

# 2702ICT Intelligent Media Systems – Assignment 1 Tourism Destinations

Individual Assignment (35%)

DUE: Tue Week 7 9am

Create an interactive web site that will help promote Queensland tourism. This application will display photos from famous Queensland tourism destinations (e.g. Great Barrier Reef, Fraser Island, Whitsunday Islands, Gold Coast etc.)

## Functional Requirements

On start-up, your application will display at least 7 Queensland tourism destinations that the user can select/click on. Once a destination is selected, your application will fetch (from flickr) and display at least 5 photo thumbnails of that destination. When a user clicks on a thumbnail, the full-size photo will be displayed in a modal. From full size display, user can go back to thumbnails. From the thumbnail display, user can go back to the list of destinations.

The photos should be fetched from Flickr via Flickr's Web API. Photos for a destination can be retrieved by performing a Flickr search API call and supply the destination as the search string.

There is a “**Recently Viewed**” feature. Once a photo (from any destination) is viewed in full-size, this photo is also added to and displayed in the Recently Viewed list. This list displays photo thumbnails in a smaller size than the thumbnail display. This list displays up to 5 thumbnails, with the most recently added thumbnail displayed at the top if the list is vertical, or on the left if the list is horizontal. Clicking on a recently viewed thumbnail will also display its full-size image in a modal and move it to the start of top of the list.

The layout of this application should be **responsive** with 3 break points:

- On a screen with width 1024px or above, there should be THREE columns.
- On a screen with width 768px or above, there should be TWO columns.
- If a screen width is below 768px, there should be ONE column.

If you wish to set the break points at different widths, please ask your tutor before the due date.

You can decide how you use the columns. But a suggestion is to use one column for displaying the categories, one for thumbnails, and one for recently viewed list.

Other functional requirements you need to implement are:

1. In your source code, the destinations should be stored in a **JSON object** and placed either at the top of your main source code file or in a separate file. The destinations your application displays can be updated by updating this JSON object.
2. The **title** of a photo, if available, should be fetched and displayed when displaying that photo as thumbnail and full-size.
3. The **date** the photo is taken should be fetched and displayed when displaying that photo as thumbnail. Note: the time should not be displayed.
4. When scaling the image for full-size display, ensure the width and height remain **in proportion**.
5. Flickr provides photos in **different sizes**. The “Large Square” size photo should be used for recently viewed list. The “Small” size photo should be used for thumbnail display. If “Large Square” or “Small” size does not exist, use the smallest photo. The “Large” size photo should be used for full-size display. If “Large” size does not exist, use the largest available photo.
6. Open-ended features:
  - a) Destination photo should be filter for “**quality**”, so that only high-quality photos are used for display. You can decide how to filter for high quality photo (or even what high quality means). An example could be that if the photo is recent or has been favoured by many people.

- b) Implement **another user interface feature** other than modal or recently viewed widget (e.g. carousel, drop down menu, or a different way to display thumbnails). You can either implement this yourself or use an API/library. Note: more marks will be awarded for self-implementation.
- c) Implement a feature that uses **another web API** (other than Flickr). Note: please avoid Web APIs that require user authentication.

You'll be judged on the originality/creativity, sophistication, complexity and impressiveness of your design and implementation, and also how well this feature fits within the theme of this assignment.

## Design

The look and feel of your web application should fit the theme of this application. You should create and implement your design to impress your client (the marker). In addition, attention to detail is expected. In coming up with your design, you should apply the design principles covered in the lecture.

Note: you should implement your own responsive layout (using media query and flexbox), and style/design. You should not use an existing CSS framework (such as Bootstrap and Foundation) to implement your layout and design.

## Documentation

1. Reflect on the process you have applied to develop your solution (e.g. how did you get started, did you do any planning, how often do you test your code, how did you solve the problems you come across). What changes would you make for assignment 2 to improve your process? Write at most half a page.
2. Describe the open-ended features and how you have implemented these features. If you did not implement any open-ended features, then describe the most difficult problems you have solved to produce your implementation. Write at most half a page.

For further details of the requirements, refer to the marking rubric. **All requirements from both the assignment specification and marking rubric must be satisfied.**

## Submission Requirements

Your submission should consist of:

1. A compressed file (.zip or .tar.gz) containing your application (including all source code). See Week 1 lecture regarding how to zip a directory and download file from Elf.
2. A PDF file containing your documentations.

These files must be submitted via the course website through the assignment 1 link.

## Assignment Demonstration and Marking

You must demonstrate and explain your work to a member of the teaching team in Week 7 class to have your submission marked.

During the demonstration, you need to show the last modified date of your file (run the command: **ls -la** in your *Javascript* directory). Make sure you do not change your code on Elf (or local environment) after you have submitted it.

**Warning: We take student academic misconduct very seriously.**