

today's-leon 仕様書

目次

1	アプリの概要・画面の仕様.....	3
2	作成理由	7
3	使用した技術・その技術を選んだ理由	8
4	制作する際にこだわった点.....	9
5	制作する際に苦労した点.....	10
6	この先取り組みたいこと.....	11

1 アプリの概要・画面の仕様

複数の人が Twitter のようにコメントできる Web アプリです。家族内でレオンに関することが自由にコメントすることができます。

Web アプリ「今日のレオン」の URL を以下に示します。

<https://todays-leon.herokuapp.com>

Github にコードを公開したページの URL を以下に示します。

<https://github.com/rikoniko/todays-leon>

以下に詳しい画面の仕様を示します。

＜トップ画面の仕様＞

ブラウザで URL を検索すると図 1 のようなトップ画面が表示されます。また、トップ画面の要素を表 1 に示します。

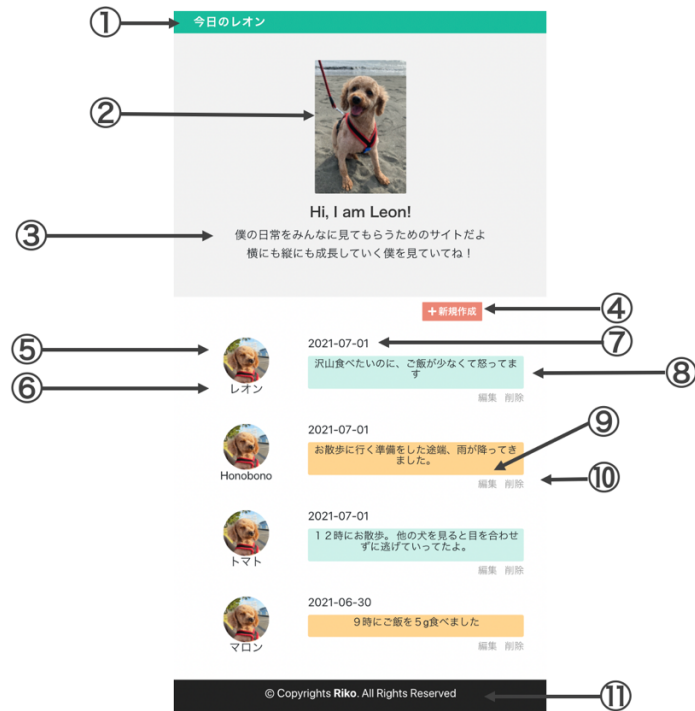


図 1 トップ画面

表 1 トップ画面の要素

番号	名称	イベント	コンポーネント	機能
1	サイト名	タップ	ボタン	トップ画面に遷移
2	レオン写真			レオンが散歩している写真
3	サイト説明			サイトの簡単な説明
4	新規作成ボタン	タップ	ボタン	新規作成画面に遷移
5	アイコン			レオンの写真がアイコンとして表示
6	ユーザー名			コメントに紐付いたユーザー名が表示される
7	日付			コメントに紐付いた日付が表示される
8	コメント			ユーザー名に紐付いたコメントが表示される
9	編集ボタン	タップ	ボタン	編集画面に遷移
10	削除ボタン	タップ	ボタン	コメントを削除する
11	コピーライティング			コピーライティングを表示

＜新規作成画面の仕様＞

図 1 の④新規作成ボタンを押すと新規作成画面に遷移し、図 2 のような画面が表示されます。また、新規作成画面の要素を表 2 に示します。

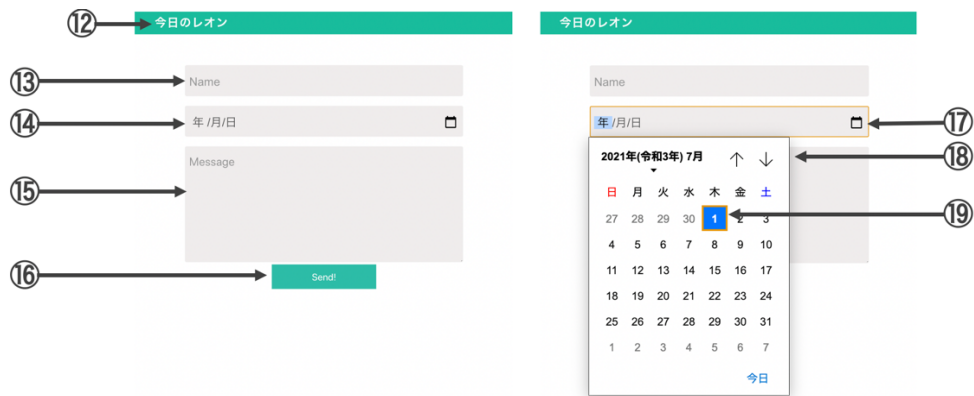


図 2 新規作成画面

表 2 新規作成画面の要素

番号	名称	イベント	コンポーネント	機能
12	サイト名	タップ	ボタン	トップ画面に遷移
13	ユーザー名	入力	テキストボックス	ユーザー名を入力
14	日付	入力	テキストボックス	日付を入力 'YYYY-MM-DD'のフォーマットで'年-月-日'の順に入力する
15	コメント	入力	テキストボックス	コメントを入力 改行は反映されない
16	送信ボタン	タップ	ボタン	コメントを送信される トップ画面へ遷移
17	カレンダーボタン	タップ	ボタン	カレンダーが表示
18	カレンダーめくる ボタン	タップ	ボタン	'↑'上矢印で前の月に戻る '↓'下矢印で次の月に進む
19	日付ボタン	タップ	ボタン	日にちを選択

＜編集画面の仕様＞

図 1 の⑨編集ボタンを押すと編集画面に遷移し、図 3 のような画面が表示されます。また、編集画面の要素を表 3 に示します。

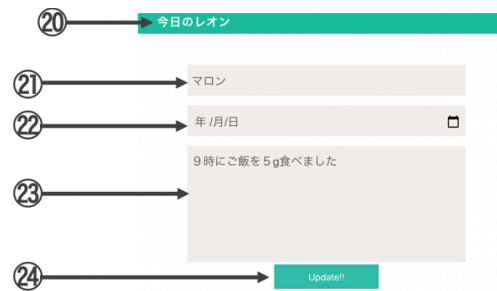


図 3 編集画面

表 3 編集画面の要素

番号	名称	イベント	コンポーネント	機能
20	サイト名	タップ	ボタン	トップ画面に遷移
21	ユーザー名	入力	テキストボックス	ユーザー名を入力 編集するコメントに紐付いたユーザー名が表示される
22	日付	入力	テキストボックス	日付を入力 'YYYY-MM-DD'のフォーマットで'年-月-日'の順に入力する 編集するコメントに紐付いた日付が表示される
23	コメント	入力	テキストボックス	コメントを入力 改行は反映されない 編集するコメントが表示される
24	更新ボタン	タップ	ボタン	コメントを更新される トップ画面へ遷移

2 作成理由

私はレオンという名前の犬を飼っています。犬は 1 日にご飯の量が決まっています。そのご飯の量は私たちが管理しなければなりません。しかし、ご飯をどのくらい与えたかを忘れてしまうことがありました。それを解決するために、いつ誰がご飯をどのくらい与えたかを記録できるようなアプリを作成しました。

3 使用した技術・その技術を選んだ理由

<フロントエンド言語>

- HTML/CSS
- JavaScript

<フレームワーク>

- Bootstrap

今回、初めて Bootstrap を使用しました。Bootstrap の使い方を学ぶために自分でコードを書くのではなく、Bootstrap を使用しました。HTML と CSS のコードの書き方は一通り学びました。

- Express(Node.js のフレームワーク)

HTTP メソッドやパスに応じて処理を記述するときに、Express を用いることでより簡単に記述できるからです。

<サーバーサイド>

- Node.js

サーバーサイドでも JavaScript での実装ができるからです。

<サーバーサイド>

- Heroku

Github で Web アプリを公開しようとしたのですが、Github はサーバーサイド言語をサポートしていないことがわかりました。なので、MySQL が使える Heroku を使用して Web アプリを公開しました。

- EJS

HTML 内に JavaScript を埋め込むような記述をすることができるからです。

- Visual Studio Code

- npm

<ツール>

- Github

コードを公開するために使用しました。

4 制作する際にこだわった点

- コメントの背景の色を交互に表示するようにしました。これは、コメントに番号をつけ、その番号が偶数と奇数かを判定し、それぞれの場合で設定を変えることで実現しました。
- ユーザーが使用するデバイスによって、画面サイズやフォントサイズを切り替えられるようにしました。Chrome のデベロッパーツールで異なるデバイスの画面の表示を確認しながら HTML/CSS を編集しました。
- 新しいコメントが上に表示され、下へ行くほど古いコメントになるようにしました。これは、`reverse()` を使用して配列を逆順することで実現しました。

5 制作する際に苦労した点

- Heroku に Web アプリをデプロイし、【heroku open】 コマンドを使ってブラウザのタブで開くと、エラーメッセージが出てしまいました。
Heroku の公式サイトを参考にして解決することができました。具体的には、アプリケーションのルートディレクトリにあるテキストファイル Procfile 内の定義を【web: npm start】 から【web: node app.js】 と変更しました。
- Node.js からデータベースに対してクエリを実行するために、【connection.query('クエリ',渡したい配列)】 と書きました。データベースでテーブルにデータを挿入するとき、VALUES に「?」を含めて、フォームからの複数の配列をクエリに渡すときに、エラーが出てしまいました。
Qiita を参考にして解決することができました。具体的には、フォームからの配列をクエリに渡す前に、変数として受け取り、【VALUES(" + 変数 + ", " + 変数 + ")】 として書くことでクエリに渡すことができました。
- Heroku で MySQL を構築して Node.js で接続後、無事にブラウザで Web アプリを見ることができましたが、しばらくするとエラーメッセージが出て見るができなくなりました。
これは、MySQL のセキュリティの仕様で定期的にデータベースとの接続が切れることが原因でした。
接続が切れたときに再接続するように設定することで解決することができました。具体的には、エラーメッセージが接続が切れたときのエラーメッセージ である「`PROTOCOL_CONNECTION_LOST`」となったときに、再接続するようにしました。
- ブランチ名を「master」として作成したのに、「main」として変更内容をリモートリポジトリに送信して、エラーが出てしまいました。
ブランチの意味を理解することが解決することができました。書かれたコードを打つだけでなく、そのコードを書くことで何が起こるかを意識することが大事だと学びました。

6 この先取り組みたいこと

- 現在はコメントが増えてしまったときに、下へスライドさせてコメントを探さなければなりません。それを解決するために、日付ごとに分けてコメントを分類する機能や「More」ボタンを作って、新規のコメント 5 件だけ表示するようにしたいです。
- 画像をアップロードできるようにしたいです。それが実現できたら、コメントに画像を添付でき、アイコンの画像を変更できるようになると思います。
- 自分が作成時に苦労したところやどのように解決したかを備忘録として記録したいです。

以上