# Assignment 3

## Due Date (Softcopy submission):

Softcopy is due on Sunday 11:59PM on 12th April 2020.

## 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 examples 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.
* All id and name attributes must have your initial (e.g., jb for Jason Bourne) as prefix.

## 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.
* Add the following Javascript files under js folder (if you have not done already):
  + xxfacade.js
* Include xxfacade to index.html

### 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.
* In the xxdatabase.js create your database for this application in a function called xxCreateDatabase( ). (Using global object is recommended) :
  + Create 2 tables: ‘type’ and ‘review’ in a function called xxCreateTables()
  + 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. You must execute drop table command for ‘type’ table, re-create it, and insert 3 records with ‘Canadian’, ‘Asian’ and ‘Others’. This table will be linked to the dropdown ‘Business Type’.
  + Hints: in your txFunction()

1. Declare sql for drop table, then call executeSql()
2. Declare sql for create table ‘type’, then call executeSql() again.
3. Declare sql for insert 1st row to ‘type’, then call executeSql() again.
4. Declare sql for insert 2nd row to ‘type’, then call executeSql() again.
5. Declare sql for insert 3rd row to ‘type’, then call executeSql()
6. Declare sql for create table ‘review, then call executeSql() again.
7. You just need one db.transaction() call and pass the txFunction().
8. You may define an array containing 3 insert sql commands and execute executeSql() by using a loop.

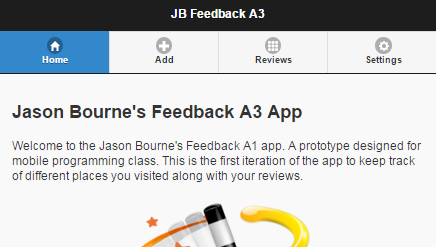
* Make a function named xxdropTables() that will drop both ‘type’ and ‘review’ tables. This function will be called when ‘Clear Database’ button under ‘Settings’ page is clicked.

### Task 2: Include CRUD operations

* In the JBFeedbackDAL file create CRUD functions for
  + ‘review’ table: xxinsert, xxselect, xxselectAll, xxupdate, xxdelete
  + ‘type’ table: xxselectAll

### Task 3: Modify ‘Home’ page

* Update the headers to indicate Assignment 3. See the screenshot below

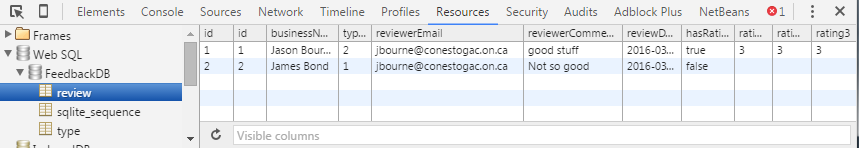


### Task 4: Modify ‘Add Feedback’ page

* In xxglobal.js add an event handler for Add Feedback pages’s ‘pageshow’ event.
* In the event handler function load the default email address from local storage and show it in the Reviewer Email textbox. Hint: It is assumed that, after the program runs, the user will go the Settings page and save the default email (this task was done in Assignment 2). The same email will 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 ‘xxupdateTypesDropdown’ 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 dropdown already defined in ‘Add Feedback’ page
    - Make ‘Others’ as selected.
* Clicking the ‘Save’ button the record will be inserted to ‘review’ table after validating.
  + Make a function in xxfacade.js file named xxaddFeedback(). 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 xxaddFeedback() function.
  + Make sure to include ‘Food quality’, ‘Service’ and ‘Value’ ONLY when ‘Do you want to add your ratings’ checkbox is checked. Check the screenshots below.

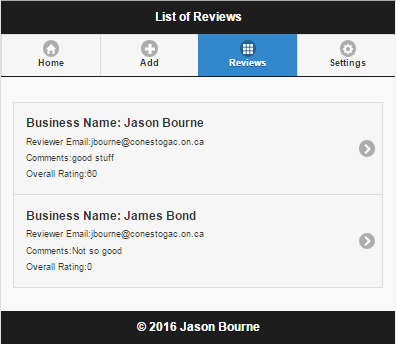
|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |



### Task 5: Update ‘Reviews’ Page

* Make a function named xxgetReviews() 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.
* Add a ‘pageshow’ event handler in xxglobal.js for ‘Reviews’ page and call the xxgetReviews() function.



### Task 6: Update ‘Modify Feedback’ Page

* Make a function named ‘xxshowCurrentReview’ in xxfacade.js file
  + Grab 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.). Make sure to show/hide ratings part based on ‘hasRatings’ value.
  + Make a ‘pageshow’ event handler for ‘Modify Feedback’ page in xxglobal.js and call xxshowCurrentReview() function
* Make a function named ‘xxupdateFeedback’ in xxfacade.js file
  + Call the ‘update’ CRUD of ‘review’ after doing validation.
  + Navigate to ‘Reviews’ page automatically.
* Make a function named ‘xxdeleteFeedback’ in xxfacade.js file
  + Call the ‘delete’ CRUD of ‘review’. Use id stored in local storage to delete the specific item.
  + Navigate to ‘Reviews’ page automatically.

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

### Task 6: Update Settings page

* Make a function named ‘xxclearDatabase’ in xxfacade.js file
  + Call the ‘dropTables’ defined the database.js file
  + Attach a click event handler for ‘Clear Database’ button and call ‘xxclearDatabase()’ function.

|  |  |
| --- | --- |
|  |  |

### Task 7: Test your app

* Use Chrome browser and provide screenshots of the following
  + Home Page
  + Add a Feedback page with data with checkbox unchecked and with Alert (Successfully inserted)
  + Add a Feedback page with data with checkbox checked and with Alert (Successfully inserted)
  + Screenshot of Developers tools showing 2 records (1 with ratings, the other without ratings)
  + Screenshot of Reviews page showing 2 items (previously inserted)
  + Modify Feedback page with successfully updated Alert
  + Modify Feedback page with successfully delete Alert
  + Settings page with alert (after clearing database)
* Save your screenshots to xxA3Screenshots.doc file.

### Task 8: Upload app

* Make a zip file xxAssignment3.zip that will contain
  + Complete project for your app
  + xxA3Screenshot.doc file.
  + After unzipping, your application must run without any modification.
* Login to eConestoga and upload to the dropbox: Assignment 3.
* For late submission: marks will be deducted per day as per specification available in the cover page with marking sheet (latest version, available under Resources in eConestoga of course PROG2430).

### Task 9: Demo your app

* Demo your app on the first class after the due date.
* Bring your hard copy during the demo containing:
  + Assignment coversheet and marking sheet
  + Printout of ALL javascript files. (Don’t print html file)
  + Printout of xxA3Screenshot.doc containing all screenshots
  + Assignment 3 marking sheet (The last page of this document)
* You must demo your project within 1 week of due date. Otherwise you will not get the marks allocated for demo.
* If you choose not to demo at all, you may drop the hardcopy of your assignment at 4G26. However, if you don’t submit the hardcopy, marks will not be uploaded.

# Marking Sheet

|  |  |  |  |
| --- | --- | --- | --- |
|  | Description | Marks Allocated | Marks Achieved |
| 1 | Home page designed as specified | 5 |  |
| 2 | Database and tables created according to spec | 15 |  |
| 3 | CRUD operations are defined in xxFeedbackDAL file and works properly | 15 |  |
| 4 | ‘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. | 15 |  |
| 5 | ‘Reviews’ page shows reviews as a listview dynamically. Information are shown according to spec. Clicking a review navigates to ‘Modify page’. Façade functions and event handlers are defined according to the spec. | 15 |  |
| 6 | ‘Modify page’ shows the selected review (the particular review clicked on ‘Reviews’ page). Update, delete works according to the spec. Façade functions and event handlers are defined according to the spec. | 20 |  |
| 7 | ‘Settings’ page works according to the spec. | 5 |  |
| 8 | 8 screenshots | 10 |  |
|  |  |  |  |
| Deduction | | | |
|  | Assignment Standard | 5 x \_\_\_\_\_\_\_\_ |  |
|  | Missing javascript file in hard copy | 5 x \_\_\_\_\_\_\_\_ |  |
|  | Id/name/function name not prefixed | 2 x \_\_\_\_\_\_\_\_ |  |
|  | Programming Standard | 1 x \_\_\_\_\_\_\_\_ |  |
|  | Late | \_\_\_\_\_\_\_\_\_\_\_ |  |
|  | Bug | 3-10 (based on severity) |  |
|  | No Demo | 40 |  |
| Total | |  |  |