

最終課題紹介資料

2020/3/5
三浦 壺生

概要

テックキャンプの最終課題にて作成したアプリケーションを紹介します。また本資料では、自身で実装した箇所、および開発を通じて得られた経験についても紹介します。

アプリケーション情報

アプリケーション概要

- フリーマーケットのアプリケーションを作成しました。
- 接続先情報
 - <http://13.231.81.144/>
 - ID/Pass
 - ID: admin
 - Pass: 2222
 - テスト用アカウント等
 - 購入者用
 - メールアドレス: buyer123@fukuoka-demo.jp
 - パスワード: buyer123
 - 購入用カード情報
 - 番号: 4242424242424242
 - 期限: 12/20
 - セキュリティコード: 123
 - 出品者用
 - メールアドレス名: seller123@fukuoka-demo.jp
 - パスワード: seller123
- Githubリポジトリ
 - https://github.com/issei1213/freemarket_sample_68a

開発状況

- 開発環境
 - Ruby/Ruby on Rails/MySQL/Github/AWS/Visual Studio Code
- 開発期間と平均作業時間
 - 開発期間: 約3.5週間
 - 1日あたりの平均作業時間: 約10時間
- 開発体制
 - 人数: 3名
 - アジャイル型開発 (スクラム)
 - Trelloによるタスク管理

動作確認方法

- Chromeの最新版を利用してアクセスしてください。
 - ただしデプロイ等で接続できないタイミングもございます。その際は少し時間をおいてから接続してください。
- 接続先およびログイン情報については、上記の通りです。
- 同時に複数の方がログインしている場合に、ログインできない可能性があります。

- 出品方法は以下の手順で確認できます

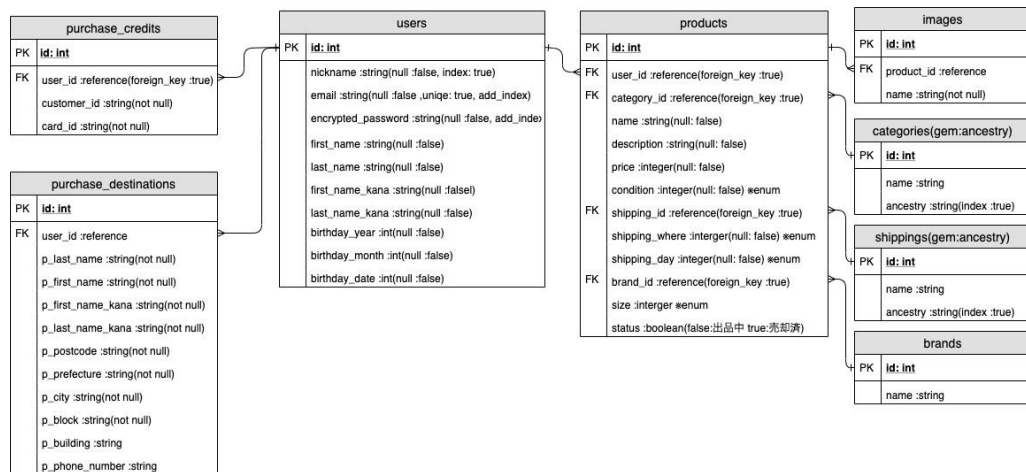
テストアカウントでログイン→トップページから出品ボタン押下→商品情報入力→商品出品

- 確認後、ログアウト処理をお願いします。

開発担当箇所

担当箇所一覧と確認方法

- DB設計



- ユーザー新規登録・ログイン(フロントエンド・サーバサイド)
 - http://13.231.81.144/users/sign_upにアクセスするとご覧いただけます。

会員情報入力

ニックネーム 必須

メールアドレス 必須

パスワード 必須

パスワード(確認用) 必須

本人確認
 安心・安全にご利用いただくために、お客さまの本人情報の登録にご協力ください。他のお客さまに公開されることはありません。

氏名 必須

氏名かな 必須

生年月日 必須

--

年

--

月

--

日

※本人情報は正しく入力してください。会員登録後、修正するにはお時間を頂く場合があります。

「次へ進む」のボタンを押すことにより、[利用契約](#)に同意したものとみなします。

次へ進む

プライバシーポリシー フリマ利用規約 特定商取引に関する表記

© FURIMA

各担当箇所の詳細

- DB設計
 - 概要
 - 必要なテーブル、カラムの選定
 - アソシエーションの決定
 - 担当内容
 - ER図の作成
 - 各テーブルの作成
 - モデルへのアソシエーションの記述
- ユーザー新規登録・ログイン
 - 概要
 - 商品出品・購入等がログインしているユーザのみに限定する為、ユーザ管理を行う機能
 - 担当内容
 - haml,scss,javascript(jQuery)を使用しトップページのマークアップ作業
 - jQueryを使用したフロントのバリデーション実装
 - ウィザード形式でのユーザ登録実装
 - Gem:deviseをカスタマイズし、ウィザード形式で実装
 - ユーザ情報を保存するモデル、コントローラー、ルーティングの作成
 - 「会員情報入力」と「商品送付先住所情報登録」の名前入力時の、自動ふりがな変換機能
- 商品出品・編集・カテゴリ機能
 - 概要
 - ユーザーが商品を出品する機能
 - 出品した商品を編集する機能
 - 担当内容
 - 出品情報を保存するモデル、コントローラー、ルーティングの作成
 - 画像投稿時の画像プレビュー表示
 - Gem:ancestryを使用し、カテゴリ・配送方法を多階層カテゴリで実装。
 - カテゴリ・配送方法はAjaxで実装
 - セレクトボックスの内容はenumを使用し実装
 - jQueryを使用したフロントのバリデーション実装

開発を通じて得られた知見

工夫した点

①チームとして工夫を行った点

メンバーが作業実施する前に、一度実装するフロント画面やサーバサイドで実装する機能を確認し、コントローラ名・アクション名を決め、共通認識を持った状態で実装することを心がけました。また、メンバーが困っていた場合、チーム内で教えあうことを意識しお互いに知識を深めていくことを徹底しました。

②個人として工夫を行った点

機能を実装する時にはデータの流れを特に意識していました。具体的にはデバッグツール (binding.pry(byebug)/better_error/binding_of_caller/debugger/console.log) をこまめに使用し、変数の状態や、メソッドの使い方を理解しながら実装していきました。また、チームメンバーが実装した機能でも、デバッグツールを使用することを忘れずに、実装方法理解しました。

苦勞した点

①DB設計

DB設計はチーム開発中で最も苦勞した点になります。私がメインになりDB設計を行いました。特にカテゴリテーブル構造がとても複雑でした。当初、カテゴリテーブルを実装する際は、ライブラリを使用せずに試みましたが、テーブル数が多くなり、データ保存を考慮した時に複雑なプログラミングを組む必要が出てきました。チーム内で協議した結果、ライブラリを使用することになりました。DB設計後は、サーバサイドの機能を実装しましたが、DB設計次第で記述量が変化することも理解できました。非常に時間がかかり、何度も修正をしましたがDB設計の重要性を理解できた経験になりました。

②ユーザ管理のカスタマイズ

今まで実装したユーザ管理では行わなかったカスタマイズを利用して、実装を行いました。今回実装したユーザ管理は、様々なサービスで使用されているウィザード形式で実装しました。最初は、アクションやページをまたぐデータの流れをイメージすることが苦勞しました。実装できた後は、様々な場面で活用できるなと思いました。また、ウィザード形式に実装方法には、テーブルで管理する方法もあったので挑戦する予定です。

③jQueryを用いたバリデーション

ユーザ情報登録・商品出品のバリデーションはサーバサイドだけではなく、ユーザー目線になった時も考慮し、jQueryを用いたバリデーションを行いました。特に苦勞した点はセレクトボックスを用いた時のバリデーションです。セレクトボックスのオプションをAjaxで実装しており、またバリデーションルールを自分で作成しないといけないこともあり、実装には苦勞しました。実装できた後は、ユーザー目線を意識できて実装できたなと感じました。バリデーションにはAjaxを用いた実装方法もあったので、挑戦する予定です。

以上