

ポートフォリオ

作成者 扇田 修至

制作にあたって

- ・制作のきっかけ

これまでに自分が学習してきたこと

(Laravel PHP, データベースの基本操作など) を用いることで

制作が可能なアプリケーションを考え、制作を決めました

フロントエンドとバックエンドに処理を分散し、

機能を実装してみたかったので React と Laravel の 2 つを用いたアプリ制作

制作物の仕様について

ユーザーによって作成されたタスク(Todoリスト,やること予定)の新規作成, 管理, 閲覧を行うことができるアプリケーション

動作は以下の操作を行うことができる。

- ・タスクの新規作成 (件名, 説明, 優先度, ジャンル, 期限)
- ・データベース上に登録されているタスクをリスト化し、一覧表示する
(その表示機能に加え 編集機能(内容変更と更新, タスクの削除)を備える)
- ・利用するユーザの登録, 編集機能(登録ユーザデータ更新, ユーザの削除)

開発環境などについて

●React (JavaScript フレームワーク)・・・ フロントエンド (ユーザが直接触れる部分)

[使用プラグイン・ライブラリなど]

- ・axios API (フロント \longleftrightarrow バックエンド間の通信, データベースとやり取りするときに使う)
- ・react-router-dom (単一のappページから各機能(component)へのリンクのルート設定)

●Laravel (PHPフレームワーク)・・・ バックエンド (裏側で動作している部分)

(このアプリでは 主にデータベースの処理命令を行う)

- ・sanctum (ログイン機能の実装に必要 認証など)

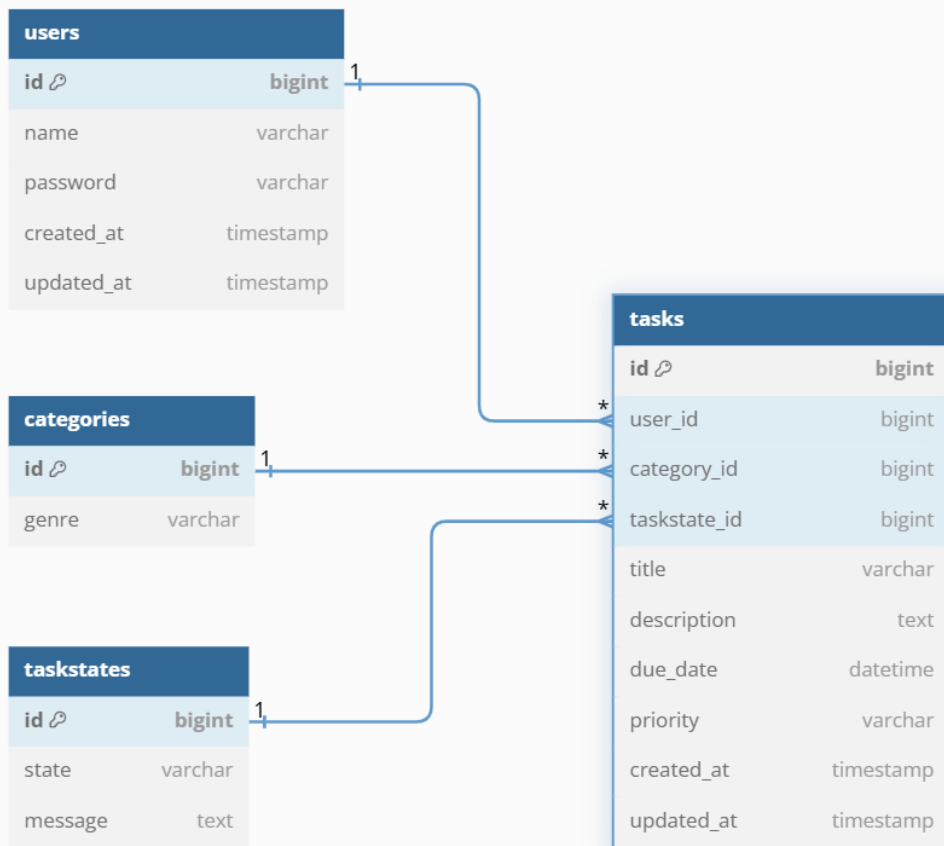
●phpMyAdmin (データベース)

[その他] TailWind CSS [画面装飾 CSS フレームワーク]

[使用ソフト] Visual Studio Code

[開発期間] 2024. 11月 - 2025. 1月 (約3か月)

データベース設計 (ER-図)



← テーブル

- ・users (ユーザの登録データ)
- ・tasks (作成したタスクのデータ)
- ・categories (カテゴリ, ジャンル分類)
- ・taskstates (タスクの状態)

リレーションにより関連する情報を tasks テーブルから別テーブルの情報を引用できるよう 1 対多の関係を設定

動作ページ①(トップページ画面, タスク新規作成)



← アプリ top ページ部分
上のメニューバーは共通部分で
各機能にリンクして移動できる



← タスクの新規作成機能
選択をクリックすると、selectboxで
項目が開き、選択することができる

各項目に入力後 作成ボタンを
クリックすることで、
データベース上にデータが
送信され 予定が作成される。

動作ページ② (ユーザの登録画面)

Vite + React

localhost:5173/usercreate

TOP 入力フォーム タスク一覧 ID個別表示 ログイン ユーザ登録 ユーザー一覧

現在のURLは /usercreate

新規ユーザー登録

ユーザー名
(UserName) 登録するユーザー名を入力

メールアドレス
(Email) 登録するメールアドレスを入力

登録パスワード
(Password) 6文字以上のパスワードを入力

※再度確認 パスワードを再度入力

登録する

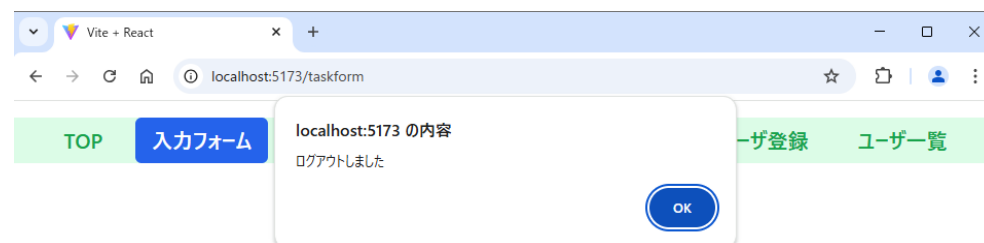
※パスワードは6文字以上の任意の文字列
メールアドレスは ~ @ ~ の形式のみ登録可

登録するパスワードは確認のため二回入力する仕様にして
最初の入力と確認用の入力内容が一致するか確認している
(一致していなければユーザは登録できない)

動作ページ③ (各種機能 リスト表示 指定ID表示, 編集, ユーザー一覧)



動作ページ④ (ログイン・ログアウトについて)



ログイン機能については Sanctum を導入し使用する。
ログインを行う際に /login に axios.post によって
ログインに必要な入力データ (mail, password) をフロント側から
API 経由でバックエンド側に送り バックエンド側で正常に処理されたらログインに成功する。
ログインに成功した場合にのみ トークンを発行し、ローカルストレージ内に保存、
そのトークンによってログイン中のユーザーを認証してユーザーを判別する。
ログアウトをした際にローカルストレージ内のトークンは消去される。

制作で工夫した点

- ・各ページ内で動作する機能についての説明をページ内に記載,
各種ボタンの動作に対し ホバーアニメーション(マウスカーソルを近づいた時に動作)を付与
使用するユーザに対し、見やすく 使いやすくなるようにユーザインタフェースに
こだわりました。

今後の改善点・課題点

- ・現状のコードだと自分のユーザID以外のタスクを閲覧できてしまう,
ログインしていなくても詳細まで閲覧できてしまうなどの問題があるため
条件(管理の権限の有無)に応じて 閲覧が可能であるという内容に書き直す必要性がある。
- ・機能の拡張(タスクに対して特定の色を指定し ラベルによる分類ができるなど)