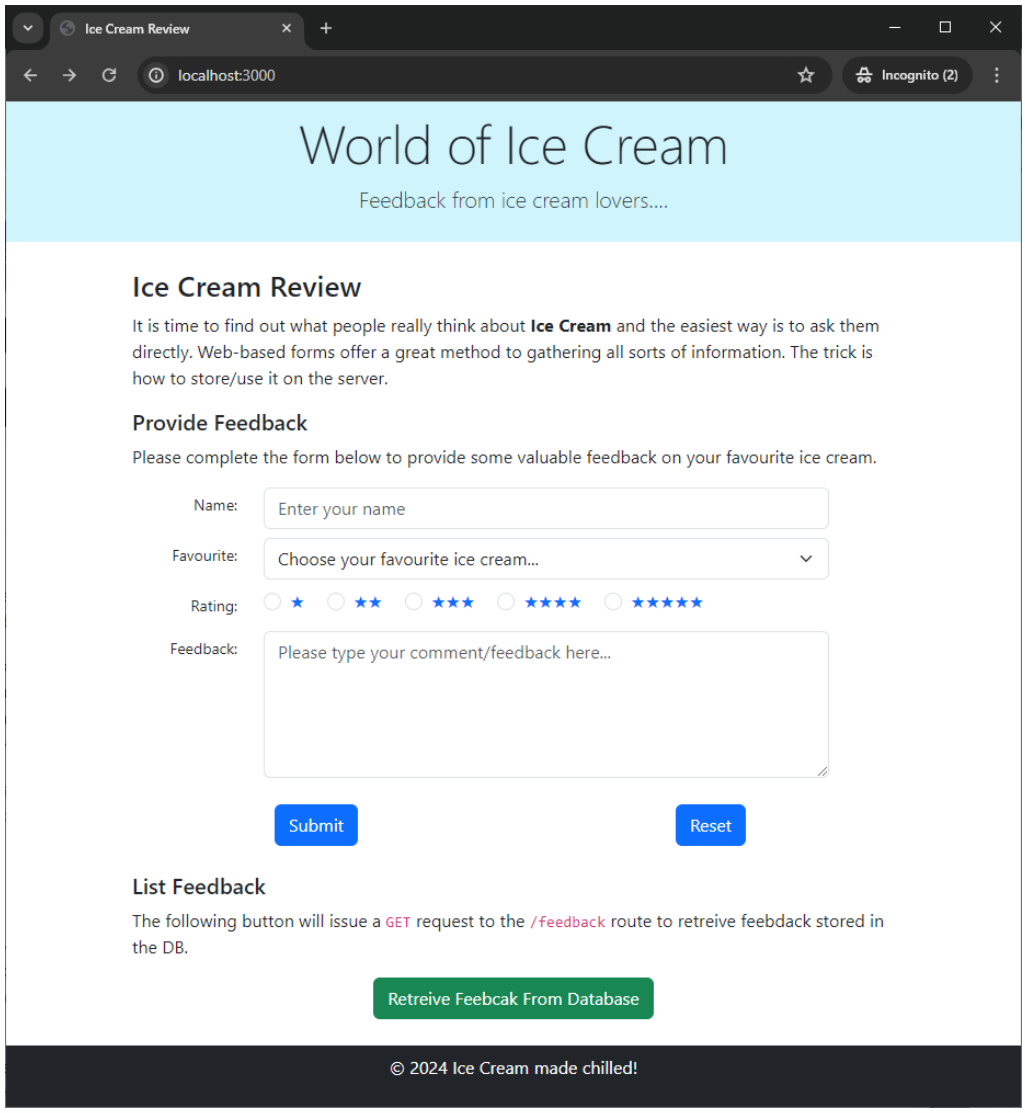
10.1P: Server Database for Website Project

Task

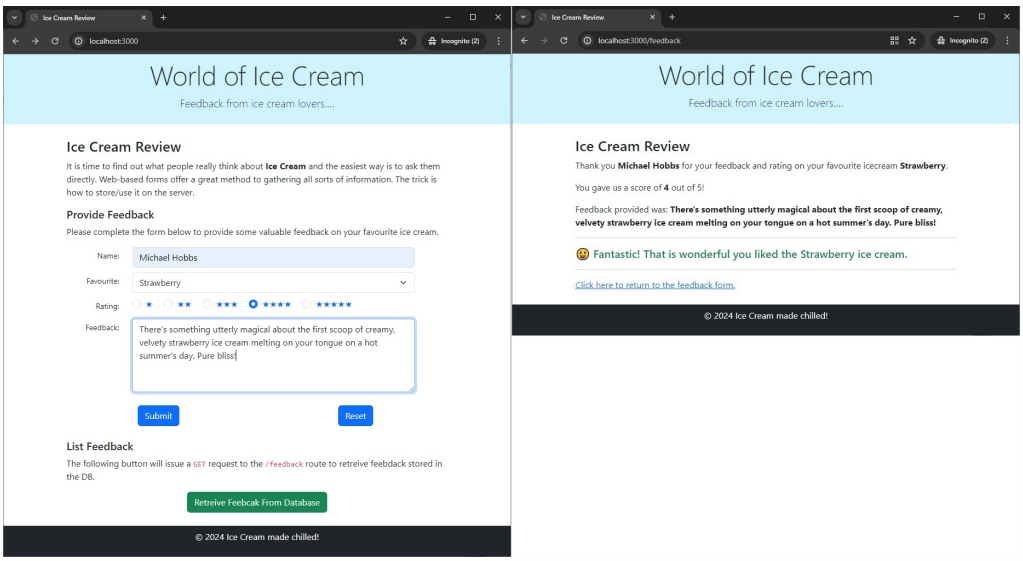
In this task, you are required to extend your web server from **Task 9-1P** to allow for the feedback to be stored in a DB for retrieval even if the web server has to be restarted.

In addition to the storing of the feedback, your web page should also provide a method to retrieve the contents of the database through the addition of another button. A sample screenshot of what your main page should look like is shown below:



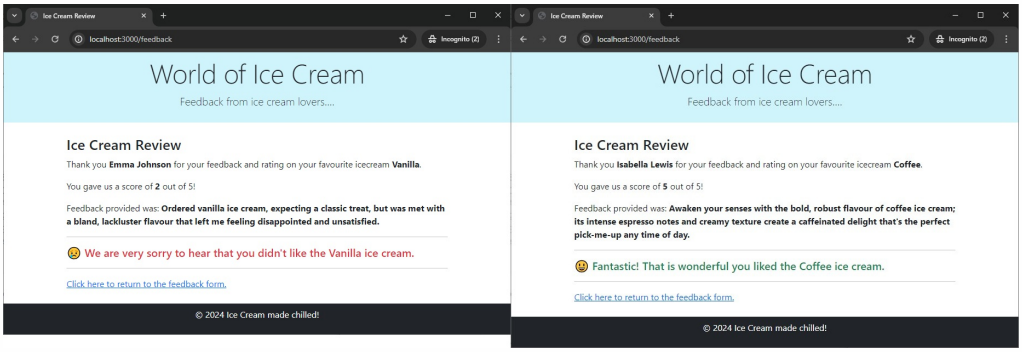
Task10.1.1 Modified form page

A sample of a feedback form entry with the response page is shown below:



Task10.1.2 Form feedback completed and response page

While samples of two additional feedback submissions is also shown below:



Task10.1.3 More form feedback completed and response page

You will need to create a web server that implements a simple *Comments Server*, with an interface to a simple database (file) and accepts requests from a user (client browser). These requests include a **POST** to store a *new* record into the database and a **GET** to display the current/updated contents of the database.

The database structure (schema) should support a record with the following fields:

Name	Type
id	INTEGER PRIMARY KEY AUTOINCREMENT
name	TEXT
icecreamtype	TEXT
rating	INT
feedback	TEXT

Steps

To complete this task, you are required to:

1. Make a standalone *Node.js* program (i.e., a `createDB.js` file) and execute it on the *Node.js* console to create and initialise a server side *SQlite3* file database

in the server folder. The database contains a table that is used to store the data sent from a form page of your own website (as described in the table above).

2. Make another *Node.js* program (i.e., a `index.js` file) in your *Node.js* server folder. This program is able to launch the server, accept the data sent from a form page of your website (in a **POST** request on the route `/feedback`), save the received data into the database table, and display the table data upon the request from the client (i.e., a **GET** request ont the route `/feedback`).

3. Make necessary changes to a form page of your website (can be stored in the template file `index.ejs`), so that it is able to:

    - send the form data to the server using a **POST** message linked to a **Submit** button
    - send a data retrieval request (e.g., a **GET** request message linked to a **Retrieve Feedback Form Database** button) to the server

4. Launch the server by executing your *Node.js* program you made.
5. Visit the form web page of your website via the local *Node.js* server (e.g., `http://localhost:3000/` which will retrieve the `index.html` page) using a web browser.
6. Enter data into the form and submit the form to the server (three or more times, by clicking a **Save User** button). The data should be saved in the server database table.
7. Within the form page, send a data retrieval request (e.g., by clicking a **Get Feedback Form Database** button) to the server. After receiving the request, the server retrieves data from the database table and displays the retrieved data in the browser.

Hints

The code snippet for the **Get Feedback Form Database** button could be:

```
<h5 class="mt-4">List Feedback</h5>
<p>
    The following button will issue a <code>GET</code> request to the <code>/feedback</code>
    route to retrieve feedback stored in the DB.
</p>
<div class="d-grid gap-5d-md-flex justify-content-md-center   mb-4">
    <a href="/feedback">
        <button class="btn btn btn-success" id="getfeedbackBtn">
            Retrieve Feedback From Database
        </button>
    </a>
</div>
```

This task is similar to the example provided in the unit site. You will note the structure of the database table will be different (slightly) and there will need to be a change to the form inputs to match the 'fields' expected in the database.

What will you submit?

You should submit:

- Source code of the template web page of your main page *form* (i.e., the `index.ejs` file)
- Source code of the template file that renders the contents of the database into a table (i.e., the `feedback.ejs` file)
- Source code of the *Node.js* file (i.e., the first `createDB.js` file) that creates a server **file** database with a table in it.
- Source code of the *Node.js* server program file (i.e., the second `index.js` file).
- Screenshot of the browser window showing the form web page with entered data.
- Screenshot of the browser window showing the retrieved data from the database table after the data retrieval request is sent to the server.