

# 課題:ECサイト

---

## ファイル構成

- root
  - app
    - static
    - src
      - model
      - Account.py
      - CartManager.py
      - Login.py
      - Shop.py
    - templates
    - main.py
  - instance
    - DB.db
  - run.py

※必要なファイル、フォルダがある場合は追加してください。

## ファイルの説明

※必要であれば、各ファイルを追加してください。

- main.py
  - Shopの商品一覧画面へ遷移
- Account.py
  - アカウントの新規作成
  - アカウントの設定変更
  - 購入履歴
- Login.py
  - ログイン
  - ログアウト
- Shop.py
  - 商品一覧
  - カートへの商品追加
- CartManager.py
  - カート内参照
    - 商品一覧
    - 合計金額
  - カートの商品削除
  - 購入手続き

## DB定義[論理]

- テーブル名 : ユーザー

- 概要：
  - ログイン情報を管理
- カラム：
  - ID：数字で連番[テーブル：個人情報と連動]
  - ユーザー名:ログインする時に利用
  - パスワード:ログインする時に利用[ハッシュ化]
  - 作成時間：レコードの作成時間
  - 更新時間：レコードの最終更新時間
  - 削除フラグ：論理削除用
- テーブル名：個人情報
  - 概要：
    - 個人情報を管理
  - カラム：
    - ID：数字で連番[テーブル：ユーザーと連動]
    - 名字：
    - 名前：
    - 住所：商品の届け先
    - 作成時間：レコードの作成時間
    - 更新時間：レコードの最終更新時間
    - 削除フラグ：論理削除用
- テーブル名：商品
  - 概要：
    - 商品を管理
  - カラム：
    - ID：数字で連番
    - 商品名：
    - 商品画像：
    - 値段(税込み)：
    - 削除フラグ：論理削除用
- テーブル名：カート
  - 概要：
    - カートを管理
  - カラム：
    - ID：数字で連番
    - カートID：カートIDでカートを管理
    - 商品ID：[テーブル：商品と連動]
    - 作成時間：レコードの作成時間
    - 更新時間：レコードの最終更新時間
    - 削除フラグ：論理削除用
- テーブル名：取引状況
  - 概要：
    - 各カートの取引を管理
  - カラム：
    - ID：数字で連番
    - カートID：[テーブル：カートのカートIDと連動]
    - 届け先：アカウントの住所が入力される

- 状況：【保留】【購入】【キャンセル】
- 作成時間：レコードの作成時間
- 更新時間：レコードの最終更新時間
- 削除フラグ：論理削除用

## 追加課題としては

- 購入が重複した場合やカートに入れるときに個数の選択が可能
- 個人情報テーブルに届け先を3つ追加
  - 購入手続き時に届け先の選択が可能