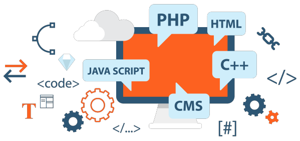
UWI
ST. AUGUSTINE
CAMPUS
FACULTY OF SCIENCE
AND TECHNOLOGY

INFO3602 Web Programming & Technologies II

Lab Tutorial - Week 7



Learning Objectives:

- Load a plugin for creating custom fields
- Create a custom post type that uses custom fields
- Create Archive pages for blog posts and custom post types
- Load custom post type content on the front end of a website
- Create basic custom WordPress queries

This lab uses the slide decks from Week 5 (W5) and Week 6 (W6)

Task 1: Load the Advanced Custom Fields Plugin

- Download the **Advanced Custom Fields** plugin via the WordPress dashboard (W5, slide 84)
- Verify you have it correctly installed via the dashboard (W5 slide 85) . This plugin allows us to create custom post types that persist and are unaffected when themes are changed.

Task 2: Create a custom post type for an Event with a custom field

- Set up the **mu-plugins** folder for enabling custom post types (W5, slide 23).
- Create the PHP file, **university-post-types.php** that will have the custom field code in the **mu-plugins** folder.
- In **university-post-types.php** add the function and code to create a custom post type called **event** (W5, slide 17)
- Examine your WordPress admin dashboard and verify you have a custom type (W5, slide 18)
- Customise the **event** post type dashboard options by :
  - using a WordPress Dashicon of your choice for the custom post type (v, slide 19)
  - renaming the custom post type in the menu to be **Events** (W5, slide 25)
  - setting appropriate labels containing the term '**Event**' when adding, viewing, editing the event type (W5, slides 25-28)
  - supporting **archives** (W5, slide 59)
  - supporting **excerpts** (W5, slide 73)
- Create a new custom field called **Event Date** that only appears in the edit page for event post types (W5, slides 85-87)
- Create **three event posts** (W5, slide 20) with months set apart by 3 months. Set one event's year to be in 2020 and the others for 2024 Months and dates are up to you. Create an **excerpt** for one of the recent events.

Task 3: Create an Archive Page for Events

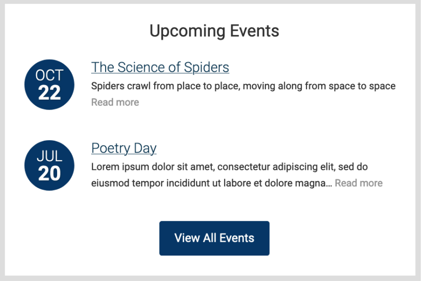
- Create an **archive page** for events called **archive-event.php** (W5, slide 61) duplicated from archive.php (*See note below)
- Customise the **banner** title and **intro** (W5, slide 61)
- Append the slug /**events** to your site's root url and open the archive page in your browser. You should see something similar to W5, slide 63.

*Note: To create an archive.php page, first duplicate index.php (blog listing page) and then customise code with a dynamic title - W5, slides 56-57. Click on an individual author's link and observe the url with the '/author/..' slug. Repeat for a category and observe how the url changes with the '/category/..' slug.

Task 4: Create an individual Event page

- Create an individual event page: **single-event.php** by duplicating single.php (W5, slide 58). This allows us to customise how an individual event page looks by tweaking the blog detail page.
- Customise the **banner title** and **intro** (W5, slide 61)
- Copy the **event-summary class** from **front-page.php** into the page and paste it within the WordPress loop together with the functional calls (Week 5, slide 62 and Week 6, Slide 59) to make the event details dynamic (month, day, title, link, excerpt).

Task 5: Load custom event post content on the front page



Upcoming Events

OCT 22 [The Science of Spiders](#)
Spiders crawl from place to place, moving along from space to space
Read more

JUL 20 [Poetry Day](#)
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna... Read more

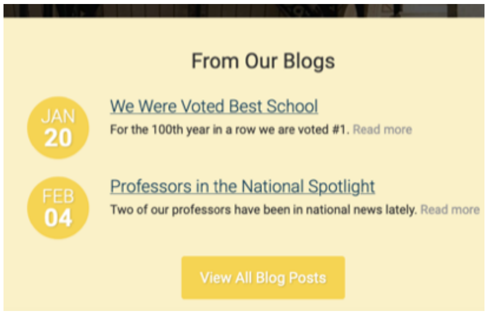
View All Events

- In the **front-page.php** file, locate the white event area by inserting some test code. (W5, slide 46)
- Create a custom WordPress query that outputs the title of event posts (W5, slide 48)
- Modify the query so that the blog posts are formatted (W5, slide 51) using the style shown on slide 50
- Remove the hardcoded content and replace with dynamic WordPress function calls that inserts:
    - the title (W5, slide 67)
    - working links to event detail pages (W5, slide 67) and archives (W5, slide 76)
    - trimmed content from the event post (first 20 words) if there are no excerpts (W5, slide 69,72)

  Refer to details on W5, slide 72. Ensure you clean up the custom query variables (W5, slide 69).
- Manually rebuild the permalink structure if links don't work (W5, slides 54-55)
- Customise the query so that dynamic event dates are used in the blue circle on the front page- (W5, slides 89-93). Goal: slide 94

Task 6: Load the two most recent blog posts on the home page of the website

- In the **front-page.php** file, locate the beige blog area by inserting some test code. (Week 6, slide 43-44)
- Test the default WordPress query to print out results of the data relative the current url (W6 slides 45-46)
- Create a custom WordPress query that outputs the two most



From Our Blogs

JAN 20 [We Were Voted Best School](#)
For the 100th year in a row we are voted #1. Read more

FEB 04 [Professors in the National Spotlight](#)
Two of our professors have been in national news lately. Read more

View All Blog Posts

recent blog posts (W6, slides 49-52)

- Customised the query further to filter blog posts under the Award category (W6, slide 53)
- Modify the query so that the blog posts are formatted (W6, slide 57) using the style shown on slide 23
- Remove the hardcoded content and replace with dynamic WordPress function calls that insert:
    - the month and day (yellow circle)
    - working links to blog detail pages
    - trimmed content from the post (first 20 words)

Refer to W6, slide 59, 62 for details. Ensure you clean up the custom query variables (W6 slide 61).

Task 7: Tidy up the design (optional)

- Insert code that makes the Blog and Event Links work in the header navigation area and the yellow style feature that indicates the page is active. (W6 slide 64 - 65)