# Major Web Project

## Assignment 2 – Design Implementation

Assignment 2 requires you to conceptualise, sketch and implement WordPress-driven dfi.dk website that is intended to deliver an optimal user experience desktop and mobile devices.

In Assignment 2 - Design Implementation, you are asked to implement your designs from Assignment A – Design Thinking, Wireframing & Storyboarding as a working, functional, WordPress site. Standards-compliant HTML and CSS can be used, as can SASS. Implement these layouts as a custom WordPress theme so that content can be deployed and updated. The layouts need to be responsive so that they are optimized for viewing on both smartphone and desktop.

The objectives of are as follows:

- Construct layouts and web pages using best-practice HTML5 and CSS3.
- Employ Responsive Web Design techniques to produce optimized layouts for smartphone and desktop.
- Familiarise yourself with WordPress.
- Understand the process of WordPress theme development, and construct a fully functioning site based on a custom theme.

This assignment is worth 17.5% of the subject mark. It is due Thursday Week 16, 5:00 PM (Thursday April 21, 2016).

---

This assignment forms the technical component of the major Web project. You will be assessed on your ability to demonstrate your understanding of HTML, CSS and WordPress.

The labs up to Week 16 will assist with some of the key technical aspects, but we expect a degree of independent learning. Specifically, we encourage you to:

- Attend the labs – they are there to help learn the skills you need to complete this.
- Refer to the lecture slides, laboratory materials and sample code on LearnIT.
- Refer to the WordPress codex, an excellent and well-documented resource for WordPress theme development: http://codex.wordpress.org/Theme_Development and http://codex.wordpress.org/Page_Templates

The assignment is in three parts and should be completed in the following order:

- **Part A** – Produce responsive static HTML and CSS mock-ups (15 marks)
- **Part B** – Implement HTML and CSS mock-ups as a dynamic, WordPress site (11 marks)

## Part A - Produce static responsive HTML and CSS mock-ups (15 marks)

In the first part you implement your designs using HTML and CSS for the static version of the site. Completing this should result in HTML and CSS mock-ups, i.e., static pages of your design implemented in HTML and CSS with appropriate placeholder content. The best advice on the resolution is at http://1200px.com.

Desktop: 1024 x 768
Mobile: (Tablet from earlier assignement)

This implementation should be responsive, i.e., layouts should change in response to the width of the browser's viewport. Since the width of the browser's viewport is different for mobile, tablet and desktop devices, this will result in a layout that is adaptive for mobile devices. Your responsive design needs to target **smartphones** and **desktop** devices. You do not need to implement a responsive design that targets tablet devices but the principle is the same. Implement the response design to your own SmartPhone's screen resolution.

Use CSS @media queries to implement responsive design. Don't use browser-sniffing or other feature detection methods but investigate what these are.

After you have completed this, your site should look like a proper movie website, i.e., provide placeholder content from the actual dfi.dk site or other online resources including:

- IMDB for movie synopses, reviews and information about directors and actors: http://www.imdb.com/
- Rotten Tomatoes for film reviews, etc: http://www.rottentomatoes.com/
- From web/sites that have reviews in English or Danish of the movies see http://b.dk or http://politiken.dk or http://www.theguardian.com/

Implement the THREE designs for the DESKTOP version of the site that you created in your first Assignment.

Marking criteria is as follows:

- Implementation of the THREE designs from Assignment 1, each for 2 marks each. For this, you will be assessed on the degree of completeness of your HTML and (responsive) CSS implementation. (6 marks)

- Quality of code, understanding of HTML and CSS, correct usage of elements and style rules, robustness of design, appropriateness of placeholder content (5 marks).

    - A site that has clear, well-documented code that demonstrates understanding of HTML and CSS with appropriate placeholder content will result in a higher mark for this component.
    - A site that has obscure, poorly documented code that demonstrates a lack of understanding of HTML and CSS with irrelevant placeholder content will result in a lower mark.

- Adherence to the visual designs from Assignment 1 (4 marks)
    - An HTML and CSS implementation that matches, or closely matches your design will receive full marks for this component.

- An HTML / CSS implementation that looks and feels different to your early design will result in a lower mark for this component.

## Part B – Implement HTML and CSS layouts as a responsive, WordPress site (11 marks)

WordPress is a CMS that can be used to run content-driven websites. WordPress provides ‘back-end’ functionality that allows end users to easily create and update content and use themes to customise how a site looks and feels. Themes give the ability to switch how a site looks and feels without altering the information structure or content of a site. *Custom themes* can be built to create customisable, content-driven WordPress sites. Part C requires you implement your HTML and CSS layouts as a functioning WordPress theme.

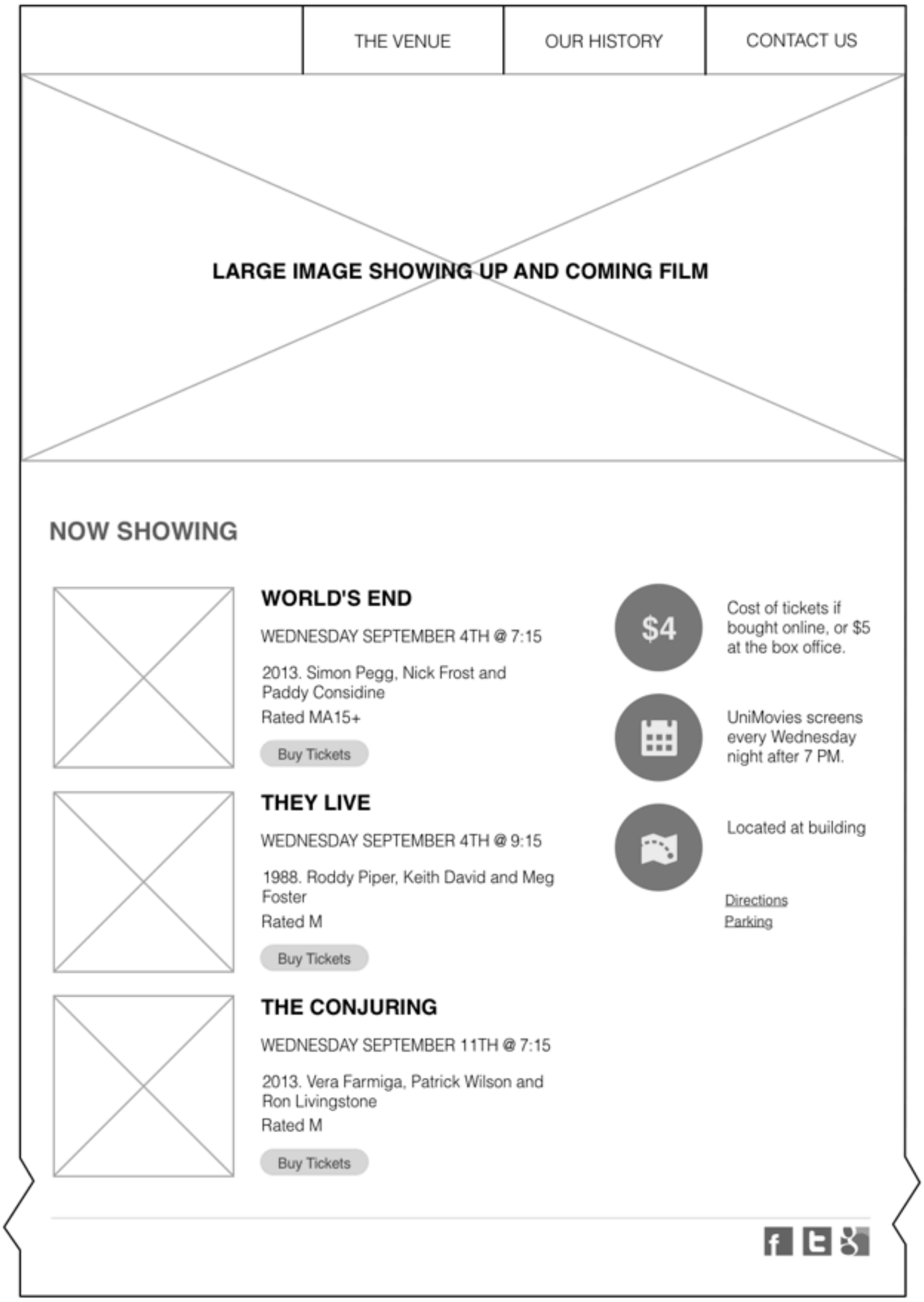
In order to understand how to implement using WordPress, it is important to understand the types of content it manages. WordPress presents 2 main types of content:

- *Posts* = frequently occurring, time-based content, e.g. blog posts and news articles, often presented in reverse order, i.e, the newest posts appear first.

- *Pages* represent separate pages or sections of a site that may be updated occasionally.

A museum website built using WordPress, for example, may use *posts* to present news about up and coming exhibitions and other events, because these events frequently occur and are *time-based*. Likewise, more static sections of the site, such as the site’s *About* or *Contact Us* sections, may be presented as *Pages* – these parts of the site, while they may be updated from time-to-time, are not so time-based as events.

To further convey how you might implement this part of the assignment, the following sample site wireframe of a University club film website is provided. The sample site wireframe demonstrates what a re-designed home page might look.



NOW SHOWING

WORLD'S END

WEDNESDAY SEPTEMBER 4TH @ 7:15

2013. Simon Pegg, Nick Frost and Paddy Considine

Rated MA15+

THEY LIVE

WEDNESDAY SEPTEMBER 4TH @ 9:15

1988. Roddy Piper, Keith David and Meg Foster

Rated M

THE CONJURING

WEDNESDAY SEPTEMBER 11TH @ 7:15

2013. Vera Farmiga, Patrick Wilson and Ron Livingstone

Rated M

Cost of tickets if bought online, or $5 at the box office.

UniMovies screens every Wednesday night after 7 PM.

Located at building

*Sample wireframe of the homepage of a re-designed film website.*
*This is a sample. Do not base your WordPress theme on this design.*

If we observe the **information content** of the above wireframe, note that:

- The *Now Showing* section of the page represents frequently updated, time-based content in chronological order. Use *Posts* to present this kind of content. Posts also contain *featured images* that showcase a thumbnail of each coming film.

- The navigation bar at the top of the design links to different sections of the site – *The Venue, Our History* and *Contact Us* - that may be updated from time-to-time. Use *Pages* to present this content.
- The design contains a large image at the top of the page that presents an enticing preview, showing a film still or poster of a coming film. Use a *featured image* or *featured header image* to present this content.
- The site contains relatively static sections, such as the content area to the right of the page that describe where the place is and how much the tickets or membership cost. As these aspects of the site are not likely to change much, they can be embedded 'directly' into the WordPress theme.

We can also observe the **structure** of the above wireframe:

- The site's *header* section consists of its top-navigation menu and featured image.
- The main *body* or *index* section of the site is represented by the series of *posts* in chronological order – the films listed under the site's *Now Showing* section.
- The static *sidebar* section contains information about the cost and location.
- The site's *footer* section at the bottom of the page.

A custom *WordPress theme* is constructed from a series of *templates* that present different aspects of its *information content*. In WordPress, separate templates may exist for:

- Presenting home-page content, such as index of the latest posts, or in this case, the list of *Now Showing* films, index.php, home.php or front-page.php.
- Presenting the header content of a site that may also contain a site's navigation – header.php.
- Presenting the footer section of a site – footer.php
- Presenting single, individual *posts* of a site – single.php
- Presenting single, separate *pages* of a site – page.php.
- Use the native Menu plugin in Wordpress and implement that in the theme (if you need and use a menu).

For this part of the assignment, you are required to implement your site as a WordPress-driven site with a custom theme based on the static HTML and CSS mock-ups that you have produced in **Part A**.

Details and examples on how to do this will be demonstrated in the tutorials in the coming weeks.

The marking criteria is as follows:

- Dynamic site content is presented in the form of *pages* and *posts*, where appropriate. (4 marks)

    - You are not permitted to use static pages to present your content. Your WordPress theme needs to be built in a way that all *page* and *post* – content can be created within the WordPress dashboard, i.e., not embedded in the theme itself.

- Correct use of template system (7 marks), specifically:
    - Use of the index template index.php, home.php or front-page.php to present home or generic page content (1 mark).
    - Separation of site structure and common elements into header.php, footer.php and sidebar.php (where appropriate) (2 marks)
    - Correct and appropriate implementation of the *WordPress loop* to present an index of chronologically ordered site content (2 marks).
    - Correct and appropriate implementation of a WordPress template to display individual *post* items using single.php (1 mark).
    - Correct and appropriate implementation of a WordPress template to display individual *pages* using page.php (1 mark).
    - Marks will not be awarded for inappropriate use of the WordPress templates, or for implementations that demonstrate a lack of understanding of the WordPress templating system.

# Submission Guidelines

The assignment is due Thursday Week 16, 5:00 PM (Thursday April 17, 2016).

**Part A** require static HTML and CSS files. These HTML and CSS files should be submitted as a ZIP file on LearnIT.

- Please ensure that your files are submitted to LearnIT no later than Thursday Week 16, 5:00.

**Part B** requires you to create a WordPress theme based on your HTML and CSS files. We will examine your WordPress theme as it resides on your directory on frwaw.itu.dk. There is no need to submit your theme files to LearnIT.

- It is your responsibility to ensure that you upload your theme files to frwaw.itu.dk by the due date.

- Your theme must be activated as the current theme on your WordPress installation, and therefore your theme must reside as a folder within WordPress's /wp-content/themes. Your theme and site should be viewable by the public by navigating to your site's URL on http://studentname.frwaw.itu.dk/. *It is not acceptable to simply put theme files as a stand-alone directory or ZIP file on* frwaw.itu.dk

Peter W. Eklund
March 14, 2016