

第63回 技能五輪全国大会

ウェブデザイン

- Web Technologies -

競技課題 M2

バックエンド

作業時間： 2時間30分

目次

| | |
|--------------------------|---|
| 目次 | 2 |
| はじめに | 3 |
| プロジェクトと課題の説明 | 3 |
| ◎課題全般について | 3 |
| ◎管理画面について | 4 |
| ◎RESTful APIについて | 4 |
| ◎不足項目実装について | 4 |
| 【管理画面の仕様に関する追加情報】 | 5 |
| 【RESTful APIの仕様に関する追加情報】 | 6 |
| 選手への指示 | 7 |
| 評価について | 7 |

はじめに

このモジュールでは、PHP（フレームワーク）とデータベースを使用してバックエンド部分（管理画面とRESTful API）を構築します。

セキュリティへの配慮を行いつつ、各機能が問題なく動作するように構築してください。

プロジェクトと課題の説明

ショッピングモール施設から、館内の案内用画面で表示する写真スライドショーのシステムを依頼されました。

今回の依頼範囲は写真の登録用管理画面とAPIサーバで、表示用UIは不要です。

管理画面では画像URL、その他メタデータの登録を行います。

案内用画面では、RESTful API経由で画像URL、およびメタデータを取得する仕様となります。

下記の要件に合うよう、管理者（システム発注元であるショッピングモール施設）が利用する管理画面と、RESTful APIを構築してください。

◎課題全般について

- データベース（userXX_m2）を使用して、用意されているスキーマを利用できる。
※ XX はゼッケン番号（例：ゼッケン番号「1」の場合、user01_m2）
- 必要なテーブル、及び必要項目は次の通り。「必要項目」とは最低限必要な項目を指し示すため、これ以外の項目が必要と判断される場合は適宜追加すること。
 - 写真の情報を格納する「写真マスタ」（テーブル名：photos）
 - id：写真ID
 - title：写真タイトル
 - place：撮影場所
 - url：画像URL
 - created_at：撮影日時
 - updated_at：更新日時

◎管理画面について

- 管理インターフェースを作成してください。
- 管理インターフェースのデザインは評価の対象外です。
- 管理画面には、次の画面が必要です。
 - メニュー画面
 - 最初に表示される画面です。
 - 「写真一覧」へのリンクが表示されています。
 - 写真一覧 インターフェース
 - データの一覧、各種ボタン（新規登録、編集、削除）を配置してください。
 - 一覧表示画面のヘッダ部分に、新規登録ボタンを配置してください。
 - 一覧表示されている各レコードの箇所に、該当のレコードを操作できる編集ボタンと削除ボタンを配置してください。
- 管理画面のURIは「/admin」とします。
- 仕様に関する追加情報は、下記に別途記載されています。

◎RESTful APIについて

- 案内用画面が、写真IDを指定して画像URLとメタデータを取得できるAPIを構築してください。
- APIのURIは「/api」とします。
- 仕様に関する追加情報は、下記に別途記載されています。

◎不足項目実装について

- 案内用画面の実装チームから、写真に対して下記のメタデータを追加してほしいと連絡がありました。必要なSQLファイル、またはマイグレーションファイルを作成してください。本追加に対するRESTful APIへの追加は不要です。
 - 写真の解像度（width, height）
 - ファイルサイズ
 - ファイルの種類（MIME TYPE）
- 作成したファイルは、提出用ディレクトリ「m2」内にある「SQL」ディレクトリに格納してください。

【管理画面の仕様に関する追加情報】

- **特徴:A1**
 - 写真情報管理
 - 管理者が、写真情報を管理できること。
- **シナリオ:A1a**
 - 写真情報の新規登録
 - 「写真情報新規登録」ボタンを押下することで、入力画面に遷移すること。
 - 必要な項目を入力後、「登録」ボタンを押下することで、写真情報がデータベースに保存されること。
 - 正常に登録された場合、「写真情報が登録されました」というメッセージが表示されること。
 - 必要な項目が不足している場合はエラーとし、「エラーが発生しました」というメッセージが表示されること。
- **シナリオ:A1b**
 - 既存の写真情報一覧（リスト）表示 ~~（リスト）表示~~
 - 写真情報一覧表示に、登録した写真情報（タイトル、撮影場所、撮影日時）、および「編集」ボタン、「削除」ボタンが表示されていること。
- **シナリオ:A1c**
 - 既存の写真情報を編集する
 - 一覧表示に配置した「編集」ボタンを押下することで、編集画面に遷移すること。
 - 編集画面には、選択した写真情報が予め入力された状態になっていること。
 - 情報を編集して「保存」ボタンを押下することで、データベース上の写真情報が更新されること。
 - 正常に更新された場合、「写真情報が更新されました」というメッセージが表示されること。
 - 必要な項目が不足している場合はエラーとし、「エラーが発生しました」というメッセージが表示されること。
- **シナリオ:A1d**
 - 既存の写真情報を削除する
 - 一覧表示に配置した「削除」ボタンを押下することで、「削除してよろしいですか？」というダイアログが表示されること。
 - 「OK」を押下した場合、データベース上の写真情報が削除されること。
 - 削除後は、写真情報一覧表示の画面が更新されること。
 - 「キャンセル」を押下した場合、写真情報一覧表示に留まること。

【RESTful APIの仕様に関する追加情報】

- **特徴:B1**

- 写真一覧情報の読み取り
 - エンドポイント: `/api/photos`
 - メソッド: `GET`
 - アプリケーションから、DBに登録されている写真情報の一覧データ読み取りができること。
 - 写真情報には、下記のデータを含めること。
 - title : 写真タイトル
 - place : 撮影場所
 - HTTPステータスコードは **特徴:B3** を参照すること。エラー発生時のエラーコードは **特徴:B3** を参照すること。

- **特徴:B2**

- 写真メタデータの読み取り
 - エンドポイント: `/api/photos/{id}.json`
 - メソッド: `GET`
 - ルーティングに注意すること。「.json」の文字列は必須とする。
 - アプリケーションから、指定された写真ID (id) に対応する写真情報のJSONデータ読み取りができること。
 - HTTPステータスコードは **特徴:B3** を参照すること。
 - idは必ず指定する。指定がない場合はエラーとし、エラーコードは **特徴:B3** を参照すること。
 - 写真情報には、下記のデータを含めること。
 - title : 写真タイトル
 - place : 撮影場所
 - url : 画像URL
 - created_at : 撮影日時
 - updated_at : 更新日時

- **特徴:B3**

- HTTPステータスコードの返却
 - データ取得成功時には、HTTPステータスコード200を返すこと。
 - エラー発生時には、HTTPステータスコード404を返すこと。

選手への指示

1. 提供されている material フォルダ内のファイルを使うことができます。
また必要に応じて、提供されているフレームワークを使用できます。
2. 「m2」という名前のサーバ上のディレクトリに作業中のアプリを保存してください。
メインファイルがindex.html または index.php という名前であることを確認してください。
3. フレームワークは Laravel、CodeIgniter、Yiiが提供されています。
SFTPでアップする際はパーミッションの設定を行う必要があります。
 - a. Laravelの場合
storage配下をすべて777に設定（例： `sudo chmod -R 777 storage/` ）
 - b. CodeIgniterの場合
writable配下をすべて777に設定（例： `sudo chmod -R 777 writable/` ）
 - c. Yiiの場合
web/assets配下をすべて777に設定（例： `sudo chmod -R 777 web/assets/` ）
4. 時間管理は自身で行ってください。
5. 注意事項：競技サーバにアップロードされたデータのみが採点対象となります。

評価について

| モジュール項目 | 配 点 |
|------------------|-----|
| 管理画面 | 10 |
| RESTful API | 10 |
| 不足項目実装 | 6 |
| ソースコード（セキュリティ含む） | 4 |
| 合 計 | 30 |