

PORTFOLIO

Hiromasa Hibino

投票アプリ

目次

プログラム概要	2 ページ
フローチャート	4 ページ
アプリケーション画面	5 ページ
ファイルの構造	11 ページ
ファイルの説明	13 ページ

プログラムの目的

アプリケーションの開発経験を積む

学んだ知識の復習、知識の定着

コードが動かないときに、自分でデバッグなどを行い解決できるようにする

コード整理の方法を学ぶ

使用技術

フロントエンド	HTML, CSS, Sass, Bootstrap, JavaScript
バックエンド	PHP
データベース	MySQL
サーバー	Xfree(レンタルサーバー)
ツール	Visual Studio Code

アプリへのリンク

<http://kunren0622.php.xdomain.jp/pollapp/>

＜ベーシック認証＞

ユーザー名 : testuser

パスワード : Eg6bnGZd

＜アプリ内ログイン用テストユーザー＞

ユーザー名 : test

パスワード : test

＜ソースコード: GitHub＞

<https://github.com/hiromasa32/pollapp.git>

主な機能

1. ログイン管理機能

ユーザー登録をしてログインをすることで、トピックへの投稿やトピックの作成が可能です。

2. 投票機能

「賛成」または「反対」の選択肢からいずれかを選び、それを選んだ理由を入力して投票できます。

3. トピック管理機能

ユーザーは自分の投稿したトピックの編集ができます。
トピックの修正、公開・非公開表示の設定が可能です。

4. グラフ表示機能

トピックごとの投票の賛成・反対の割合を円グラフで表示することで、視覚的にトピックへの意見を把握できます。

5. レスポンシブデザイン対応

PC、スマートフォンのどちらからでも閲覧・投票が可能です。

フローチャート

トピックへの投票・トピックの管理ができるアプリケーション。

会員登録 or ログイン

ログイン後は、ログイン情報がセッションに保持される。

トップ画面(トピック一覧画面)

一覧から投票したいトピックを選択する

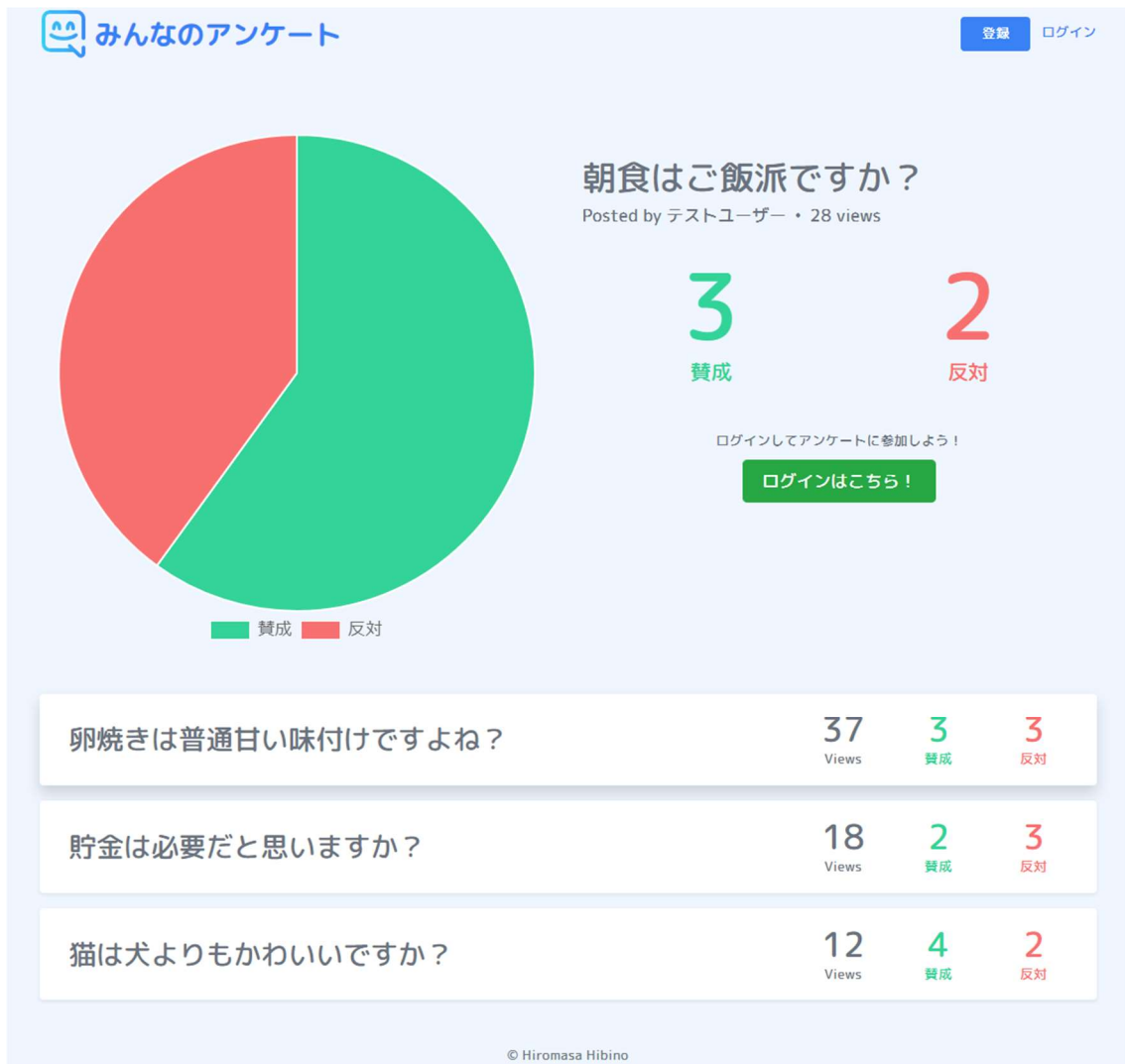
投票する

「賛成」「反対」のいずれかを選択、コメントを入力して投票。

ユーザーのトピック管理

- ・ トピックの新規作成
- ・ トピックの修正
- ・ トピックの公開・非公開表示の設定

トップ画面（トピック一覧表示画面）



ログインをしていない状態では、トピックの閲覧のみ可能です。

最新のトピックが大きく表示されます。

トピックを選択することで、対象のトピックについて投票できる画面に移動します。

登録画面

新規のユーザー登録を行う画面です。

入力された情報のチェックを行い、全ての項目が条件通り入力されている場合、登録ボタンが押せます。

ユーザーID: 4 文字以上 10 文字以内の半角英数字で入力

パスワード: 4 文字以上の半角英数字で入力

ニックネーム: 10 文字以内で入力

ユーザーID

パスワード

ニックネーム

[ログインへ](#)

ユーザーID

 ✓

パスワード

 ✓

ニックネーム

 ✓

[ログインへ](#)

ユーザーID

 ⚠

4文字以上で入力してください。現在の文字数は1文字です。

パスワード

 ⚠

4文字以上で入力してください。現在の文字数は3文字です。

ニックネーム

 ⚠

値の入力が必須です。

[ログインへ](#)

ログイン画面

以前にユーザー登録した場合、その情報を使用してログインします。

入力された情報のチェックを行い、全ての項目が条件通り入力されている場合、ログインボタンが押せます。

ユーザーID: 4文字以上 10文字以内の半角英数字で入力

パスワード: 4文字以上の半角英数字で入力

The image displays three sequential screenshots of a login form, illustrating the validation process. Each form contains two input fields: 'ユーザーID' (User ID) and 'パスワード' (Password), along with two buttons: 'アカウント登録' (Account Registration) and 'ログイン' (Login).

- Top Left Screenshot:** Shows the initial state with empty input fields. The 'ログイン' button is disabled.
- Top Right Screenshot:** Shows a successful login state. The 'ユーザーID' field contains 'test1' and the 'パスワード' field contains '.....', both with green checkmarks. The 'ログイン' button is now active.
- Bottom Screenshot:** Shows an error state. The 'ユーザーID' field contains 'tes' and has a red error message: '4文字以上で入力してください。現在の文字数は3文字です。' (Please enter 4 or more characters. The current number of characters is 3). The 'パスワード' field is empty and has a red error message: '値の入力が必須です。' (Value input is required). The 'ログイン' button is disabled.

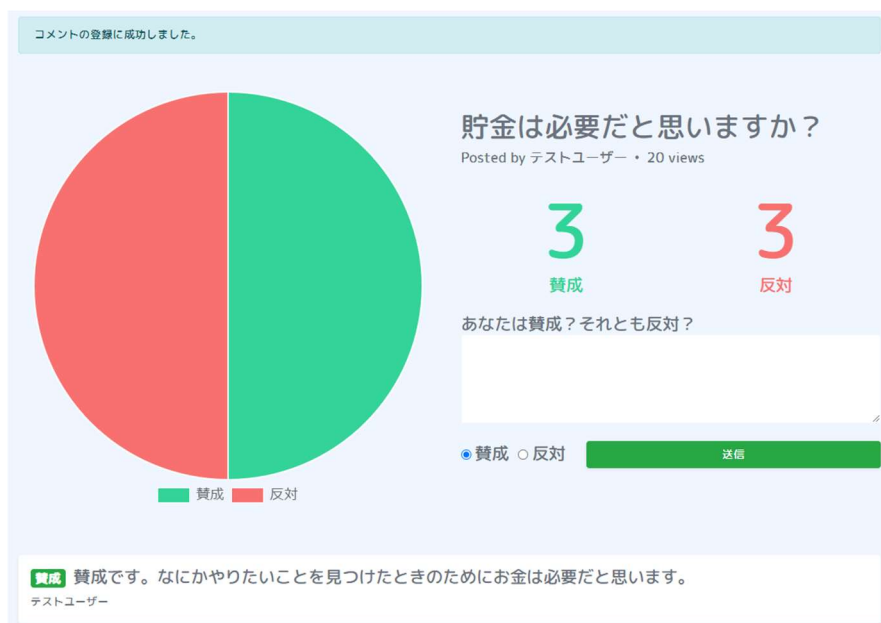
投票画面

ログイン後に投票することが可能です。

トピックに対してのコメントと「賛成・反対」を選択して送信ボタンを押すことで投票できます。



投票後は「賛成・反対」の選択した側に数がカウントされて、コメントが追加されます。



トピック新規作成画面

「投稿」ボタンをクリックすることでトピック新規作成画面へ移動する。
トピックタイトル、ステータス(表示・非表示)を選択してトピックを新規作成できる。

トピック作成

タイトル

習いごとは、できるだけ小さい頃からした方がいいと思いますか？



ステータス

公開



送信

戻る

トピックの登録に成功しました。

過去の投稿

公開

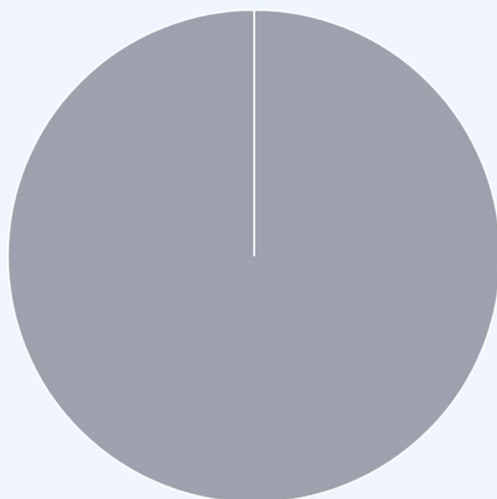
習いごとは、できるだけ小さい頃からした方がいい
と思いますか？

0
Views

0
賛成

0
反対

投稿が完了するとトップ画面に新規トピックが追加されます。



まだ投票がありません。

習いごとは、できるだけ小さい頃
からした方がいいと思いますか？

Posted by テストユーザー ・ 0 views

0
賛成

0
反対

あなたは賛成？それとも反対？

☒ 賛成 ☐ 反対

送信

トピック管理画面

ユーザー自身が投稿したトピック一覧が表示されます。

過去の投稿

公開 朝食はご飯派ですか？	28 Views	3 賛成	2 反対
公開 卵焼きは普通甘い味付けですよね？	37 Views	3 賛成	3 反対
公開 貯金は必要だと思いますか？	18 Views	2 賛成	3 反対
公開 猫は犬よりもかわいいですか？	12 Views	4 賛成	2 反対

ユーザーは自分の投稿したトピックの編集ができます。
トピックの修正、公開・非公開表示の設定が可能です。

トピックの編集

トピックの編集

タイトル

ステータス

公開

送信

戻る

ファイル構造

MVC モデルに分けてディレクトリを作成

[php]ディレクトリ

index.php : ウェブアプリケーションのエントリーポイントとなるファイル。

config.php : 定数が定義してあるファイル

[models]ディレクトリ : データの管理

abstract.model.php

commnet.model.php

topic.model.php

user.model.php

[views]ディレクトリ : 画面の入出力

404.php

home.php

login.php

register.php

[topic]ディレクトリ

archive.php

detail.php

edit.php

[controllers]ディレクトリ : 処理を行う

home.php

login.php

logout.php

register.php

[topics]ディレクトリ

archive.php

create.php

detail.php

edit.php

[partials]ディレクトリ : 複数のページで共通して使用するファイル

header.php

footer.php

topic-header-item.php

topic-list-item.php

[db]ディレクトリ : データベースに関連するファイル

comment.query.php

datasource.php

topic.query.php

user.query.php

[libs]ディレクトリ : 認証機能、ルーティング機能など

auth.php

helper.php

message.php

router.php

[css]ディレクトリ : スタイルシート

[bootstrap]

style.css

style.scss

[images]ディレクトリ : 画像ファイル

logo.svg

[js]ディレクトリ : JavaScript ファイル

form-validate.js

pie-chart.js

[vender]ディレクトリ

 chart.js

【データベース】(Xfree サーバーの MySQL を使用)

<データベース名>	kunren0622_pollapp
<テーブル名>	
「topics」	投稿内容を管理する
「comments」	投稿された内容に対してのコメント欄を管理する
「users」	ユーザー情報を管理する

各ファイルの説明

■ config.php

よく使用するパスは define() で定数として定義して利用する

■ index.php

「header.php」, 「router.php」, 「footer.php」を組み合わせてページを表示する。

require_once で各ファイルを読み込む

header 部分は header.php を読み込む。

ログイン状態によって表示される内容を切り替える。

body 部分は router.php の route() 関数によって表示を動的に切り替える。

【controllers】ディレクトリ

■ home.php

topic.query.php の fetchPublishedTopics() 関数を実行して、トピック情報を取得。

views/home.php を実行する。

■ register.php

[GET 送信の場合]

get() 関数が実行されて、view/register.php を呼び出す。

登録ページが表示される。

ログインをしていない場合は、こちらのページが表示される。

[POST 送信の場合]

登録ページで入力して登録ボタンを押すと、POST 送信を行い `post()`関数が実行される。
`auth.php` の `regist()`関数でユーザーID とパスワード、ニックネームの入力チェックをする。
その後、データベースと接続して同じユーザーがいないか確認する。
問題なければデータベースに登録される。
登録成功時は、データがセッションに保存されてログインが保持された状態でトップ画面に戻る。

■ `login.php`

[GET 送信の場合]

`get()`関数が実行されて、`view/login.php` を呼び出す。
ログイン入力ページが表示される。
ログインをしていない場合は、こちらのページが表示される。

[POST 送信の場合]

ログイン入力ページで入力してログインボタンを押すと、POST 送信を行い `post()`関数が実行される。
`auth.php` の `login()`関数でユーザーID とパスワードの入力チェックをする。
その後、データベースと接続して情報を取得。一致すればログイン成功。
ログインデータがセッションに保存されてトップ画面に戻る。

■ `logout.php`

`auth.php` の `logout()`関数が実行される。
`abstract.model.php` の `clearSession()`関数が実行され、セッションが削除される。
その後、トップ画面に戻る。

■ `topic/archive.php`

セッションを利用してデータベース情報を取得する。
ユーザートピックが 1 つ以上あれば、`views/topic/archive.php` の画面を表示。
ない場合はメッセージが表示される。

■ `topic/detail.php`

データベースから `topics` テーブルと `comments` テーブルの情報を取得して、`view/topic/detail.php` の画面を表示。
トピックのコメント入力後に送信ボタンを押すと、POST 送信され、コメントが登録される。

■ `topic/create.php`

セッション情報を利用して、`view/topic/edit.php` を実行。トピックの新規作成画面を表示する。

トピック入力後に送信ボタンを押すと、POST 送信されてトピックが登録される。

■ topic/edit.php

セッション情報を利用して、view/topic/edit.php を実行してトピックの編集画面を表示する。
トピック変更後に送信ボタンを押すと、POST 送信されてトピックの変更が反映される。

【views】ディレクトリ

■ home.php

取得したトピック情報を使用して、トピックの中央画面と一覧画面の表示を行う

トピックの中央画面 : topic-header-item.php

トピックの一覧画面 : topic-list-item.php

■ register.php

アカウント登録の画面

■ login.php

ログイン認証の画面

■ 404.php

404 ページ

■ topic/archive.php

ログインユーザーが投稿したトピック一覧を表示する。

topic-list-item.php を foreach で繰り返し呼ぶことで、トピックを表示している。

■ topic/detail.php

トピックの詳細表示画面

トピックに対しての詳細な内容が表示される。

■ topic/edit.php

第二引数が true の場合は、トピックの編集画面が表示。

false の場合は、トピックの作成画面が表示される。

【partials】ディレクトリ

■ header.php

header 部分を出力する。

ログアウト時は、「登録」「ログイン」ボタンが表示される。

ログイン時は、「投稿」「過去の投稿」「ログアウト」ボタンが表示される。

helper.php の the_url()関数でパスを取得して、指定のページに遷移する。

■ footer.php

コピーライト@の表記

JavaScript ファイルの読み込み

■ topic-header-item.php

トップ画面の中央に新規トピックを大きく表示する。

左側の円グラフは chart()関数で表示。

右側のコメント部分は topic_main()関数と comment_form()関数で行う。

comment_form()関数は投稿の入力フォームの表示になるので、ログイン状態のときのみ表示する。

■ topic-list-item.php

トップ画面の下にトピック一覧を foreach で繰り返し表示する。

トピックタイトル、ビューの数、賛成数、反対数を表示。

トピックタイトルをクリックすると、トピック id と同じ詳細ページ(detail.php)へ移動する。

【models】ディレクトリ

■ abstract.model.php

親クラスとなる。継承して利用している。

setSession()関数: セッションに登録

getSession()関数: 登録したセッション値を取得

clearSession()関数: セッションのクリア

getSessionAndFlush()関数: セッション値を取得した後、クリアする

■ comment.model.php

トピックのコメントについてのチェック

abstract.model.php を継承している。

validateAgree()関数: 賛成・反対の入力チェック

validateBody()関数: コメントの入力チェック、100 文字以内

■ topic.model.php

トピック投稿・編集のチェック

abstract.model.php を継承している。

validateId()関数: id のチェック

validateTitle()関数: トピックタイトルの入力チェック

validatePublished()関数: ステータス(賛成・反対)のチェック

■ user.model.php

ユーザー登録・ログインで使用する。auth.php

abstract.model.php を継承している。

validateId()関数: ユーザーID の入力チェック

validatePwd()関数: パスワードの入力チェック

validateNickname()関数: ニックネームの入力チェック

【libs】ディレクトリ

■ auth.php

login()関数: ログイン管理

regist()関数: 登録管理

isLogin()関数: ログインしているか確認

logout()関数: ログアウト管理

requireLogin()関数: ログイン状態の確認をする。していない場合、ログインページへ遷移。

hasPermission()関数: クエリを実行して、指定のトピックが存在するか確認する。

requirePermission()関数: 編集権限があるか確認。false でログインページへ遷移。edit.php

■ helper.php

get_param()関数: 引数が true の場合は、\$_POST で取得。false の場合は、\$_GET で取得。

redirect()関数: 条件によってホーム画面や一つ前の画面にリダイレクトする

the_url(): 引数のパスを出力する

is_alnum()関数: 引数が数字とアルファベットで構成されているかをチェックする

escape()関数: 配列、オブジェクト、単一データのエスケープを行う

■ router.php

route(\$rpath, \$method)の引数によって表示するページを動的に切り替える。

controllers 内の \$rpath で指定されたファイルを読み込んで呼び出す。

第一引数(\$rpath)の値は

\$rpath = parse_url(\$_SERVER['REQUEST_URI'])となる。

第二引数(\$method)の値は GET, POST のいずれかになる。

読み込んだ php ファイルの `get()`関数、`post()`関数のどちらを実行するか決定する。

■message.php

「error」「info」「debug」のメッセージ管理を行う

`push()`関数: セッションにメッセージを追加する

`flush()`関数: メッセージを表示する

`init()`関数: セッション内のメッセージを初期化する

【db】ディレクトリ

■datasource.php

データベースへ PDO 接続

`select()`関数: 全てのデータを取得

`selectOne()`関数: データを 1 つ取得

`execute()`, `begin()`, `commit()`, `rollback()`: トランザクション処理用の関数

■topic.query.php

topics テーブルのクエリ全般

トピックの操作を行う

`fetchByUserId()`関数

topics テーブルを対象に `user_id` に当てはまるデータを全て取得する。

例: `user_id` が「test」の場合、test の topice テーブルのデータを取得できる。

`archive.php` など、対象ユーザーのトピック一覧データを取得する時に使用する。

`fetchPublishedTopics()`関数

topics テーブルと users テーブルを対象にする。

`del_flg` が 1 以外で `published=1` (公開) に当てはまるデータをトピック id の降順に並びかえて取得する。

取得するデータは、トピックテーブルの全ての情報と、users テーブルの `nickname`。

`home.php` で全てのトピックデータを表示する時に使用する。

`fetchById()`関数

topics テーブルと users テーブルを対象にする。

topics テーブルの `id` がマッチして、`del_flg` が 1 以外、`published=1` (公開) に当てはまるデータを 1 件取得する。

取得するデータは、トピックテーブルの全ての情報と、users テーブルの `nickname`。

edit.php, detail.php など対象の 1 件のデータが必要なときに使用する。

incrementViewCount()関数

topics テーブル内の対象 id の views の値を+1 する

detail.php でページが表示されたときに値を 1 加算する。

isUserOwnTopic()関数

編集権限があるか確認するために使用。

トピックテーブル内の指定された id と user_id が存在するかカウントする。

update()関数

topics テーブルの対象 id の title(トピックタイトル)、published(賛成・反対)を変更する

edit.php で使用。

insert()関数

topics テーブルにトピックを 1 件挿入する。

title, published, user_id は入力値が反映される。

create.php で使用。

incrementLikesOrDislikes()関数

topics テーブル内の指定の topic_id を対象に likes, dislikes のいずれかを 1 加算する

detail.php の登録完了時に実行する。

■ comment.query.php

comments テーブル全般

トピックのコメントについてのクエリ

fetchByTopicId()関数

comments テーブル内の topic_id が一致したデータを取得する。

comments テーブルのデータ全てと users テーブルの nickname を取得。

detail.php でトピックのコメント一覧を表示するときに使用する。

insert()関数

topic_id, agree, body, user_id が登録される。

detail.php で入力したコメントを POST 送信するときに実行される。

■user.query.php

users テーブル全般のクエリ

登録したユーザー情報を操作する。

fetchById()関数

users テーブル内の指定の id を全て取得する。

id=test の場合、test ユーザーの情報を取得する。

auth.php のユーザー登録やログイン認証で実行される。

insert()関数

users テーブルに id, pwd, nickname を挿入する。

auth.php のユーザー登録時に実行される。

【JavaScript】ディレクトリ

■chart.js

JavaScript でグラフ(チャート)を描画するためのライブラリ

円グラフの表示に使用した。

■pie-chart.js

グラフの設定を行った

■form-validate.js

登録フォーム、ログインフォームの入力値チェック

【CSS】ディレクトリ

■Bootstrap

デザインは Bootstrap を使用

レスポンス、要素の配置、色

■SCSS

色の変数定義

Bootstrap の値を上書きして使用