

最終課題紹介資料

2019/5/31
氏名：山本一貴

概要

TECH::EXPERT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

- フリーマーケットアプリケーションである「メルカリ」のコピーサイトを作成しました
- 接続先情報
 - URL <http://13.112.45.101/>
 - ID/Pass
 - ID: mercari
 - Pass: teama
 - テスト用アカウント等
 - 購入者用
 - メールアドレス: buyer@test.com
 - パスワード: buybuy
 - 購入用カード情報
 - 番号：4242424242424242
 - 期限：01月23年
 - ユーザー名：山田太郎
 - セキュリティコード：1111
 - 出品者用
 - メールアドレス名: seller@test.com
 - パスワード: sellsell
- Githubリポジトリ
https://github.com/yamamotokazutaka/freemarket_sample_50a1

開発状況

- 開発環境
 - ruby/Ruby on Rails/MySQL/Github/AWS/sublimeText3
- 開発チーム
- 開発期間と平均作業時間
 - 開発期間：4週間
 - 1日あたりの平均作業時間：7時間
- 開発体制
 - 人数：5人
 - 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。
- 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

マークアップ

- 商品詳細ページ
- マイページ
- 商品検索ページ

サーバーサイド

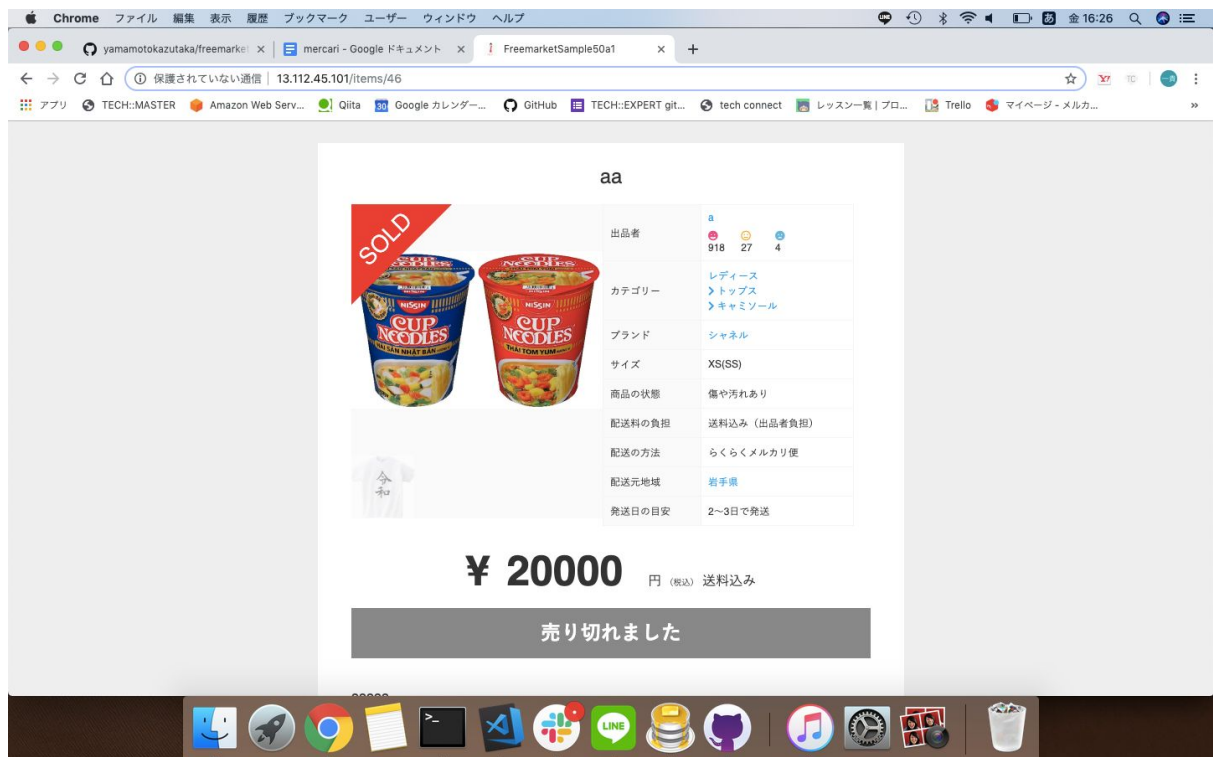
- デプロイ（上記IPアドレスにアクセス）
- 商品詳細表示（商品詳細画面に移った後に各情報が決め打ちではなく商品毎のデータに変わっていることを確認）
- 商品検索①（トップページの検索窓からキーワードを入れて検索→キーワードが名前に含まれている商品が表示されていることを確認）
- 商品検索②（商品検索①で表示されているviewのサイドバーから詳細検索を実施）

各担当箇所の詳細

マークアップ

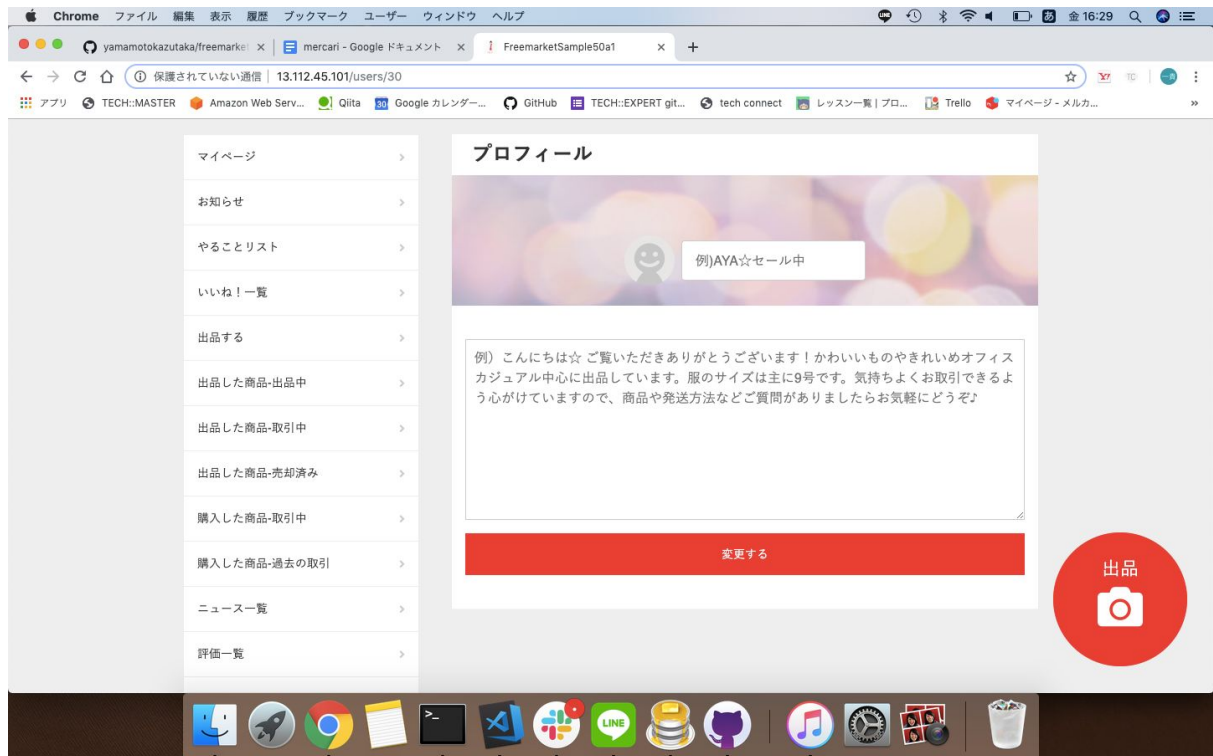
- 商品詳細ページ

<http://13.112.45.101/>にアクセスしたのち任意の商品画像をクリックしていただくとアクセスできます。



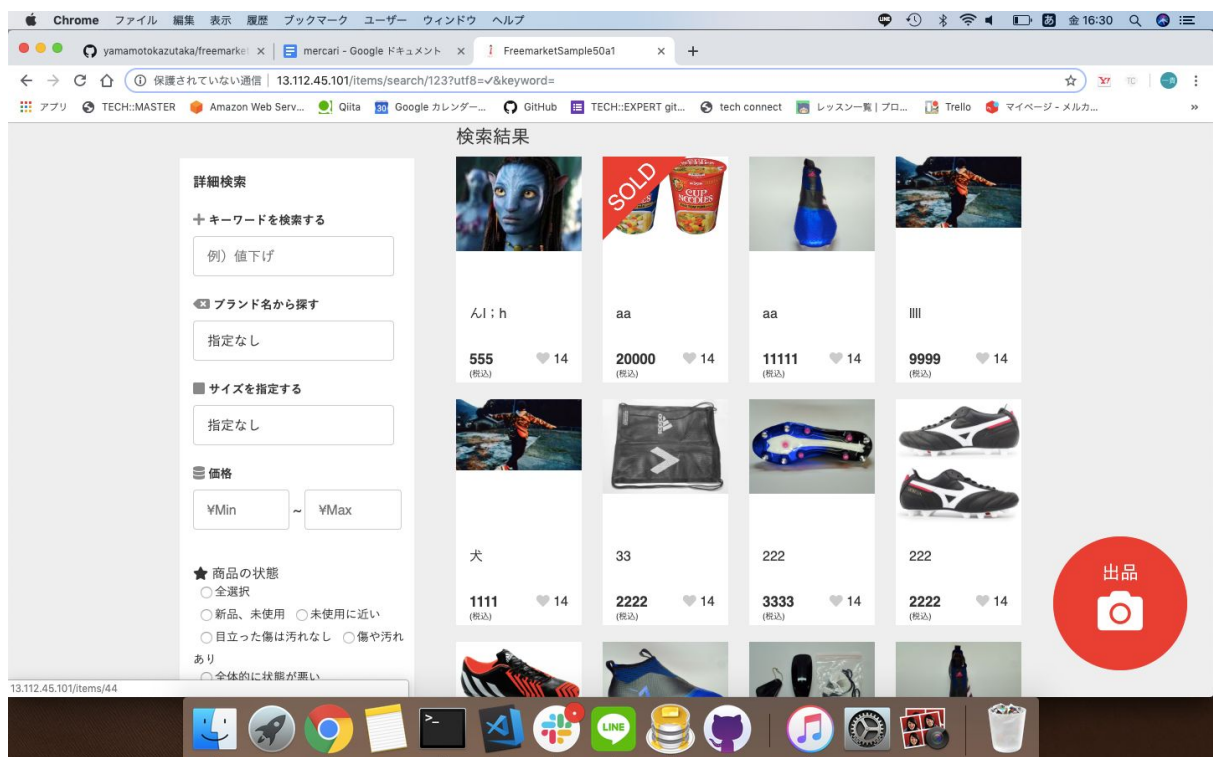
- マイページ

<http://13.112.45.101/>にアクセスしたのちログイン（メールアドレス: buyer@test.com、パスワード: buybuy）していただくとマイページボタンから入ることができます。



- 商品検索ページ

URL <http://13.112.45.101/>にアクセスした後ヘッダー部分の検索窓から任意のキーワードを入力していただくとアクセスできます。



サーバサイド

- デプロイ
 - 概要

AWSを用いてIPアドレスを設定
 - 内容

capistranoを使ってデプロイ作業を自動化
メルカリ内にて投稿された画像をAWSのS3に保存できる様に設定
BASIC認証をかけてセキュリティ性を向上
- 商品詳細表示
 - 概要

商品ページにアクセスした後商品の個別の情報を表示
 - 内容

DBから任意の商品の情報を取り出し表示
- 商品検索①
 - 概要

商品トップページから商品のタイトルを検索できる様に設定
 - 内容

whereメソッドとLIKE句を用いてあいまい検索を実装
Gem kaminariを用いて商品検索後のページにpaginationも実装
- 商品検索②
 - 概要

Gem ransackを用いて複数検索を実装
 - 内容

商品検索①を行なった後・ブランド・サイズ・価格・商品状態・配送料の負担・販売状況で絞り込み検索ができる様に実装

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

- ・開発ブランチを作ってチームで挙動を確認してからmasterにデプロイを行う様にした。
- ・本番環境へのデプロイを細かく（2～3日毎に）行い動作確認を行なった
- ・マークアップを10日で終わらせてサーバーサイドに時間を割く様にした。
- ・マークアップを担当した人と同じ人がその箇所のサーバーサイドの開発も担当することでスムーズに作業ができる様にした。
- ・

②個人として工夫を行った点

- ・本番環境への自動デプロイを細かく（2～3日毎に）行い動作が正しく動くかどうかを確認した。

苦労した点

①デプロイ時

cssが当たらずhtmlしか表示されないという問題が発生しました。それまでの記述を何度も確認したが間違いが見当たらなかったためaws-ec2のインスタンス、unicorn、mysqlを全て再起動することでアセットファイルをコンパイルすることができcssが当たりました。デプロイはエラーが見えないことが多いからこそエラー時に共通して確かめること（インスタンス再起動,unicoenをkill,secrets.ymlが存在しているか等）を控えておくことが重要だと感じました。

②商品検索機能実装時

gem ransackを用いて開発することは分かっていたが知識が曖昧でメルカリに組み込むことができませんでした。そこでQiita記事の [\[Rails\]ransackを利用した色々な検索フォーム作成方法まとめ](#)に載っているミニアプリを自分でも作ってransackの使い方を学んでからメルカリに応用しました。アプリを作るので時間はかかりましたがgemに対する理解が深まりスムーズに実装することができました。ただ答えとなる記事を見つけにいくのではなくどうすればその事柄について理解できるのか自分が実装したいことに対して落とし込むことができるのかという視点を持つことが重要だと思いました。

③GitHub

チームの方向性としてmasterを守るために開発ブランチを作って作業をしていくことが決まりましたが開発当初はGitHubに対する理解が浅く全員が開発ブランチで作業を行なっていました。そのため各々がコードレビューのためのプルリクエストを出すことができないという状況が生じてしまいました。開発ブランチから各々の作業ブランチを切ること自分たちが行いたかった作業をすることができました。GitHubは個人作業ではなかなか理解ができないのでとても良い学びになりました。

以上