



Learning Objectives:

- Create dynamic page banners with custom image sizes and custom fields
- Add and display custom subtitles on interior pages of the site
- Customise WordPress to operate with custom roles and permissions
- Create and test user accounts using default and custom WordPress roles
- Set up open registration on a WordPress site with working Sign Up functionality
- Reset a user's password from the Admin Dash
- Customise a user profile with a gravatar
- Control the display of the WordPress admin bar for particular roles
- Customise the default WordPress Login Screen

This tutorial references the slides from Week 7 and Week 8 for the different tasks

Task 1: Use a dynamic background image on the page banner

- Create a new custom field called Page Banner that has within it two fields: Page Banner Subtitle and Page Banner Background Image (Week 7, slides 56- 59).
- In functions.php, create a new custom image size for the page banner (Week 7, slide 60)
- Upload a page banner image and banner subtitle for one of the professors (Week 7, slide 61)
- Display the page banner image and subtitle entered from the WordPress admin on the professor page (Week 7, slides 62-63). Verify this works.
- Print out the contents of the variable returned by the function call `get_field('page_banner_background_image')` using the `print_r` function (Week 7, slide 65). What type of variable is it?
- Use the custom image size for the page banner on the page (Week 7, slides 66-67). Looks nice.

Task 2: Refactor the code from Task 2 into a reusable `pageBanner()` function

- In functions.php, create a function called `pageBanner()` (Week 7 slide 68)
- Replace the code in `single-professor.php` with the `pageBanner()` function call. (Week 7, slide 69-70). Verify no change on the front-end.
- In `page.php`, replace the generic page banner div with the `pageBanner()` function call (Week 7, slides 71). Observe that pages without a background image specified from the WordPress admin are given black banner images.
- Customise the `pageBanner()` function to accept an array and then reference the contents of the array so that the title is used from the array (Week 7, slide 75).
- In `page.php`, pass in an array to the function with some values for the hardcoded banner title and subtitle (Week 7, slide 76). Observe the effect on the front end (Week 7, slide 77). Remove the hardcoded value and observe the effect again.
- Edit the function to cater for when no title is passed in (Week 7, slides 78-79).
- Repeat for the subtitle (Week 7, slides 80-81)
- Repeat for the background image (Week 7, slides 82-83)
- Neaten up by catering for pages that supply no arguments (Week 78, slide 84). Give an example of such a page (if it exists on your website). Add a few banner images to liven up your pages.

Task 3: Create User Accounts with Default WordPress Roles

- Create a new user account from the WordPress admin toolbar on the home page (Week 8 slide 7). The username/password should be something of your choice, and role should be Subscriber.
- Create a second user account (momuffet) from the WordPress dashboard (Week 8 slide 8) using the details shown in Week 8 slide 9. Give the user account Editor permissions (Week 8 slide 11)
- Observe your list of user accounts and roles on the dash (Week 8 slide 12). You should see 3 accounts.
- Test out the accounts in a private browser (Week 8 slides 13-14) and observe the different views presented for the different roles (Week 8 slides 15-16)

Task 4: Create Custom WordPress Roles

- Install and activate the User Role Editor by Members plugin for creating custom roles (Week 8 slides 19-20)
- Create a new role (Week 8 slide 22) called **Event Planner** (Week 8 slide 23).
- Set up custom permissions for this role so that it only manages **Event** posts. (Week 8 slides 25-27)
- Observe the new options on the dash (Week 8 slide 28)
- Enable all permissions for this role for the **Event** type (Week 8 slide 29) so that the user can create, edit their events and those from other users.
- Restore all **Event** permissions for the admin role (Week 8 slides 34-37).
- Create a second new role called **Programs Coordinator** (Week 8 slide 38)
- Set up custom permissions for this role so that it manages **Program** posts (Week 8 slide 39) and enable all permissions so that the user can create, edit their programs and those from other users.
- Restore all **Program** permissions for the admin role (Week 8 slide 42).

Task 5: Modify Existing User Accounts with Custom WordPress Roles

- Modify the momuffet user account so that the role is now Event Planner (Week 8 slide 30)
- Verify that the user account was updated with the new role (Week 8 slide 31)
- Create an Event post using this account (Week 8 slide 32) and verify that it is rendered on the front page (Week 8 slide 33)
- Assign a second role to this user account, Programs Coordinator, (Week 8 slide 40)
- Verify that the user account was updated with the new role (Week 8 slide 41)

Task 6: Set up Open Registration for Subscribers

- Enable open registration from the WordPress dash under the Settings option. Ensure that the role is subscriber (Week 8 slide 46)
- Test out this feature using a private browser (Week 8 slide 47)

Task 7: Add Functionality to the SignUp Button

- Add the appropriate command to the SignUp button code so that users are directed to the site's sign up page when pressed (Week 8 slide 48)
- Test that the button works using another user account where the username is lucas (Week 8 slides 49-50)
- Add the logic in header.php to hide the sign-up button (Week 8 slides 56-58) when the user is logged in, display a button that says Log Out (Week 8 slide 59), and show the user's picture as a gravatar (Week 8 slides 61-62).
- Tidy the URLs for the buttons (Week 8 slide 66)

Task 8: Reset a User Account Password

- From the WordPress admin, go to -> Users-> All Users and reset the password for lucas (Week 8 slides 51-52)

Task 9: Hide/Show the WordPress admin bar for different roles

- In functions.php, add code to hide the admin bar when a subscriber is logged (Week 8 slides 63-65)

Task 10: Customise the Default WordPress Login Screen

- Open the login screen by adding /wp-login to the end of your site's homepage url (Week 8 slide 69)
- In functions.php, add code that redirects users to the homepage instead of the WordPress site when the WordPress logo is clicked on the login screen (Week 8 slide 70)
- Locate the code that controls the WordPress logo using custom CSS (Week 8 slides 71-72)
- Add in custom CSS in style.css that replaces this default code (Week 8 slides 73-74)

Post samples of your own custom work from Tasks 2 and 10 on the myelearning forum for 1 bonus day .