

お気に入り動画管理アプリケーション (2021年7月26日版)

氏家 せいじ

概要：本稿は、お気に入り動画管理アプリケーションの内容をまとめたものである。本稿の構成としては、第1章に「機能概要」、第2章に「利用技術」、第3章に「システム詳細」、第4章に「実装」、第5章に「まとめ」という構成となっている。

Favorite Movie Management Application (version 2021/7/26)

UJIE SEIDY

1. 機能概要

お気に入り動画管理アプリケーションの機能概要は以下の通りである。

- ・ログイン
ログインは、入力したアカウント名とパスワードがデータベースに登録してあるのかを調べ、あるならば個人ページに遷移する機能である。
- ・新規登録
新規登録は、入力したアカウント名とパスワードをデータベースに登録する機能である。
- ・登録動画一覧
登録動画一覧は、動画登録で登録した内容を全て表示する。
- ・登録動画一覧（検索）
登録動画一覧（検索）は、登録動画一覧にある入力欄からチャンネル名を検索し、該当するチャンネル名のみ表示する機能である。
- ・動画登録
動画登録は、入力したチャンネル名とタイトルと url をデータベースに登録する機能である。
- ・ログアウト
ログアウトは、セッションを削除し、トップページに遷移する機能である。

以上がお気に入り動画管理アプリケーションの機能概要である。

2. 利用技術

この章では、お気に入り動画管理アプリケーションを制作するにあたり、利用した技術（主に言語）について述べる。

2.1 HTML

HTML は、Hyper Text Markup Language の略称であり、Web ページを制作するために開発された言語である。また、ハイパーテキストを記述するためのマークアップ言語の1つでもある。機能としては、ハイパーリンクや画像等のマルチメディアを埋め込むハイパーテキストとしての機能や、見出しや段落といったドキュメントの抽象構造、フォントや文字色の指定などの見た目の指定、などといった機能がある。HTML は1から始まり、改良が加えられたことで現在 HTML5 が使用されている。

2.2 PHP

PHP は、PHP: Hypertext Preprocessor の略称である。最初の文字が「頭字語の頭字語」になっているため「再帰的頭字語」と呼ばれる。また、PHP は、動的に Web ページを生成するために開発されたサーバーサイドのスクリプト

ト言語である。他のプログラミング言語と比較して仕様や文法が簡単であり、MySQL などのデータベースとの連携が容易なことなどから、WordPress を含めた Web アプリケーションの開発にもよく使われる。PHP の主な特徴として HTML で記述したファイルの中に埋め込んで使うことがあげられる。HTML で記述された内容の途中に PHP のプログラムを記述するというものである。これによって PHP・HTML を別々のファイルとして扱うのではなく、1 つのファイルにまとめることが可能となる。

2.3 CSS

CSS は、Cascading Style Sheets の略称であり、ウェブページのスタイルを指定するための言語である。ワープロソフトなどで作成される文書も含めて、文書のスタイルを指定する技術全般をスタイルシートという。HTML や XHTML などで作成されるウェブページにスタイルを適用する場合には、スタイルシート言語の 1 つである CSS が一般的に利用されている。また、HTML と組み合わせて使用する言語である。HTML がウェブページ内の各要素の意味や情報構造を定義するのに対して、CSS ではそれらをどのように装飾するかを指定する。例えば、ウェブページがスクリーンに表示される際の色・サイズ・レイアウトなどの表示スタイルや、プリンタなどの機器で印刷・出力される際の出力スタイル、音声で読み上げられる際の再生スタイルなど、ウェブページをどのようなスタイルで表示・出力・再生するかについて指定することができる。

2.4 JavaScript

JavaScript は、JS と省略されて使用されることがあり、動的な Web ページを作成する事のできるプログラミング言語で、通常はブラウザ上で実行される。JavaScript を使うと、ユーザーのアクションに応じたコンテンツの表示の他、ブラウザ上で表示される地図やグラフィックアニメーションなども表示する事が可能である。JavaScript は、開発当初 LiveScript と呼ばれていたが、その当時人気のあったプログラミング言語の Java に影響を受け、JavaScript という名前に変更された。JavaScript と Java にいくつかの共通点はあるが、まったく別の言語である。

2.5 MariaDB/MySQL

MySQL は、現在はオラクルが所有し、オープンソースとして開発が続けられている。MySQL はデータベースを保存する形式を選択できるところが特徴で、MyISAM 形式は速度が要求される処理に適している上に、現在の標準である InnoDB は MyISAM になかったトランザクション処理に対応している。このように業務に応じて選択できる点、高速であるということ、PHP などサポート言語が多く使いやすいことなどから世界で最も利用されているデー

タベースとなっている。また、レンタルサーバで付属しているデータベースとしても MySQL が多い。

MariaDB は、MySQL の生みの親 Michael "Monty" Widenius 氏が立ち上げたプロジェクトで開発が続けられているデータベースである。MySQL や MariaDB の名称は彼の娘たちの名前が由来で名付けられた。最初の娘の名前が「My」、次の娘の名前が「Maria」であった。MySQL が企業に買収されたのをきっかけに MariaDB は生まれた。MariaDB はフリーでオープンソースであることや、MySQL に対する高い互換性から、Linux 系の OS が標準でサポートするデータベースが MySQL から MariaDB に変更された。XAMPP の管理画面は MySQL と表示されているが、内部は MariaDB が動作している。

2.6 Apache

Apache とは、世界的に最も普及している Web サーバ (HTTP サーバ) ソフトウェアの一つ。Apache Software Foundation (Apache ソフトウェア財団) が開発しており、オープンソースソフトウェアとして公開している。古くから開発・公開されてきた人気の高い Web サーバで、安定性や機能の豊富さ、対応ソフトウェアの多さなどが評価され、企業から個人まで様々な用途で利用されている。Linux や Windows を始め様々な OS 上で動作し、いくつかの著名な OS では、その OS での動作に最適化された高性能なプログラムを利用することができる。Apache はオープンソースとしてソースコード自体が配布されているため、実行可能 (バイナリ) 形式へのコンパイル時にあらかじめモジュールを組み込み、Web サーバ本体に一体化させて動作させる「静的リンク」を行うことができる。別にコンパイルしたモジュールを実行時に連携させる「動的リンク」より導入は手間だが、高速に動作させることができる。

2.7 XAMPP

XAMPP (ザンプ) とは、Web アプリケーションの実行環境を構築するソフトウェアである。環境構築するのは、1 つのソフトウェアをインストールだけでしたら容易であるが、複数のソフトウェアとなると手間がかかる。XAMPP は、コマンド入力もなく詳細設定ファイルも変更せず、実行環境をボタンをクリックするだけで環境構築が可能である。

XAMPP は Apache Friends によって開発された。Apache サーバとの親和性も高い。5 つの単語の意味を組み合わせると XAMPP と名づけられた。

- X 「クロスプラットフォーム」
- A 「Apache」
- M 「MariaDB (旧バージョンは MySQL)」
- P 「PHP」
- P 「Perl」

3. システム詳細

この章では、お気に入り動画管理アプリケーションのシステム詳細について述べる。

まず、全てのファイルを列挙しその後ひとつずつ解説を行う。最後にデータベース設計について解説を行う。

このアプリケーションは、以下のファイル構成となっている。

- index.html
- PHP
 - delete.php
 - display.php
 - display(search).php
 - login.php
 - logout.php
 - movieregister.php
 - mypage.php
 - register.php
- CSS
 - display.css
 - display(search).css
 - index.css
 - login.css
 - logout.css
 - movieregister.css
 - mypage.css
 - register.css
- IMAGE
 - hanahana_settei6.gif
- JS
 - mypage.js

3.1 index.html

index.html はドキュメントルートとして設定されているファイルである。

3つの<div>タグで背景を分割している。また、<div>で分割している、content というクラスでは、<h1>タグを用いたタイトルの表示と、タグを用いたメニューの表示を行なっている。さらにメニューの表示は、menu というid でさらに分割し、<a>タグでログイン画面か新規登録に遷移するようになっている。

<head>内には、JS が埋め込まれており、mypage.php の画面でログアウトが選択された時、ユーザーにログアウトをしたことをアラートで表示するプログラムが記述されている。

3.2 PHP

3.2.1 delete.php

delete.php は、削除ボタンが押された項目にあるデータを消去する。try の中でデータベースに格納されているid を delete from movies where id = :id で消去する。この段階でそのid の列の動画がデータベースから消去される。エラーが発生したら catch の部分が実行されエラーメッセージとともにトップページへ戻される。全ての消去が問題なく行われた場合、登録動画一覧へと遷移する。

3.2.2 display.php

display.php は、ログインしているアカウントが登録した動画のチャンネル名と、動画名を表形式で全て表示する。3つの<div>タグで背景を分割している。そして、<h1>タグで、ページのタイトルを表示している。さらに、<div>タグで分割された、チャンネル名検索欄を作成し、検索ボタンを押すことで、display(search).php へ遷移するように作成した。また、チャンネル名検索欄は<form>タグが使われていて、入力をPOSTで受け取っている。

<head>の中でJSを動かしており、削除ボタンが押された時にその列の表示を削除するプログラムとなっている。

php側では、まずセッションを開始し、ログインデータがなければトップページに戻るよう作成した。そして、try の中でデータベースに接続し、その中の現在ログイン中のアカウント名(username)と一致するデータを選択する。また、そのデータを表を作成していく中で表示する。表示はforeachを用いて行うが、もし表示するデータが一件も見つからなかった場合、「登録動画がありません。」という表示を行う。ユーザーが登録したurlをクリックした時、別タブを開き、そのURLに遷移する。エラーが発生したら catch の部分が実行されエラーメッセージとともにトップページへ戻される。問題なく動作した場合ユーザーの入力まち、あるいは、操作待ち状態になる。

3.2.3 display(search).php

display(search).php は、display.php と同じ背景設定と、ページタイトル、チャンネル名検索欄で作成されている。二つのファイルの違いは、display.php のチャンネル名検索欄で入力されたチャンネルの動画のみ表示されるという点である。

html 内に記述されたphpでセッションを開始し、ログインデータがなければトップページに戻るよう作成。display.php で入力されたチャンネル名をPOSTで受け取り\$channel_searchに格納する。もし何も入力がなければ、全てのデータを表示する。そして、try の中でデータベースに接続し、その中のチャンネル名が検索したいチャンネル名(\$channel_search)と一致するデータを選択する。また、そのデータを表を作成していく中で表示する。表示は

foreach を用いて行うが、もし表示するデータが一件も見つからなかった場合、「登録動画がありません。」という表示を行う。ユーザーが登録した URL をクリックした時、別タブを開き、その URL に遷移する。エラーが発生したら catch の部分が実行されエラーメッセージとともにトップページへ戻される。問題なく動作した場合ユーザーの入力まち、あるいは、操作待ち状態になる。

3.2.4 login.php

login.php は、3 つの <div>タグで背景を分割し、box クラスのフォームでアカウント名とパスワードの入力画面を設置した。ログインボタンで php を実行しその下にはトップページに遷移するように <a href>タグをつけた。

php 側では、まずセッションを開始する。ログインボタンが押されてから、POST で受け取ったアカウント名入力値とパスワード入力値、ハッシュ化したパスワード入力値、認証失敗:0、認証成功:1、をそれぞれの変数に格納する。データが渡ってきた場合 try を実行する。try では、PDO インスタンスの生成と SQL の実行をする。実行後 SQL の実行結果を連想配列 \$data に格納する。\$data が一件のみだったら認証成功 (入力値が正しいか否かは SQL での照合で完了済み) というプログラムになっている。そして、認証が終わった段階で、DB 捜査関連の変数を初期化する。その後、もしログイン認証ができていたら POST で受け取ったアカウント名を \$_SESSION["username"] に格納し、mypage.php へ遷移する。ログイン認証ができていなかったら (アカウント情報が間違っていたら)、「ログイン情報が間違っています」とアラートする。try に入る前にそもそもデータが渡って来なかったときは、エラーメッセージとともにトップページへ戻される。

3.2.5 logout.php

logout.php は、比較的単純な作りになっている。mypage.php のログアウトという項目を押すと実行される。セッションを開始し、空の配列を \$_SESSION に格納する。そして、session_destroy() で、現在のセッションに関連づけられた全てのデータを破棄する。最後に「logout」というステータスを持った状態で、index.html へ遷移する。

3.2.6 movieregister.php

movieregister.php は、入力されたチャンネル名、動画名、URL をデータベースに入れるプログラムである。3 つの <div>タグで背景を分割し、box クラスの form を置いた。中身は、まず、h1 でページタイトルを表示しチャンネル名、タイトル、URL を <input>タグを用いて入力欄を作成した。そして登録ボタンを押すと、php が実行され、入力された値は、POST で受け取るようにした。また、「トップページへ戻る」を押すことで mypage.php へ遷移するよ

うになっている。

php 内では、まず、セッションを開始し、セッションにユーザーネームが格納されているかを確認する。入っていないければ、ログインエラーとみなし index.html へ戻す。そして、POST で受け取った、チャンネル名、動画名、URL をそれぞれの変数に格納する。try では、データベースに受け取った値をそれぞれ格納する。エラーが発生したら catch の部分が実行されエラーメッセージとともにトップページへ戻される。問題なく動作した場合は、「登録しました」とアラートし、再び入力待ち、あるいは、操作待ち状態になる。

3.2.7 mypage.php

mypage.php は、ログインが成功したアカウントが最初に遷移するページである。3 つの <div>タグで背景を分割し、また「header」、「dc-menu」、「dc-menu-trigger」、「menu-overlay」CSS 用にさらに分割しメニュー欄を制作した。メニュー欄は、「登録動画一覧」、「動画登録」、「ログアウト」の 3 つがあり、「登録動画一覧」は、「display.php」へ遷移し、「動画登録」は、「movieregister.php」へ遷移し、「ログアウト」は、「logout.php」へ遷移する。

php 内では、セッションを開始し、セッションにユーザーネームが格納されているかを確認する。入っていないければ、ログインエラーとみなし index.html へ戻す。

3.2.8 register.php

register.php は、会員登録を行うプログラムである。3 つの <div>タグで背景を分割し、box クラスの form を置いた。中身は、まず、h1 でページタイトルを表示しアカウント名、パスワードを <input>タグを用いて入力欄を作成した。そして登録ボタンを押すと、php が実行され、入力された値は、POST で受け取るようにした。また、「トップへ戻る」を押すことで mypage.php へ遷移するようになっている。

php 内では、POST で受け取った、アカウント名、パスワードをそれぞれの変数に格納する。try では、データベースに受け取った値をそれぞれ格納するが、データベース内に既に同じアカウント名が格納されていたら、「このアカウント名はすでに使用されています。」とアラートする。同じアカウント名がなかった場合のみ、受け取ったアカウント名と、受け取ったパスワードを、ハッシュ化した値を格納する。エラーが発生したら catch の部分が実行されエラーメッセージとともにトップページへ戻される。問題なく動作した場合は、「登録しました」とアラートし、index.html へと遷移する。

3.3 CSS

3.3.1 display.css

3 つに分割したバックグラウンドにアニメーションをつけ、keyframes を用いて緑と青のグラデーションに動くように作成した。そして、表は、text-align: center にすることで中央に表示させた。また、リストタブの「・」は、list-style-type: none で非表示にした。

3.3.2 display(search).css

display.css と全く同じ作りになっており、3 つに分割したバックグラウンドにアニメーションをつけ、keyframes を用いて緑と青のグラデーションに動くように作成した。そして、表は、text-align: center にすることで中央に表示させた。また、リストタブの「・」は、list-style-type: none で非表示にした。

3.3.3 index.css

背景は、display.css と同じで、3 つに分割したバックグラウンドにアニメーションをつけ、keyframes を用いて緑と青のグラデーションに動くように作成した。全体を中央に寄せ、リストのドットを非表示にした。そしてカーソルが当たったメニューに hanahana_settei6 の画像が表示される。

3.3.4 login.css

背景は、display.css と同じで、3 つに分割したバックグラウンドにアニメーションをつけ、keyframes を用いて緑と青のグラデーションに動くように作成した。ログインフォームのエリア内のスタイル、ユーザー ID とパスワード入力ボックスのスタイル、入力ボックスがフォーカスされたときのスタイル（幅が大きくなる&枠の色が変わる）、Login ボタンのスタイル、Login ボタンをマウスオーバーしたときのスタイルを作成した。

3.3.5 logout.css

背景は、display.css と同じで、3 つに分割したバックグラウンドにアニメーションをつけ、keyframes を用いて緑と青のグラデーションに動くように作成した。

3.3.6 movieregister.css

背景は、display.css と同じで、3 つに分割したバックグラウンドにアニメーションをつけ、keyframes を用いて緑と青のグラデーションに動くように作成した。登録フォームのエリア内のスタイル、チャンネル名と動画名、URL 入力ボックスのスタイル、入力ボックスがフォーカスされたときのスタイル（幅が大きくなる&枠の色が変わる）、登録ボタンのスタイル、登録ボタンをマウスオーバーしたときのスタイルを作成した。

3.3.7 mypage.css

背景は、display.css と同じで、3 つに分割したバックグラウンドにアニメーションをつけ、keyframes を用いて緑と青のグラデーションに動くように作成した。また、クラス分けした、「header」、「.dc-menu-trigger」、「.menu-overlay」でメニュー欄を中央に寄せ、色を指定し、リストの「・」を非表示にした。

3.3.8 register.css

背景は、display.css と同じで、3 つに分割したバックグラウンドにアニメーションをつけ、keyframes を用いて緑と青のグラデーションに動くように作成した。会員登録フォームのエリア内のスタイル、ユーザー ID とパスワード入力ボックスのスタイル、入力ボックスがフォーカスされたときのスタイル（幅が大きくなる&枠の色が変わる）、登録ボタンのスタイル、登録ボタンをマウスオーバーしたときのスタイルを作成した。

3.4 IMAGE

3.4.1 hanahana_settei6.gif

index.html のメニューのどれかにカーソルが当たった時に表示される画像。光るハイビスカスである。

3.5 JS

3.5.1 mypage.js

mypage.js は、ログインした後に遷移する mypage.php のメニューに動きをつけるために作成した。ハンバーガーメニューとなっており、クリックすると横にスワイプしメニューが表示される。

3.6 DB

このアプリケーションは「registeredmemberdb」というユーザーで2つのテーブルを作成した。1つは、users テーブルであり、もう1つは、movies テーブルである。users テーブルは、username と password というカラムを設定している。password はハッシュ化されたものが格納される。username は、varchar(100) で設定し、password は、varchar(255) で設定した。また、movies テーブルは、id と username と channel と title と url というカラムを設定している。id は、int(255) で設定し主キーにしている。また、オートインクリメントをつけ重複が無いようにしている。username は、varchar(100) で設定し、channel は、varchar(100) で設定し、title は、varchar(255) で設定し、url は、varchar(255) で設定している。

4. 実装

4.1 画面遷移

このアプリケーションは「localhost」とurlを入力すると、「index.html」から始まる。そこには、2つのメニューがあり、「ログイン」を押すと、「login.php」へ遷移し、「新規登録」を押すと、「register.php」へ遷移する。「login.php」、「register.php」は、どちらも「index.html」に遷移するリンクがあり、「login.php」は、ログイン後「mypage.php」へ遷移する。「mypage.php」には、3つのメニューがあり、「登録動画一覧」を押すと、「display.php」へ遷移し、「動画登録」を押すと、「movieregister.php」へ遷移し、「ログアウト」を押すと、「logout.php」へ遷移する。「display.php」、「movieregister.php」は、どちらも「mypage.php」に遷移するリンクがあり、「display.php」は検索機能を利用すると、「display(search).php」へ遷移する。「logout.php」へ遷移後「index.html」へ遷移し最初のページへ戻る。

以下に、簡単な図でまとめる。

```
index.html
  ->login.php
    ->mypage.php
      ->display.php
        ->display(search).php
          ->mypage.php
            ->movieregister.php
              ->mypage.php
                ->logout.php
                  ->index.html
                    ->index.html
  ->register.php
    ->index.html
```

4.2 実装環境

- macOS
 - バージョン 11.2.3
 - MacBook Air(13-inch, 2017)
 - プロセッサ 2.2 GHz デュアルコア Intel Corei7
 - メモリ 8 GB 1600 MHz DDR3
 - 起動ディスク Macintosh HD
 - 内蔵ディスプレイ 13.3 インチ (1440 x 900)
- XAMPP
 - バージョン 8.0.3-0
 - 起動サーバー (manager-osx)
 - * MySQL Database
 - * ProFTPD
 - * Apache Web Server
- Google Chrome

- Visual Studio Code
 - バージョン 1.58.0

5. まとめ

このアプリケーションを制作するに至った経緯としては、日常生活で、YouTubeをよく見るが、高く評価した動画を検索できないことが不便だと感じた。そこで、その動画をチャンネルごとにまとめることで、自分だけのお気に入り動画を集められると思い、このお気に入り動画管理アプリケーションの制作に至った。このアプリケーションを制作するにあたって、4つのものを目標に制作した。

- (1) 外部からの攻撃を考慮すること。
- (2) ビジュアル面に気を配ること。
- (3) 会員機能をつけること。
- (4) データの取り扱いに気を配ること。

1つ目の「外部からの攻撃を考慮すること。」は、クロスサイトスクリプティングに考慮し、受け取った値を、htmlspecialcharsで囲むことで、入力欄で入力されたHTMLタグを誤って処理するのを防ぐことができる。完璧に全ての攻撃を防げるようにはできていないためまだまだ改善の余地がある。

2つ目の、「ビジュアル面に気を配ること。」は、CSSとJSを用いて、見た目を変更した。何もないと質素な感じを与えてしまうため、動的なデザインを用いて、背景を動かしたり、ハンバーグメニューを横にスライドさせたりなど、視覚的な面にも手を加えたものを制作した。

3つ目の、「会員機能をつけること。」は、中間の段階で会員機能を実施していたため、その機能は最低限つけることにした。教科書で実装していた会員登録機能は、メールアドレスを利用するものであったため、それを無くしアカウント名とパスワードのみの登録にした。会員機能をつけることで、アカウントごとに動画を管理することを可能にし、第3者が登録した動画を勝手に消したり、知らない動画を勝手に追加されることを防ぐことが可能である。

4つ目の、「データの取り扱いに気を配ること。」は、パスワードに考慮することである。データベースに直接パスワードを格納してしまうと、データベースの中身を見られてしまった場合、パスワードが容易に第3者の手に渡ってしまう。それを防ぐために、sha256を用いて、ハッシュ化を行なった。こうすることで、データベースの中身を見られてしまった場合でも、パスワードを盗まれることなく、データを扱うことができる。今回のアプリケーション開発は、初の開発であったが、基本的な会員機能の実装と、少しではあるが、セキュリティ面に考慮したものを実装できた。他にも、URLを押した時にそのページで遷移してしまうと、次の自分が登録した動画を見るときに、1度ブラウザバックをしないといけなくなってしまうため、使いやすさを考慮して、" target="_blank" rel="noopener noreferrer"

でセキュリティ対策をした上で、別タブで遷移するようにした。以上がお気に入り動画管理アプリケーションの解説である。

参考文献

- [1] PHP7+MariaDB/MySQL マスターブック、マイナビ