پروژه شاپینگ لیست (Online shop) ورژن شی گرا

توسعه دهنده : بابک زارع

در این مستند راجع به کل پروژه یعنی کلاس ها، متد ها، فانکشن ها و Flow برنامه توضیحاتی داده شده برای استفاده ی راحت تر توسعه دهندگان. هنوز بخش هایی از پروژه مانند متد هایی که ادمین میتواند استفاده کند باقی مانده اند. اما پروژه به درستی کار میکند.

*این مستند و همینطور پروژه هنوزبه طور کامل تکمیل نشده است و کلاس Product و یک سری فانکشن ها که در فولدر helpers قرار دارند باید توضیح داده شوند. ممنون که صبر میکنی 😊*

هدف اصلی پروژه و توضیحات کلی:

این پروژه در واقع یک شبه فروشگاه آنلاین میباشد که کاربر در آن میتواند اکانت خود را بسازد، لاگین کند و وارد صفحه اصلی فروشگاه شود. در صفحه اصلی فروشگاه شما چندین دستور دارید که میتوانید از آنها استفاده کنید. یکسری دستورات هم وجود دارند که فقط توسط یوزر هایی با رنک ادمین میشود از آنها استفاده کرد. **این دستورات شامل:**

Add 🡪 'additem' , 'add'

Charge 🡪 'charge' , 'addmoney'

Balance 🡪 'balance', 'wallet', 'mymoney', 'mywallet'

Remove 🡪 'remove' , 'delete', 'rm' , 'del'

Change Password 🡪 'changepassword', 'changepass'

Category 🡪 'category' , 'categories'

Help 🡪 'helpme' , 'help', 'h'

Prods 🡪 'prods', 'products', 'product

My Discount 🡪 'mydiscount' , 'mycoupon, 'discounts', 'coupons'

Clear 🡪 'clear'

Buy 🡪 'buy'

Exit 🡪 'ex', 'exit', 'q', 'quit'

Logout 🡪 'logout'

Show 🡪 'Show'

دستورات رنک Admin بدلیل آنکه هنوز در دست توسعه میباشد اینجا تعریف نشده.

به ترتیب از بالا به پایین عمل دستورات را توضیح میدهیم:

Add : وارد بخش اضافه کردن محصول به سبد خرید میشود. از کاربر در ورودی به ترتیب، دسته بندی یا کتگوری محصول، نام محصول، تعدادی که از محصول کاربر میخواهد را میگیرد و درصورت موجود بودن آنرا به سبد خرید کاربر که cart نام دارد (یک دیکشنری است) اضافه میکند.

Charge : این دستور برای شارژ کیف پول کاربر میباشد. کاربر در ورودی مبلغی را وارد میکند و حساب اش شارژ میشود.

Balance : مقدار موجودی کیف پول کاربر را به او نمایش میدهد.

Remove : این دستور برای حذف یک محصول از سبد خرید است. از کاربر در ورودی به ترتیب: دسته بندی محصول، نام محصول و مقداری که میخواهد حذف کند را میگیرد و در صورت وجود محصول در سبد خرید مقدار داده شده را از سبد کم میکند.

Change Password : این دستور برای تغییر رمز عبور کاربر است و کاربر میتواند با استفاده از آن رمز عبورش را تغییر دهد.

Category : دسته بندی محصولات را به کاربر نشان میدهد ( به صورت کلی ) کاربر محصولات را نمیبیند ولی تعداد محصولات موجود در دسته بندی را میتواند ببیند.

Help : به کاربر دستورات موجود در برنامه را نشان میدهد.

Prods : به کاربر محصولات موجود در فروشگاه را نشان میدهد ( و دسته بندی آنها )

My discount : کد های تخفیف کاربر را به او نمایش میدهد ( **در حال توسعه** )

Clear : ترمینال را پاک میکند

Buy: کاربر با وارد کردن این دستور وارد مرحله نهایی کردن خرید خود میشود. در این مرحله ابتدا از کاربر پرسیده میشود که آیا میخواهی از کد تخفیف استفاده کنی؟ و کاربر با بله یا خیر میتواند به این ورودی پاسخ بدهد. درصورت ورود کد تخفیف مبلغ تخفیف به فاکتور اعمال میشود و از قیمت فاکتور کم میشود. در نهایت محصولات به کاربر نمایش داده میشوند و از کاربر برای نهایی کردن خرید خود یک تاییدیه میگیرد که درصورت وارد کردن 'finish' خرید نهایی میشود.

Exit : دستوری برای خروج از برنامه میباشد.

Logout : این دستور برای Logout کردن از برنامه میباشد. و کاربر وارد مرحله اول برنامه یعنی Create an account یا

Login میشود

Show : سبد خرید را به کاربر نمایش میدهد. درصورتی که مقدار پول کاربر کمتر از مبلغ فاکتور بود، مبلغ فاکتور به رنگ قرمز نمایش داده میشود

**کلاس های برنامه:**

* **User ( در فولدر auth تعریف شده است )**
* **Product ( در فولدر product تعریف شده است )**

کلاس User :

این کلاس همانطور که از اسم اش پیداست برای user است. به این صورت که در flow برنامه بعد از لاگین کردن کاربر و گذراندن مراحل احراز هویت، یک instance از این کلاس ساخته میشود و در متغیری به اسم current\_user تعریف میشود.

***Instance arguments:*** self, username, password, balance

Username : یوزر نیم کاربر

Password : پسورد کاربر

Balance : مقدار پول کاربر

***Instance methods:***

***این بخش صرفا یک توضیح کلی راجع به متد ها و پارامتر های آنها است. در ویدیو نحوه انجام کار متد توضیح داده میشود.***

**Add\_purchase\_history()** : یک پارامتر به اسم cart که یک دیکشنری است میگیرد. این متد درواقع سبد خرید کاربر را میگیرد و آنرا به users.json اضافه میکند. این کار برای این است که بعد از هربار خرید سبد خرید قبلی به دیتابیس اضافه شود و کاربر بتواند در آینده خرید های قبلی خودش را ببیند.

**Use\_discount()** : یک پارامتر به اسم discount میگیرد که یک استرینگ است. این متد کد تخفیف داده شده را از دیتابیس users.json حذف میکند. فرآیند چک کردن کد تخفیف که آیا در لیست کد های تخفیف کاربر وجود دارد یا نه در flow برنامه انجام میشود.

**get\_percent\_of\_discount()** : یک پارامتر به اسم discount میگیرد که یک استرینگ است. این متد درصد تخفیف کد تخفیف داده شده را از دیتابیس users.json میخواند و یک integer که همان درصد تخفیف است را برمیگرداند. فرآیند چک کردن کد تخفیف که آیا در لیست کد های تخفیف کاربر وجود دارد یا نه در flow برنامه انجام میشود.

**Check\_discount()** : یک پارامتر به اسم discount میگیرد که یک استرینگ است. این متد کد تخفیف داده شده را از دیتابیس چک میکند و اگر در دیتابیس users.json وجود داشت True برمیگرداند. در غیر این صورت False برمیگرداند.

**Add\_discount()** : دو پارامتر discount که همان کد تخفیف است، و پارامتر percent که همان درصد تخفیف میباشد را میگیرد. این متد با استفاده از پارامتر ها یک key value به users.json اضافه میکند. درواقع کد تخفیفی جدید به کد های تخفیف instance اضافه میشود.

**Get\_role()** : رول یا همان رنک instance را از users.json میخواند و یک استرینگ برمیگرداند.

**Set\_role()** : یک پارامتر role را میگیرد که استرینگ است. توجه شود که این استرینگ یا باید 'admin' باشد یا باید 'user' . درغیراین صورت ارور میدهد. این متد رنک جدید را به instance اعمال میکند.

**check\_password()** : پارامتری به نام value میگیرد که یک استرینگ است. value را میگیرد و با پسورد instance آن را بررسی میکند. درصورتی که پسورد داده شده با پسورد instance یکی نبود. یک ValueError raise میکند.

**Add\_balance()** : پارامتری به نام value میگیرد که یک اینتیجر است. این متد اینتیجری که میگیرد را به عنوان مبلغ شارژ پول به حساب instance اضافه میکند. درواقع به users.json ارجاع میکند و value را به balance اینستنس اضاف میکند.

**Decrease\_balance()** : پارامتری به نام value میگیرد که یک اینتیجر است. این متد در ابتدا چک میکند که آیا مبلغ داده شده از مقداری که instance دارد بیشتر است یا نه. درصورتی که بیشتر بود این مقدار داده شده را از users.json کم میکند و در انتها یک True برمیگرداند. درغیراینصورت False برمیگرداند.

**Get\_balance()** : موجودی حساب instance را برمیگرداند.

**Change\_password()** : یک پارامتر میگیرد که همان پسورد است و یک int است. این متد چک میکند که آیا پسورد وارد شده با پسورد instance یکی میباشد یا خیر. درصورتی که باهم برابر نبودند، فرآیند تغییر پسورد instance را از users.json انجام میدهد و در انتها یک True برمیگرداند. درغیراین صورت False برمیگرداند.

**کلاس متد ها Class methods :**

**Login()** : این کلاس متد username و password را به عنوان پارامتر میگیرد. نقش این متد احراز هویت میباشد. این متد با صدا زدن استاتیک متد does\_username\_exist() (که در ادامه توضیح داده میشود) یوزرنیم را چک میکند که آیا در users.json وجود دارد یا خیر. اگر وجود داشت، پسورد داده شده را با پسورد درون users.json چک میکند. درصورت مطابقت، True برمیگرداند به منظور اینکه احراز هویت موفقیت آمیز بود. در غیر این صورت False برمیگرداند.

**Create\_account()** : این کلاس متد username, password, balance را به عنوان پارامتر میگیرد. نقش این متد ساخت اکانت جدید ( اضافه کردن اکانت جدید به users.json ) میباشد. درصورتی که username داده شده از قبل در users.json وجود نداشته باشد، حساب جدید را که در واقع یک دیشکنری است را به users.json اضافه میکند و در انتها True برمیگرداند. درغیر این صورت False برمیگرداند.

**استاتیک متد ها Static methods** :

**Read\_password\_from\_file()** : username را به عنوان پارامتر میگیرد و پسورد آن یوزر نیم را از users.json برمیگرداند.

**Does\_username\_exist()** : username را به عنوان پارامتر میگیرد و بررسی میکند که آیا در users.json وجود دارد یا خیر. درصورتی که وجود داشت True برمیگرداند در غیر این صورت False بر میگرداند.

**Read\_balance\_from\_file()** : این متد فقط درمواقعی استفاده دارد که هنوز instance ای از کلاس User ساخته نشده اما ما نیاز داریم که بالانس یوزر مورد نظر را بخوانیم. Username را به عنوان پارامتر میگیرد و در users.json بالانس آن یوزر را برمیگرداند.

**ادامه دارد...**