ES Theme Editor & Media manager

Version 1.0 draft

# Overview

The purpose of this project is to make a standalone theme editor and media manage which we will then “plug in” to our ES platform system (which allows for a mixture of sites – typically ecommerce or corporate but also various other configurations)

The overall solution will be based on existing shopify and wordpress examples as set out in the wordpress project here <https://www.freelancer.com/projects/php/Shopify-style-theme-editor/details>

The system will be a JS integration running as a full screen popup / overlay / page with an “Exit” or close button as you can see in both wordpress and shopify.

While it doesn’t have to be 100% the same – indeed some things have to be different – we are looking for a modern and professional solution, so you need to use a modern, generic theme for both functions of the system.

# Our system

Our system is multi site platform much like shopify except that as well as shops there are other types of sites – e.g. corporate but also community etc. it is a different system with a different background and options but just in terms of general description.

We have an existing theme system with a range of tools to amend content in the admin but the options are in different places and it can be confusing for a user to know what to edit – there are also aspects which cannot be edited at the moment – so we want to address this with the theme editor

So for example look here <https://hogmerch.com/>

1. The top part the page is largely static
2. The banners can be controlled via a banner system – but with no options
3. Latest products is ALWAYS linked to featured category 1
4. Then there are other hard coded banners and sliders
5. Then a footer which is partially hard coded and partially driven by dynamic content like pages and categories

Where we want to be is

1. General page options – as defined in the settings of the theme editor with a range of options which can vary by theme – supporting different content types like textbox, textarea, tick, dropdown (with values) and range slider (for sizing) coloru picker (for colours)
   1. General page options would include ability to apply colours etc
2. The menu would be driven by our existing menu widget or the older category drive approach as now
3. For al page sections – like shopify we will have blocks and the user can add and choose the block type
4. The banners will use the existing banner system but with a wordpress like streamlined approach – so banners can be updated in either place
5. Latest products – it will be the ability o select – via a browse tool like shopify – a category (in shopify this is a collection)
6. Various content types supported so image and text, image only, text only, carousell etc like shopify

Then a similar sort of approach on internal pages where we use existing blocks but sort of bring it into the editor so that options and content can be changed (for not at least) with the old tools or with the new one

**Its VERY IMPORTANT** that at least the basics work with the existing theme approach and then then customers can upgrade as needed or items can be hooked in – so what we need is a flexible theme model using e.g. JSON

# Code quality

The system should be PHP OOP using universal functionality that will work on any regular hosting platform (PHP 7+ but compatible with PHP 7.4 and 8) – you need to have logs and error handling so that we can investigate queries. Code needs to be centralised with good application of reuse. Classes and methods need to be suitably named and grouped.

Try to avoid raw queries so we can have a central place to apply fixes – ensure incoming parameters are checked for quotes and security problems.

In general everything will be PHP and where JS is used we prefer JQUERY with JSON – so e.g. functions will take a JSON object and return a JSON object.

Like PHP JS should also be OOP with JS objects – e.g. var result = MediaManager.Show({parameter:value,parameter2:value})

If (result.url)….

# Coding rules

Every developer has their own style but for consistency we want to ensure that at least you:

1. The system should be OOP
2. **NO GLOBAL VARIABLES please – use objects with properties (static if applicable)**
3. Properties are functions and DB columns should be title case (e.g. Surname, AddressLine1)
4. Table names (if applicable) lowercase (customers, orders)
5. Local variables are lower case or camel case ($user, $customerList)
6. Code should be “in scope” – i.e. don’t write code that relies on a variable in an include that’s buried somewhere – this is tied with global variables – instantiate objects within view of use etc
7. Other parameters – e.g. js/url should be titlecase
8. All code needs to be centralised as much as possible
9. Logs should be in the database or if in files in a data location
10. Config needs to be in a config folder for ease of releasing

# Integration

I am expecting to have a JS integration as is very common these days so something like

<script src=”modules/display/uitools/theme\_editor.js”></script>

<script>$(‘btn’).click(function(){ThemeEditor.Show()})</script>

<script src=”modules/display/uitools/media\_manager.js”></script>

<script>$(‘btn’).click(function(){MediaManager.Show()})</script>

Authentication will either be by tying in with our session or via a token system – either is fine so long as the system remains secure. I think the token system is less confusing for you.

# User experience

1. File uploading needs to have progress bars
2. Ideally it should be possible to undo mistakes where possible (restore previous version with a list of revisions)
3. If there is an issue there needs to be suitable error – e.g. file too large
4. All activity logged so we can investigate queries.

# Design

There is no specific requirement for design but I am expecting a modern and generic design much like shopify and wordpress has. The system needs to work on desktop and mobile (as much as possible given the limited space) – similar to those platforms.

# Media manager

The media manager is much like the wordpress one with the following general functionality

1. Upload files - very much like wordpress –
   1. drag and drop upload widget
   2. select to upload bulk files with a progress bar into the media manager –
   3. should also allow uploading of ZIP (with extract option) and
   4. importing from URL –
   5. Videos, images and embeds supported including SVG – embeds would have to be paste into a text box.
   6. PDF upload should also be supported with an option to generate thumbs from the first page.
2. Categorised - gallery categorisation - partially linked to file system directories - so you can filter for a category and place the images there
3. Media library - much like wordpress a list of all images in scope with search box and filters by date and type
4. Attributes - again much like wordpress - ability to delete the image, set attributes such as title and alt and class or have a description (for internal reference), show/copy teh URL for use elsewhere, optional display options
5. Option to select multiple items like wordpress (see parameters)
6. Some options to rotate/crop/adjust images like wordpress has and also resize - these should be returned in the response – In many case these parameters go into the URL but where not they should be separate values.
7. We will be using this popup with CK editor so it needs to work with it (sample to be supplied for us to implement) as well as a JS integration with a success callback function that has the filename etc for action
8. Ability to zip and bulk download files (tick / select all / select folder).
9. There should be a remove option – including batch remove
10. There should be some way to determine where / if the image has been used

Public methods: MediaManager.Show();

Parameters:

1. Root Folder – a root folder – e.g. Products – which serves as the main root for the results
2. Show Attributes- whether attributes can be specified or just basic file uploading
3. Media Types – what media types are supported – e.g. web image only, SVG, Video, Embed
4. Popup Mode – a property to determine what its for – e.g. for a CK editor, for a Product image, for a Gallery page, for a Banner.
5. Delete allowed

Returns

1. Local file path from domain
2. Full file path with URL
3. Any attributes and configuration - e.g. an alt text

This media manager will then be used

1. Directory – showing all categories and allowing the user to navigate and bulk/single upload files – allowing for management of content and also
2. With the theme editor below – to pick images where needed for it
3. With our own features to e.g. select an image to go on a banner or embed in a page.
4. With CK editor for editing page content.

# Theme editor

The media manager effectively has 3 tools:

1. Ability to import and export themes
   1. Import will add the theme to a list – where one theme is active – very similar to shopify with a version and date – see options in export
   2. Export will export a theme – it should be possible for the theme to be
      1. Just design templates
      2. Also settings – e.g. the name of the store and other settings defined
      3. Also content – i.e. pages, banners etc but not products or orders
      4. Also data – i.e. customers and orders and products.
      5. Full site – i.e. everything – the whole database dump and all images
   3. Both would basically be zips of all content – if there are too many entires (e.g. thousnda of products and images) then the system should detect this and gracefully fail to export and suggest a manual copy via support.
   4. Only one theme is active at a time but these can be swapped round
   5. Have a duplicate option like shopify has
2. Code editor – so much like the shopify one with a tree of templates grouped by folder and then a code style editor with code recognition (as good as you can supply)
   1. Should have something like “older versions” which allows you to go back
   2. Ctrl+s should save
   3. Line numbers are important
   4. In my system there is a default theme and then a per project theme – so the system should show the union of files in both systems but when a default theme page is modified it should be copied to the project theme automatically
   5. To date our themes are just PHP code – as part of this we will have to introduce a JSON configuration – see below
   6. Ability to add sections like shopify
   7. The “assets” upload will basically be the media manager
   8. An option to view the page being edited.
   9. On exit - warn the user of unsaved changes (if applicable)
3. UI editor (shopify called customise)
   1. It doesn’t have to be 100% like shopify but should generally be that appraoch
   2. Should have the different screen views like shopify (desktop, mobile, tablet)
   3. Should have undo like shopify
   4. Ctrl+s to save
   5. The preview should update with changes (well work out how to achieve this)
   6. Ability to select an area like shopify have – e.g. homepage or pages or products etc – I will give you a list
   7. It should be possible to add elements like with shopify – e.g. blocks, and sliders and collections – these elements should have a visibility option
   8. The options should be theme defined like shopify using a JSON configuration
   9. Drag drop reordering of elements.
   10. On exit warn the user of unsaved changes (if applicable).

Preview will be supported by basically calling a certain URL which will then load the specified theme.

You don’t need to design a theme but you do need to update our default theme to have all the functionality

Public methods: ThemeEditor.Show();

Parameters:

1. InitialPage
2. AllowPageChange
3. EditorMode

Returns

1. Cannot think of anything at this stage

Theme support

1. Our current themes are PHP based and we want to support them in the editor as much as possible – even if it’s a limited – well have to look at how this would work as the project unfolds
2. For new themes we want to support liquid – ideally to ultimately support shopify templates themselves (this would be in the future) – we need to see it working with existing and new examples with both liquid and PHP
3. A new theme would have a JSON like configuration (and also some will be in in the db) and we need to see samples of this

# Integration

Once we have tested the system with unit tests and standalone examples we will integrate it into our main platform inhouse – on your side we just need your support to answer questions or fix issues / tweak until the integration is complete.

# Authentication / data connection

Initially a centralised configuration will allow for connection to the database and you will access the db directly using those settings – we will need to pass a token which will identify which user is logged in. Note the token with parameters will need to allow for connection to multiple databases as is applicable for the multi site configuration.

# License / ownership

This has to be an Exceed IT Systems Ltd system where we have full ownership and can take this forward to resell or develop in any way we wish without any royalties – you are responsible to ensure any third party modules are royalty free and have no license charges or relevant restrictions.

# Milestones, Budget and Timeline

The agreed budget is $2500 with a delivery time of 25 days.

Obviously, we want this delivered ASAP but we can understand if it takes a bit longer – though if it’s a lot longer please discuss with us.

Unless there is some significant change in requirements though the budget is fixed – if you feel something is extra discuss it with us (don’t do it and then look to charge for it after) – in short any change to the price has to be agreed in advance.

As explained having half a system doesn’t help us so we propose the following:

1. We will start with the media manager (which is the smaller task and is a dependency for the theme editor anyway)
2. We will escrow $250 initially
3. Once we have a working prototype we will escrow another $500
4. Once we feel the prototype is working correctly you will give us the code and we will release $500 and escrow another $250 (so that $500 total is in escrow and you have been paid $500 at this stage)
5. We will integrate on our side while you work on the theme editor
6. For the theme editor we will do the same approach so that once the theme editor is ready we will already have the media manager installed and tested and we upon receipt of code and some basic checks we will release all but $500 of the $2500 agreed
7. Once we have fully integrated the theme editor (assumed to be a few days) and its all working well pay the remaining $500

**Note I am not releasing any funds until I have received, reviewed, and tested the code.**