

COMPX322-23A Assignment One

Due Date: Monday March 27th, 10am

Asynchronous Event Viewer

PART ONE

For this coursework you are to implement a web application that displays information about local events. Your implementation will require use of the following:

Validated HTML	Cascading Style Sheets (CSS)
JavaScript	DHTML
Server-side PHP scripting	Interaction with a MySQL database
AJAX and Fetch	Retrieval of data from an external API

Application description

The *minimum* requirements for the application are as follows:

- it consists of a **single** html page with a separate CSS document to control layout and appearance
- when the page is first loaded it shows a list of local events, this must use an **AJAX** request to get data from a MySQL database
- clicking on one of the event names will display full information for that event, retrieved via another **AJAX** request
- a user can request weather information for the event location, this is retrieved from OpenWeatherMap () using a **fetch** request
- a user is able to update event information by entering new text, this must use an **AJAX** request
- a user can return to the home page, which refreshes the event list, using an AJAX request
- all functionality is achieved within a single page (url) without the user needing to refresh the page in their browser or navigate to a new page. This is where the use of asynchronous requests and DHTML is essential

What you need to do (in whichever order you think appropriate)

- a) Create an account at <https://openweathermap.org/api> (you can make 1000 requests per day free of charge)
- b) Import the provided MySQL table into your database: event.sql
- c) Design and implement the layout of your web page using well-formed and valid HTML in combination with one or more CSS

stylesheets which should be stored in separate file(s).

- d) Implement PHP scripts to retrieve and add the data needed by your application
- e) Implement any required JavaScript code that is needed to support your application. This includes code to handle asynchronous AJAX and fetch requests and responses to and from the server. Make sure you follow the instructions above for the use of AJAX and fetch. Keep your JavaScript code in a separate file.
- f) NOTE: your PHP scripts should return data to your HTML page using AJAX, you should **not** display content directly from your PHP scripts

Important – for this assignment do not use JavaScript libraries (e.g. JQuery) for the AJAX requests.

PART TWO

Sign up for a lab session to demonstrate your solution. You will be required to answer some questions about your code and show how you would refactor part of it based on a given change request. You can book an early lab session as soon as you have all of the required functionality complete (you don't need to wait until your solution is totally finished). Final demo slots will be available to book in the supervised labs the week before the deadline.

What to submit and how

Your code for this assignment must be submitted electronically on Moodle. Make sure your connection scripts are set for use in the lab environment. Assuming that all parts of your application are within a directory called **compx322assn1** within your **course_html** directory

Compress the directory into **compx322assn1.tar.gz** (or **gzip** or **zip**)

Upload the compressed file to the **Assignment 1 Submission** link on Moodle. No other mechanism for submission will be accepted.

Make sure all work submitted is your own, you should be familiar with the University regulations on plagiarism. You are free to discuss solutions with your classmates but this is an **individual** assignment so you should prepare all of the code on your own. This also means you should not use AI solutions to write your code.

How your work will be assessed

The assignment will be marked out of **100** as follows:

• Application meets minimum functional requirements	30 marks
• Asynchronous requests are used to retrieve, update and display all data without a page reload	20 marks
• AJAX and fetch are used for the requests as required	10 marks
• PHP scripts correctly handle data and interact with MySQL server	10 marks
• HTML/CSS used appropriately to create usable layout and clear presentation	10 marks
- correct use of HTML elements	
- CSS used for functional and aesthetic design purposes	
• All code is suitably commented	10 marks
	90 marks
• Lab demonstration (mandatory)	10 marks
TOTAL	100 marks

The lab demonstration will require you to explain your code, answer some questions about your code and explain how you could refactor it for a given requirement change.

Note: marks will be deducted if we have to edit your PHP scripts to correctly connect to the UoW MySQL server so make sure you test in that environment and submit code that will successfully connect.