

به نام خدا

پروژه برنامه نویسی پیشرفته

استاد داود آبادی

سال 1403-1404

اعضای گروه: فرزاد دهقان منشادی(402521252) و علی نقی نژاد(402522157)

هرکس قسمت هایی که خودش زده بود رو به صورت doc نوشته و توضیحی از مراحل کار داده است که در زیر اول فرزاد دهقان و بعد علی نقی نژاد به ترتیب قسمت خود را نوشته اند.

اول پروژه فایل .gitignore را ساختیم تا به سری مشکلات ترم پیش دوباره تکرار نشه. با زدن ui صفحه ی sign up شروع کردم و بعدش سراغ توابع validation و منطق کار کردن sign up رفتم اینجا بود که دیدم نمیدونم دقیقا باید چی کار کنم و تقریبا یک روز صرف تحقیق در مورد پاس کاری متغیرها و چجوری ارتباط دیتابیس و ui و منطق برنامه کردم که بیشتر اون با کمک و مشورت chatgpt و مقداری هم سرچ توی نت بود. بعد از این شروع به نوشتن یک نقشه و نمایش پاس کاری و ارتباط پوشه های مختلف روی کاغذ کردم و بعد این ساختار را توی پروژه پیاده کردم.

اول داشتیم اطلاعات user توی فایل json ذخیره می کردیم ولی خیلی زود متوجه شدیم اگه بخوایم به همین روال پیش ببریم کارمون خیلی سخت تر میشه و سر reporting , search خیلی اذیت خواهیم شد و این موضوع هم بود که توی فاز دوم باید دوباره کاری کنیم و همه این ها رو پاک کنیم و دیتابیس بیاریم بالا واسه همین از همینجا شروع به زدن برنامه با database کردیم. توی زدن ui ها ما در مرحله اول می رفتیم نمونه های خارجی و ایرانی ای که توی این زمینه زده شده بود رو می دیدیم و بعد به chatgpt می گفتیم که یک ui با این اطلاعات بزنه و در نهایت با در نظر گرفتن جفت موارد خودمون توی qt designer شروع به زدن می کردیم و بعد با یک دستوری توی ترمینال اون رو به فایل پایتون تبدیل می کردیم. اولین کار بعد از تبدیل کد تمیز کردن اون بود چون یک سری خط های کد مثل نامگذاری المنت ها توی پایتون نیاز نبود و qt به هیچ عنوان از Enter استفاده نمی کرد.

معمولا توی استفاده از قابلیت های qt وقتی به مشکل می خوردیم اولین گزینه مون برای سینتکس ها chatgpt بود ولی چند بار در طول پروژه هم مجبور شدیم بریم doc خود qt را بخوانیم.

بعد از تمام شدن قسمت ورودی برنامه وصل کردن login , signup به هم اولین قسمتی که زدیم record income , record cost بود و چون کاملا شبیه هم بودن یکی رو زدیم و اون یکی رو کپی کردیم.

اول پروژه هرکی داشت توی branch خودش push می کرد ولی اینجا که رسیده بودیم نمیدونم چرا ولی داشت ارور میداد و حتی یک سری مجبور شدم سر اشتباهی که با git به وجود اومده بود به چند commit قبل برگردم از اول بزنم و از اینجا به بعد هر دو توی master کد هایی که میزدیم را push می کردیم.

ما برای forgot password یک ماژول زدیم که بیاد و در صفحه اول username , سوال امنیتی که در هنگام ثبت نام ازش پرسیده شده را بگیره که 1. بریم و ایمیل اش رو از دیتابیس بخونیم و به ایمیل اش یک رمز موقت بفرستیم تا بهش اجازه دیدن رمزش رو بدیم 2. احراز هویت user هم با استفاده از چک کردن سوال امنیتی بررسی کنیم. (سوال امنیتی ما رنگ مورد علاقه شخص بوده)

در ادامه برگشتیم و برای نگه داشتن رمز دوم و ایمیل مبدا از environmental variables استفاده کردیم تا امنیت حساب ایمیل توی گیت هاب حفظ بشه.

مرحله بعد ui و منطق قسمت ثبت کردن category رو زدیم.

اولین صخره سختی که توی پروژه بهش خوردم قسمت search بود که هم توی doc یکم ناواضح بود هم خودش سخت بود و بیشتر از یک روز روش وقت گذاشتم که بیشتر اون صرف برطرف کردن انواع باگ هایی بود که انگار هرچی جلوتر میرفتم به جای اینکه کمتر بشن از توی دل هم زاییده میشدند.

در قسمت login ابتدا از کاربر username و password را میگیرد و در صورتی که اشتباه باشد باکس ها مربوطه قرمز میشود که از این طریق کاربر متوجه میشود که رمز یا یوزرنیم را اشتباه وارد کرده است و بعد از سه بار اشتباه وارد کردن دکمه ورود قفل میشود و به مدت ۶۰ کاربر نمی تواند ورود کند و این کم شدن زمان به کاربر نمایش داده میشود و در قسمت forget password اگر که کاربر رمز خود را فراموش کرده باشد با وارد کردن پاسخ سوال امنیتی و یوزرنیم با استفاده از کتابخانه smtp برای کاربر ایمیلی ارسال میشود که حاوی رمزی است که در صورتی که کاربر رمز را درست وارد کند به او رمز نمایش داده میشود و سپس کاربر به صفحه login منتقل میشود و برای بررسی رمز و یوزرنیم ui به شکل مستقیم با دیتابیس در ارتباط نیست و ابتدا درخواستش را به controller منتقل میکند و سپس اطلاعات از controller به ui منتقل میشود.

بعد از ورود به برنامه هفت تا آیکون به کاربر نمایش داده میشود که با کلیک بر روی هر قسمت وارد آن قسمت میشود و قسمتی هایی که مربوط به تراکنش ها است درخواست ها به transactions controller فرستاده میشود و بقیه قسمت ها مثل تنظیمات بیشتر با user controller در ارتباط است و در هر کدام از قسمت ها علاوه بر بررسی اطلاعات ورودی بعضی جاها از try except به دلیل ورودی های خیلی خاص استفاده شده است و در صورتی که کاربر قسمتی را اشتباه وارد کند به شکل messagebox به کار ارور مربوط به هر بخش نمایش داده میشود و در قسمت هایی مثل income cost reports که کاربر با تاریخ سروکار دارد به کاربر table نمایش داده میشود.

در بعضی جاها که بعضی تابع ها خیلی تکرار می شد برای جلوگیری کلاس و یا interface استفاده شده و برای بعضی قسمت ها باید یک متد و یا قسمت خاص از آبجکت ها که دارای متد یکسان هستند تغییر کند از duck typing استفاده شده و بعضی قسمت ها برای جلوگیری از اضافه کردن تابع ها از lambda استفاده شده است به طور مثال برای وصل کردن دکمه ها به تابع هایی که args دریافت میکنند.

در قسمت report ابتدا فیلد ها خواسته شده توسط کاربر پر میشود و در صورتی کاربر هیچ فیلدی انتخاب نکرد هیچ نموداری به کاربر نمایش داده نمیشود و کاربر باید از بین روز و ماه و سال و بازه یکی را انتخاب کند و اگر نکند به کاربر هشدار داده میشود که یک را انتخاب کند و در صورت درستی سه تا نمودار که مربوط به هر کدام از درآمد ها و هزینه ها به کل درآمد و هزینه ها نمایش داده میشود و نمودار سومی نیز نمایش داده میشود که نشان دهنده در نسبت درآمد و هزینه ها نسبت به هم هست که کاربر می تواند درصد آن ها و اینکه هر کدام مربوط به چه منبعی است ببیند.

در قسمت تنظیمات برای اضافه کردن تصویر از QFileDialog استفاده شد که کاربر می تواند فایلی را از سیستم انتخاب کند و بعد از آن اطلاعات مربوط به تصویر به دیتابیس اضافه شود و هر بار که کاربر دوباره ورود میکند به کاربر نمایش داده شود و برای حالت دارک و لایت هنگامی که کاربر تغییرات را اعمال کرد صفحه بسته می شود و صفحه لاگین نمایش داده میشود و وقتی دوباره کاربر وارد شود تغییرات اعمال میشود و صفحه به دارک و یا لایت تغییر پیدا میکند.

مدل نوشته شدن پروژه:

ما برای زدن پروژه از مدل VCM استفاده کردیم و پوشه های ما به صورت زیر بود:

- Controllers
- Database
- Models
- Pictures
- Utils
- Views
- Backups

درون پوشه controller دو فایل برای user و تراکنش ها هست که در واقع رابط بین قسمت backend و front برنامه ما هست که در قسمت back ما models , utils, database را داریم که models به نوعی قالب بندی user و تراکنش های ما هست. utils شامل email sender که با استفاده از کتابخانه smtplib به ایمیل کاربر رمز موقت می فرسته و validation که شامل یک کلاس هست که تمام تابع های validation ما توش قرار داره و در نهایت save که برای اضافه کردن منطق شی گرای به پروژه اضافه و از آن به عنوان interface در کد هایمان استفاده کردیم.