

人文情報学科 Web サイト用 Wiki の作成について

1048055 竹村茜音

目 次

1	はじめに	1
1	1 制作するもの	1
2	2 Wiki とは	2
3	3 利用対象者	3
4	4 類似するシステム	4
2	2 計画	5
1	1 必要な機能	5
3	3 作成過程	7
1	1 ページ構成	8
2	2 新規作成	8
3	3 記事編集	9
4	4 メニュー編集	11
5	5 ログイン	11
6	6 ログアウト	13
7	7 画像アップロード	14
8	8 CSS	15
4	4 アンケートについて	16
1	1 アンケートの内容	16
2	2 アンケートの結果	16
5	5 おわりに	19
1	1 自己評価	19
2	2 今後の課題	20

1 はじめに

(1) 制作するもの

私が大谷大学の人文情報学科に入学したいと思った理由は幼いころからコンピュータに触れることが好きで、もっとコンピュータについて深く学びたいという気持ちがあったからだ。大谷大学に入学し、1年生では Word や Excel などのコンピュータの基礎について学習してきたが、2年生から HTML⁽¹⁾や CSS⁽²⁾などのホームページ作成について学習し、ホームページのシステムやデザインに興味を持った。ゼミを選択する際、福田ゼミの「人の役に立つ Web アプリケーションの作成」というテーマを知り、自分の知識を増やし人の役に立つものを作成したいという思いを持ち、福田ゼミを選択した。ゼミに所属して2年間、Python⁽³⁾をはじめ PHP⁽⁴⁾や MySQL⁽⁵⁾などを学んできた。システムに関してはほとんど無知だった私は福田ゼミに所属し、はじめは授業の早さについていくことに必死であったが、先生や先輩方から丁寧に教えて頂いたおかげで少しずつ理解することができ、どんどんシステムについて学ぶことの楽しさを知った。

卒業制作を作り始めるにあたり、今まで学習してきた知識を生かしなおかつテーマに沿った卒業制作を作るには、どのようなものを作成すれば良いか考えた。私は人文情報学科に所属して4年間様々なことを学んできたが、TeX⁽⁶⁾やアプリの授業よりも HTML や PHP や CSS のようなホームページ作成の授業に楽しさを感じたためホームページ作成に関連し、今まで学習してきた知識を生かしたことを卒業制作にしたいと思った。具体的にどのような卒業制作にしようか悩んでいる私に福田先生が提案して下さったのが、人文情報学科のオリジナルサイト(図1参照)のことである。

人文情報学科オリジナルサイトとは、大谷大学のホームページ内にあり人文情報学科メニューをクリックしたところに、「オリジナルサイトへ」とリンクが表示されており、福田先生と福田ゼミの卒業生によって作成されているページである。よって現在の人文情報学科オリジナルサイトは福田先生と福田ゼミの学生が編集・更新している。私自身福田ゼミに所属した当初、福田ゼミの3年生で「教員・ゼミ」のページを編集したことがあるのを思い出した。福田先生から昨年度の卒業生が Wordpress というソフトで人文情報学科の Web サイトを実用化しているが、あまり上手くいっていないという話を聞いた。そこで福田先生と相談させて頂き、ソフトは使用せずに PHP と CSS を使用して人文情報学科 Web サイトの Wiki 化を卒業制作にすることが決定した。

(2) Wiki とは

Wiki の定義は IT 用語辞典によると

Web ブラウザから簡単に Web ページの発行・編集などが行える、Web コンテンツマネジメントシステム (CMS) の 1 つ。Web サーバにインストールして Web ブラウザから利用する。

複数人が共同で Web サイトを構築していく利用法を想定しており、閲覧者が簡単にページを修正したり、新しいページを追加したりできるようになっている。編集者をパスワードなどで制限したり、編集できないよう凍結したりすることもできる。HTML の知識がなくてもリストやリンクを簡単に作成できるように、簡易な整形書式が定められている。

とある⁽⁷⁾。

Wiki は上記の説明のように複数人が共同で Web サイトを構築していくシステムのことで、どのページのデザインも同じように統一されているというのが一般的である。しかし人文情報学科オリジナルサイトでは、内容や動作が様々なためページによって少しずつデザインを変えていく必要がある。また一般的な Wiki では簡易な整形書式が定められているため HTML の入力是不必要なのだが、人文情報学科オリジナルサイトでは、新しくページを作成する際は HTML を入力してもらい、編集の際は HTML を表示することにする。これは管理の利用対象者が人文情報学科の先生であるからこそ、HTML の入力は容易にできるレベルという前提で作成する。

よって、一般的な Wiki の機能は残しつつ人文情報学科オリジナルサイトに特化したものに改善していくことにする。また現在の人文情報学科オリジナルサイトとの違いを付けるために、これから先は現在のオリジナルサイトは「現在のサイト」と表記し、また今作成しているオリジナルサイトは「新しいサイト」と表記することにする。

(3) 利用対象者

人文情報学科オリジナルサイトの閲覧者は人文情報学科に在学する全ての学生はもちろん、インターネットを使う全ての人々が対象である。このことは現在のサイトも、私が新たに作成するものも同じで変更はない。オリジナルサイトの管理の面から考えると、現在「学科ニュース」の編集や更新を行っているのは福田先生 1 人である。「教員・ゼミ」のページは私も経験したように、年度始めに福田ゼミの 3 年生がチームに分かれて各々の先生のページを編集している。ページには、先生のインタビュー・担当科目・研究業績・オフィスアワーの時間などたくさんの

項目があり、年度ごとに変更も多い。そのため先生にアポイントを取って各々の先生にインタビューをして、そのインタビューをまとめて編集するととても手間がかかってしまうのだ。サイトを Wiki 化することにより先生たちが各々でページを編集してもらうことを目的としている。各々の先生に編集してもらうことで、3 年生も授業に専念できるようになるだろう。

よって、管理の面から考えると人文情報学科オリジナルサイトの利用対象者は、人文情報学科の先生である。

(4) 類似するシステム

現在使われている人文情報学科オリジナルサイトを基にして 1 つ 1 つのページ内容とメニューリンクは作成する。ただ右側のメニュー欄など不必要と思われるものは削除しつつ良いところは残し、できるだけシンプルで見やすいものを目指す。管理の面では、現存のものはオフライン⁽⁸⁾でしか管理することができず、大谷大学のパソコンからでないとな編集や更新をすることができない。編集・更新を行いたいときにすぐ行えないという問題点があるため、オリジナルサイトを Wiki 化してそこを改善していく。

現在のサイトは決まった人しか編集や更新ができない上、オフラインで管理するので手間がかかり更新が億劫になりがちである。実際「教員・ゼミ」のページでは先生たちの最近の研究実績など、更新が滞っているページも少なくはない。しかし Wiki 化をすることによりオンライン⁽⁹⁾で編集ができるので、大谷大学内でなくても好きな場所で好きな時間に思いついたときに更新できる仕組みになっているので、先生が各々で自分のページを編集し、スムーズに Web サイトの運営をできるよう

にしたいと考える。現在のサイトは更新が滞っているページがあるため、Wiki 化することで更新回数が増えることを期待している。更新回数が増えることにより、閲覧数も増えていけばなお良い。

2 計画

(1) 必要な機能

人文情報学科オリジナルサイトを Wiki 化するにあたって大きく違うのが、管理体系がオフラインかオンラインかである。Wiki 化にあたって福田先生から参考として指示を頂いた。

1. どこかのページを開いて、それを別名で保存して新しいページを作る。その新しいページへのリンクは自分で張る。
2. 左メニュー編集用のページも作る。
3. 編集や保存は、ログインしたときだけに表示される。
4. CSS については、今のところ、3 年生のゼミで作る予定。それをベースに作り直す。
5. バックアップの仕組みを考える。
6. トップページは新着情報、新着情報の詳細ページへのリンク
7. 先生の紹介、ゼミの紹介、先生のインタビューは同じテンプレート
8. 卒論のリストのページ用に 1 つテンプレート。ゼミのページからリンク。
9. コース紹介は、各コースの紹介で、簡単にゼミを紹介し、あとはゼミのページへのリンク
10. 入試情報は、入試日程、オープンキャンパス、自己推薦入試のページ。基本的には新着詳細と同じテンプレート。
11. 学科の紹介は、各コースの簡単な紹介と各コースへのリンク。

上記のとおりである。これらを踏まえて必要な機能を考えてみる。

(i) 編集

編集は記事編集とメニュー編集に分けて考える。

まず記事編集では、全てのページの右下に編集リンクを作成する。編集リンクをクリックすると編集ページに飛び、ヘッダーやメニューの HTML は表示せずそのページの中身の HTML だけ表示し、中身だけ編集できるようにする。そのためにヘッダーやメニューといった外枠のファイルに、編集できる中身のファイルを読み込んで表示することにする。編集の際に内容が変化してしまうとファイル名と内容が一致なくなり、リンクを貼る際にわかりづらくなっては使いにくいと考え、ファイル名を変更して保存できる仕組みにする。そしてメニュー編集は、新規作成あるいは記事を編集した場合のみメニュー編集のリンクを表示することにする。メニュー編集は基本的に新規作成や編集をしたあとに手を加えるため普段の表示は行わない。

(ii) バックアップ

人文情報学科オリジナルサイトでは、日々新しい情報が更新されている。特にトップページの「更新情報」ではこまめな更新が必要とされる。日々更新していく中で編集の際に前のデータに上書きする仕組みだと、もし間違えて編集を確定してしまった場合、前のデータが消えてしまい前のデータに戻すことが不可能となる。前のデータに戻すという作業はとても手間がかかり、完全なる復元は難しいと考える。私自身ゼミの授業で間違えたデータを上書きしてしまい前のデータに戻す作業に手間取った経験が何度もあり、バックアップの重要性をひしひしと感じているので、バックアップは必ず作成しようと考えた。バックアップ機能を付け、新しく保存したデータと以前のデータ 2 つとも保存するように

する。編集でファイル名を変更して保存する場合でも、万が一のことを考えてバックアップされるようにする。

(iii) ログイン

不特定多数の人が閲覧する人文情報学科オリジナルサイトで新規作成や編集を常に表示し続けると、いたずらで書き込む人や情報を改ざんする人、データを消去する人が出てこないとも言い切れない。新規作成や編集を信頼できる人のみに限る必要があるためパスワードを設置し、限られた人のみログインができるようにする。ログインしている間のみ新規作成や編集のリンクを表示する。

(iv) 画像アップロード

現在のサイトの「教員・ゼミ」のページや「校舎紹介」のページには画像が表示されている。新しい先生が増えたり校舎の外観が変わったりしたときに新たに画像を挿入する必要がある。現在のサイトでは画像を直接フォルダに保存しているが、新しいサイトではオンラインで操作するので画像を直接フォルダに保存することができなくなる。そこで画像アップロードの機能を付けることにする。この機能もログインしている間のみ表示する。

また福田先生から参考として指示を頂いた 6 番～11 番はデザインやページ内容の話であり、現在のサイトとほぼ変わらないので現在のサイトのものをそのまま使用することにする。そして授業で学習したときと同様に HTML4.01 仕様に準拠して、UTF-8 の文字コードで作成している。

3 作成過程

ここからは作成した順を追って過程を見ていく。

(1) ページ構成

基本的なページの表示は page.php に、他のページを読み込んで表示する。読み込む用のページは、page.php の階層に data フォルダを作成してその data フォルダの中に保存している。そのためリンク表記は

page.php?file_name ファイル名.php

となる。file_name で読み込む用の他のページは<body> ~ </body>の間のみ入力しており、page.php で読み込んでいるページはどのページも同じレイアウトで表示することになる。(図 2 参照)

page.php には編集のリンクを付け

```
<a href="henshu.php?file_name=<?php echo $file_name;?>">編集リンク</a>
```

と書く。これによりどのページを読み込んでも編集のリンクが表示され、編集のリンクをクリックしたページを編集することが可能である。

(2) 新規作成

記事を書き込むにあたり、shinki.php と shinki_kakunin.php を作成した。shinki.php(図 3 参照)には入力欄と、ファイル名を入力できるテキストボックスを作成し、入力欄のテキストを\$data、ファイル名のテキストは\$file_name を付ける。\$data と\$file_name を shinki_kakunin.php に送り、shinki_kakunin.php(図 4 参照)では実際にどう反映されるかプレビューを表示し、確認して保存する。保存先は data フォルダの中なので、「data/」とファイル名の、.php の部分は勝手に読み込むよう

にしており、利用者はファイル名をつけるだけで良い。(図 5 参照)

もし既存のファイル名と同じファイル名をつけた場合そのまま保存すると、既存のファイルに上書きしてしまうことになる。記事編集にバックアップの機能はあるが新規作成にバックアップの機能は付けていないので、上書きしてしまうと以前の記事が完全に消去されてしまうため

```
$file_name = "data/" . $file_name . ".php";  
if (file_exists($file_name))  
echo "<h1>$file_name が存在します</h1>";  
else  
echo "$file_name は存在しません";
```

と shinki_kakunin.php にタグを書き同じファイル名が存在することを知らせるため、その旨を表示することにした。

shinki.php の入力欄の上にある参考タグは、<h2>や<h3>のクラスセレクトアがどう反映されるか表記している。(図 6 参照) これがあることでどのタグをどのように使用するか頭の中で構成を考えながら入力することが可能である。

(3) 記事編集

表示方法は page.php のときと同様に henshu.php、henshu_kakunin.php を外枠で表示し他のページを読み込むことにした。henshu.php に data フォルダのページを読み込むことで全て編集ページのレイアウトで表示できる。記事を編集するには記事ページの右下に表示されているリンクをクリックする。編集のリンクは、
<a href="henshu.php?file_name=<?php echo \$file_name;?>">この

ページを編集する

と書き、\$file_name の中身の\$data を henshu.php で読み込み表示し (図 7 参照) 中身を編集したのち henshu_kakunin.php でどのように表示されるか確認する (図 8 参照)。記事編集の際にファイル名と記事内容が違ってくる、メニュー編集の際にどのファイル名かわからなくなってしまうことを考慮してファイル名を変更できる機能を作成した。新規作成のときと同様に、ファイル名を入力できるテキストボックスを設置しテキストボックス内には現在のファイル名を表示しておく。(図 9 参照)

他にはバックアップ機能を付けることにした。理由はバックアップ機能がない場合以前の記事に上書きしていくしかないが、その方法だと間違えて上書きしてしまった場合以前の記事に戻すことが難しいからである。現在記事を保存している data フォルダと同じ階層に、backup フォルダを作成した。

```
if (isset($_POST['honbun'])) {  
    extract($_POST);  
    if (file_exists("data/" . $file_name)){  
        if (rename("data/" . $file_name , "backup/" . $file_name) ==  
false) {  
            print "バックアップできていません。";  
        }  
    } else {  
        print "ファイル${file_name} を保存しました。";  
    }  
}
```

}

これで以前の記事は backup フォルダに移動し、編集された記事は data フォルダに保存される。もしファイル名を変更した場合でも、以前のファイル名の記事は backup フォルダへ移動し、新しく名前がつけられた編集後の記事は data フォルダに保存されるのである。backup フォルダはセキュリティ保護のためアクセスできないことにする。またどのページを編集したかということを知りやすくするため、編集のページを保存するとその編集したページに戻るようリンクした。

(4) メニュー編集

メニューを編集するリンクは、新規作成と記事編集を保存した際に表示することにした。(図 10 参照)メニューを編集する機会は、記事を新規作成したときか記事を編集、あるいは記事のファイル名を変更したときしかないと考えたからである。メニュー編集も記事編集と同様に、henshu.php に menu.php を読み込み表示し編集を行う。はじめの段階では新規作成やログインボタンも menu.php の中に表示していたが、メニュー編集の際に新規作成やログインボタンが意図的に削除できてしまうことを考慮して menu.php のタグの中には表示せず、編集できるメニュー部分は require で読み込み、編集されたくない部分は page.php に書き込むことにした。

(5) ログイン

人文情報学科 Web サイトを Wiki 化するにあたって一番苦労したのがログインである。ログインは cookie を利用する。cookie とは、Web サイトの提供者が Web ブラウザを通じて訪問者のコンピュータに一時

的にデータを書き込んで保存させるしくみのことである。一度ログインをするとそのサイト内でならどのページに移動してもログアウトするまで、cookie が保存されたまま、つまりログインされたままになるのである。よって、cookie を利用してログインページを作成することにした。まず cookie.php を作成し define でパスワードを設定する。そしてその下に

```
if (isset($_POST["password"]) && $_POST["password"] == PASSWORD) {  
    setcookie("JBKYOIN", "odjgg");  
    $login = true;  
} else if (isset($_COOKIE["JBKYOIN"]) && $_COOKIE["JBKYOIN"]  
== PASSWORD) {  
    $login = true;  
}
```

と書き、cookie を JBKYOIN としパスワードは odjgg とした。

実際にログインするのはメニュー欄にある、「ログイン」である。パスワードが認証されればこのパスワードを知っている人はこの cookie を所有していると読み込み、セットで自分のコンピュータに保存する。あとは require で page.php、login.php などの全ての枠組みページに読み込むとどのページに移動しても cookie は保存されており、ログイン状態が継続されたままになるので、ページを移動することにパスワードを入力する必要がない。実際にパスワードを入力するために login.php を作成した。ログインページ (図 11 参照) ではパスワードを入力するテ

キストボックスがあるが type 属性を password 指定することで、実際にパスワードを表示しないようにした。

ログイン中の場合のみ編集リンク、メニュー欄では、新規作成・画像アップロード・ログアウトを表示し、ログインしていない場合はそれらを表示しないように if 条件で分類した。また、ログイン中の場合のみ表示するメニュー欄のリンクはログイン前とログイン後の違いがわかりやすいように<h3>のクラスセレクタの文字色 (#F5A21B) で表示した。

(図 12 参照)

ログイン ID とパスワードを 1 人 1 人が取得しそれぞれ違う ID でログインすることも考えたが、ID は取得せず同じパスワードで誰でもログインできるようにした。人文情報学科 Web サイトにはそれほど危険な個人情報は入っておらず、そこまで厳重なセキュリティにする必要はないと考えたからである。セキュリティをより安全にするため福田先生に定期的にパスワードを変更してもらい、福田先生がそれぞれの利用者の先生にパスワードを知らせてもらうことにする。

(6) ログアウト

はじめはログインページに、ログインのボタンとログアウトのボタンが存在していた。そのためログイン中であっても、ログインページにアクセスしログアウトのボタンを押さなければログアウトすることができず 2 度手間であった。ログイン中の間はログインページに飛ぶリンクをログアウトボタンに変更するために cookie_logout.php を作成し

```
$logout == false;
if(isset($_COOKIE["JBKYOIN"])){
```

```
setcookie("JBKYOIN", "");  
}  
header("Location: ./page.php");
```

と書き直すことにより、cookie は所有しているがパスワードを所有していないことになりワンクリックでログアウトできるようにした。ログアウトするとメニュー欄の新規作成・画像アップロードが表示されなくなり、ログインボタンが表示され、ページごとの編集リンクも表示されなくなる。

(7) 画像アップロード

「教員・ゼミ」のページには先生の顔写真をそれぞれ表示しており、「校舎紹介」のページにもそれぞれの教室の画像を表示している。もし新しい先生が赴任された場合や教室の画像と現在の教室が異なる場合、新しく画像をアップロードしなければならない。そこで、image.php と画像を保存する image フォルダと upload.php を作成した。image フォルダの階層は、data フォルダや backup フォルダと同様である。image.php で選択した画像が image フォルダに保存される仕組みの画像アップロードを作成した。(図 13 参照)「ファイルを選択」ボタンをクリックすると、ウィンドウが表示されるので、そこからアップロードする画像を選択することができる。(図 14 参照)アップロードが成功すると upload.php に移動し、どの画像をアップロードしたかを表示して画像に間違いがないか利用者が確認できるようにした。(図 15 参照)どの画像をアップロードしたかをすぐ表示することで、間違えた画像をページに添付してしまい再び編集や画像アップロードを行わなければい

けないことがないようにした。表示した画像の下にテキストボックスを設置し、テキストボックス内のタグをコピーして画像を表示したいところにペーストするだけで画像を表示することができる。簡単に画像が添付できることと、フォルダの階層は新しいサイトを作成している私は理解しているが、初見で階層は分かりにくいことを考えこのような仕組みにした。

(8) CSS

当初 CSS は計画のところで述べたように、3年生のゼミで作るそれをベースに私が作り直す予定であった。しかし3年生の授業が思うように進まず CSS が間に合わなかったのである。福田先生に相談すると現在のサイトの CSS を使用しても良いとのことだったので、現在のサイトの CSS と私が作成する CSS の2つを使用することにした。現在のサイトの CSS で基本的なデザインやメニューを設定し、私が新たに作成する CSS でヘッダーやフッターの位置設定や、<h2>や<h3>などの文字色のクラスセクタやテーブルのクラスセクタを設定した。ページごとによって必要なテーブルの高さや横幅やセルの大きさが違うので文章が変なところで改行されて文章が読みにくいことや、逆に文字が少なくてセルが広すぎる問題が起きた。読みやすくなるようにテーブルごとにクラスセクタで大きさを設定した。そのテーブルに見合った大きさを設定するので、テーブルがすっきりして見える。

4 アンケートについて

(1) アンケートの内容

実際に新しいサイトを3・4年生の福田ゼミ生を対象に使ってもらいアンケートを実施する。同時に何人もの人が同じページを編集するとエラーになってしまう可能性が高いため、ある人には「コース紹介」を編集、ある人には「学科紹介」を編集、ある人には「新規作成」など、1人1人違うページを編集・更新してもらうことにする。アンケートの内容は以下の通りである。

1. デザインは見やすいか
2. サイトは使いやすかったか
3. 良かった点
4. 改善したほうがよい点

(2) アンケートの結果

まず新しいサイトのデザインについてアンケートを行った結果ほぼ全員が「デザインが統一されていて見やすい」と回答した。とても喜ばしい結果であり、色を統一しつつメリハリをつけ現在のサイトに構造を似せたことが理由だと考える。アンケートの良かった点についてもデザインに関することが多く書かれていた。よってデザインはこのままでいくことにする。

次に使いやすさについてのアンケートでは、「使いにくい」や「使い方が分かりづらい」という意見が多数あった。第一に、登録する前の確認ページが確認ページだと気付かず終わってしまい、登録ボタンを押さず消してしまったという意見である。短い記事にする場合は、`henshu_kakunin.php` で保存ボタンが上にあるので気付くことができる

が、長い記事の場合は保存ボタンがスクロールしないと表示されていることに気付かないので記事を消してしまう可能性があるということに気付いた。そこで、henshu_kakunin.php の 1 番上に目立つように<h1>で「まだ保存されていません」と表記した。(図 16 参照) <h1>のクラスセレクトは赤色フォントにしているのでより目立つことと思う。この赤色フォントに気付き保存ボタンの押し忘れが減ったり、長い記事の場合保存ボタンを探してくれることを望む。

第二に編集の仕方や、メニューの編集の仕方が分かりづらいという意見である。これは読み込んだページにたくさんタグが書かれており、どのタグがどのように表示されているか初見では分かりづらいということと、全体的に説明不足であることを予測した。新しいサイトを作成している私は理解しているので、初見の人の立場に立たなければいけないことに気付いた。そこでテキストボックスを設け、<h2>や<h3>がどのように表示されるかのクラスセレクトの説明や、画像添付のタグを表記した。なぜ編集画面でファイル名がテキストボックスで表示されるのか、という意見もあったので、ファイル名を変更できるテキストボックスの上に「ファイル名も変更できます」と説明を追加した。(図 9 参照)

第三はログインボタンについてである。これは作成過程のログインの部分でも述べたがアンケートにも書かれていたので改めて述べることにする。ログインについての意見としては、現在ログイン中かどうかかわらないということと、メニューを編集する際にログインボタンや新規作成ボタンが意図的に削除できてしまうという意見である。まずログイン中の場合に、<h3>でログイン中である表示をした。ぱっと目のつくところがないと意味がないので、メニュー欄の 1 番上に表示することにした。ログインボタンが削除できてしまうという意見をもらい menu.php を

少し書きかえることにした。編集可能なメニュー部分は require で読み込み、「ログインボタン」や「新規作成」などの編集されたくない部分は menu.php の中には表示せず、menu.php を require で読み込んでいる page.php などの外枠のページに書き込むことにした。これでメニュー編集欄には表示されず、削除されることもない。

改善点についてのアンケートでは、私が主観的に見ても大して気にならないことが、客観的に見ると改善点だと感じる事が多く意見としてあった。

まず、ログインしたらログインページに飛びリンクをログアウトボタンに変更する点である。これも作成過程のログアウトの部分でも述べたがアンケートにも書かれていたので改めて述べる。アンケートの結果を読み編集するまで、ログインページに、ログインのボタンとログアウトのボタンが存在していた。そのためログイン中であっても、ログインページにアクセスしログアウトのボタンを押さなければログアウトすることができず 2 度手間であった。そしてログイン中でもメニュー欄に「ログイン」と表記されているので、ログイン中かどうかわかりづらいという意見もこの点から出てきたに違いない。ログイン中の間はログインページに飛びリンクをログアウトボタンに変更することにより、ワンクリックでログアウトできるようにした。

他にも上記の使いやすさについてのアンケートで述べた、「使いにくい」や「使い方が分かりづらい」という意見が、改善したほうがよい点の部分に多く書かれており、同じような意見が多数存在した。客観的に見てもらうことで、私 1 人では気付かなかった部分にたくさん気付くことができ、アンケートの意見を反映した新しいサイトはより使いやすいものになっただろう。

今回のアンケートを行って良い意見がほとんどではあったが、その中にもちゃんと改善点を指摘してくれる意見が多くあった。この意見を素直に受け止め、全ての人要望に応えることで Web サイトは利用者がより使いやすいサイトになっていくと感じた。それと同時に理想だけでは思い通りのものは作成できず、知識と技術が必ず必要となってくると感じた。

5 おわりに

(1) 自己評価

今まで簡単な Web ページまたは先生に習って Web サイトを作成してきたが、今回人文情報学科オリジナルサイトの規模は今までに経験したことのないもので、福田先生から助言は頂いたが自分でページ構成を考え工夫して作った Web ページは今回が初めてで、今までとの規模の違いに戸惑うことが多かった。作成する側となり、はじめて「人の役に立つ Web アプリケーションの作成」というテーマがいかに難しいかということを実感した。福田先生の要望や、自分が思う機能・デザイン・レイアウトなど、他の人がどのようにすれば使いやすいと感じるかを考えながら作成する必要があった。しかし、役に立つであろうと考えた機能があってもそれを実現できる知識や技術が足りず、いかに自分の知識や技術が乏しいものであったかを痛感することになった。しかし、だからこそ自分が作成できる範囲で、どのようにすれば人の役に立つ機能が実現できるかを深く考えることができた。

新規作成や編集はゼミで学んだことを生かして進めていくことが出来た。しかし cookie やバックアップの仕組みは初めてで思ったより作業がはかどらず時間をかけすぎてしまい、現在のサイトから新しいサイト

へリンクが十分に反映できなかったことは自分の計画の甘さが原因である。これから先、今回のようなことがあった時は、作成の進み具合や残り時間をしっかり把握し、時間を逆算しながら計画や作業を実行することが重要だと考えた。しかし自分で色々な資料を探し理解できるまで調べることができたので、自分の知識として身につけたいと思う。今後の課題もあり、より良いものにするため改善する点はまだまだあるので完璧な完成とはまだ言えないが、1つのものを作り上げたという事実は私の自信になった。まだまだわからないことばかりではあるが、今回の作成を通して、知識の幅も今まで以上に広げることができたのではないかと思う。

(2) 今後の課題

サイト全体にエラーはなく新規作成や編集など個々の機能も問題なく動くので、新しい人文情報学科オリジナルサイトとして使用できるだろうと思う。しかし、初めて操作する人からするとまだまだ説明不足で使いづらいのではないかと思う。アンケートを実施した時点でたくさん「使いにくい」や「使い方が分からない」という意見があり、いくつかの部分には説明を入れたり、テキストボックスを作り使いやすくしたりと工夫してみたが、まだまだ説明不足なところが多いに違いない。

今後の課題として言えるのは、

1. ファイルの階層がややこしい
2. 編集画面でのタグが見づらい
3. CSS のクラスセレクタが分かりづらい
4. セキュリティの仕組みが甘い

である。

まずファイルの階層がややこしい点である。私はファイルの階層をウィンドウで見ながら新しいサイトを作成していたので理解しているが、オンラインでサイトを操作しても階層はわかりにくいだらうと思う。そのためリンクを作成するときに php で file_name を読み込むのを忘れてしまったり、data フォルダの中にあるファイルということがわからなかったりして、エラーがでてしまっただけでは閲覧できないので、新しいサイトの使い方説明ページを作成するといいかもしれない。次に編集画面でのタグが見つらい点である。ただ羅列してるだけだとどこに何についてのことが書いてあるか把握できないし、編集したいところを羅列してるタグの中から見つけるのも時間がかかってしまい編集の効率が悪くなってしまう。もっとコメントで説明を入れたり、Tab を使ったり、行間を空けたりして見やすくするべきであった。ページ数が 50 ページ以上になるので現時点では見やすくできなかったが、これからもタグを見やすく編集していこうと思う。そして次に CSS のクラスセレクタが分かりづらい点である。作成過程のところで各テーブルごとにクラスセレクタを使用して分かりやすくしたつもりが、クラスセレクタを増やすぎてどのクラスセレクタがどのテーブルに使用したかわかりづらくなってしまった。そして新しく CSS を挿入する際にどのクラスセレクタが使用されているかを把握できていないところは解決しなければいけない点である。CSS をオンライン上で編集できる仕組みも必要かと思う。最後にセキュリティの仕組みが甘い点については URL がフォルダの階層をそのまま表しているのが攻撃される可能性がある。それは現在のサイトも同様なのだが、新しいサイトではそのような URL を表示しない仕組みを考えるべきである。セキュリティの点については全体的に私の勉強不足であるので、もっと勉強してより良いセキュリティの仕組

みを考えていきたい。

時間があれば再びアンケートを実施し、改善した点が使いやすいかということと、より改善したらいい点など意見を集めたいと思う。サイトの機能は問題なくても、説明不足の点が多いというのは主観的にサイトを作成してきたからだと思うので、できれば改善点を他の人に言われてから気付くのではなく、自分で客観的に見て気付いていきたい。まだたくさんの方が時間があるため、現在のサイトから新しいサイトへリンクが十分に反映できなかった点も含め時間の許す限り問題解決に努め、現在のサイトより良い新しいサイトを作成していきたい。

注

- (1) HyperText Markup Language の略
- (2) スタイルシート：Cascading Style Sheets の略
- (3) プログラミング言語の中の 1 つ
- (4) プログラミング言語の中の 1 つ：Hypertext Preprocessor の略
- (5) データベースを管理・運用するためのシステム
- (6) スタンフォード大学の Donald E. Knuth 教授 (当時) が開発した
組版ソフトウェア
- (7) <http://e-words.jp/w/Wiki.html> より
- (8) コンピュータや通信機器がネットワークに接続していない状態
- (9) コンピュータや通信機器がネットワークに接続されており、ネット
ワークを通じてサービスを受けられる状態

文献表

西沢直木

2012.07.30 『PHP プログラミングの教科書』ソフトバンククリエイ
ティブ株式会社

大藤幹

2011.09.13 『10 日で覚える CSS/CSS3 入門』大日本印刷株式会社

『IT 用語辞典 e-Words』

<http://e-words.jp/>

『PHP マニュアル』

<http://www.php.net/manual/ja/index.php>

『TAG<index>』

——人文情報学科 Web サイト用 Wiki の作成について——

<http://www.tagindex.com/index.html>