PHP応用MVC作成

MVC作成

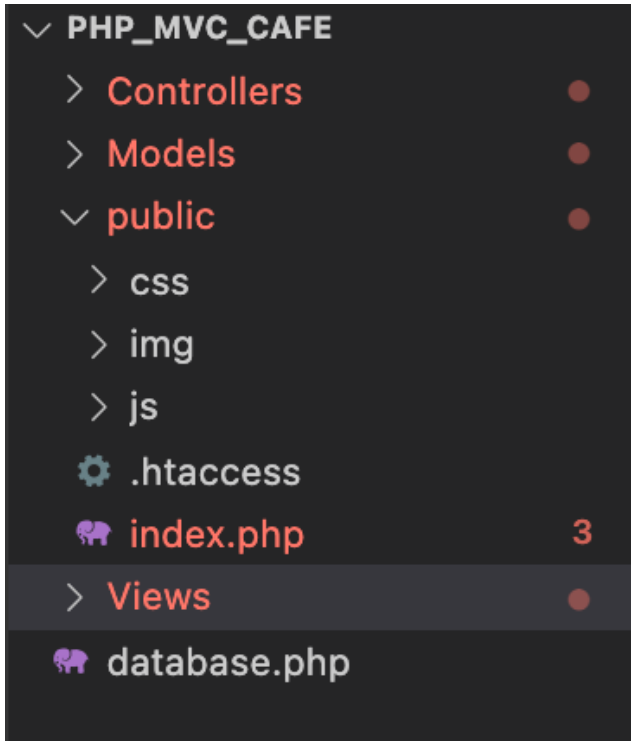
- 事前準備
  - テーブル作成
  - フォルダ構成
- お問い合わせ機能
  - .htaccessの追加
  - Database.phpの追加
  - Create
  - Read
  - Update
  - Delete

テーブル作成

- 以下DBとテーブルを作成してください。
- 制約は忘れずに設定すること(もし不明な場合、ご自身で調べて設定してください)
  - DB名: casteria
  - テーブル名:contacts

カラム名	型	論理名	制約	備考
id	int	システムID	主キー	オートインクリメント
name	varchar(50)	氏名	NotNull	
kana	varchar(50)	フリガナ	NotNull	
tel	varchar(11)	電話番号		
email	varchar(100)	メールアドレス	NotNull	
body	text	お問い合わせ内容		
created_at	datetime	送信日時		CURRENT_TIMESTANP

フォルダ構成の説明



ダウンロードしたzipファイルのPHP応用_演習の中身にあらかじめ最低限のフォルダは作成していますが、お問い合わせ機能実装前の事前準備完了時点でのフォルダ構成は以下のようになります。

- Controllers: 処理振り分けを担当
- Models: データ操作やDB接続を担当
- Views: 見た目(html等)を担当
- public: 最初に読み込まれるルート
  .htaccess :Apacheの設定ファイル
- database.php :DB接続情報を記載

次のステップで.htaccessとindex.phpの追加を行います。

Database.phpの追加

database.php

database.php

```php
<?php
define('DB_HOST', 'localhost'); // データベースのホスト名又はIPアドレス
define('DB_USER', 'root');      // MySQLのユーザ名
define('DB_PASSWD', 'root');    // MySQLのパスワード
define('DB_NAME', 'casteria');  // データベース名
```

データベースの接続に必要な情報を定数として定義しModelで呼び出してください。

.htaccessの記述とindex.phpの追加

.htaccess

```
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^(.*)$ index.php [QSA,L]
</IfModule>
```

Publicフォルダ配下に.htaccesssというファイルを作成し左画像のように記述してください。

一つ一つの記述に関してはご自身で確認してください。

index.php

```
<?php
define('ROOT_PATH', str_replace('public', '', $_SERVER["DOCUMENT_ROOT"]));
$parse = parse_url($_SERVER["REQUEST_URI"]);
// ファイル名が省略されていた場合、index.phpを補填する
if(mb_substr($parse['path'], -1) === '/') {
    $parse['path'] .= $_SERVER["SCRIPT_NAME"];
}
require_once(ROOT_PATH.'Views'.$parse['path']);
```

Publicフォルダ配下にindex.phpというファイルを作成し左画像のように記述してください。

*上記ファイルの追加が完了したら、MAMP/XAMPのドキュメントルートをpublicに変更してください。

お問い合わせ機能

- 事前準備まで終えたら実際にオブジェクト指向を用いてMVCを作成していきましょう。
- イメージが付きやすいようにダウンロードしてきたフォルダの中にサンプルを用意してあるのでModel、Controller,Viewの作り方の参考にしてください。

お問い合わせ機能（Create)

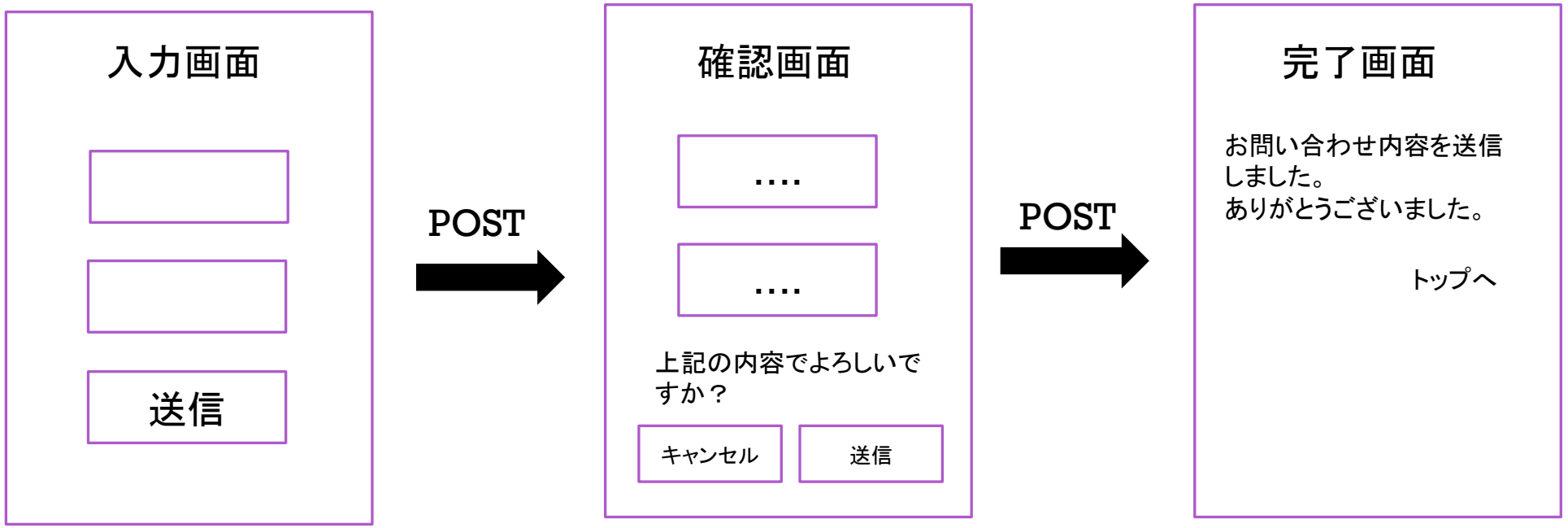
まず初めにお問い合わせからデータの登録する機能を追加します。
要件は以下の通りです。

- 入力画面で入力した内容をデータベースに保存できること
- PHP/Javascript両方でバリデーションを実装すること

氏名	空チェック、10文字以内(半角、全角区別なし)
フリガナ	空チェック、10文字以内(半角、全角区別なし)
電話番号	数字(0-9)かどうか(必須入力ではない）
メールアドレス	空チェック、メールアドレス(xxx@xxx)かどうか
お問い合わせ内容	空チェック、(確認画面で改行を読み込むこと)

- バリデーションに引っかかった後に値を保持していること、エラーメッセージが表示されること
- エスケープ処理がされていること
- トランザクションを使用すること
- お問い合わせ画面→確認画面→完了画面を作成すること
- 確認画面で入力された値が表示されること
- 確認画面、完了画面へのダイレクトアクセスの禁止
- レイアウトに関しては特に指定はないが整っていること

お問い合わせ機能(Create)の画面構成



*キャンセルボタン押下時には入力画面に戻り値が保持されていること

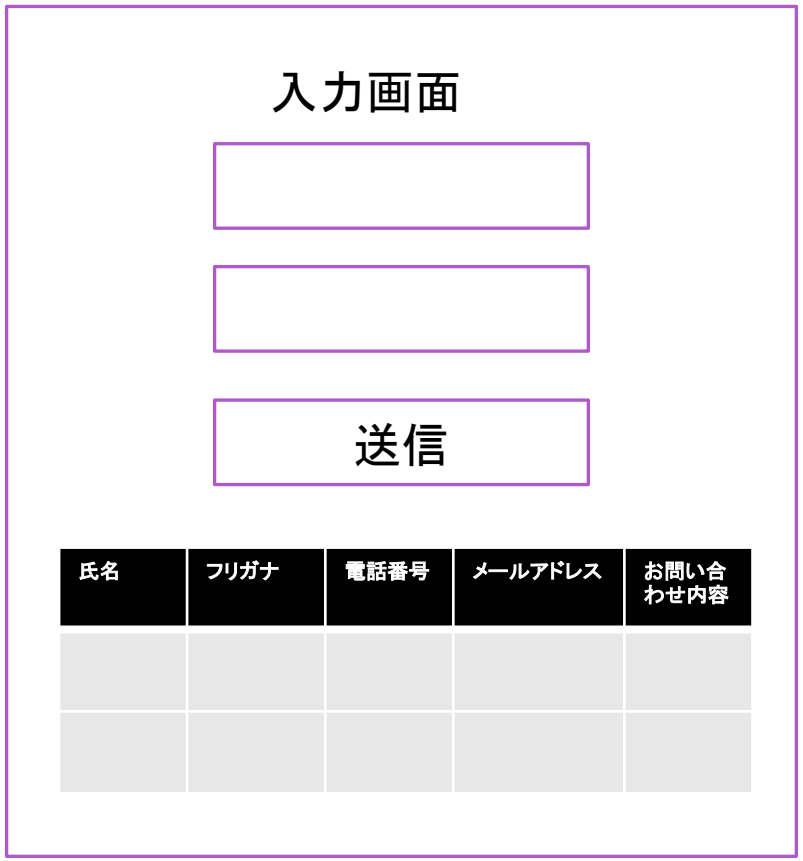
お問い合わせ機能（Read)

前項で作成したデータを一覧表示させる機能を追加します。
要件は以下の通りです。

- 氏名、フリガナ、メールアドレス、電話番号、お問い合わせ内容が一覧表示できていること
- お問い合わせ内容に改行も反映させれること
- テーブルのレイアウト崩れがないこと

*テーブルの表示イメージについては次のページに記載

お問い合わせ機能(Read)の画面構成イメージ



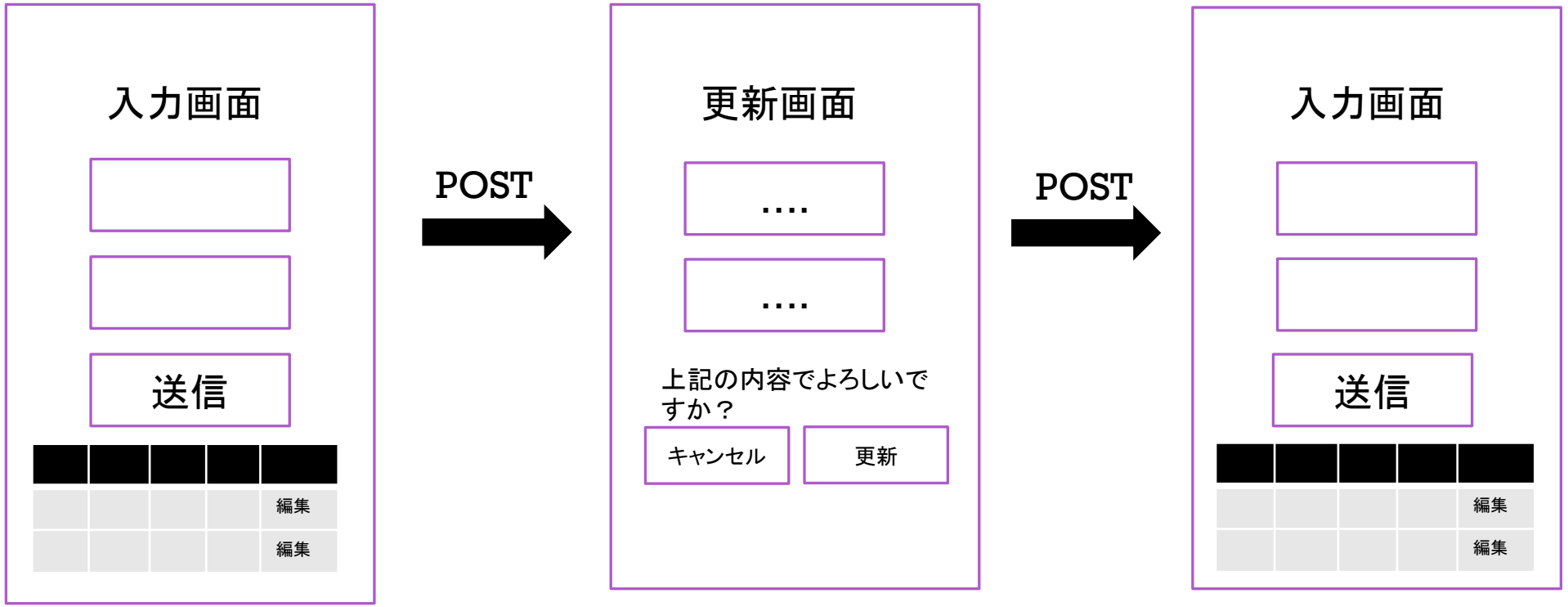
左図のようにお問い合わせ内容の入力画面したに保存したデータをテーブル形式で一覧表示させてください。

お問い合わせ機能（Update)

登録したデータの更新機能を追加します。
要件は以下の通りです。

- テーブルに編集ボタンがあること
- 編集ボタンを押下すると選択したデータの詳細画面が表示されること
- 詳細画面にデータが正しく表示されていること
- データの内容を編集し更新ボタンを押すと入力画面に遷移しデータが更新されていること
- データの登録時と同じバリデーションが適用されていること

お問い合わせ機能(Update)の画面構成



お問い合わせ機能（Delete)

登録したデータの削除機能を追加します。
要件は以下の通りです。

- テーブルに削除ボタンがあること
- 削除ボタンを押下すると削除前の確認のポップアップが表示されること
- OKを押下するとページがリダイレクトされデータがテーブルから削除されていること

お問い合わせ機能(Delete)の画面構成