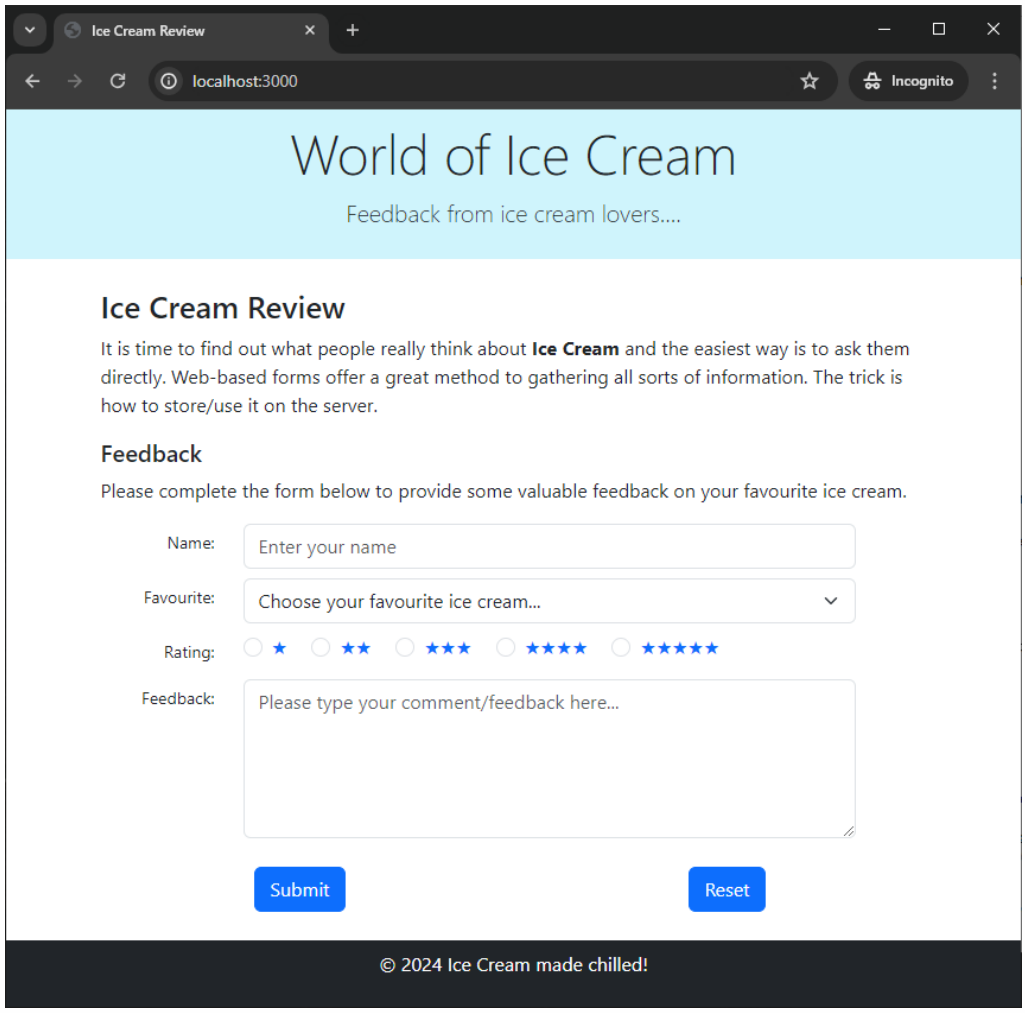
9.1P: Send a Response to the Received Data

Tasks

In previous weeks we have looked at how we can handle `GET` requests in an express server. Here, we will look at how to handle `POST` requests, as generated when submitting a web form. You are supplied with the code that presents the following web page form for accepting some information regarding a user's favourite ice cream, including their name, a rating out of 5 and some feedback:

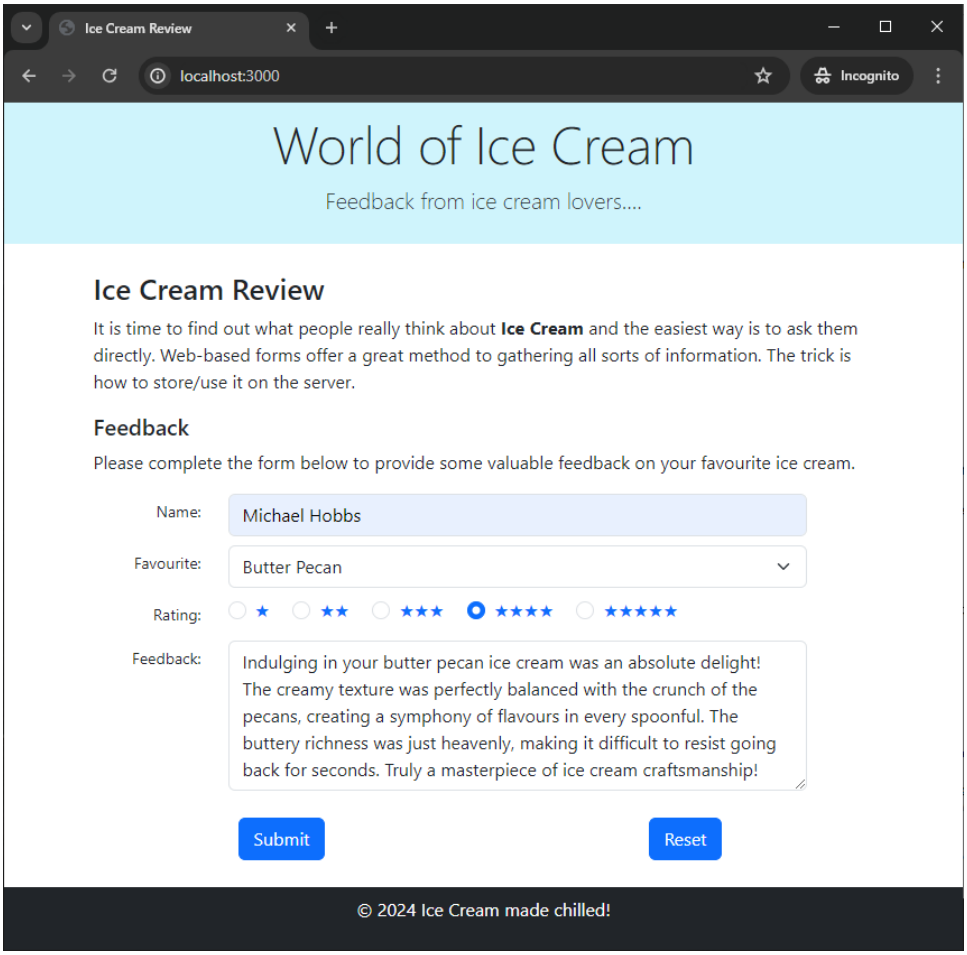


Task9.1.1 Example form layout

In this task, your web page should use **Encapsulated JS** templates for the *main form page* and for the *thank you* response page. The template files for the main form page (broken up into `header.ejs`, `index.ejs`, `footer.ejs` sections) is provided on the Ontrack website, in the task 9.1P page through the *Resources* link found in the bottom right-hand corner of the page.

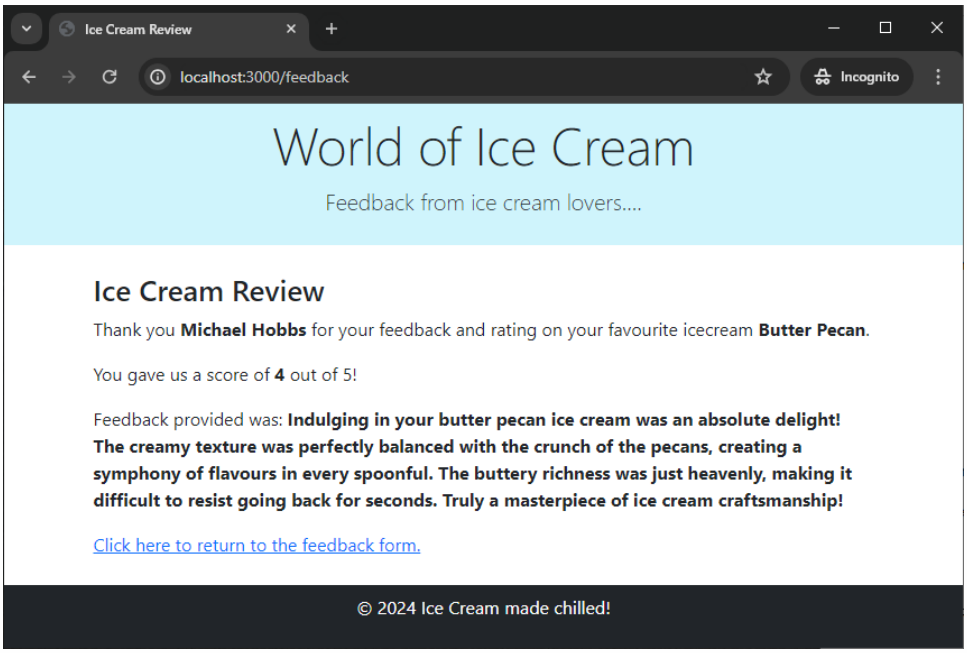
You are required to:

1. Save the provided EJS template files into a new `views` folder in your project directory (following the example provided in the Week 08 content page 8.10 Using templates).
2. Make a *Node.js* application program (i.e., an `index.js` file) in your *Node.js* server folder. Add a `GET` handler for the default `/` route to render the `index.ejs` template with an appropriate `title` parameter.
3. Create a new `thankyou.ejs` template file (that can also *include* the `header.ejs` and `footer.ejs` templates to maintain consistency in presentation). The template should include fields that can be populated by arguments it is provided for the *name*, *favourite ice cream*, *rating* and *feedback*.
4. Add a new `POST` handler to accept data from the **form page**. The route address can be found in the form's `action` value. The request handler should:
    - extract the forms fields (from the *body* of the *request*)
    - render a `thankyou` template for the response page. The form's fields can be passed as parameters to the template to populate the *response* message.
5. Visit the form web page via the local *Node.js* server (e.g., `http://localhost:3000/`) using a web browser; which should render the template `index.ejs` file **NOT** a static file.
6. Enter some data into the form, like the following:



Task9.1.2 Example form with input

7. When the "**Submit**" button is clicked, then a response page (message) generated from the `thankyou.ejs` template and should be displayed in the browser, like the following:



Task9.1.3 Response from the server

What will you submit?

You should submit:

- Source code of the *Node.js* file (i.e., the `index.js` file).
- Source code of the *template* file for your **Thank You** page (i.e., the `thankyou.ejs` file).
- Screenshot of the browser window showing the form web page with entered data.
- Screenshot of the browser window showing response message after the "**Submit**" button is clicked.