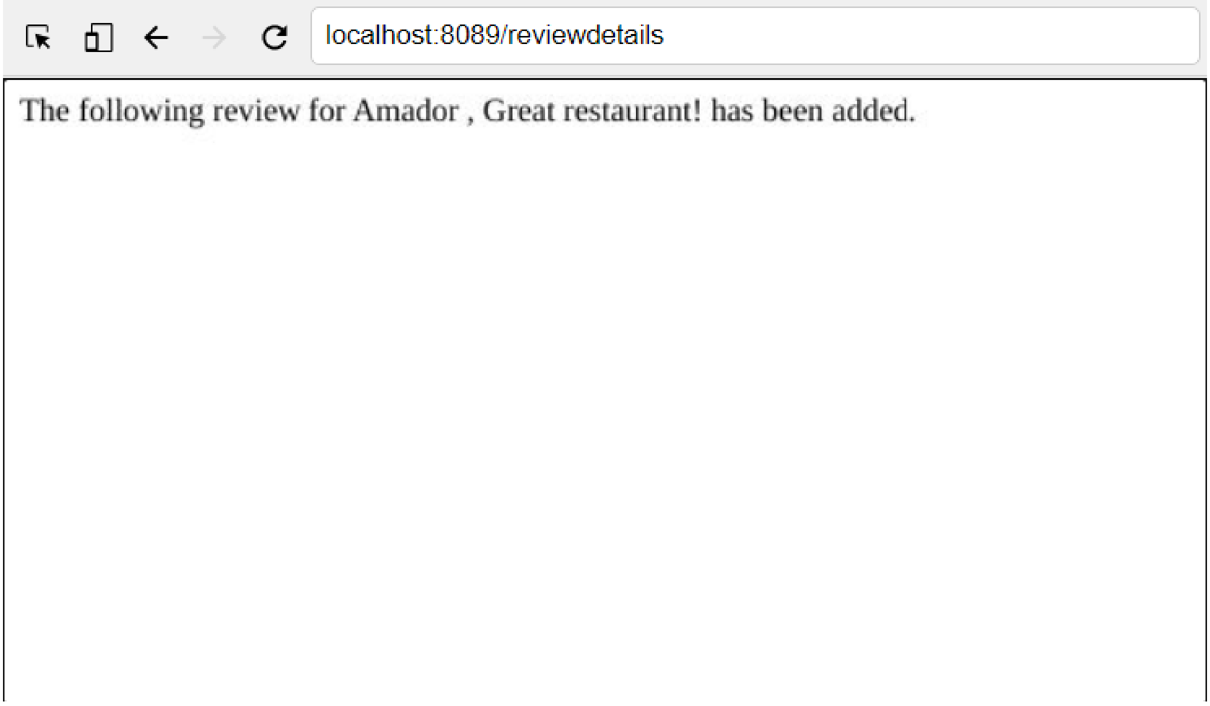
Restaurant Name: Amador

Review: Great restaurant!

Please click on the page that you wish to access below.

Home page
One Star Michelin Restaurants
Three Stars Michelin Restaurants

Image 4.21: Web Application


localhost:8089/reviewdetails

The following review for Amador , Great restaurant! has been added.

Image 4.22: Web Application

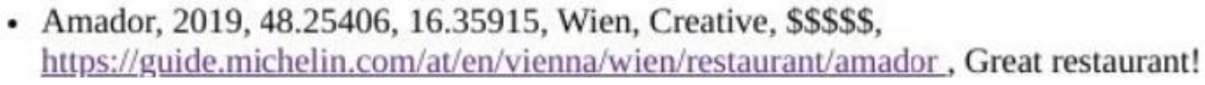

- Amador, 2019, 48.25406, 16.35915, Wien, Creative, $$$$$, https://guide.michelin.com/at/en/vienna/wien/restaurant/amador , Great restaurant!

Image 4.23: Web Application

Although this method works, the flaw is that it can only contain one review at a time as there cannot be two values in a single attribute of a single row. If another user overwrites the previous review, the previous comment will be gone forever. To solve this issue, I believe we can create an individual page for each restaurant instead of just one web page with a list of the entire table and add a comment section that does not interfere with the tables OneStarMichelin and ThreeStarsMichelin.

The URL to my Coursera Lab is,
https://hub.labs.coursera.org:443/connect/sharedvwbdrlkn?forceRefresh=false&path=%2F%3Ffolder%3D%2Fhome%2Fcoder%2Fproject