

出席確認アプリの改良について

寺地龍一

目 次

1	はじめに	1
1	1 制作物について	1
2	2 利用対象者について	1
3	3 制作（改良）前の確認	1
4	4 必要な技術	3
2	構想	4
1	1 挙げられた改良点について	4
2	2 改良の構想	4
3	3 作業するうえでの計画	6
3	制作過程	7
1	1 構成	7
2	2 行程	7
4	まとめ	19
1	1 評価	19
2	2 今後の改善点	20
5	おわりに	22

1 はじめに

(1) 制作物について

私は大谷大学人文情報学科において情報デザインコースの福田ゼミに所属し、「人の役に立つ Web アプリケーションの作成」というテーマに則った制作物を作りたいと考えていた。どのようなアプリケーションを作るかと四苦八苦している私に、福田先生のほうから卒業生が制作していった出席確認アプリの改良を頼みたいとの申し出があった。

制作物の発想に難航して他のゼミ生が制作している中、まだその段階に入ってなく焦っていた私にとってその申し出は鶴の一声であった。断る理由などあるはずもなく、私は喜んで着手することにしたのである。これが、私が「出席確認アプリの改良」を卒業論文のテーマにしようとした経緯である。

(2) 利用対象者について

利用対象者としては、もちろん福田先生ないしは他の講師、講義に出席した生徒である。受講する生徒の数が多い講師ほど使ってほしいと考えている。が、Web を介しての出席確認なので PC が置いていない講義や人文生に配布されている iPad を持っていない場合などには対応できないという弱みが存在する。なので人文情報学科の講義にしか利用できないが、それでも少ない利用回数のなかでこのアプリケーションの有用性が理解してもらえるようなものを制作および改良をしていきたい。

(3) 制作（改良）前の確認

「卒業生の論文をしっかりと熟読するように。」というアドバイスと、現在講義で実際に使用されているその出席確認アプリケーションの構造

——出席確認アプリの改良について——

データを福田先生からいただいた。まずその構造データがどのように成り立っているのか把握しようとしてファイルを開いてみたのはいいものの、書いてあるコードが何で、どこにどう繋がっているのかという理解が困難であった。

そこで卒業論文を先に読むことが先決であると思い立ち、大学のゼミの卒業論文ページからダウンロードして論文を読んでもらうことにした。その下準備がなければ構造データの内容を理解できないと考えたからである。熟読して、問題点や改良してほしい点などが述べられている箇所をメモしながら、次は構造データを閲覧してみることにした、というよりも解析と呼んだほうが近いものであったのだがそこは割愛する。

プログラムコードを見るだけでは不十分と感じ、一貫してプログラムを動かして仕組みの理解を試みた。そうすることでより一層の理解が得られ、コードのどこがどう作用しているかなどが効率良く解るのでないかと考えたからである。その他にも、福田先生の授業で実際に使われている授業出席登録のページ(資料1参照)も閲覧し卒業生の制作との差異を理解する。

学生側のページは、出席の入力ページである出席確認ページ(資料2参照)と出席登録ページ(資料3参照)の2ページである。教員用のページはTOP画面である教員管理ページ(資料4参照)があり、そのページ内に配置されているリストから出席確認表(資料5参照)、出席簿(資料6参照)のページへと移れるようにされている。出席の削除は、削除の確認(資料7参照)と削除(資料8参照)の2行程に分けている。間違えて削除ボタンを押してしまった場合の保険であると思われる。

これらのプログラム群を一貫して、実際に動かしながら理解を深めて

いく。

(4) 必要な技術

大元のページであるところは標準的な JavaScript コードで書かれており、その他の場所は PHP(Hypertext Pre-processor) で構成されている仕組みであった。デザインは出席ページと管理ページで CSS(CascadingStyleSheets) プログラムを分けて作成されていた。PHP は、福田ゼミのなかでも 1 番よく使うプログラミング言語であるために構造データの飲み込みが早かった。

しかし、Javascript や CSS は大学 3 年の頃に学んだきり復習していなかったため勉強する必要があることがわかった。PHP のなかにも難解なものもあるために Web 上の資料などを参照して制作を進めることにした。プログラムのコードを読んでいくにつれて PHP や Javascript でもないものが存在した。それは正規表現というもので、これも大学 3 年の最初に少し触った程度で理解に乏しく非常に難解であった。卒業者が作成していった正規表現を学ぶための Web ページなどを活用して時間をかけて 1 つずつ読み解いていく。

2 構想

(1) 挙げられた改良点について

(i) ログイン画面について

卒業論文を読んだうえで、卒業者が急務の改良点について述べている部分があった。まず第1にログイン画面である。現状だとログイン画面が存在せず、そのまま管理者画面ページが表示されてしまうのである。URLを知らなければ画面遷移自体が無理であるが念のために対処する。

(ii) 出席表での個人の出席カウント

前作の場合は日付ごとに右から学生番号・日付・名前そして削除ボタンという構造で表が成り立っている。表については資料5を参照してほしい。この出席表の表し方だと生徒一人一人の出席した数がカウントされず、全体を通して非常に見づらいという欠点があった。そのために表を改良してほしい、との案が述べられていた。

この2項目が重点的に改善してほしい点であると書かれており、自分で実際にこの卒業者が制作した出席確認アプリケーションの管理画面で確認してみて、この2つが改良するうえでの重点になるだろうことを理解した。

(2) 改良の構想

(i) ログイン画面

前述した通り管理ページがそのまま表示されてしまう作りになっているためにトップ画面にログインシステムを設け、そこでパスワードを入力してもらうことで管理者画面へ遷移する仕組みを作成する。このプロセスを挟むことによってパスワードを打つ手間は増えることになるが、

不正アクセスなどをされるケースを予防することができる。そして入口を作るということは必然的に出口も作らなくてはいけないため、ログアウト処理もできるように改良する。

(ii) 出席表

改良案として出されていた出席のカウンタのみならず、情報デザインコースらしくもっと見やすくかつ見栄えがするようなものを制作したい。具体的には日付ごとにタブ付けし、それらをクリックすればその日付のページへ遷移して日付ごとの出席表を閲覧する、という形にしたい。表については、日付を省き左から氏名・出席番号・感想・カウンタ数・削除ボタンという表記にしたいと構想している。

(iii) デザインの見直し

デザインは CSS(CascadingStyleSheets) が管理しているために、まずオリジナルのものを作り、それを既存のものから差し替えて新しいデザインのものに作り変える。作る上で気をつけたいことは、パッと見て機能が解るものとすることである。構造が解りにくくは取っつきにくく、PC を初めて触る人に至ってはそれだけで使うのも嫌がる可能性を考慮した結果である。福田先生の方からも「とにかく、解りやすいものを。」との指示があったため「見やすいデザイン」を大前提に作ることに決定した。

(iv) 出席簿

前作のメインである「出席簿右クリックでダウンロードして Excel に貼り付けて下さい。」という文字にカーソルを持っていき、そこをクリックすることでダウンロードできるという機能はそのままに、新しいプログラムに合わせてコードを変えて対応するように制作する。以上の構想で出席確認アプリケーションを改良していく予定とする。

(3) 作業するうえでの計画

アプリケーションの改良をするというもののため、そもそもあるプログラムがどうやって、どのようにプログラムが作用して動いているのかを熟知している必要がある。そのためにまず出席アプリケーションの作者である中西陽子氏の卒業論文を読み、そのうえでプログラムコードの意味を読み解いていく。解らない部分はノートに図を描き、どのようにして動いているのか考える。

次に、ログインとログアウトのプログラムを作成する。この2つは元のプログラムを改良するのではなく、新規にプログラムを作成しそれを組み込んでいくことになる。前プログラムを大々的に改良するのではないため初期に手をつける。出席表のカウントと出席簿は、出席表に出席簿が連動しているため出席表のカウント、および日付のリスト分けを作成した後に出席簿の改良に着手することにする。CSSの見直しは、全ての作業が終わった後に始める。CSSによる大々的なデザインの変更は大変なようでいて、プログラムのバグなどの致命的なエラーなどが検出される危険性が低いからである。

この順番で出席確認アプリケーションの改良をおこなう。進行具合やファイルのエラーなどにより計画変更も起こり得るが柔軟に対処する。

3 制作過程

(1) 構成

ファイル構成は、学生用は出席確認ページ (index.html) と出席登録ページ (toroku.php) の 2 ページである。教員用のページはまずログイン処理をおこない、TOP 画面である教員管理ページ (kanri.php) へ遷移するようにする。そのページ内の左にあるリストから出席確認表 (info_table.php)、出席簿 (shussekiibo.php) のページへ飛ぶことができる。同時にログアウト処理もおこなえるようにページ内のどこかにログアウトの項目を追加する。

出席確認表のページをクリックして飛ぶと、出席した日付ごとにリスト分けされている日付画面 (hiduke.php) が現れる。その日付をクリックしたページ上にある出席者の削除ボタンを押すことで出席の削除ができる。削除の確認 (info_table_kakunin.php) と削除 (info_table_delete.php) の 2 行程に分けている。間違えて削除ボタンを押してしまった場合の保険である。デザイン関連は CSS を使用し、学生用の CSS(index.css) と教員用の CSS(kanri.css) とに分けてデザインする。

(2) 行程

行程は、最初に管理者側のログイン画面と並行してログアウト処理を作成した。

(i) ログイン画面

最初にログイン画面 (login.php) を作成する。新規にページを作るといってもそこまで大仰なプログラムを組むのではなく、管理者ページの TOP 画面である教員管理ページからプログラムのコードを流用する。

——出席確認アプリの改良について——

プログラムの中にある html タグの上に下記のコードを使用することによりログイン認証が行える。

```
<?php
session_start();
$errorMessage = "";
$viewUserId = htmlspecialchars($_POST["userid"],
ENT_QUOTES);
if (isset($_POST["login"])) {
if ($_POST["userid"] == "test" && $_POST["password"]
== "pass") {
session_regenerate_id(TRUE);
$_SESSION["USERID"] = $_POST["userid"];
header("Location: kanri.php");
exit;
}
else {
$errorMessage = "ユーザ ID あるいはパスワードに誤りがありま
す。";
}
}
?>
```

上記を見れば解るように、最初にセッションを開始している。ログインというシステムを使用する場合、クッキーというクライアント（ユーザー）側に情報を保持させる場合と、セッションというサーバー自体に

情報を保持する場合に分けられる。

何故クッキーを選ばなく、セッションの方を採用したかという、クッキーはクライアント側にデータが保存されるため、改ざんする事が出来る。また、クッキーに保存されたデータは同じサーバーの別アプリからもアクセス出来るのでセキュリティ上問題がある可能性があるのである。

セッションの場合はクライアント側にはセッション ID のみが保持されるので、詳細情報はサーバー側に保持されるため比較的安全である。当然、セッション ID が盗まれれば同じ事だが、セッション ID は自動的に予測不可な ID がその都度振られるため、比較的安全と考えてもよい。

時間的に見た場合、比較的長いスパンで保持したいもの（不連続な接続、例えばログイン名など）はクッキー、時間的には短時間でその都度変化していくもの（例えば通販のカート内の数量や価格等）はセッション、と言う様な使い分けが出来るのである。

ログインに失敗した場合は、`$errorMessage` にエラーメッセージを格納し、画面に表示している。ログインに成功した場合は、セッションハイジャック対策として、`session_regenerate_id()` を使って、セッション ID を再発行してから、`kanri.php` を表示するようにした。ユーザー ID は”test”、パスワードは”pass”としている。ここで書くユーザー ID とパスワードは `method` 属性が POST であるため、それらを入力するために `form` 要素を使用する。

`method` 属性は GET と POST の 2 種類あるが、ログイン処理では POST を使用した。何故かという、GET のパラメータを使う場合、

ブラウザ依存でURLの長さの制限に引っかかり、すべてのGETのパラメータが送られない場合があるからである。その点、POSTのパラメータの長さの制限はサーバー依存であるため、フォームからの入力は何らかの制限がない限り、POSTを使うというのがセオリーだからである。ログイン画面がどのようなものかは資料9を参照とする。form要素のコードの始めは以下ようになる。

```
<form id="loginForm" name="loginForm" action="<?php print  
($_SERVER['PHP_SELF']) ?>" method="POST">
```

ログイン画面で入力してもらうのはユーザーIDとパスワードだけであるが、プログラムにパスが指定されているため特定されにくく不正アクセス行為の予防になるのではないだろうか。

ログインに成功した場合、教員管理ページ(kanri.php)へ遷移するようにして、尚且つ管理ページ上部に「ようこそ、"userid"さん」と表記するようにした(資料10参照)。プログラムの書き方は以下の通りである。

```
<?php  
session_start();  
// ログイン状態のチェック  
if (!isset($_SESSION["USERID"])) {  
    header("Location: logout.php");  
    exit;  
}  
echo "ようこそ" . $_SESSION["USERID"] . "さん";  
?>
```

この機能を追加したことで、ログイン処理が上手く作動していることが確認できる。はじめに、ログイン状態のチェックを行っている。具体的には、`$_SESSION["USERID"]` に値が格納されているかを調べている。`login.php` でログインに成功すると、`$_SESSION["USERID"]` にユーザ ID を格納しているため、この値が入っていないという場合に不正なアクセスとみなして、ログアウト画面を表示するようにしている。ログイン画面の作成は以上の内容である。

(ii) ログアウト画面

ログアウト画面 (`logout.php`) では、セッション情報の破棄を行うようにした。`$_SESSION["USERID"]` に値が格納されているかどうかで意図的にログアウトしたのか、セッションタイムアウトでログアウトになったのかを自動で判断して、メッセージを出力するようにした。詳しいプログラム内容は資料 11 を参照とする。

また、ログアウト画面にログイン画面へのリンクを貼って、再度ログインする折に簡単に行えるようにした。出席の削除と違い、このログアウトは確認画面が存在しないため保険として用意した。ログアウト自体は教員管理ページ (`kanri.php`) のページ内の左にあるリストからおこなう。そのためにリストの下にログアウトのタグを新しく作る。出席確認表 (`info_table.php`)、出席簿 (`shussekiibo.php`) のページにも同様の改良をおこない、どのページからでもすぐログアウトが出来るようにした。ログアウトの画面および処理は以上の内容である。

(iii) 出席確認表ページの改良

次に出席確認表ページ (`info_table.php`) の改良をおこなった。改良前の場合だと、学生のデータを取り出して学生番号と日付、名前のみを取り出し表に表示されるようになっていた。日付は新しい順に上から並ん

——出席確認アプリの改良について——

であり、更新すると新しいデータが追加されるようになっていた。しかし、これらはテーブル表示で更新するたびに延々と下に伸びていくという悪い点も内包していたのである。

これをどうにかできないかと考えた結果、日付ごとにリスト分けしてその日付をクリックすることで出席確認表があるページへ飛ぶように改良することにしたのである。そのためには別に日付ページ (hiduke.php) を作り、そちら側に学生から送られてきた日付を除く学生番号、名前、感想を取り出してテーブル表示にする必要がある。感想の横には削除ボタンも取り付けて出席の削除も行えるようにする。出席確認表ページ (info_table.php) に記載するコードは以下の通りである。

```
<?php
function make_table() {
$dir_path = "./data";
$dirs = scandir($dir_path);
print("<ol>");
foreach ($dirs as $dir) {
if (preg_match("/^\d{8}$/", $dir, $match)) {
$hiduke = $match[0];
print("<li class='main_list'><a href='hiduke.php?hiduke
=$hiduke'>$hiduke</a></li>");
}
}
print("</ol>");
}
?>
```

`$dir_path` で学生側のデータへアクセスして取り出してから、それを `$dirs` という変数に変換している。そして取り出した日付だけを `ol` リストに変えて書いていくようにしている (資料 12 参照)。学生の出席データを閲覧するには、その見たい日付をクリックして出席表がある `hiduke.php` に飛んでいける。しかし、これだけのコードを書いただけでは表示されないので `body` 内に以下のコードを書く。

```
<?php
make_table();
?>
```

このコードを書くことにより `function make_table()` が有効となり、`ol` リストで日付のテーブルが作成される。出席表確認ページの改良は以上の内容である。

(iv) 日付ページ

次に、出席確認表の新しいメインである日付ページ (`hiduke.php`) を作成した。このページは学生用の出席確認ページから送られてきた学生番号、名前、感想を取り出してテーブル表記で表示することを目的としている。構想していたものとの差異は、福田先生と相談して出席者のカウントを省くことにした点である。日付ごとにリスト分けし、その中に出席者のテーブル表を作成したほうがわかりやすく見やすいものになるのではないかと考えたからである。

テーブル表の表記に関しては、日付とキーワードを除く学生番号・名前・感想を左からテーブルで表記し、学生番号順に上から並べるようにした。更新すると新しい出席者のデータが追加され、番号順に自動的に並べられるようにした。出席確認表ページから `method` 属性の `GET` で

情報を受け取るコードを書く。method 属性の GET に関しては (i) で説明しているのでそちらを参照してほしい。そのうえで scandir で出席者の表示してほしいデータを抽出している。

しかし、そのままテーブル表を作成すると文字化けやテーブル表自体に謎の空欄が発生した。詳しく調べてみると、".DS_Store" というデータがプログラムの邪魔をしていることが判明した。データ整理時に必ずこのファイルが発生しているためにプログラムを更新、つまり新しい出席者のデータを閲覧しようとするたびにエラーおよびバグが発生したのである。foreach 以下に以下のコードを書くことによってその問題を解決した。

```
if ($file != "." && $file != ".." && $file != ".DS_Store"
    && $file != ".txt") {
```

このコードを加えたことによってエラーが検出されなくなり、謎の空欄も表示されなくなった。ほぼ構想していたものと同じものが仕上がったといえるのでないかと考える。詳しいプログラム内容は資料 13 を参照とする。日付ページの作成は以上である。

(v) 削除について

日付ページを増やしたことによりそれと連動する削除プログラムのコードが機能しなくなったため、これらを再び使用できるようにプログラムを書き直す必要がある。削除確認ページは print する変数を変えるだけでちゃんと確認し作動する。日付ページでプログラムして送られた変数を使うと削除のプログラムも動作に成功した。成功すると、メインの下に出席確認表ページへと戻れるリンクを貼った。プログラムの詳細は資料 14 を参照とする。

(vi) 出席簿の改良

次は出席簿の改良をおこなった。前プログラムをそのまま起動すると 70 行目でエラーが起こっており、最初にそのバグ取りをおこなった。原因としては table で出力していた preg-split という関数の表記の記入ミスであった。この関数のなかには正規表現が使われているために見落としをしてしまったのだと思われる。その他にも黒い四角形や白い四角形の空間がページ内のテーブル周辺に出現するなどのバグが発生した。

しかしそれらは CSS が他の table タグと被っているために起こっているバグであり、新規に table タグを増やすか、それとも table 表記をやめれば解決する問題であった。私は出席簿ページでの table 表記をやめ、従来のメインである「出席簿右クリックでダウンロードして Excel に貼り付けてください。」と表示して、そこから出席簿データをダウンロードしてもらうことにした。プログラムの詳細は資料 15 を参照とする。

理由としては、第 1 に table 表記は日付ページ (hiduke.php) で既に表示しているためである。第 2 に出席簿をダウンロードしたファイル (資料 16 参照) が福田先生から「参考に授業で取った出席のファイルです。」と頂いた出席簿ファイル郡と表示に違いがあったためである (資料 17 参照)。

これは、学生用のページである出席確認ページから飛べる出席登録ページ (toroku.php) に福田先生が使っているプログラムと違いがあったためだと判明した。福田先生と相談した結果、先生が使われているファイルの変換方法へと統一してほしい、と依頼され出席簿の改良より出席登録ページの改良を先におこなうことにした。

出席登録ページの改良では、プログラムの中にある PHP の上部に

```
<?php
if ($error != "") {
die($error);
} else {
extract($_POST);
$date = date('Ymd');
$info = "<shusseki>\n<bango>$studentno</bango>\t<name>$name</name>\n<subject>$subject</subject>\n<pw>$pw</pw>\n</shusseki>";
$dir_name = "./data/$date";
```

とコードを書き直すことによって解決した。\$info 以下のコードが食い違っており、それらを新しいものに置き換えている。PC 側の日付を自動で取得し、尚且つ入力してもらった生徒番号、名前、感想内容、パスワードを関数 dir_name で変換した日付に名前を変えている。この下のコードに、その変換された名前で作成するコードを書いている。つまり、data というファイルの中に日付ごとのファイルが作成されて、その中に出席者の各データが入っているということである (資料 18 参照)。プログラムの詳細は資料 19 を参照とする。

出席登録ページ (toroku.php) での登録処理コードを新しいものに置き換えはしたが、日付ページ (hiduke.php) に table 表記されていた出席確認表には古いコードで作成されたものしか表示されないため、こちらも手を加える必要がある。資料 13 参照、73 行目でのコードを

```
if (preg_match("/<bango>(.*?)</bango>.*<name>(.*?)</na
```

```
me>.*<subject>(.*?)<\/subject>/s", $data, $matches))
```

とコードを書き直すことにより、変更された出席登録ページ (toroku.php) に対応した出席確認表が表示される。古いコードで作成された出席者のデータは残念ながらテーブルに表示はされない。出席簿および出席登録ページの改良は以上である。

(vii) CSS の見直し

全てのファイルの誤作動などのエラーがないか確認した後、デザイナーの大元となる CSS の改良へと着手した。CSS のフォルダが 1 つあり、その中に学生の出席登録ページ用のデザイン (index.css) と教員管理ページ用のデザイン (kanri.css) がある。出席登録用のデザインは白を基調にしたおとなしく目に優しいデザインにした (資料 20 参照)。改良するまえとさほどデザインは変更せず、不満だと思える点はなかったので背景を変えるだけに留まった。

教員管理ページのデザインは以前のデザインと比べてブログ風のデザインに変更して誰が使っても取っつきやすく改良した。出席登録ページに使ったものと異なった背景のイメージを使用し差別化をおこなった。ヘッダーの画像ファイルには福田ゼミでの制作、という意味を込めて福田先生の好きな猫の画像にした (資料 21 参照)。出席確認表、出席簿、ログアウトのリストを作りそれを教員管理ページの左に配置した。出席確認表のページは日付ごとにリスト表記され (資料 22 参照)、日付ページでは出席者が生徒番号・名前・感想・削除の順でテーブル表記されている (資料 23 参照)。削除の確認、確認後の削除ページのレイアウトについては資料 24、25 を参照してほしい。出席簿ページはレイアウトを変えたのみに留まった (資料 26 参照)。CSS の見直しについては以上である。

(viii) 問題点への対策

問題としては大きく2つに分けて述べられる。1つ目は(iv)でも述べているように、データ整理時・更新時に生成される".DS_Store"というデータが出席表に文字化けや謎の空欄を表しているという問題である。この解決策については(iv)で述べているために省く。

2つ目はファイルなどの細かいバグは多かったが殊更に多かったといえるものはデザインの要であるCSSである。新しくプログラムコードを書くたびにエラーが出て何が問題であるかを突き詰めていくとCSSとの互換性であった、というのが最多であった。日付ページ(hiduke.php)の日付リストなども教員管理ページ用のデザイン(kanri.css)と初期は整合性が取れなく、CSSで設定していたリストの設定が反映されてしまっていた。そのため対策として、CSSに書かれている

#main liの部分を

#main ul liのコードを付け加えることによって対策した。

しかし、それ以降もファイルの改良を重ねていくにつれてCSSとの整合性のズレが広がっていったため、福田先生と相談した結果、CSS自体を1から作り直すことに決定した。過去自分が作成したブログのCSSを応用したため、真っさらなCSSから始めたというわけではない。

出席登録アプリケーションの全てのファイル郡をその新規に作成したCSSに対応すべくタグを全てを変えていったが、その過程でもエラーは頻発した。Web上にCSS Validation Serviceというページがあり、そのページで自分が書いたCSSを直接入力することにより、どこからエラーが出ているのかが解る仕組みとなっている。このページを使用して1つ1つエラーを潰していき、(vi)でのデザインに至ったのである。以

上が大きな2つの問題に対しておこなった対策である。

4 まとめ

(1) 評価

自己評価としては、出席確認アプリケーションの制作者である中西陽子氏の提示していた「今後の課題や改善点」で挙げられていたログイン画面を作成したのに加えて、出席確認表ページの修正やその中に日付ページを作成し日付ごとに出席者のデータテーブルを表記したので、構想をしていたものよりも良いものが作れたのではないかと自己評価したい。しかし、福田先生と相談した結果出席者の総合カウント表示は出席確認表には入れず省いたが、どうにかしてカウント表示を加えてみたかったためその作成が出来なかったのが残念である。

後は、ログイン画面を作成したのはそれでいいのだが、ログインした教員用に授業の出席者の各データを分けることができず、出席データが統一化されてしまっていた。CSSのデザインの変更に時間をかけすぎたなどの様々な要因が重なった結果とはいえ、気づくのが早ければ対処できたのかもしれないと思うと悔やまれる。

私は人文情報学科に転学科してきたので2年しか情報系の勉学を学べなかったが、それでもその短い期間で得た知識を総動員してこの出席確認アプリケーションの改良をおこなった。エラーやバグは改良をするうえでの全てのファイルで出ており、その都度過去に学んだ資料を引っ張りだしてきてそれを応用して解決した。

PHPのコードも難しいものを使っており、3回生の折に学んだPHPに対する資料を熟読しては試してを繰り返しおこなった。それでも解ら

ない場所は Web ページで検索し、その単語の意味を理解して 1 つ 1 つ対処していった。CSS などは 3 回生の折にも少ししか触れていなく、特に理解が乏しかったため制作に 1 番時間がかかった。その甲斐あってブログ風の見栄えがよいものに仕上がったと自信をもっていえる。

PHP や CSS などの乏しかった知識が前以上に増えたことは確実にあり、制作は有意義なものであったと考える。ただ、最初の出席管理アプリケーションという制作物に対する理解やそのプログラムの仕組みの解明、新しく作成した CSS に時間をかけすぎたことにより、当初は完成したものを実際に学生に使用してもらい、また教員側にもログインから出席管理までの流れなどをテストすることができなくなってしまった。このことにより、自分が使用する分には見えない視点の意見や問題点などの燻り出しができなかったのが残念である。

(2) 今後の改善点

実際に利用してみてもの課題や改善点などは山積みかと思うが、大きな改善点としては 3 つに分けられるのではないかと考える。

(i) 出席のカウント

今回の改良では出席のカウントは福田先生との相談で省いたが、カウント表が必要な教員がいる可能性もあるので追加してもらおう。左のリストに出席カウントの欄を新しく作り、カウントページへ飛べるようにする。そこに左から、出席者の生徒番号、現在の出席状況などが書かれている表を作成する。時間配分のミスによりカウント表をどう作成するか資料を集められなかった。カウント表の作成か追加をすれば、よりよいアプリへと昇華できると考えている。

(ii) 出席データの分割化

ログイン画面を作成して上手く動作するのはいいのだが、そのまま教員管理ページへ遷移して出席確認ページへ移ると出席データが統一化されているのが分かる。個人で利用するぶんには何も問題はないのだが、この出席確認アプリケーションは複数で利用するのが目的であるために出席データを各教員ごとに分割化できればと考えている。教員ごとにログイン ID やパスワードを変えられるようにする仕組みがあればさらによいと思う。

どのように分割化するかなどの技術や知識が乏しかったことや時間配分のミスで資料も集められなかった。早期にこの問題に気づいていたのなら対策をできたかもしれないと思うと残念である。改善できるなら、出席者のデータを教員ごとに分割して出席確認表ページで表示できるようにして、ログイン ID やパスワードの変更タグも教員管理ページに加える処理をおこなうことで飛躍的に使用人口が増えると考えている。

(iii) 出席削除からのリンク

日付ページから出席者を削除した折に、メインに出席確認表へと飛べるリンクが出現する(資料 24 参照)。それを、出席者を削除した確認のためにその出席者を削除した日付のページへと飛べるように改善してほしい。CSS などに時間を割いたために時間が余りなく、改善するのに手が届かなかった。

この 3 点が今後の改善点として最も大きいものでないかと考える。(ii) を改善するだけでも、このアプリケーションを利用する教員が増える率は上がるのではないだろうか。

5 おわりに

この出席確認アプリケーションの改良は元からあったものに手を加えるわけであるため、制作を始めた折は簡単に事が終わっていろいろな機能をつける時間が余るのでないかと安易に考えていた。しかしそんな事はなく実際はその逆で、元からあったプログラムがどのような仕組みで動作しているのかが逐一分かっていなければ改良など出来なかったのである。重要な部分を不用意に消去してしまえば、それだけで出席確認がおこなえなくなる危険性をはらんでいるからである。そのため、まずプログラムの理解に時間を費やしたのだが、その解読が難航したのも時間ロスに繋がっている。

しかし、ここで慎重にプログラムを解読したからこそ出席確認がおこなえなくなるほどの失敗は起こさなかったのは僥倖であったといえるのでないだろうか。PHP は 3 回生で学んだことが活かせるところは多かったのだが、正規表現・CSS は 3 回生の折に少し触った程度であり、その部分を理解するのが鬼門であった。

特に正規表現は 3 回生で福田ゼミに入るためにおこなったテストで出題されたその時に、参考資料で学んで以来だったために解読に苦戦した。ただ、苦戦した甲斐があって一層プログラムによる理解が深まり、正規表現だけでなく CSS にも意欲的に取り組むきっかけとなった。デザインも、事務的なものから親しみやすいブログ風のデザインへと変更してどんな人でも使いたくなるようなデザインにするのに苦心した。

ただ、やはりプログラムの理解・解読による時間ロスのために客観的に評価を下してもらおうと考えていた、福田先生の講義で実際にプロセスを踏んで使ってもらったうえで、アンケートを取ることがおこなえなかったのは悔やまれる。アンケートを取れていれば、自分では気づけな

——出席確認アプリの改良について——

い細かいところの指摘などや新しい改良案も取り入れることが可能だったかもしれないためである。

福田ゼミのテーマである「人の役に立つものをつくる」というものに沿ったものづくりをできたと私はそう考える。利用する人のことを第1に考えてプログラムを組んだりデザインをしたりなどはやったことがなかったため、卒業制作という形ではあったが今回そのように心がけて作成できたのは良い経験であった。もし修正や改善するようなら、そのような場所を直してより使いやすく、利用者に「便利だ」と言ってもらえるようなアプリケーションに変えていってもらえたら幸いである。

文献表

- ・「授業出席登録」 <http://tiber.que.ne.jp/otani/shusseki/>
- ・「とほほの WWW 入門」 <http://www.tohoho-web.com/www.htm>
- ・「HTML クイックリファレンス」 <http://www.htmq.com/>
- ・「PHP マニュアル」 <http://www.php.net/manual/ja/>
- ・「正規表現オンライン教材」 <http://tiber.que.ne.jp/otani/RE/index.html>
- ・「CSS オンライン教材」 <http://tiber.que.ne.jp/otani/CSSkyozai/>
- ・「CSS ValidationService」 <http://jigsaw.w3.org/css-validator/>