Assignment 3

Due Date:

You must demonstrate your program during class time. (It is an individual assignment)

	Date	***Time***
Softcopy	27-March-2022(Sunday)	11:59PM
Demo	Week 14 (11-14 April 2022)	During class time
Demo cut-off date	23-April-2022 (Friday)	5:00PM

Students are expected to demo their projects during specified demo time which takes place during regular class. In special circumstances, if you miss the demo during class time, please make an appointment and finish your demo before the cut-off date mentioned above. Failing to do so will cost you demo marks specified in the marking sheet (the last page of this document)

Topic:

Incorporate Database, Tables and all CRUD operations. This iteration will complete the project.

Before you begin:

- You must complete Assignment 2 before proceeding.
- You must know how to search and use the online jQuery Mobile Docs to enhance your mobile app.
- Make yourself familiarize with the hands on done in the lab during class time.
- The assignment specification and screenshots are made assuming a student's name Jason Bourne (and initial is JB). Replace Jason Bourne with your full name, JB with your own initial. Also replace xx with your own initials.
- Make sure to fulfill all programming and assignment standards. The Standards Summary is available at eConestoga.

Study material:

- Please complete AvengersDB-Iteration 3 hands on program before proceeding with this assignment.
- Please check the associated video to better understand the flow of execution of Assignment 3

Problem Specification:

Task 1: Create Assignment 3 project

- Create an empty project and name it: XXFeedbackA3 (where XX is your initials).
- Copy all the files from your assignment 2 project.

Task 1: Create database and tables

- Check "Introduction to Foreign Key Constraints" section of https://www.sqlite.org/foreignkeys.html for more info about foreign key. Also: [SQL FOREIGN KEY Constraint (w3schools.com)]()
- In the xxdatabase.js create your database for this application in a function named createDatabase( ). (Using global object is recommended) :
    - Create 2 tables: 'type' and 'review' in a function named createTables()
    - Read the next step before creating these tables.

```
"CREATE TABLE IF NOT EXISTS type( "
                + "id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,"
                + "name VARCHAR(20) NOT NULL);";

"CREATE TABLE IF NOT EXISTS review( " +
                "id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT," +
                "businessName VARCHAR(30) NOT NULL," +
                "typeId INTEGER NOT NULL," +
                "reviewerEmail VARCHAR(30)," +
                "reviewerComments TEXT," +
                "reviewDate DATE," +
                "hasRating VARCHAR(1)," +
                "rating1 INTEGER," +
                "rating2 INTEGER," +
                "rating3 INTEGER," +
                "FOREIGN KEY(typeId) REFERENCES type(id));";
```

    - Note: very important: Table 'type' is a lookup. Once the table is created you must initialize it by inserting 5 records with 'Others, 'Canadian', 'Asian', 'European' and 'Australian'. The order is not important (primary key of each of the records may be different based on the order you use). This table will be linked to the dropdown 'Business Type' in both add and modify pages.
    - This initialization will take place when the program starts.
    - After that, the review table will be empty, and type table will have 5 records. If you do not take precaution, every time you reload the program, it will insert 5 more records resulting in duplicates.
    - By following the following sequence while <u>every time</u> loading your application you may avoid this issue:
        - drop 'type' table,
        - recreate it,
        - insert 5 records.
          > Please note that there are other ways, but the process mentioned above is the easiest to implement. See some hints below:

- Hints: in your txFunction() of createTables() function
  1. Declare sql for drop 'type' table, then call executeSql() with necessary params.
  2. Declare sql for create table 'type', then call executeSql() again with necessary params.
  3. Declare sql for insert 1st row to 'type' (Others), then call executeSql() again with necessary params.
  4. Declare sql for insert 2nd row to 'type' (Canadian), then call executeSql() again with necessary params.
  5. Declare sql for insert 3rd row to 'type' (Asian), then call executeSql() with necessary params.
  6. Declare sql for insert 4th row to 'type' (European), then call executeSql() again with necessary params.
  7. Declare sql for insert 5th row to 'type' (Australian), then call executeSql() again with necessary params.
  8. Declare sql for create table 'review', then call executeSql() again with necessary params.
  9. All these steps can be done inside your txFunction() within the createTables() function
  10. Please note, you just need one db.transaction() call and pass the txFunction() to do all operations mentioned above.

Task 2: Drop tables

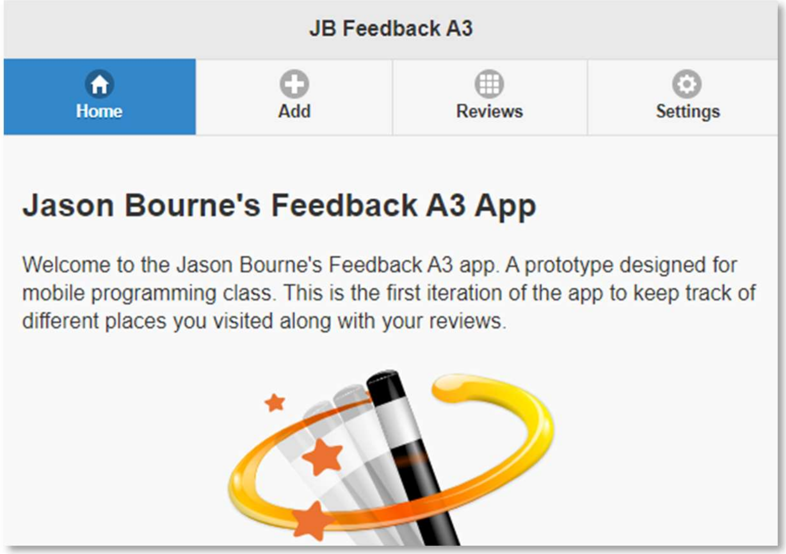
- Make a function named dropTables() that will drop both 'type' and 'review' tables. This function will be called when 'Clear Database' button under 'Settings' page is clicked. Note, both tables can be dropped with a single transaction.
- Look into AvengersDB-Iteration 3 hands on program posted in eConestoga.

Task 3: Include CRUD operations

- In the xxDAL file create CRUD functions for
  - 'review' table: insert, select, selectAll, update, delete
  - 'type' table: selectAll

Task 4: Modify 'Home' page

- Update the headers and home page to indicate Assignment 3. See the screenshot below



Jason Bourne's Feedback A3 App

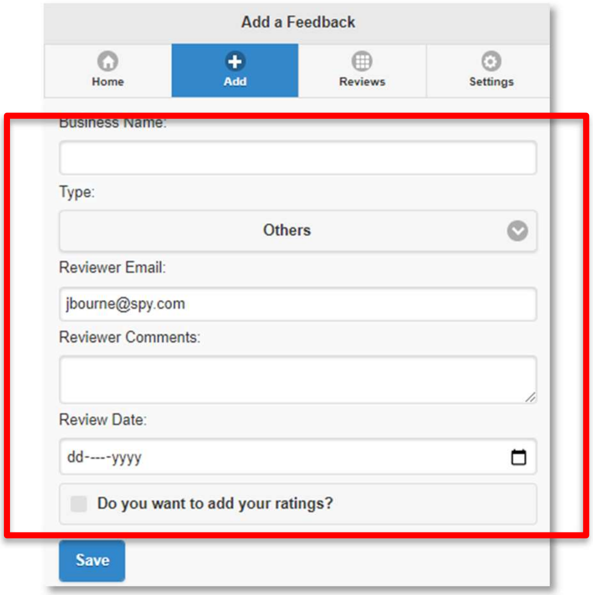
Welcome to the Jason Bourne's Feedback A3 app. A prototype designed for mobile programming class. This is the first iteration of the app to keep track of different places you visited along with your reviews.

Figure 1: Home page

Task 5: Modify 'Add Feedback' page

- In xxglobal.js add an event handler for Add Feedback pages's 'pageshow' event.
- In this event handler function load the default email address from local storage and show it in the Reviewer Email textbox. Hint: You may assume that, after the program runs, your instructor will go to the Settings page and save the default email (this task was done in Assignment 2). The same email is expected to show up on the Add page.
- The 'Type' dropdown will be automatically populated from 'type' table when the page is shown. Make sure the items are not coming from hard coded '<selects>'.
  - Make a function named 'updateTypesDropdown' in xxfacade.js file and execute it when Add page is shown. In this function
    - Call the 'selectAll' CRUD of 'type'
    - Append each item to the "Type" dropdown
    - Make 'Others' as selected by default.
    - HINTS: Consider using id of type as the value of each <option> element. It will be helpful while adding a new record to 'review' table.
- Clicking the 'Save' button, the record will be inserted to 'review' table after validating.
  - Make a function in xxfacade.js file named addFeedback(). This function will do the validation and call the insert CRUD for 'review' (defined in Task 2) while passing the inputs.
  - Add a click event handler in xxglobal.js for 'Save' button and call the addFeedback() function.

  - Make sure to include ‘Food quality’, ‘Service’ and ‘Value’ when ‘Do you want to add your ratings’ checkbox is checked. If it is unchecked, ‘Food quality’, ‘Service’ and ‘Value will be 0. Check the screenshots below for clarification.
  - Anytime if ‘Do you want to add your ratings’ checkbox is unchecked the ratings block will hide. When the checkbox is checked it will show the ratings block with 0 for all fields. This applies for Add and Modify pages. Please check the video.
- Any time if we visit Add a Feedback page from any other page (hint: ‘pageshow’ event) the following initialization will take place to the Add form:
  - Business Name will be empty
  - Type will be ‘Others’
  - Reviewer email will be filled with the defaultEmail from local storage.
  - Reviewer Comments will be empty.
  - Review date will be empty
  - Do you want to add your ratings checkbox will be unchecked and ‘Food quality’, ‘Service’, ‘Value’, and ‘Overall Rating’ will be hidden.



Business Name:

Type: Others

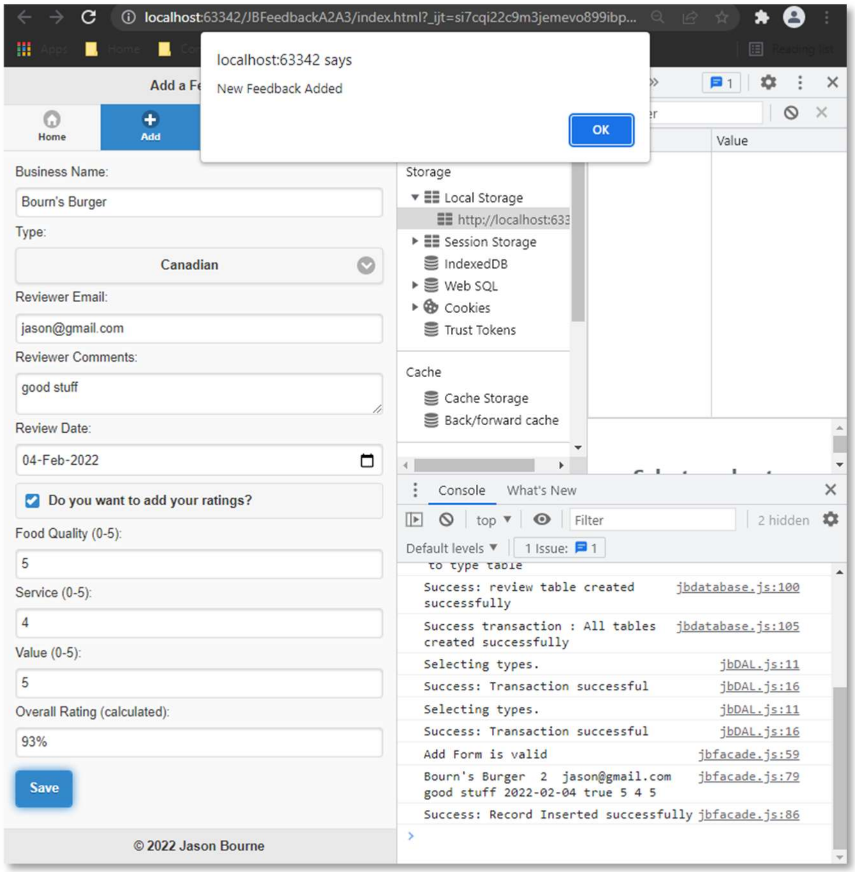
Reviewer Email: jbourne@spy.com

Reviewer Comments:

Review Date: dd----yyyy

□ Do you want to add your ratings?

Figure 2: Initial look of Add a feedback page



localhost:63342/JBFeedbackA2A3/index.html?_ijt=si7cqi22c9m3jemevo899ibp...

localhost:63342 says

New Feedback Added

Business Name: Bourn's Burger

Type: Canadian

Reviewer Email: jason@gmail.com

Reviewer Comments: good stuff

Review Date: 04-Feb-2022

☑ Do you want to add your ratings?

Food Quality (0-5): 5

Service (0-5): 4

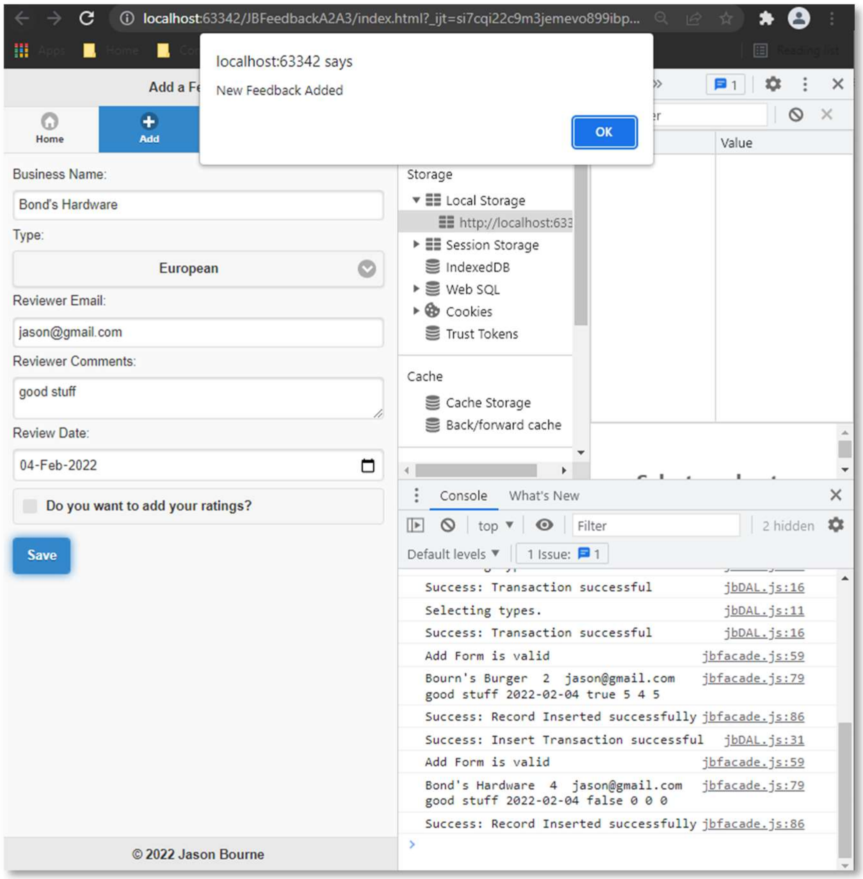
Value (0-5): 5

Overall Rating (calculated): 93%

© 2022 Jason Bourne

```
  to type table
Success: review table created       jbdatabase.js:100
successfully
Success transaction : All tables    jbdatabase.js:105
created successfully
Selecting types.                          jbDAL.js:11
Success: Transaction successful           jbDAL.js:16
Selecting types.                          jbDAL.js:11
Success: Transaction successful           jbDAL.js:16
Add Form is valid                      jbfacade.js:59
Bourn's Burger  2  jason@gmail.com     jbfacade.js:79
good stuff 2022-02-04 true 5 4 5
Success: Record Inserted successfully jbfacade.js:86
```

Figure 3: Save clicked with ratings and alert shown



localhost:63342/JBFeedbackA2A3/index.html?_ijt=si7cqi22c9m3jemevo899ibp...

localhost:63342 says

New Feedback Added

OK

Business Name: Bond's Hardware

Type: European

Reviewer Email: jason@gmail.com

Reviewer Comments: good stuff

Review Date: 04-Feb-2022

□ Do you want to add your ratings?

Save

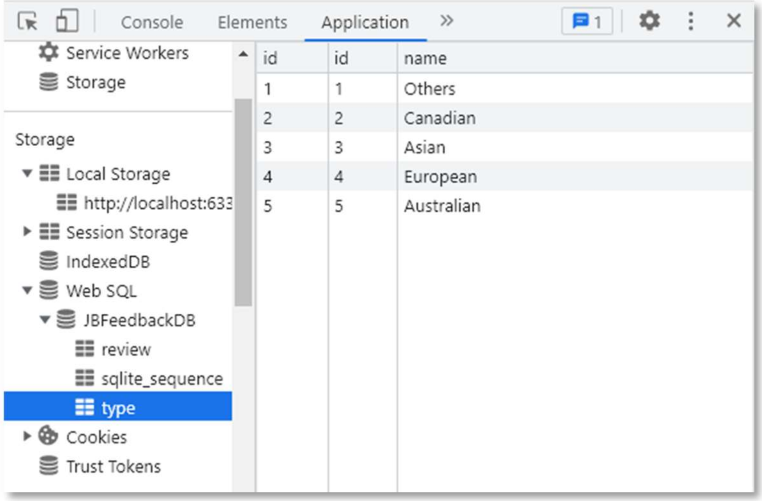
© 2022 Jason Bourne

```
Success: Transaction successful              jbDAL.js:16
Selecting types.                             jbDAL.js:11
Success: Transaction successful              jbDAL.js:16
Add Form is valid                         jbfacade.js:59
Bourn's Burger  2  jason@gmail.com        jbfacade.js:79
good stuff 2022-02-04 true 5 4 5
Success: Record Inserted successfully jbfacade.js:86
Success: Insert Transaction successful       jbDAL.js:31
Add Form is valid                         jbfacade.js:59
Bond's Hardware  4  jason@gmail.com       jbfacade.js:79
good stuff 2022-02-04 false 0 0 0
Success: Record Inserted successfully jbfacade.js:86
```

Figure 4: Save clicked without ratings. Alert shown

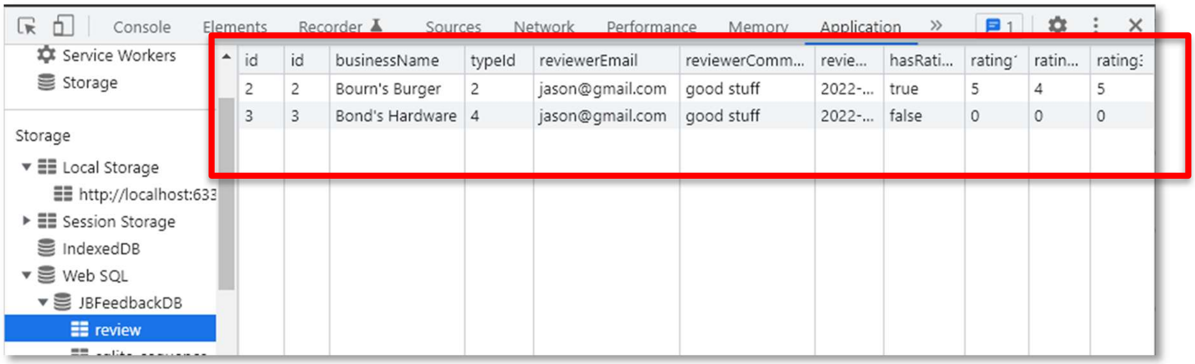
- Make sure to get the id (from type table) (can be obtained from 'value' of the dropdown items, see HINTS previously mentioned) of the selected item from the dropdown while assigning to the typeId field of review table. For example, In the following screenshot, for the 1st record,
  - id: 2
  - businessName: Bourn's Burger
  - typeId: 2
    - this value is acting as a foreign key and in relational database it links to the record of 'type' designated by this id.
    - Your id and name fields of 'type' table may be different based on the order you used to insert the 5 records. In this example 2 is representing 'Canadian'

- Please check the image below:



id	id	name
1	1	Others
2	2	Canadian
3	3	Asian
4	4	European
5	5	Australian

Figure 5: type table

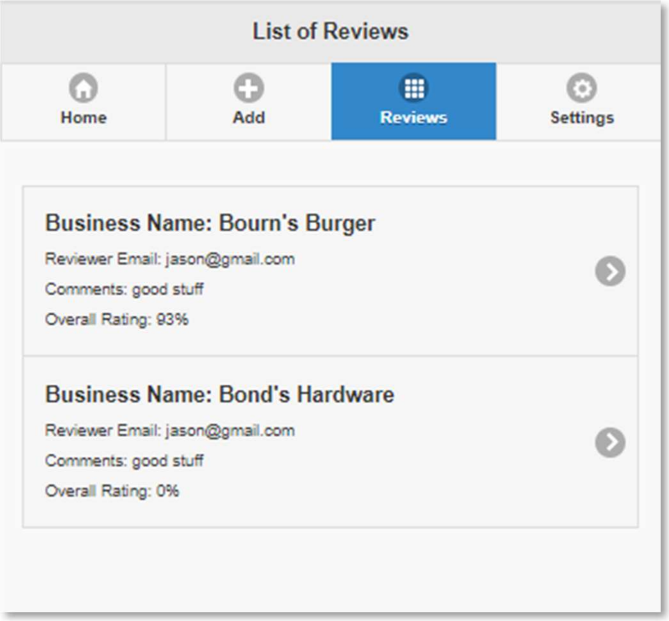


id	id	businessName	typeId	reviewerEmail	reviewerComm...	revie...	hasRati...	rating	ratin...	rating
2	2	Bourn's Burger	2	jason@gmail.com	good stuff	2022-...	true	5	4	5
3	3	Bond's Hardware	4	jason@gmail.com	good stuff	2022-...	false	0	0	0

Figure 6: review table containing 2 records

Task 6: 'Reviews' Page

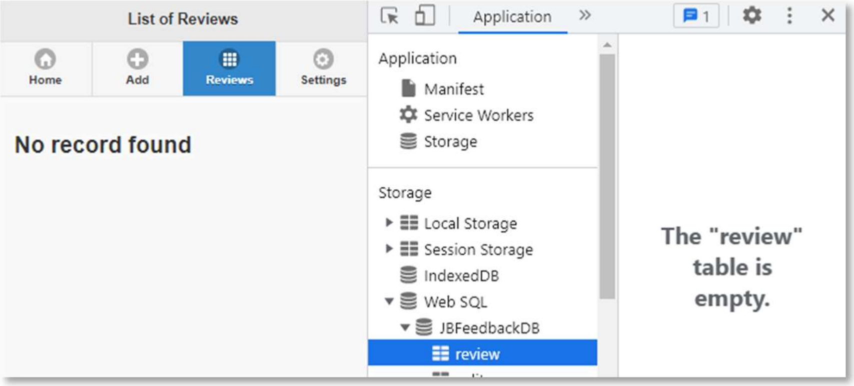
- Make a function named getReviews() in xxfacade.js
  - Call the CRUD function 'selectAll' of 'review' and generate a listview item for each row.
  - Include an extra attribute 'data-row-id' and assign the 'id'
  - Include only Business name, Reviewer email, comments and overall ratings in each list view item.
  - You have to recalculate overall ratings only if the 'hasRatings' field is true. Attach a click event handler dynamically for each list view item.
  - When each list view is clicked save their corresponding 'data-row-id' value to local storage and navigate to 'Modify Feedback' Page.
  - Show "No record found" on Reviews page if there is no record in the table.
- Add a 'pageshow' event handler in xxglobal.js for 'Reviews' page and call the getReviews() function.



Business Name: Bourn's Burger
Reviewer Email: jason@gmail.com
Comments: good stuff
Overall Rating: 93%

Business Name: Bond's Hardware
Reviewer Email: jason@gmail.com
Comments: good stuff
Overall Rating: 0%

Figure 7: List of reviews showing 2 items



No record found

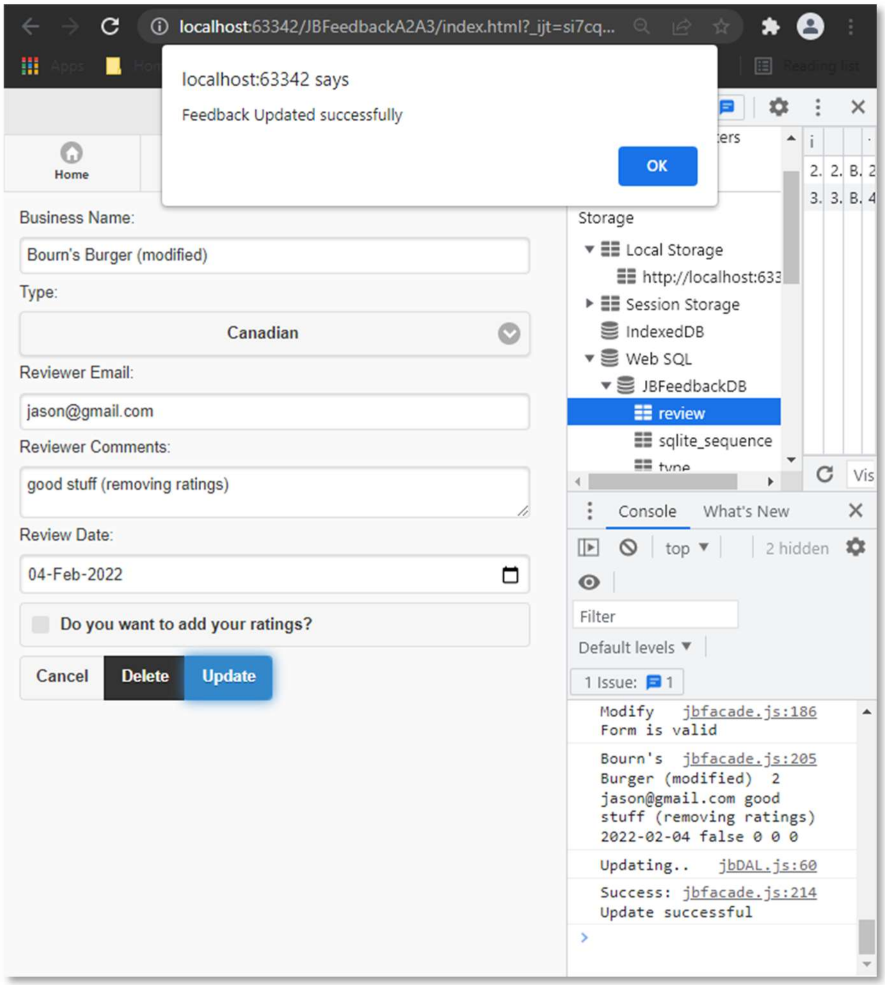
The "review" table is empty.

Figure 8: List of reviews showing No record found when review table is empty

Task 7: 'Modify Feedback' Page

- Make a function named 'showCurrentReview' in xxfacade.js file
  - Fetch the id stored in local storage
  - Use that id as the parameter for 'select' CRUD of 'review'
  - Show the values of each of the fields of the returned row in specific input controls (e.g., textbox, combobox, radiobutton etc.). Important: Make sure to show/hide ratings part based on 'hasRatings' value.
  - Remember to set each rating 0 if the ratings checkbox is unchecked while updating a record.
  - Make a 'pageshow' event handler for 'Modify Feedback' page in xxglobal.js and call showCurrentReview() function
- Make a function named 'updateFeedback' in xxfacade.js file

   - Call the 'update' CRUD of 'review' after doing validation. Show an alert.
- Make a function named 'deleteFeedback' in xxfacade.js file
   - Call the 'delete' CRUD of 'review'. Use id stored in local storage to delete the specific item. Show an alert.
   - Navigate to 'Reviews' page automatically.



localhost:63342/JBFeedbackA2A3/index.html?_ijt=si7cq...

localhost:63342 says

Feedback Updated successfully

Business Name: Bourn's Burger (modified)

Type: Canadian

Reviewer Email: jason@gmail.com

Reviewer Comments: good stuff (removing ratings)

Review Date: 04-Feb-2022

□ Do you want to add your ratings?

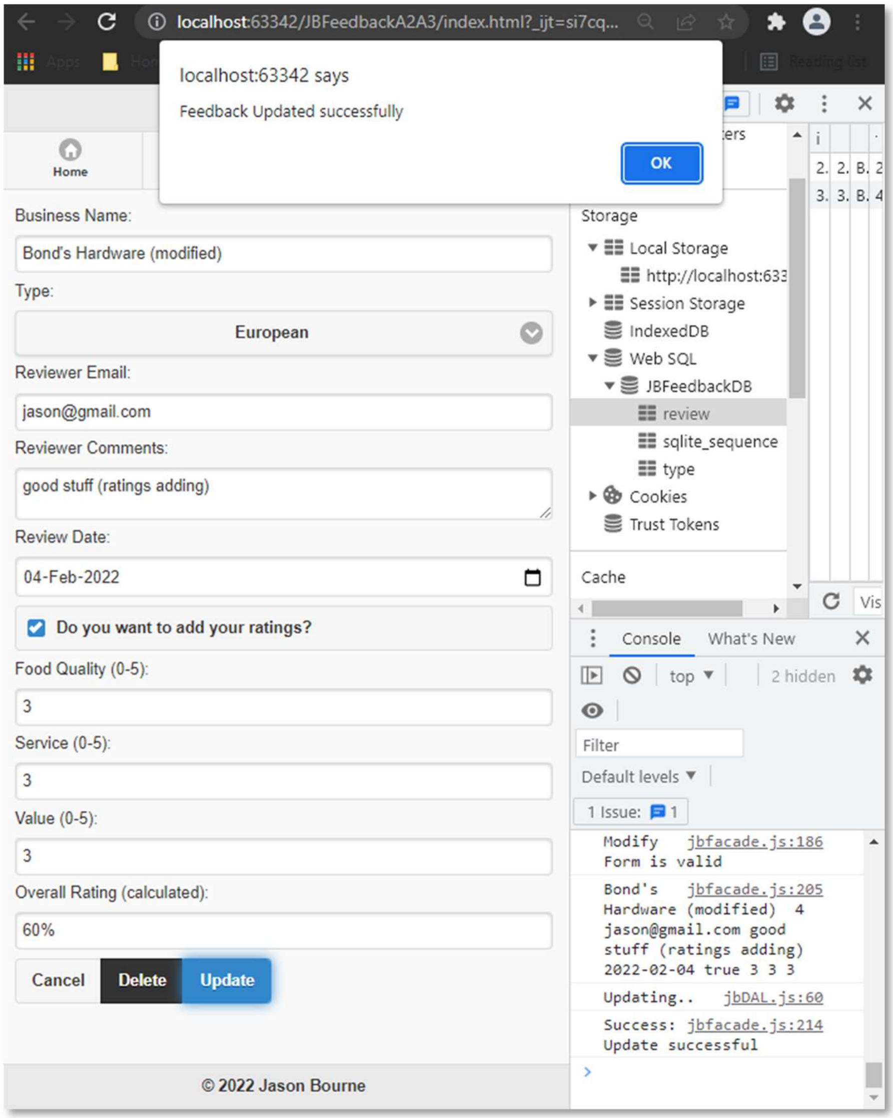
```
Modify          jbfacade.js:186
Form is valid

Bourn's         jbfacade.js:205
Burger (modified)  2
jason@gmail.com good
stuff (removing ratings)
2022-02-04 false 0 0 0

Updating..      jbDAL.js:60

Success:        jbfacade.js:214
Update successful
```

Figure 9: Updated with rating removed, alert shown



localhost:63342 says

Feedback Updated successfully

Business Name: Bond's Hardware (modified)

Type: European

Reviewer Email: jason@gmail.com

Reviewer Comments: good stuff (ratings adding)

Review Date: 04-Feb-2022

☑ Do you want to add your ratings?

Food Quality (0-5): 3

Service (0-5): 3

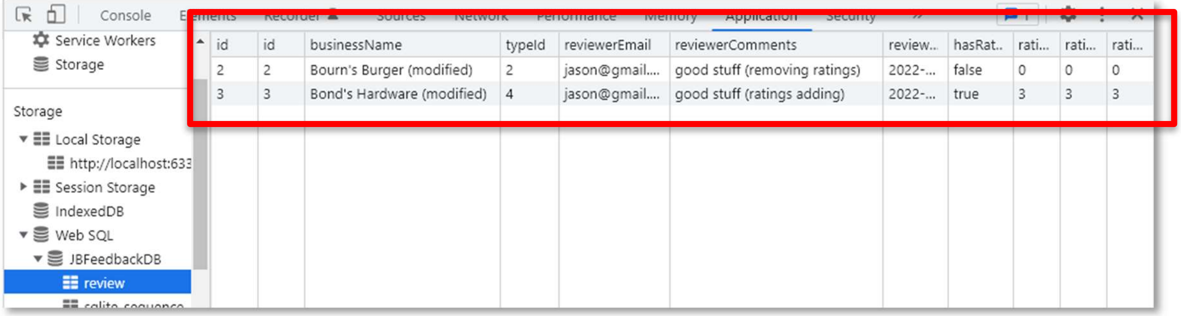
Value (0-5): 3

Overall Rating (calculated): 60%

© 2022 Jason Bourne

```
Modify      jbfacade.js:186
Form is valid
Bond's      jbfacade.js:205
Hardware (modified)  4
jason@gmail.com good
stuff (ratings adding)
2022-02-04 true 3 3 3
Updating..     jbDAL.js:60
Success: jbfacade.js:214
Update successful
```

Figure 10: Updated, with ratings added, alert shown

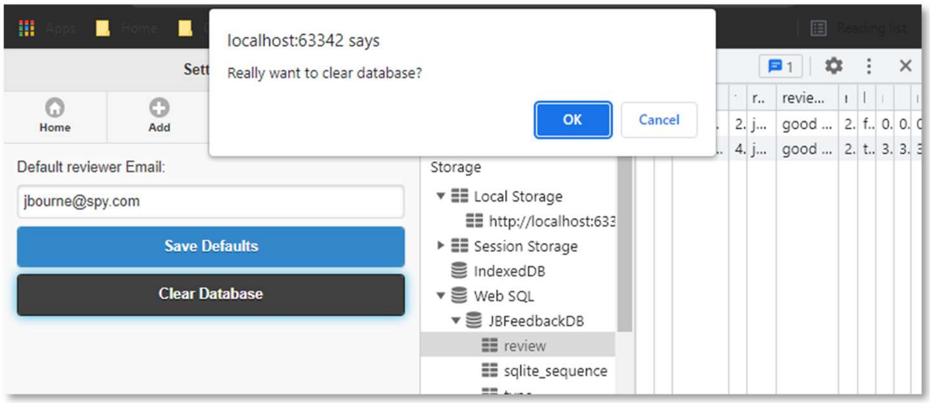


id	id	businessName	typeId	reviewerEmail	reviewerComments	review...	hasRat...	rati...	rati...	rati...
2	2	Bourn's Burger (modified)	2	jason@gmail....	good stuff (removing ratings)	2022-...	false	0	0	0
3	3	Bond's Hardware (modified)	4	jason@gmail....	good stuff (ratings adding)	2022-...	true	3	3	3

Figure 11: review table after the updates. Notice the ratings part.

Task 8: Update Settings page

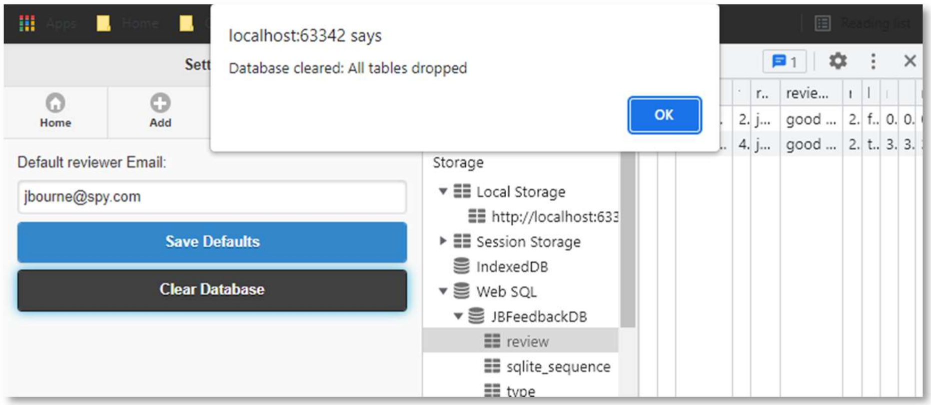
- Make a function named 'clearDatabase' in xxfacade.js file
  - Call the 'dropTables' defined the database.js file
  - Attach a click event handler for 'Clear Database' button and call 'clearDatabase()' function. Show an alert.



localhost:63342 says

Really want to clear database?

Figure 12: Clear Database clicked



localhost:63342 says

Database cleared: All tables dropped

Figure 13: Database is cleared

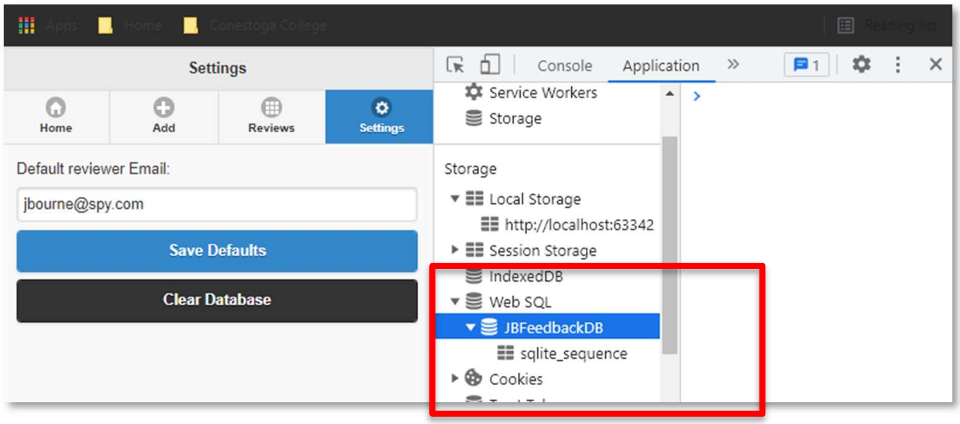


Figure 14: Tables have been dropped

Task 9: Test your app

- Please test your program carefully. No screenshot is required. Your program will be tested during demo.

Task 10: Upload app

- Make a zip file xxAssignment3.zip that will contain
  - Complete project for your app
  - After unzipping, your application must run without any modification.
- Login to eConestoga and upload to: Assignment 3.
- For late submission: marks will be deducted per day as specified below.

Task 11: Demo your app

- Demo your app during class time. See the announcement at the beginning of this document. Check Schedule Summary-2021 document available under "Content->Course Information" module for details.
- Please feel free to contact your instructor if you have any question.

Marking Sheet

	Description	Marks Allocated	Marks Achieved
1	Home page designed as specified	5	
2	Database and tables created according to spec	15	
3	Type table is initialized properly	5	
4	CRUD operations are defined in xxDAL file and works properly	15	
5	‘Add Feedback’ page: ‘type’ dropdown is populated from database with ‘other’ as default, default email is placed in ‘reviewer email’, save button works properly. Façade Functions and event handlers are defined according to the spec, Form initializes properly, show hides behaves according to spec	15	
6	‘Reviews’ page shows reviews as a listview dynamically. Information is shown according to spec. Clicking a review navigates to ‘Modify page’. Façade functions and event handlers are defined according to the spec.	15	
7	‘Modify page’ shows the selected review (the particular review clicked on ‘Reviews’ page), Type dropdown populated from ‘type’ table with correct item selected. Update, delete works according to the spec. Façade functions and event handlers are defined according to the spec.	20	
8	‘Settings’ page works according to the spec.	5	
9	Review table shows proper foreign key in Developer’s tool (id of Type)	5	
Deduction			
	Runtime errors	15 x_______	
	Assignment Standard (proper project name, see the standard documents for detail)	5 x ________	
	Programming Standard	1 x ________	
	Late Submission (softcopy)	_________days	
	JavaScript file not prefixed	2 x ________	
	Bugs (including requirements mentioned in this specification)	3-10 based on severity	
	Failing to answer to questions during demo	5-10 based on severity	
	No Demo	60	

	Total		

Late Penalty (Softcopy submission)

Days Late	Penalty %
1	5
2	10
3	20
4	30
5	45
6	60
7	80
8	100

Please note: How "Days Late" is calculated: Your assignment is due on Sunday 11:59pm. You are considered 1 day late if you submit on anytime on Monday (until 11:59PM). If you submit anytime on Tuesday (until 11:59PM). you are considered 2 days late.