

2015年度 卒業論文

# WordPress テンプレートテーマ制作ツールの 開発

大阪産業大学 デザイン工学部 情報システム学科  
情報教育システム研究室

12H041 小西大祐

# 目次

1	はじめに	1
2	目的	2
3	WordPress	3
4	テーマの制作方法	4
4.1	フルスクラッチ	4
4.2	既存テーマのカスタマイズ	4
4.3	WordPress テーマジェネレーター	4
5	システムの説明	6
5.1	動作環境	6
5.2	仕様	6
5.3	動作	9
6	考察	13
6.1	利点	13
6.2	問題点	13
7	まとめ	15
8	今後の課題	16
8.1	テーマの色を変更可能にする	16
8.2	制作するテーマのレスポンス対応	16
8.3	各画面の制作を可能にする	16
8.4	他の CSS フレームワークへの対応	16
付録 A	付録 1	19
A.1	付録 1.1	19
付録 B	ソースコード	19
B.1	フロントエンド	19
B.2	バックエンド	48

# 1 はじめに

近年、WordPress [1] を用いたサイトの制作が増加傾向である。WordPress を用いたサイト制作は一般的に、ヒアリング、デザイン、コーディング、WordPress の構築の流れで行われる。制作者は様々で、制作会社であれば全て自社で賄える場合が多いが、フリーランスなどの小規模な所では全て行うのが難しい場合がある。特にテンプレートテーマ\*1の制作においてヒアリングからデザインだけ、またはヒアリングからコーディング\*2までを行い、PHP のコードに置き換える部分を外部委託する事がよくある。しかしこの場合委託先を探すことに時間や費用がかかる他、情報の伝達ミスなどの問題が起こる可能性がある。そこで、そういった制作者でも短期間でWordPress のテーマを制作できるツールを開発する事でこの問題の解決を試みた。

第 2 章では本研究の目的について述べる。第 3 章では WordPress について述べる。第 4.3 章では WordPress のテーマジェネレーターについて述べる。第 5 章では本研究で開発したシステムの動作環境や仕様について述べる。第 6 章では本研究の考察について述べる。第 7 章には研究のまとめについて述べる。第 8 章では今後の課題についてまとめる。

---

\*1 ページを動的に生成する際に使用するページのテンプレート

\*2 静的なサイトの状態までのマークアップ

## 2 目的

本研究の目的は、なるべくコードを書くこと無く、自分好みの見た目のオリジナルテンプレートテーマを制作出来るシステムを開発することである。またこれを用いることで、これまでかかっていた WordPress を用いたサイト制作の時間を短縮することも目的である。なるべくコードを書くこと無く制作するためには、GUI を用いた操作を可能にすることが有効だと考える。また制作者の環境は様々なため、Web システムとして開発する事である程度対応する必要があると考える。これらを踏まえたシステムの開発を試みた。

### 3 WordPress

WordPress [1] は Automattic 社制作の PHP 製オープンソースのブログ/CMS プラットフォーム<sup>\*3</sup>である。企業のコーポレートサイトや教育機関のサイトから個人のブログなど幅広い Web サイトにおいて採用されている。その採用件数は近年増加傾向であり、今では Web 全体の約 24%<sup>\*4</sup>の Web サイトに採用されている。最新版は 4.3.1(2015 年 9 月 17 日現在) である。

テンプレートテーマ (以下「テーマ」という) を制作する事で、サイトの見た目を自分好みに変更する事が出来る。このテーマを元に動的にページが生成される。また、プラグインを追加することで機能の拡張を行い、多言語化対応や問い合わせフォームの設置などが行える。

---

<sup>\*3</sup> Web コンテンツを構成するテキストや画像などを管理し、サイトを構築するソフトウェアのこと

<sup>\*4</sup> W3Tech 2015 年のデータ

## 4 テーマの制作方法

主な WordPress のテーマの制作方法は以下である。

- フルスクラッチ
- 既存テーマのカスタマイズ
- テーマジェネレータの利用

なかでも既存テーマのカスタマイズを行う事が多い。既存テーマのカスタマイズを行う際、スターターテーマと呼ばれるテーマをベースにしたり、公式テーマディレクトリ [2] で公開されているものをカスタマイズして制作する。

### 4.1 フルスクラッチ

既存のものを利用すること無く、テーマを制作する方法である。既存のものから影響を受けること無く自由にテーマを制作する事が出来る。しかし、どうしても工数がかかってしまう他、WordPress 側で推奨されていないコードを用いるなどのセキュリティ面の問題が発生してしまう可能性がある。

### 4.2 既存テーマのカスタマイズ

\_`s` [3] や Bones [4]、bathe [5] といったスターターテーマ\*<sup>5</sup>や WordPress の公式テーマディレクトリ [2] に掲載されているテーマを用いて新たなテーマを制作する方法である。

また、こういった既存のテーマを元にコードを上書きする方法と、既存のテーマを元に子テーマを制作する方法があり、後者の方法を採用する事が多い。この方法は親テーマの機能を引き継ぎ、必要なコードのみ追加して子テーマを制作するため、工数が少なく済む。また、親テーマの更新の恩恵を受けることができるため、WordPress の新バージョンへの対応といった点で対応が行いやすい。

しかし、デザインや機能が元にしたテーマの影響を受けてしまう。また、公式テーマディレクトリには約 1678 件のテーマが掲載されており\*<sup>6</sup>、その中から自分の作りたい物を実現しやすいテーマを探すのは困難である。

### 4.3 WordPress テーマジェネレーター

WordPress テーマジェネレーターとは、WordPress のテンプレートテーマを生成出来るツールのことである。

テーマジェネレーターは Web 上で利用できるものと、ローカル環境にダウンロードして利用するものがある。既存の WordPress のテーマジェネレーターは、ローカル環境にダウンロードして使用するものが主流である。

WordPress のテーマジェネレーターは様々な機能を有しているが、主な機能として以下が列挙できる。

- レイアウトの変更機能
- 色の変更機能
- パーツの配置機能
- リアルタイムプレビュー

WordPress のテーマジェネレーターとして以下のシステムが存在している。

---

\*<sup>5</sup> テーマのベースとして利用することを考えて制作されたテーマのこと。ベーステーマとも呼ばれる。

\*<sup>6</sup> 2013 年 2 月現在の情報

#### WordPress Theme Generater

Web 上でリアルタイムプレビューしながら WordPress のテーマを制作できる。タイトルや色、レイアウトなどを編集することが可能。テーマファイルは既にある程度作られたものが用意されており、それをカスタマイズする事でテーマの制作を行う。WordPress の ver.2.1~3.7 までに対応したテーマしか制作できない。

#### WP THEME GENERATOR

Web 上で既存のテーマを元に編集を行いテーマの制作を行うことができるシステム。異なるデバイスを想定しながらテーマの制作を行うことができ、パーツの追加やフォントの変更など可能な事はとても多い。パーツの追加などを全て上部固定のメニューから操作を行うため、特にカラムの操作ではどこを編集しているのかが分かりにくい。

#### templatetoaster

WordPress や Joomla!<sup>\*7</sup>、Dorupal<sup>\*8</sup>をはじめ多くの CMS のテンプレートテーマ制作をサポートしている。レスポンス対応やメニュービルダーなど多くの機能がある。Windows(XP~10) しか対応していない。

#### Artisteer

元になるテーマが提案され、自分でパーツを置いてテーマを制作する事が可能なシステム。Windows しか対応していない。

---

<sup>\*7</sup> 多機能で世界中の多くの政府のサイトでも採用されているオープンソースの CMS

<sup>\*8</sup> ポータルサイトの構築に非常に適したオープンソースの CMS

## 5 システムの説明

本研究では、コーディングの知識の無い人でも自分好みのテーマを制作できるシステムを開発した。以下に本研究で開発したシステムの動作環境や仕様について述べる。

### 5.1 動作環境

本システムは以下の環境での動作を確認している。その他の環境での動作確認は行っていないため、動作の保証はできない。

#### 5.1.1 システム環境

- PHP ..... 5.6.7
- MySQL ..... 5.5.42
- Apache ..... 2.2.29
- WordPress ..... 4.3.1
- jQuery ..... 2.1.4
- underscore.js ..... 1.8.3
- Node.js [6] ..... 5.0.0
- Node モジュール
  - http ..... 0.0.0
  - url ..... 0.11.0
  - jschardet ..... 1.4.1
  - iconv ..... 2.1.11
  - fs-extra ..... 0.26.2
  - archiver ..... 0.21.0

#### 5.1.2 制作者の推奨環境

- Google Chrome 46+
- FireFox 42+

## 5.2 仕様

### 5.2.1 フロントページの制作が可能

WordPress のテーマを制作する際、最低限制作しなくてはならないファイルは以下の 2 つである。

- style.css
- index.php

style.css は WordPress で生成されるページのデザインを管理しているファイルである。index.php はメインのテンプレートであり、テンプレート階層<sup>\*9</sup> [7] に示されているとおり、全てのページの専用ファイルの代わりとして利用することができる。(テンプレート階層を図 1 に示す) なお、index.php よりも上位の専用ファイルがある

---

<sup>\*9</sup> ページ生成の際に使用するファイルを選択する階層

場合はそちらが優先して使用される。

今回は複雑なレイアウトを考えて組むことが多い front-page.php を作成し編集可能にした。テーマを制作するとなると front-page.php の他、archive.php や single.php などのファイルも変更する必要がある。しかしながらページの切り替えにファイルを対応させることが難しかった為、本システムではプロトタイプとして front-page.php とそれに関連するファイルを含め、以下のファイルを制作可能なファイルとした。

- header.php
- footer.php
- style.css
- functions.php
- front-page.php

### 5.2.2 制作するテーマのベース

完全にオリジナルの WordPress テーマとなると設計が難しく、また後のメンテナンス性や拡張性が落ちる可能性が有るため、Automattic, inc が提供している `_s` [3] をベースとなるテーマに採用した。これによりテーマ制作者が後にテーマを拡張する必要が出てきた際、ドキュメントをもとに拡張できるようになった。

また、現在 Web 制作の現場でよく利用されている CSS フレームワークである Bootstrap3 [8] をベースのテーマに組み込んだ。これによりテーマ内でのグリッドレイアウトの実装やボタン、モーダルパネルなどの UI コンポーネントを容易に追加を可能にした。本システムでは、この `_s` に Bootstrap3 を組み込んだものをベーステーマとし、直接書き換える形でオリジナルのテーマ生成を行う。

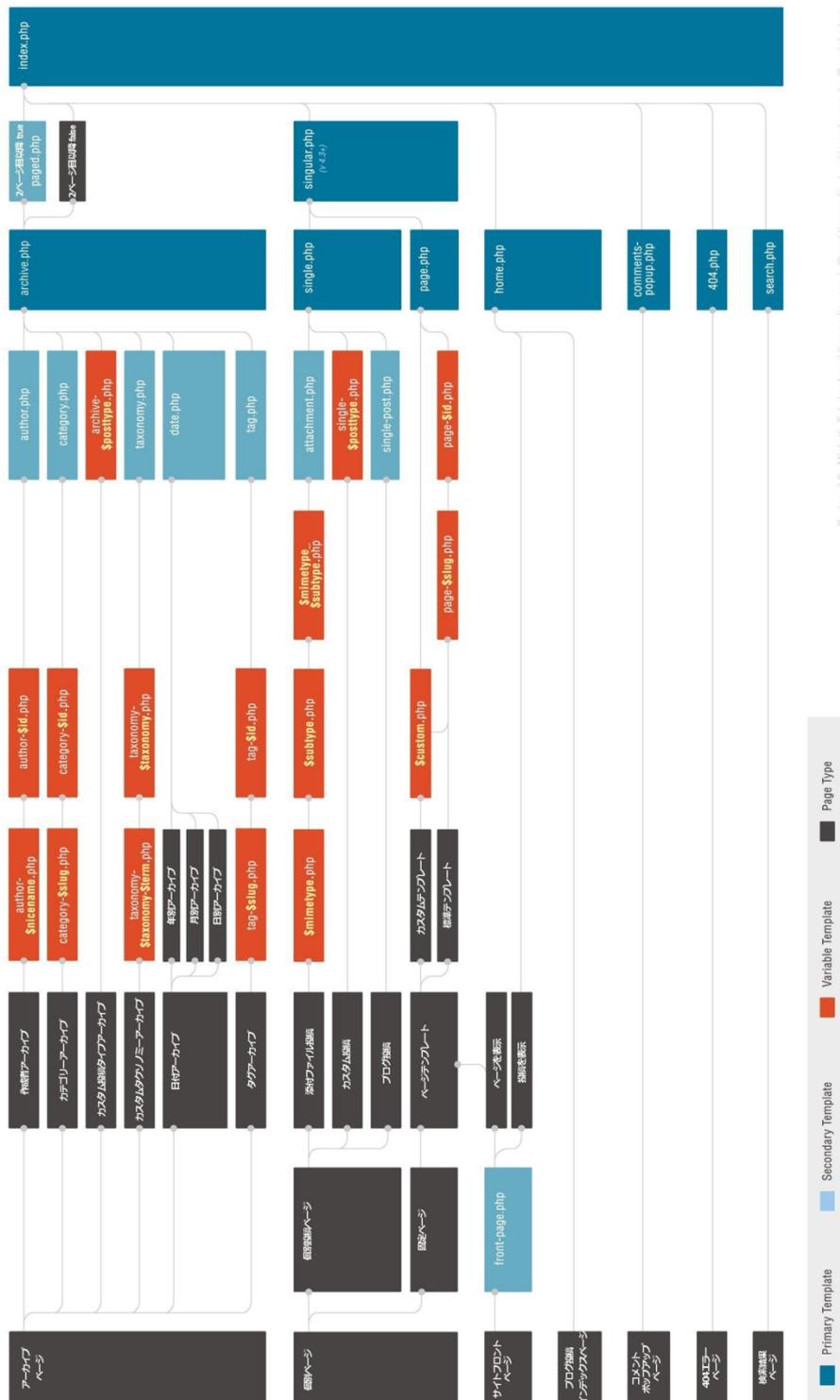
### 5.2.3 生成されるテーマ

本システムで生成されるテーマでは、以下の機能を導入する事が出来る。

- カスタムメニュー
- ウィジェット
- サムネイル

これ以外の機能はプラグイン等を使用して実装する事になる。

また、本システムによって生成されるテーマは、テーマ名を設定することができず、全て「wp-themegen」になってしまう。そのため、制作者はダウンロード後に style.css に記述されているテーマ名を編集する必要がある。



Designed by Michelle Schulte at markthemedia.com | @markthemedia | wphierarchy.com by @amiribrahim

図 1 WordPress のテンプレート階層 (WordPressCodex から引用) [7] なお、情報量が多く読みやすくするため横向きにしている

## 5.3 動作

### 5.3.1 ファイルへの書き込みまでの処理の流れ

パーツの追加からファイルへの書き込み、ページを更新するまでの動作の流れは以下のステップとする。

1. 制作者がパーツをドロップする
2. 追加された情報をバックエンドに渡す
3. 情報を受け取る
4. 情報を最適な形に編集する
5. ファイルに書き込む
6. プレビューエリア更新する

この動作の流れを図 2 に示す。

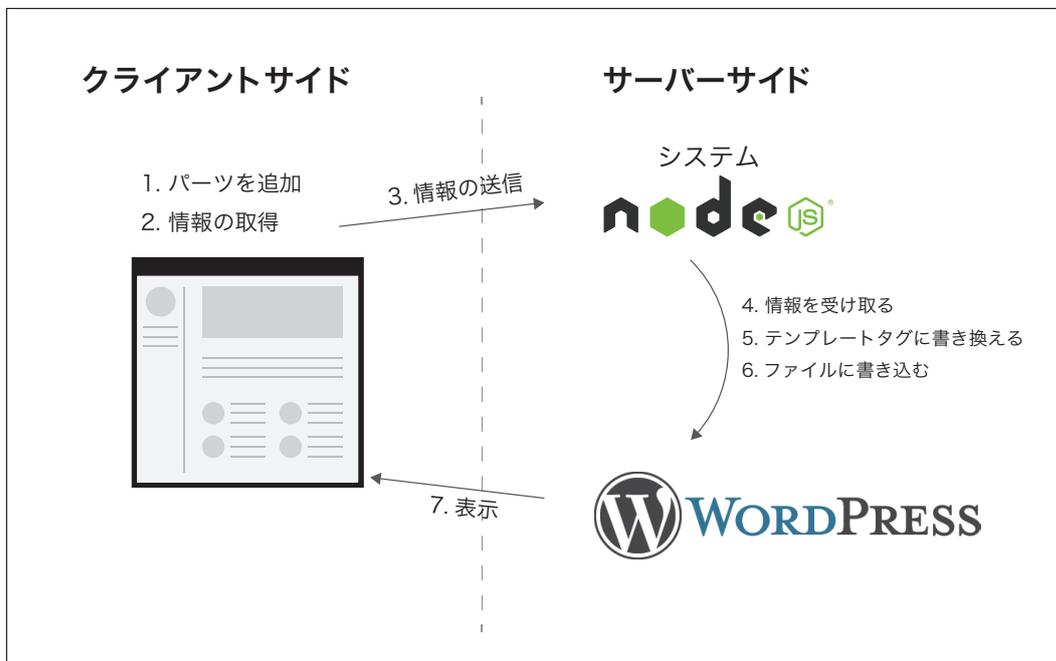


図 2 パーツを追加しファイルを書き込み、ページを更新するまでの処理の流れ。クライアント側で制作者がパーツを追加。その情報を取得しサーバー側に送信。Node.js を用いたシステムで情報を受け取る。受け取った情報を必要に応じて WordPress のテンプレートタグに置き換えて、ファイルに書き込む。それを WordPress を用いて表示する。

### 5.3.2 プレビューエリア

実際に制作しているテーマが見れた方が編集時に便利な為プレビューエリアを設けた。制作時と実際の環境との差異をなるべく無くす為に、バックエンドで WordPress を動かして制作中のテーマを有効化したうえで実行結果を表示している。また、様々な投稿やウィジェットが入ってきたときにも対応できるよう、テーマユニットテストデータ\*10をインポートしている。これにより、様々な投稿やウィジェット、メニューなどのパーツを追加時に瞬時に結果を見ながらテーマを制作する事が可能である。

### 5.3.3 GUI による直感的なパーツの挿入

直感的な操作を可能にするため HTML5 の Drag & Drop API [9] を利用し、見出しや画像をはじめ WordPress 特有のパーツをドラッグアンドドロップで挿入可能にする UI を実装した。これにより、PHP などのコーディングの知識がない人でもテーマへのパーツの追加を容易にした。制作者にパーツを追加する際に追加するパーツの情報を入力してもらう。それを予め用意しておいたパーツ出力用のコードのテンプレートに当てはめ、パーツを追加した部分にパーツの要素を追加する事ができる。

---

\*10 テーマのテストに用いる投稿やウィジェットなどのデータ



図 3 左側のサイドバーから色の濃い四角を右側のカラムの中にドラッグアンドドロップで追加する。



図 4 見出しを追加したときの設定画面。見出しのレベルや表示する見出しの内容を入力する。背景の色が違うのは、設定パネルに注視させるためにマスクをかけているためである。

#### 5.3.4 パーツ情報の送信

フロント側からサーバー側のシステムに対して追加するパーツ情報の送信を送信している。送信には Ajax<sup>\*11</sup> [10] を用いて行っている。これによりページ遷移する事無くサーバー側のシステムにデータを送信している。

#### 5.3.5 テーマファイルの生成

ファイルの生成はプレビューにも関わり、制作者をあまり待たせず処理を行う必要があったため、ノンブロッキングで非同期処理が行えることからサーバーサイドに Node.js を採用した。フロント側から Ajax により POST 送信された情報を Node.js の http モジュールを用いて取得する。取得したデータを整形した後、WordPress の テーマファイルに書き込むことでテーマファイルの生成処理を実現している。

#### 5.3.6 テーマファイルの圧縮

WordPress のテンプレートテーマのファイルはその仕様上とても数が多く、1つ1つのファイルをダウンロードするのは困難である。そのためテーマファイルの圧縮ファイルを生成する機能を実装した。これには Node.js の archive モジュールを用いた。ダウンロードボタンのクリック時に現在のテーマファイルの中身を全てまとめ、それを zip 圧縮したファイルを生成し、ダウンロードするという機能を実装した。

---

<sup>\*11</sup> Web ブラウザ内で HTTP の非同期通信を行う処理

## 6 考察

本研究では、PHP などのコーディングの知識が無い人でも、簡単に自分の制作したいオリジナルテーマを制作できるツールを Web 上に制作した。以下に本研究で開発したシステムの考察を示す。

### 6.1 利点

本研究で開発したシステムの利点を示す。

#### 6.1.1 直感的な操作でテーマが制作できる

コーディング知識の乏しい制作者でも、コードを書くこと無く GUI による直感的な操作のみで容易にパーツを追加し、WordPress のテンプレートテーマを制作する事が出来る。

また、実際の環境に近い状況でテーマの制作が出来るため、より多くの状況に対応出来るテーマの制作が出来ると考える。

#### 6.1.2 環境構築の手間を省く事が出来る

WordPress の開発を行う上で面倒なのが、環境の構築である。従来は仮想サーバー等を用意する他、類似システムのようにシステムをダウンロードする必要がある。しかし、本システムでは Web 上で制作を行うことができるため仮想環境などの開発環境を用意する必要が無い。その代わりに、Google Chrome や FireFox といったモダンなブラウザを用意する必要があり、それを用意するだけでテーマの制作にとりかかる事ができる。これにより OS などの環境に殆ど依存することなく WordPress のテーマファイルを制作する事ができる。

#### 6.1.3 コンポーネントの追加が容易

ベースのテーマに CSS フレームワークの Bootstrap を組み込んである。そのため、Bootstrap の公式サイトを見る事によりボタンやナビゲーション、モーダルパネルなど必要に応じてコンポーネントの追加を行うことができる。これにより、容易に WordPress のテーマを制作する事が可能である。

#### 6.1.4 モックアップの制作にも利用できる

Web 制作の現場では、ワイヤーフレームや Web カンプ、CSS フレームワークを利用したモックアップ等を用いてクライアントの考えとのズレを少なくするという事がしばしば有る。しかしながらこういったものではシステムを交えた物を提供することは難しい。そこで、本研究で制作したシステムを利用する事で、本来より早い段階でシステムも交えた WordPress サイトのモックアップをクライアントに提供することが出来るのではないかと考える。

### 6.2 問題点

本研究で開発したシステムと生成されるテーマファイルの問題点を示す。

#### 6.2.1 `.s` の機能を越えた機能実装は行えない

本システムではベースに使用した `.s` で有効化されていたカスタムメニュー<sup>\*12</sup>やウィジェット<sup>\*13</sup>といった必要最低限の機能を利用したテーマの制作しか行えない。そのため、システム上で他の機能を有効化したり、PHP を記

<sup>\*12</sup> 管理画面から項目を設定することができるメニューの機能

<sup>\*13</sup> 管理画面からカテゴリーやタグのリストなどを追加出来る機能

述し既存の機能を修正したり、新たに機能を追加することはできない。そういった場合は一度制作を終わらせてダウンロードした後に拡張してもらう必要がある。

#### 6.2.2 複数人での同時作業が出来ない

本システムでは作業を行う制作者が1人と仮定している。そのため既に作業中の制作者がいるにも関わらず、複数人の制作者が利用を始めてしまうと、テーマの初期化が行われ作業途中のテーマファイルが上書きされてしまう。

現状その回避を行う為の処理が実装できていないため、そういった事態を回避できる仕組みを考え、実装する必要があると考える。

#### 6.2.3 他人との共有が出来ない

制作の現場では、制作の途中でメンバーとの共有を行い意見を仰ぐということが少なくはない。しかしながら、本システムでは制作途中のテーマの情報を第三者に共有する機能を有していない。そのため、今後そういったことにも対応できるようにする必要があると考える。

## 7 まとめ

本研究を行うにあたり、WordPress のテーマ制作はとても時間がかかり、制作者側も PHP のコードに対する理解が無いと制作が困難といった背景があった。これの解決のため、本研究では Web 上に制作者の環境による影響を受けず、GUI による簡単な操作で WordPress のテンプレートテーマを制作出来るツールを構築した。本システムを利用することで、PHP などのコーディングの知識のない制作者でも簡単に WordPress のテーマを制作し利用する事ができると考えられる。しかしながら、本システムを用いることでテーマ制作にかかっていた工数を短縮するという点の検証できておらず解決したとはいえない。

## 8 今後の課題

### 8.1 テーマの色を変更可能にする

現状、制作時にテーマ内の色を変更する事が出来ない。現状色の変更をしたい場合、自前の WordPress の環境でテーマを有効化した後、テーマカスタマイザー<sup>\*14</sup>を利用する事でテーマの色を変更する事が可能である。しかしながら、それではあまり実用的なテーマでは無いため、今後カラーピッカーなどで容易に色を変更できる機能を実装する必要がある。

### 8.2 制作するテーマのレスポンス対応

近年スマートフォンやタブレットといった PC 以外からのサイトへのアクセスが増えている。そのため制作するサイトはどのデバイスからも見やすいものにするためレスポンス対応を行う事が多い。しかしながら今回構築したシステムにより制作されるテーマはレスポンス対応を行っていない。

テーマのベースに Bootstrap を組み込んでいるため、カラムの実装時にスマートフォンやタブレット用のカラム幅を指定するスタイルを書き込めるようにすればよいと考える。しかし、現在各端末からの見え方を反映し最適な形でプレビューや編集を行う方法が見つかっておらず機能の実装できていないため今後の課題と考える。

### 8.3 各画面の制作を可能にする

今回の制作したシステムではフロントページしか制作することが出来ない。しかしながら、WordPress では、フロントページ以外にも投稿一覧用のページ、投稿記事用のページ、固定ページなどそれぞれに対応したファイルがある。それを元にページがレンダリングされるため、それぞれのファイルが必要となる。そのため制作出来る画面を切り替える機能の実装が必要だと考える。

### 8.4 他の CSS フレームワークへの対応

今回は多くの制作者が利用しているため Bootstrap を採用した。しかしながら、制作の現場では Foundation<sup>\*15</sup>や Material Design Lite<sup>\*16</sup>など他の CSS フレームワークも使用されている。こういった他の人気のあるフレームワークにも対応する事でより多くのサイトのパターンにも対応でき、多くの人に利用してもらえるようになると思う。

---

<sup>\*14</sup> WordPress の ver.4.3 から実装されたテーマの編集機能

<sup>\*15</sup> Zurb 社製の CSS フレームワーク

<sup>\*16</sup> Google 社製のフレームワーク

## 謝辞

本研究を進める上で、ご指導頂いた大垣斉准教授、情報教育システム研究室のメンバー、卒業生の方々には深く感謝致します。

## 参考文献

- [1] Automatic inc. Wordpress. <https://ja.wordpress.org/>.
- [2] Wordpress テーマディレクトリ. <https://ja.wordpress.org/themes/>.
- [3] Automatic inc. .s. <http://underscores.me/>.
- [4] eddiemachado. bone. <https://github.com/eddiemachado/bones>.
- [5] ixkaito. bathe. <http://github.com/wp-bathe/bathe>.
- [6] 松島造道. はじめての Node.js -サーバーサイド JavaScript で Web アプリを開発する. ソフトバンククリエイティブ, 2013.
- [7] テンプレート階層—wordpress codex 日本語版. <https://wpdocs.osdn.jp/%E3%83%86%E3%83%B3%E3%83%97%E3%83%AC%E3%83%BC%E3%83%88%E9%9A%8E%E5%B1%A4>.
- [8] Bootstrap. <http://getbootstrap.com/>.
- [9] ネイティブ html5 ドラッグ&ドロップ - html5 rocks. <http://www.html5rocks.com/ja/tutorials/dnd/basics/>.
- [10] Ajax とは | asynchronous javascript + xml - 意味/解説/説明/定義 : it 用語辞典. <http://e-words.jp/w/Ajax.html>.

(2015 年 11 月 30 日現在)

## 付録 A 付録 1

### A.1 付録 1.1

## 付録 B ソースコード

### B.1 フロントエンド

```
<!DOCTYPE html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>WordPress テーマのオーサリングツール</title>
    <meta name="description" content="WordPress のテーマを GUI を使って制作するツール">
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css">
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <header class="l-gloval-header">
      <div id="partsnav-btn"><span class="menu-line"></span><span class="menu-line"></span><span class="
      <div class="l-brand-area">
        <h1 class="brand">WP-Themegen</h1>
      </div>
      <div class="header-items">
        <button class="btn-download"><i class="fa fa-download"></i>ダウンロード</button>
      </div>
    </header>
    <main class="l-maincontent">
      <iframe id="result-iframe" src="../wptg/" width="100%" height="100%"></iframe>
    </main>
    <aside class="l-partsnav">
      <h2>パーツ (仮)</h2>
      <ul class="part-items">
        <li id="loop" draggable="true" class="part-item"><i class="part-item__icon fa fa-repeat"></i><span>ループ</span></li>
        <li id="widget" draggable="true" class="part-item"><img src="" alt="ウィジェットの画像"><span class="part-item__caption">ウィジェット</span></li>
        <li id="custommenu" draggable="true" class="part-item"><i class="part-item__icon fa fa-list"></i><span>カスタムメニュー</span></li>
      </ul>
      <h2>ビジュアル (仮)</h2>
      <ul class="part-items">
        <li id="headerTitle" draggable="true" class="part-item"><i class="part-item__icon fa fa-header"></i><span>ヘッダー</span></li>
      </ul>
    </aside>
  </body>
</html>
```

見出し</span></li>

```
<li id="picture" draggable="true" class="part-item"><i class="part-item__icon fa fa-picture-o">
```

画像</span></li>

```
<li id="code" draggable="true" class="part-item"><i class="part-item__icon fa fa-code"></i><spa
```

```
<li id="article" draggable="true" class="part-item"><i class="part-item__icon fa fa-edit"></i></>
```

記事</span></li>

```
</ul>
```

```
</aside>
```

```
<div class="modal">
```

```
<header class="modal-header">
```

```
<h2 class="modal-title"></h2>
```

```
</header>
```

```
<div class="modal-content"></div>
```

```
<footer class="modal-footer">
```

```
<button id="submitBtn" class="submit-btn">決定</button>
```

```
</footer>
```

```
</div>
```

```
<div class="modal-overlay"></div>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.3/underscore-min.js"></script>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
```

```
<script src="js/script.min.js"></script>
```

```
<script id="temp_headtitle" type="text/template"><h<%- level %>><%= content %></h<%= level %>></scr
```

```
<script id="temp_post" type="text/template"><p><%- content %></p></script>
```

```
<script id="temp_widget" type="text/template">
```

```
<div id="secondary" role="complementary" class="widget-area"><?php dynamic_sidebar( 'sidebar-1' )
```

```
</script>
```

```
<script id="temp_customnav" type="text/template">
```

```
<nav id="site-navigation" role="navigation" class="main-navigation">
```

```
<button aria-controls="primary-menu" aria-expanded="false" class="menu-toggle">
```

```
<?php esc_html_e( 'Primary Menu', 'wp_themegen' ); ?>
```

```
<?php wp_nav_menu( array( 'theme_location' => 'primary', 'menu_id' => 'primary-menu' ) );
```

```
</button>
```

```
</nav>
```

```
</script>
```

```
<script id="temp_loop" type="text/template">
```

```
<?php
```

```
if ( have_posts() ) :
```

```
while ( have_posts() ) : the_post();
```

```
?>
```

```
<?php get_template_part( 'template-parts/content', get_post_format() ); ?>
```

```
<?php
```

```
endwhile;
```

```
the_posts_navigation();
```

```

        else :
            get_template_part( 'template-parts/content', 'none' );
        endif;
    ?>
</script>
</body>
</html>

/*! normalize.css v3.0.2 | MIT License | git.io/normalize */
/**
 * 1. Set default font family to sans-serif.
 * 2. Prevent iOS text size adjust after orientation change, without disabling
 *    user zoom.
 */
html {
    font-family: sans-serif;
    /* 1 */
    -ms-text-size-adjust: 100%;
    /* 2 */
    -webkit-text-size-adjust: 100%;
    /* 2 */
}

/**
 * Remove default margin.
 */
body {
    margin: 0;
}

/* HTML5 display definitions
   ========================================================================= */
/**
 * Correct 'block' display not defined for any HTML5 element in IE 8/9.
 * Correct 'block' display not defined for 'details' or 'summary' in IE 10/11
 * and Firefox.
 * Correct 'block' display not defined for 'main' in IE 11.
 */
article,
aside,
details,
figcaption,
figure,

```

```

footer,
header,
hgroup,
main,
menu,
nav,
section,
summary {
    display: block;
}

/**
 * 1. Correct 'inline-block' display not defined in IE 8/9.
 * 2. Normalize vertical alignment of 'progress' in Chrome, Firefox, and Opera.
 */
audio,
canvas,
progress,
video {
    display: inline-block;
    /* 1 */
    vertical-align: baseline;
    /* 2 */
}

/**
 * Prevent modern browsers from displaying 'audio' without controls.
 * Remove excess height in iOS 5 devices.
 */
audio:not([controls]) {
    display: none;
    height: 0;
}

/**
 * Address '[hidden]' styling not present in IE 8/9/10.
 * Hide the 'template' element in IE 8/9/11, Safari, and Firefox < 22.
 */
[hidden],
template {
    display: none;
}

```

```

/* Links
===== */
/**
 * Remove the gray background color from active links in IE 10.
 */
a {
  background-color: transparent;
}

/**
 * Improve readability when focused and also mouse hovered in all browsers.
 */
a:active,
a:hover {
  outline: 0;
}

/* Text-level semantics
===== */
/**
 * Address styling not present in IE 8/9/10/11, Safari, and Chrome.
 */
abbr[title] {
  border-bottom: 1px dotted;
}

/**
 * Address style set to 'bolder' in Firefox 4+, Safari, and Chrome.
 */
b,
strong {
  font-weight: bold;
}

/**
 * Address styling not present in Safari and Chrome.
 */
dfn {
  font-style: italic;
}

/**
 * Address variable 'h1' font-size and margin within 'section' and 'article'

```

```

* contexts in Firefox 4+, Safari, and Chrome.
*/
h1 {
    font-size: 2em;
    margin: 0.67em 0;
}

/**
 * Address styling not present in IE 8/9.
 */
mark {
    background: #ff0;
    color: #000;
}

/**
 * Address inconsistent and variable font size in all browsers.
 */
small {
    font-size: 80%;
}

/**
 * Prevent 'sub' and 'sup' affecting 'line-height' in all browsers.
 */
sub,
sup {
    font-size: 75%;
    line-height: 0;
    position: relative;
    vertical-align: baseline;
}

sup {
    top: -0.5em;
}

sub {
    bottom: -0.25em;
}

/* Embedded content
===== */

```

```

/**
 * Remove border when inside 'a' element in IE 8/9/10.
 */
img {
  border: 0;
}

/**
 * Correct overflow not hidden in IE 9/10/11.
 */
svg:not(:root) {
  overflow: hidden;
}

/* Grouping content
===== */
/**
 * Address margin not present in IE 8/9 and Safari.
 */
figure {
  margin: 1em 40px;
}

/**
 * Address differences between Firefox and other browsers.
 */
hr {
  box-sizing: content-box;
  height: 0;
}

/**
 * Contain overflow in all browsers.
 */
pre {
  overflow: auto;
}

/**
 * Address odd 'em'-unit font size rendering in all browsers.
 */
code,
kbd,

```

```

pre,
samp {
    font-family: monospace, monospace;
    font-size: 1em;
}

/* Forms
===== */
/**
 * Known limitation: by default, Chrome and Safari on OS X allow very limited
 * styling of 'select', unless a 'border' property is set.
 */
/**
 * 1. Correct color not being inherited.
 *    Known issue: affects color of disabled elements.
 * 2. Correct font properties not being inherited.
 * 3. Address margins set differently in Firefox 4+, Safari, and Chrome.
 */
button,
input,
optgroup,
select,
textarea {
    color: inherit;
    /* 1 */
    font: inherit;
    /* 2 */
    margin: 0;
    /* 3 */
}

/**
 * Address 'overflow' set to 'hidden' in IE 8/9/10/11.
 */
button {
    overflow: visible;
}

/**
 * Address inconsistent 'text-transform' inheritance for 'button' and 'select'.
 * All other form control elements do not inherit 'text-transform' values.
 * Correct 'button' style inheritance in Firefox, IE 8/9/10/11, and Opera.
 * Correct 'select' style inheritance in Firefox.

```

```

*/
button,
select {
    text-transform: none;
}

/**
 * 1. Avoid the WebKit bug in Android 4.0.* where (2) destroys native 'audio'
 *    and 'video' controls.
 * 2. Correct inability to style clickable 'input' types in iOS.
 * 3. Improve usability and consistency of cursor style between image-type
 *    'input' and others.
 */
button,
html input[type="button"],
input[type="reset"],
input[type="submit"] {
    -webkit-appearance: button;
    /* 2 */
    cursor: pointer;
    /* 3 */
}

/**
 * Re-set default cursor for disabled elements.
 */
button[disabled],
html input[disabled] {
    cursor: default;
}

/**
 * Remove inner padding and border in Firefox 4+.
 */
button::-moz-focus-inner,
input::-moz-focus-inner {
    border: 0;
    padding: 0;
}

/**
 * Address Firefox 4+ setting 'line-height' on 'input' using '!important' in
 * the UA stylesheet.

```

```

*/
input {
  line-height: normal;
}

/**
 * It's recommended that you don't attempt to style these elements.
 * Firefox's implementation doesn't respect box-sizing, padding, or width.
 *
 * 1. Address box sizing set to 'content-box' in IE 8/9/10.
 * 2. Remove excess padding in IE 8/9/10.
 */
input[type="checkbox"],
input[type="radio"] {
  box-sizing: border-box;
  /* 1 */
  padding: 0;
  /* 2 */
}

/**
 * Fix the cursor style for Chrome's increment/decrement buttons. For certain
 * 'font-size' values of the 'input', it causes the cursor style of the
 * decrement button to change from 'default' to 'text'.
 */
input[type="number"]::-webkit-inner-spin-button,
input[type="number"]::-webkit-outer-spin-button {
  height: auto;
}

/**
 * 1. Address 'appearance' set to 'searchfield' in Safari and Chrome.
 * 2. Address 'box-sizing' set to 'border-box' in Safari and Chrome
 *    (include '-moz' to future-proof).
 */
input[type="search"] {
  -webkit-appearance: textfield;
  /* 1 */
  /* 2 */
  box-sizing: content-box;
}

/**

```

```

* Remove inner padding and search cancel button in Safari and Chrome on OS X.
* Safari (but not Chrome) clips the cancel button when the search input has
* padding (and 'textfield' appearance).
*/
input[type="search"]::-webkit-search-cancel-button,
input[type="search"]::-webkit-search-decoration {
  -webkit-appearance: none;
}

/**
 * Define consistent border, margin, and padding.
 */
fieldset {
  border: 1px solid #c0c0c0;
  margin: 0 2px;
  padding: 0.35em 0.625em 0.75em;
}

/**
 * 1. Correct 'color' not being inherited in IE 8/9/10/11.
 * 2. Remove padding so people aren't caught out if they zero out fieldsets.
 */
legend {
  border: 0;
  /* 1 */
  padding: 0;
  /* 2 */
}

/**
 * Remove default vertical scrollbar in IE 8/9/10/11.
 */
textarea {
  overflow: auto;
}

/**
 * Don't inherit the 'font-weight' (applied by a rule above).
 * NOTE: the default cannot safely be changed in Chrome and Safari on OS X.
 */
optgroup {
  font-weight: bold;
}

```

```

/* Tables
===== */
/**
 * Remove most spacing between table cells.
 */
table {
    border-collapse: collapse;
    border-spacing: 0;
}

td,
th {
    padding: 0;
}

body {
    font-size: 16px;
    color: #464646;
}

html,
body {
    height: 100%;
}

.l-gloval-header {
    position: fixed;
    top: 0;
    left: 0;
    width: 98%;
    height: 80px;
    margin-bottom: 20px;
    padding: 0 1%;
    background-color: #fbfbfb;
    box-shadow: 0 2px 3px #dedede;
    overflow: hidden;
}

.l-brand-area {
    margin: 15px 0;
    float: left;
}

```

```

.l-maincontent {
  width: 100%;
  height: 100%;
  margin: 0 auto;
}

.l-partsnav {
  width: 200px;
  position: fixed;
  left: -220px;
  top: 80px;
  padding: 0 10px;
  background-color: #D8DAD9;
}

.l-partsnav.is-active {
  position: fixed;
}

#partsnav-btn {
  display: inline-block;
  width: 40px;
  float: left;
  margin: 15px 0;
  margin-right: 30px;
}

.menu-line {
  display: block;
  border: 2px solid #252525;
  margin: 10px 0;
}

.part-items {
  list-style-type: none;
  padding: 0;
  border-bottom: 1px solid #cacaca;
  overflow: hidden;
  padding-bottom: 20px;
}

.part-item {
  width: 48%;

```

```

margin: 5px 2% 5px 0;
padding: 10px;
float: left;
box-sizing: border-box;
background-color: #bec1c0;
transition: background-color .3s ease-in-out;
cursor: move;
}
.part-item:hover {
background-color: #d8dad9;
}
.part-item:nth-child(even) {
margin-right: 0;
}
.part-item__icon {
display: block;
font-size: 50px;
text-align: center;
}
.part-item__caption {
display: block;
font-size: 13px;
text-align: center;
}

[draggable] {
-moz-user-select: none;
-khtml-user-select: none;
-webkit-user-select: none;
-ms-user-select: none;
user-select: none;
/* Required to make elements draggable in old WebKit */
-khtml-user-drag: element;
-webkit-user-drag: element;
}

.columns {
border: 1px solid #d1d1d1;
margin: 10px;
width: 100%;
min-height: 50px;
font-size: 12px;
}

```

```

.modal {
  display: none;
  position: fixed;
  left: 25%;
  top: 0px;
  width: 50%;
  z-index: 1;
  margin: 0 auto;
  background-color: #fff;
  border: 1px solid #d1d1d1;
  border-radius: 3px;
  box-shadow: rgba(220, 220, 220, 0.4) 0px 3px 6px 3px;
}
.modal-active {
  display: block;
}
.modal-header {
  padding: 10px;
  border-bottom: 1px solid #d1d1d1;
}
.modal-title {
  font-size: 24px;
  margin: 0;
}
.modal-content, .modal-footer {
  padding: 10px;
}
.modal .modal-item {
  margin: 10px 0;
}
.modal .submit-btn {
  display: inline-block;
  background-color: #2ED24E;
  border: 2px solid #2ED24E;
  transition: background-color .3s ease-in-out, color .3s ease-in-out;
  padding: .7rem 1rem;
  color: #fff;
  font-size: 16px;
  cursor: pointer;
}
.modal .submit-btn:hover {
  background-color: #fff;
}

```

```

    color: #2ED24E;
}
.modal-overlay {
    display: none;
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(100, 100, 100, 0.6);
}
.modal-overlay-active {
    display: block;
}

.brand {
    font-size: 20px;
    margin: 10px 0;
}

.header-items {
    float: right;
    margin: 15px 0;
}

.gridbtn {
    margin-right: 5px;
    background-color: #fff;
    border: none;
    font-size: 2em;
    padding: 3px;
}

.btn-download {
    display: inline-block;
    background-color: #2ED24E;
    border: 2px solid #2ED24E;
    transition: background-color .3s ease-in-out, color .3s ease-in-out;
    padding: .7rem 1rem;
    color: #fff;
    font-size: 16px;
    cursor: pointer;
}

```

```

.btn-download:hover {
  background-color: #fff;
  color: #2ED24E;
}

.column-active {
  border: 1px solid #2ED24E;
}

(function() {
  var CustomNav, HeaderTitle, Loop, Post, TempPart, Widget, handleDragEnd, handleDragEnter, handleDragL
  extend = function(child, parent) { for (var key in parent) { if (hasProp.call(parent, key)) child[k
  hasProp = {}.hasOwnProperty;

  TempPart = (function() {
    function TempPart() {}

    TempPart.prototype.storeTempData = function(tempid, data) {
      return this.partData = _.template($('#tempid').html())(data);
    };

    TempPart.prototype.appendCode = function(part) {
      return iframe.find('.columns').append(part);
    };

    return TempPart;

  })();

  HeaderTitle = (function(superClass) {
    extend(HeaderTitle, superClass);

    function HeaderTitle() {
      this.partData = {
        level: 0,
        content: ''
      };
      this.setPartData();
      this.appendCode(this.storeTempData('#temp_headtitle', this.partData));
    }

    HeaderTitle.prototype.getPartData = function() {
      this.level = $('#.modal-item__lebel').find('option:selected').val();

```

```

    return this.content = $('>modal-item__text').val();
};

HeaderTitle.prototype.setPartData = function() {
    this.getPartData();
    if (!(_.isEmpty(this.level) && _.isEmpty(this.content))) {
        this.partData.level = this.level;
        return this.partData.content = _.escape(this.content);
    } else {
        return console.log('empty');
    }
};

return HeaderTitle;

})(TempPart);

Post = (function(superClass) {
    extend(Post, superClass);

    function Post() {
        this.partData = {
            content: ''
        };
        this.setPartData();
        this.appendCode(this.storeTempData('#temp_post', this.partData));
    }

    Post.prototype.getPartData = function() {
        return this.content = $('>modal-item__textarea').val();
    };

    Post.prototype.setPartData = function() {
        this.getPartData();
        if (!_.isEmpty(this.content)) {
            return this.partData.content = _.escape(this.content);
        } else {
            return console.log('empty');
        }
    };

    return Post;

```

```

})(TempPart);

Widget = (function(superClass) {
  extend(Widget, superClass);

  function Widget() {
    this.partData = {
      content: ''
    };
    this.setPartData();
    this.appendCode(this.storeTempData('#temp_widget', this.partData));
  }

  Widget.prototype.getPartData = function() {
    return this.content = $('#modal-item__textarea').val();
  };

  Widget.prototype.setPartData = function() {
    this.getPartData();
    if (!_.isEmpty(this.content)) {
      return this.partData.content = _.escape(this.content);
    } else {
      return console.log('empty');
    }
  };

  return Widget;
})(TempPart);

CustomNav = (function(superClass) {
  extend(CustomNav, superClass);

  function CustomNav() {
    this.partData = {
      content: ''
    };
    this.setPartData();
    this.appendCode(this.storeTempData('#temp_customnav', this.partData));
  }

  CustomNav.prototype.getPartData = function() {
    return this.content = $('#modal-item__text').val();
  };

```

```

};

CustomNav.prototype.setPartData = function() {
  this.getPartData();
  if (!_isEmpty(this.content)) {
    return this.partData.content = _.escape(this.content);
  } else {
    return console.log('empty');
  }
};

return CustomNav;

})(TempPart);

Loop = (function(superClass) {
  extend(Loop, superClass);

  function Loop() {
    this.partData = {
      content: ''
    };
    this.setPartData();
    this.appendCode(this.storeTempData('#temp_loop', this.partData));
  }

  Loop.prototype.getPartData = function() {
    this.content = $('#modal-item__check').val();
    return console.log(this.content);
  };

  Loop.prototype.setPartData = function() {
    this.getPartData();
    if (!_isEmpty(this.content)) {
      return this.partData.content = _.escape(this.content);
    } else {
      return console.log('empty');
    }
  };

  return Loop;

})(TempPart);

```

```

$('.modal').find('#submitBtn').on('click', function() {
  if ($('.modal').hasClass('pt-headerTitle')) {
    return new HeaderTitle();
  } else if ($('.modal').hasClass('pt-textarea')) {
    return new Post();
  } else if ($('.modal').hasClass('pt-widget')) {
    return new Widget();
  } else if ($('.modal').hasClass('pt-custommenu')) {
    return new CustomNav();
  } else if ($('.modal').hasClass('pt-loop')) {
    return new Loop();
  } else {
    return console.log('part error!');
  }
});

```

```

handleDragStart = function(e) {
  return e.stopPropagation();
};

```

```

handleDragOver = function(e) {
  if (e.preventDefault) {
    e.preventDefault();
  }
  this.classList.add('columns-over');
  e.dataTransfer.dropEffect = 'move';
  return false;
};

```

```

handleDragEnter = function(e) {
  return this.classList.add('columns-over');
};

```

```

handleDragLeave = function(e) {
  return this.classList.remove('columns-over');
};

```

```

handleDrop = function(e) {
  var dropItemId, modal;
  e.preventDefault();
  if (e.stopPropagation) {
    e.stopPropagation();
  }

```

```

}
this.classList.remove('columns-over');
dropItemId = e.dataTransfer.getData('Text');
if (dropItemId === 'loop') {
    modal = new window.settingModal.Modal('追加するループの設定');
    modal.addContent(window.settingModal.parts.rendCheckBox());
    modal.addPartClass('loop');
    modal.show();
} else if (dropItemId === 'widget') {
    modal = new window.settingModal.Modal('追加するウィジェットの設定');
    modal.addContent(window.settingModal.parts.rendInpText());
    modal.addPartClass('widget');
    modal.show();
} else if (dropItemId === 'custommenu') {
    modal = new window.settingModal.Modal('追加するカスタムメニューの設定');
    modal.addContent(window.settingModal.parts.rendInpText());
    modal.addPartClass('custommenu');
    modal.show();
} else if (dropItemId === 'headerTitle') {
    modal = new window.settingModal.Modal('追加する見出しの設定');
    modal.addContent(window.settingModal.parts.rendHeaer(), window.settingModal.parts.rendInpText());
    modal.addPartClass('headerTitle');
    modal.show();
} else if (dropItemId === 'picture') {
    modal = new window.settingModal.Modal('追加する画像の設定');
    modal.addContent(window.settingModal.parts.rendUploadBtn());
    modal.addPartClass('picture');
    modal.show();
} else if (dropItemId === 'article' || 'code') {
    if (dropItemId === 'article') {
        modal = new window.settingModal.Modal('追加する文章の設定');
    } else {
        modal = new window.settingModal.Modal('追加する記事の設定');
    }
    modal.addContent(window.settingModal.parts.rendTextArea());
    modal.addPartClass('textarea');
    modal.show();
}
return false;
};

handleDragEnd = function(e) {
    return [].forEach.call(cols, function(col) {

```

```

        return col.classList.remove('columns-over');
    });
};

iframe = document.getElementById('result-iframe');

iframe.onload = function() {
    var cols;
    cols = this.contentDocument.getElementsByClassName('columns');
    return [].forEach.call(cols, function(col) {
        col.addEventListener('dragenter', handleDragEnter, false);
        col.addEventListener('dragover', handleDragOver, false);
        col.addEventListener('dragleave', handleDragLeave, false);
        return col.addEventListener('drop', handleDrop, false);
    });
};

partDragStart = function(e) {
    this.classList.add('part-item_draging');
    e.dataTransfer.setData('Text', e.target.id);
    return e.stopPropagation();
};

partDragEnd = function(e) {
    return [].forEach.call(parts, function(part) {
        return part.classList.remove('part-item_draging');
    });
};

parts = document.querySelectorAll('.part-item');

[].forEach.call(parts, function(part) {
    part.addEventListener('dragstart', partDragStart, false);
    return part.addEventListener('dragend', partDragEnd, false);
});

$('#result-iframe').on('load', function() {
    var gridMenu, removeGridMenu;
    iframe = $('#result-iframe').contents();
    iframe.on('click', '#addRowBtn', function() {
        iframe.find('.row-active').parent().append('<div class="row"> <div class="columns col-number-0 co
        return removeGridMenu();
    });
});

```

```

iframe.on('click', '#removeRowBtn', function() {
  if (iframe.find('.row').length > 1) {
    iframe.find('.row-active').remove();
    return removeGridMenu();
  } else {
    iframe.find('.row-active').append('<div class="alert alert-warning" role="alert">
最後の行なので削除出来ません</div>');
    return removeGridMenu();
  }
});
iframe.on('click', '#addGridBtn', function() {
  var activeColumns, columnNumber, columnNumberVal, newColumns;
  columnNumber = iframe.find('#columnNumber');
  columnNumberVal = Number(columnNumber.val());
  if (columnNumberVal < 12) {
    columnNumberVal++;
    columnNumber.val(columnNumberVal);
    activeColumns = iframe.find('.row-active > .columns');
    activeColumns.removeClass(function(index, className) {
      return (className.match(/bcol-md-\S+/g) || []).join(' ');
    });
    newColumns = Math.floor(12 / columnNumberVal);
    activeColumns.addClass("col-md-" + newColumns);
    return iframe.find('.row-active').append("<div class=\"columns col-number-\" + (columnNumberVal
  }
});
iframe.on('click', '#reduceGridBtn', function() {
  var activeColumns, columnNumber, columnNumberVal, newColumns;
  columnNumber = iframe.find('#columnNumber');
  columnNumberVal = Number(columnNumber.val());
  if (columnNumberVal > 1) {
    columnNumberVal--;
    columnNumber.val(columnNumberVal);
    activeColumns = iframe.find('.row-active > .columns');
    activeColumns.last().remove();
    activeColumns.removeClass(function(index, className) {
      return (className.match(/bcol-md-\S+/g) || []).join(' ');
    });
    newColumns = Math.floor(12 / columnNumberVal);
    return activeColumns.addClass("col-md-" + newColumns);
  }
});
gridMenu = "<ul class=\"grid-menu\">\n <li id=\"addRowBtn\">列を追加</li>\n <li id=\"removeRowBtn\">列を削除</li>\n</ul>";

```

```

列を削除</li>\n <li id=\"adjustColumnBtn\">カラム数<input type=\"text\" id=\"columnNumber\" value=\"1\"
</span>\n <span id=\"reduceGridBtn\"> </span>\n </li>\n</ul>";
removeGridMenu = function() {
    iframe.find('.column-active').parent('.row').removeClass('row-active');
    iframe.find('.column-active').removeClass('column-active');
    return iframe.find('.grid-menu').remove();
};
return iframe.on('click', '.columns', function() {
    var self;
    self = $(this);
    iframe.find('.alert').remove();
    if (self.hasClass('column-active')) {
        return removeGridMenu();
    } else {
        iframe.find('.column-active').removeClass('column-active');
        iframe.find('.grid-menu').remove();
        self.addClass('column-active');
        self.parent('.row').addClass('row-active');
        return iframe.find('.row-active').after(gridMenu);
    }
});
});

if (!window.settingModal) {
    window.settingModal = {};
}

window.settingModal.Modal = (function() {
    function Modal(modalTtitle) {
        this.$target = $('<div>.modal</div>');
        this.$target.find('<div>.modal-title</div>').text(modalTtitle);
        console.log('constructor at Modal');
        this.$target.find('<div>.submit-btn</div>').on('click', (function(_this) {
            return function() {
                console.log('submit!');
                _this.resetContent();
                _this.resetPartClass();
                return _this.hide();
            };
        })(this));
        this.$target.next('<div>.modal-overlay</div>').on('click', (function(_this) {
            return function() {
                _this.resetContent();
            };
        })(this));
    }
    return Modal;
})();

```

```

        _this.resetPartClass();
        return _this.hide();
    };
})(this));
}

Modal.prototype.addContent = function() {
    var i, results;
    i = 0;
    results = [];
    while (i < arguments.length) {
        this.$target.find('.modal-content').append(arguments[i]);
        results.push(i++);
    }
    return results;
};

Modal.prototype.resetContent = function() {
    this.$target.find('.modal-title').empty();
    return this.$target.find('.modal-content').empty();
};

Modal.prototype.addPartClass = function(partClass) {
    return this.$target.addClass('pt-' + partClass);
};

Modal.prototype.resetPartClass = function() {
    return this.$target.removeClass(function(index, className) {
        return (className.match(/\/bpt-\S+/g) || []).join(' ');
    });
};

Modal.prototype.show = function() {
    console.log('show at Modal');
    this.$target.addClass('modal-active').animate({
        'top': '110px'
    }, 400);
    return $('<div>.modal-overlay').addClass('modal-overlay-active');
};

Modal.prototype.hide = function() {
    this.$target.removeClass('modal-active');
    return $('<div>.modal-overlay').removeClass('modal-overlay-active');
};

```

```

    };

    return Modal;

  })();

  if (!window.settingModal.parts) {
    window.settingModal.parts = {};
  }

  window.settingModal.parts.rendHeaer = function() {
    var rendHeaer;
    rendHeaer = '<div class="modal-item"> <label>見出しレベル</label> <select class="modal-item__lebel" >
    return rendHeaer;
  };

  window.settingModal.parts.rendInpText = function() {
    var inpTxt;
    inpTxt = '<div class="modal-item"> <label>内容 <input class="modal-item__text" type="text" value="" >
    return inpTxt;
  };

  window.settingModal.parts.rendTextArea = function() {
    var rendTextArea;
    rendTextArea = '<div class="modal-item"> <label for="inptxtarea">内容 <textarea class="modal-item__
    return rendTextArea;
  };

  window.settingModal.parts.rendCheckBox = function() {
    var rendCheckBox;
    rendCheckBox = '<div class="modal-item"> <label><input class="modal-item__check thumb" type="checkbox" >
サムネイル</label> </div>';
    return rendCheckBox;
  };

  window.settingModal.parts.rendUploadBtn = function() {
    var rendUploadBtn;
    rendUploadBtn = '<div class="modal-item"> <p>アップロードするファイルを選択</p> <button class="btn-uplo
ファイルを選択</button> </div>';
    return rendUploadBtn;
  };

  $(''.btn-download').click(function() {

```

```

var $page, dumpPage, pageData, pageDomObj, serializer;
iframe.find('.column-active').parent('.row').removeClass('row-active');
iframe.find('.column-active').removeClass('column-active');
iframe.find('.grid-menu').remove();
$page = $('#result-iframe').contents();
pageDomObj = $page[0];
serializer = new XMLSerializer();
dumpPage = serializer.serializeToString(pageDomObj);
pageData = {};
pageData.dom = dumpPage;
pageData = JSON.stringify(pageData);
return $.ajax({
  url: 'http://127.0.0.1:1337/node/js/server.js',
  type: 'POST',
  cache: false,
  dataType: 'json',
  data: pageData,
  success: function(data) {
    return console.log('success');
  },
  error: function(XMLHttpRequest, textStatus, errorThrown) {
    $("#XMLHttpRequest").html("XMLHttpRequest : " + XMLHttpRequest.status);
    $("#textStatus").html("textStatus : " + textStatus);
    return $("#errorThrown").html("errorThrown : " + errorThrown.message);
  },
  complete: function(data) {
    return console.log("ajax finish");
  }
});
});

```

```

$(function() {
  var line, maincontent, partsNav, partsNavBtn;
  partsNav = $('.l-partsnav');
  partsNavBtn = $('#partsnav-btn');
  line = $('.menu-line');
  maincontent = $('.l-maincontent');
  return partsNavBtn.click(function() {
    partsNavBtn.toggleClass('is-active');
    if (partsNavBtn.hasClass('is-active')) {
      line.eq(1).animate({
        "opacity": 0
      }, 500);
    }
  });
});

```

```

line.first().css({
    'transform-origin': 'left top',
    'transform': 'rotate(45deg)',
    'transition-duration': '500ms'
});
line.last().css({
    'transform-origin': 'left bottom',
    'transform': 'rotate(-45deg)',
    'transition-duration': '500ms'
});
partsNav.addClass('is-active').animate({
    'left': '0'
}, 300);
return maincontent.animate({
    'padding-left': '220px'
}, 300);
} else {
line.eq(1).animate({
    'opacity': 1
}, 300);
line.first().css({
    'transform-origin': 'left top',
    'transform': 'rotate(0deg)',
    'transition-duration': '500ms'
});
line.last().css({
    'transform-origin': 'left bottom',
    'transform': 'rotate(0deg)',
    'transition-duration': '500ms'
});
partsNav.animate({
    'left': '-220px'
}, 500);
return maincontent.animate({
    'padding-left': '0'
}, 500);
}
});
});

}).call(this);

```

## B.2 バックエンド

```
var http = require('http'),
    fs = require('fs-extra');

var jschardet = require('jschardet'),
    Iconv = require('iconv').Iconv;

var archive = require('./archive.js'),
    convertTag = require('./convertTag.js');

console.log('ready...');

// http サーバーの作成・起動
http.createServer(function(req, res) {
  res.writeHead(200, {
    'charset': 'utf-8',
    'Content-Type': 'text/html',
    'Access-Control-Allow-Origin': '*' // クロスドメインからの ajax 処理の許可
  });

  res.end('Hello World\n');

  if(req.method == 'POST') {
var body = {};

req.on('data', function (data) {
  // 受け取ったデータの文字コード判別
  var detectResult = jschardet.detect(data);

  // データの変換 (?->utf-8)
  var iconv = new Iconv(detectResult.encoding, 'UTF-8//TRANSLIT//IGNORE');
  var convertedString = iconv.convert(data).toString();

  // JSON のパース
  body = JSON.parse( convertedString );
});

req.on('end',function(){
  body.dom = convertTag.getWpPart(body.dom);
  body.dom = convertTag.phpTags(body.dom);
```

```

// file への書き込み
fs.outputFile('output.txt', body.dom, function(err) {
  if (!err) {
    return console.log('wrote file!');
  } else {
    return console.log(err);
  }
});

// 圧縮処理の実行
archive.zip();

console.log('end');
});
}
}).listen(1337, "127.0.0.1");

console.log('Server running!! at http://127.0.0.1:1337/\n');

exports.getWpPart = function(domData) {
  // doctype から body までを include タグ (header 用) に置換する
  (function() {
    var re = /<!doctype html>(.*\n)+\s*<body[^\>]+\>/i;

    domData = domData.replace(re, '<?php get_header(); ?>');
  })();

  // script から html 閉じまでを include タグ (footer 用) に置換する
  (function(){
    var re = /<script[^\>]*>(.*\n)+<\</body><\</html>/i;

    domData = domData.replace(re, '<?php get_footer(); ?>');
  })();

  return domData;
};

exports.phpTags = function(domData) {
  // PHP 開始タグの修正
  domData = domData.replace(/<!--\?php/g, '<?php').replace(/\/\?-->/g, '?>');

  return domData;
};

```

```

// ファイルの圧縮処理
var archiver = require('archiver'),
    fs = require('fs-extra');

exports.zip = function() {

    const OUTPUT_PATH = "./dest/", // 出力先
        FILE_NAME = "wp_themegen.zip";

    // zip ファイルのストリームを生成して、archiver と紐付ける
    var archive = archiver.create( 'zip', {} );
    var output = fs.createOutputStream( OUTPUT_PATH + FILE_NAME );
    archive.pipe(output);

    // js/を起点とした圧縮対象の場所、出力先でファイルを配置する場所
    archive.directory(__dirname + '/../public/', './');

    archive.on('error', function(err){
        throw err;
    });

    // archive.finalize() で、zip 圧縮完了すると、
    // ストリームのクローズイベントが発火する
    output.on("close", function(){

        // zip ファイルのサイズ
        var archive_size = archive.pointer() + " bytes";
        console.log('End archive dir & files.\nFile size is ' + archive_size);
    });

    // zip 圧縮実行
    archive.finalize();
};

```