

ITECH3224/6224

World Wide Web Technology 2

Faculty of Science

Assignment 2 – Dynamic review site

Introduction

This is an individual assignment in which you will explore the use of data interchange between web servers and browsers, dynamic construction of page content, and API design, using PHP, JavaScript, XML, and JSON.

Details of the weight of the assignment and due date are given in the course description.

Please note: you may adapt the application to the review of anything you wish – *within tasteful boundaries* – but you should keep the basic schema the same.

For example, you could do reviews of guitars, microbreweries, running tracks, nightclubs, coffees, tv shows, shoes, games, chicken parmigiana or similar. Adjust table names and foreign keys appropriately.

Task Description

Introduction

Raccoon Reviews is a website that allows raccoon enthusiasts to connect over their shared interest.

Each raccoon has a dedicated page that shows a photo and a name, a list of reviews, and some form of average rating.

Users can submit new reviews of a raccoon, including their name, some review text, and a rating.

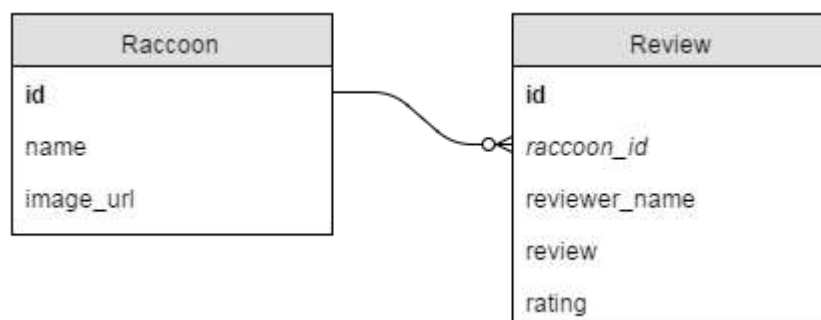
Database

A simple database with two tables contains data about raccoons, and their reviews.

The database has the following structure:

Raccoon(id, name, image_url)

Review(id, *raccoon_id*, reviewer_name, review, rating)



Each record in the Raccoon table refers to a single raccoon, and each record in the Review table is a review of a single raccoon. Each Raccoon may have many reviews.

The assignment tasks are closely associated with the lab work of topics 7 to 10. Code and examples from lectures and labs should be a useful guide throughout this assignment. The assignment requires a number of files and a report to be produced. The report should respond to written tasks that are included below.

Initial Task

Create the database using your MySQL skills. You will need at least 4 raccoons, with at least 2 reviews each.

You **may** normalize the database if you wish, to include a dedicated "reviewer" table.

Invent your own data – name raccoons after your family, favourite bands, explorers, movie characters, or mythological figures, for example.

Use appropriate data types – in particular **Review.review** should be a TEXT field to allow long entries.

Review.rating is an integer value from 1 to 5 inclusive, where 5 indicates that this is an amazing raccoon.

Submit your SQL file as part of your assignment.

XML and JSON markup

Mark up the complete data using XML tags and save it as an .xml file. Check that the file is well-formed and report the method used. Draw a diagram showing the structure of the DOM tree associated with the .xml file, using data from any ONE record. Include the root element, node types, elements and attributes.

Similarly, mark up the complete data using JSON and save it as a .json file. Check that it is valid JSON and report the method used to validate.

Submit both files as part of your assignment.

Back-end API

Create a RESTful XML or JSON API (your choice), implementing, at minimum, the following functionality. Use the HTTP method specified in parentheses:

- List all raccoons (*GET*)
 - Should return raccoon id, name and image_url for all raccoons, without reviews
 - Use an optional querystring/GET parameter to allow sorting by name & average rating
- Retrieve all details for a single raccoon (*GET*)
 - including reviews
- Create a new review/rating (*POST*)
- Delete a review/rating (*DELETE*)
- Update an existing review/rating (*PUT*)

Follow HATEOAS (Hypermedia as the Engine of Application State) practices. Include a note in your report about how you have followed HATEOAS.

Under Apache, this step will require configuring a .htaccess file to allow using clean URLs under the **api** folder (without a .php extension)

Front-end

Create an HTML/CSS/JS page which uses JavaScript, the DOM and your back-end API to:

- Display a menu of all raccoons. Allow sorting the menu by name and rating
- Display an individual raccoon's details when the menu item is selected, including the image and the average rating

- Allow the user to submit a new review/ranking, which will be displayed immediately in the review list, and upon which the raccoon's average will be updated from the server
- Periodically (every 30 seconds), poll the details for the current raccoon and display any new reviews

Note: This task may require finding images to use. Consider using Wikimedia Commons to find images with permissive licenses.

All functionality should be implemented using JavaScript, the DOM and the API, without reloading/refreshing the browser page.

Bonus task (Optional!)

Derive your averaging and sorting code based on the approach listed here:

<http://www.evanmiller.org/ranking-items-with-star-ratings.html>

Additional task for ITECH6224 students

On the topic of “Session hijacking and session fixation” identify four (4) relevant, independent resources. Use these resources to write an essay discussing the topic in your own words.

Take care to cite appropriately. Word count should be 450-550 words (excluding references)

Report contents

1. The theme of the review site (eg. Raccoon Review)
2. Statement of completion
3. DOM diagram
4. Essay task (ITECH6224 only)
5. Give details of any assistance received (apart from teaching staff).

Further details

Refer to the Course Description for details of submission to Moodle, late assignments, extensions, special consideration, plagiarism, student support, presentation of academic work and adopted reference style.

Submission

All files should be zipped and uploaded to Moodle by the due date and time.

ITECH3224/6224 World Wide Web Technology 2

Faculty of Science

Assignment 2 Marking Guide

Name _____ ID _____ Marker _____

Criterion	Maximum	Obtained
.sql file – requirements satisfied	1	
.xml file	2	
- Accurate	1	
- Well-formed	1	
.json file	2	
- Accurate	1	
- Syntactically valid	1	
Back-end API	6	
- Clean RESTful (resource-based) design, uses HATEOAS approach	1	
- List & Details (1), New review/rating (1), Delete (1), Update (1)	4	
- Sort based on query parameter	1	
Front-end (Must use API to construct)	4	
- Menu	1	
- Details, including image	1	
- Submit a new review/ranking, including list and average update	1	
- Automatic update	1	
Written task (<i>ITECH6224 only</i>)	5	
- Content	3	
- Adherence to academic standards	2	
Report contents	2	
- Theme and statements of completion/acknowledgements. Document standards.	1	
- DOM diagram	1	
Bonus!	2	
- Sort using Bayesian confidence	2	
Quality of code:	3	
- Modularity - file and code structure. Not monolithic.	1	
- Indentation/layout, naming scheme	1	
- Use of comments, including quality and accuracy. Comments do not simply narrate code but illuminate intent and design decisions.	1	
Totals		
- ITECH3224	20	
- ITECH6224	25	
Final Mark (scaled, correct to 1 decimal place. Maximum of 20 marks)	20	

Comments: