

第21回 若年者ものづくり競技大会

ウェブデザイン

- Web Technologies -

フロントエンド

作業時間： 2時間30分

はじめに

このモジュールでは、回転寿司店のタッチパネル注文フロントエンド画面を作成する。

プロジェクトの説明

富山県内で回転寿司店を営む企業から依頼があり、タッチパネル注文フロントエンド画面を構築することになった。

来訪客は、メニューをカテゴリ別に絞り込み、商品の詳細を確認したうえで、数量を指定してカートに追加できるものとする。カートでは、商品の数量変更、削除、合計金額の確認を行えるものとする。

注文内容はAPIを利用して送信し、送信に成功した場合は注文完了画面を表示すること。

このシステムは来訪客向けの画面であるため、店舗側が注文を受信・管理する画面の実装は範囲外とする。

追加課題として、来訪客が自身の注文履歴と注文状況を確認できる画面を実装しても良い。

サイトの横幅は800pxで固定し、画面中央に配置すること。レスポンシブデザインは考慮しなくて良い。

APIについて

このサイトにおいて必要なデータを取得、格納できるAPIが提供される。

取得系APIのデータは全て固定である。

更新系APIについては、実際にはデータベースの更新は行われなため、正しく動作しているかどうかは、HTTPステータスコードによって判断する。

メニューおよびカテゴリはAPIから取得すること。

カート内の商品、数量および合計金額は、フロントエンド側で状態管理すること。

エンドポイント一覧

- メニュー関連
 - メニュー取得・表示API（GET /api/menus）
- カテゴリ関連
 - カテゴリ一覧取得API（GET /api/categories）
- 注文関連
 - 注文送信API（POST /api/orders）
 - 注文履歴取得API（GET /api/orders）

APIの動作については、Bruno、またはPostmanで確認することができる。

機能要件

1. メニュー一覧画面

メニュー取得・表示

- GET /api/menusを利用してメニュー一覧を取得する
- 各商品には以下の情報を表示する
 - 商品名
 - 価格
 - カテゴリ
 - 画像
 - 売り切れ状態
- データ取得中はローディング状態を表示する
- データ取得に失敗した場合はエラーメッセージを表示する
- 売り切れの商品は、利用者が売り切れであることを判別できる表示とする
- 売り切れの商品はカートへ追加できないようにする

カテゴリ絞り込み

- GET /api/categoriesを利用してカテゴリ一覧を取得する
- 選択したカテゴリに属する商品のみを表示する
- すべての商品を表示できる選択肢を用意する
- カテゴリの切り替え後、画面を再読み込みせずに表示内容を更新する

2. 商品詳細表示

商品を選択したとき、商品詳細をモーダルで表示する。

- 以下の情報を表示する
 - 商品名
 - 画像
 - 価格
 - 説明
- 注文数量を選択できるようにする
- 選択した数量の商品をカートへ追加できるようにする
- モーダルを閉じる操作を提供する
- タッチ操作で選択しやすいボタンおよび入力領域とする

3. カート

カートの内容はフロントエンド側で状態管理する。

- 商品をカートへ追加できる
- 商品ごとに数量を変更できる
- 商品をカートから削除できる
- 商品ごとの小計を正しく計算する
- カート全体の合計金額を表示する
- カートが空の場合は、その状態が分かるメッセージを表示する

4. 注文内容確認

注文送信前に、カートに追加された注文内容を確認できる画面または表示領域を提供する。

- 注文する商品名、数量、小計を表示する
- 注文全体の合計金額を表示する
- 注文内容を修正するため、カートへ戻る操作を提供する
- 注文を確定して送信する操作を提供する

5. 注文送信

`POST /api/orders`を利用して注文内容を送信する。送信するデータの形式およびレスポンスの詳細は、別途提供されるAPI仕様書を参照すること。

- カートが空の場合は注文を送信できないようにする
- 注文送信中は送信ボタンを無効化する
- 注文送信中であることが分かる表示を行う
- 二重送信が発生しないようにする
- APIから成功レスポンスが返された場合は注文完了画面へ遷移する

6. 注文完了画面

注文送信に成功した場合は、注文完了画面を表示する。

- 注文が受け付けられたことを明確に表示する
- APIから返されたメッセージと注文番号を表示する
- メニュー一覧画面へ戻る操作を提供する
- 新しい注文を開始するときは、送信済みのカート内容を初期化する

追加課題

以下は上位者向けの追加課題とする。

A. 注文履歴

GET /api/orders?table_no=12を利用して、指定されたテーブルの注文履歴を表示する。

- 注文済みの商品を表示する
- 商品ごとの数量を表示する
- 注文のステータスを表示する
- データ取得中はローディング状態を表示する
- データ取得に失敗した場合はエラーメッセージを表示する

B. 多言語切り替え

日本語と英語の表示切り替え機能を実装する。

- 画面上から言語を切り替えられる
 - 商品とカテゴリの英語表示は不要
- メニュー、ボタン、メッセージなどのUI表示を切り替える
- 画面遷移後も選択した言語を保持する

選手への指示

- 提供される「materials」フォルダ内の「moduleA」フォルダ内のファイルを使うことができる
- パッケージマネージャー（npm）が提供されているが、パッケージのインストールを行うことはできない
- ライブラリを利用する際は、提供されているライブラリを適切なフォルダに配置し読み込みを行うこと
- 競技サーバ内の「/var/www/html/XX_module_a」フォルダ内に「ビルド後」のファイルをアップロードすること
 - 完成したウェブサイトは「http://userXX.skilljapan.info/XX_module_a」で表示確認をすること
 - ※ 「XX」はゼッケン番号
（例：ゼッケン番号「1」の場合、http://user01.skilljapan.info/01_module_a）
- 競技サーバ内の「/var/www/html/XX_module_a」フォルダ内に「_src」フォルダを作成し、ソースコードのデータをアップロードすること
 - ソースコードの採点は「_src」内のファイルを元に実施するため、フォルダ・ファイルが存在しない場合は採点不可となる
 - 「node_modules」など容量の大きい不要なファイルがアップロードされている場合は減点対象となる
- 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るので注意すること