

به نام خدا

پروژه برنامه نویسی پیشرفته :

مهدی دهقانی و امین شیروانی

از ابتدای پروژه ما از رابط گرافیکی و PYQT استفاده کردیم و در برنامه QT Creator فرم ها و صفحه های مختلف مورد نیاز را طراحی میکردیم و سپس با استفاده از UIC از ان ها در کد استفاده کردیم.

ما برای هر فرم یک پوشه و یک فایل جدا گانه در نظر گرفتیم تا بتوانیم برای هر فرم کد عکس و فایل ui منحصر به فرد داشته باشیم و نیز ماژول بندی را رعایت کرده باشیم تا پروژه تمیز تر باشد و دارای خوانایی بهتری باشد.

چون هر دو تا حد خوبی به CSS تسلط داشتیم به اکثر فرم ها بک گراند Linear Gradient دادیم و با استفاده از متد CSS setStyleSheet های زیادی را به هر کدام از دکمه ها و line edit ها و غیره اد کردیم تا گرافیک برنامه ظاهر خوبی به خود بگیرد.

پروژه ما از نظر ماژول بندی به سه قسمت تقسیم میشود که قسمت اول شامل فرم ها استایل ها و متد های به خصوص خود هستند که هر کدام جدا گانه در پوشه های خود قرار دارند.

در قسمت دوم ما هسته مرکزی پروژه یعنی فایل Connector را نوشتیم که ارتباط بین تمامی فرم ها را ایجاد میکند و این قابلیت را به ما میدهد که بتوانیم به دکمه های موجود در داخل هر فرم سیگنال بدهیم و تعیین کنیم که هر کدام از دکمه ها چه کاری را انجام دهند اکثریت کد پروژه در این قسمت نوشته شده است و دارای متد های مختلف مربوط به هر کدام از فرم های پروژه است.

و در قسمت سوم ماژول هایی مانند نشان دادن پیام های مختلف تایمر و هسته اصلی ذخیره اطلاعات که در ابتدا فایل های json و کلاس json بود که بعد از برخورد با مشکلاتی غیر قابل حل و زمان بر با کلاس و فایل دیتابیس جایگزین شد.

در ابتدا ما تصمیم گرفتیم تا از فایل json برای ذخیره اطلاعات استفاده کنیم زیرا هم باعث شد تا ما کار با فایل های json را یاد بگیریم و با متد های کتابخانه json در پایتون آشنا شویم و اطلاعات بیشتری در مورد این فرمت فایل یاد بگیریم. این فرمت فایل به دلیل داشتن دیکشنری های تو در تو بعد از دیتابیس گزینه بسیار خوبی برای ذخیره اطلاعات بود.

ما در 4 فایل json تمامی اطلاعات را ذخیره کردیم در فایل اول اطلاعات شخصی تمامی کاربران در فایل دوم اطلاعات درآمد هر کاربر و در فایل سوم اطلاعات هزینه هر کاربر و نهایتاً در فایل چهارم کتگوری های وارد شده توسط هر کاربر را ذخیره کردیم.

ابتدا کار با فایل های json بسیار راحت بود و میتوانستیم با آن ها همه چیز را ذخیره کنیم تا به بخش search و report رسیدیم و در این مرحله به مشکل خوردیم و نمیتوانستیم به راحتی با فایل های json کار کنیم و در آن ها سرچ کنیم و تصمیم گرفتیم تا پروژه را ریفتکتور کنیم و فایل ها را به جای json روی database ذخیره کنیم. که در ابتدا زیاد باب میل نبود زیرا نمیخواستیم که در فاز اول به روی دیتابیس سوییچ کنیم و با json کار را به اتمام برسانیم اما به دلیل زمان بر بودن و سختی این امر موفق نشدیم.

با سوییچ کردن به روی دیتابیس و تغییر ساختار پروژه و پاک کردن فولدر و کلاس json و اضافه کردن کلاس دیتابیس کار بسیار راحت شد و بعد از آن با طراحی مجدد صفحه سرچ و گزارش گیری این دو بخش نیز به پایان رسیدند.

بعضی از ویژگی های منحصر به فرد پروژه که در داک ذکر نشده بود:

* در قسمت `forgot password` ما با استفاده از کتابخانه `smtplib` یک تایید رندوم به ایمیل فرد ارسال میکنیم تا فرد آن را در قسمت کد تایید وارد کند و اگر آن را درست وارد کند و عبارت `captcha` را نیز درست وارد کند پسورد فرد را به اون نشان خواهیم داد.

* در صفحه اصلی پروژه تایم لایو به کاربر نشان داده خواهد شد. و مقدار زمانی که کاربر در پنل خود بوده محاسبه میشود وقتی دکمه خروج در صفحه اصلی را بزند مقدار زمانی که فرد در پنل خود بوده است را به او نشان میدهیم.

* استفاده مکرر از `CSS`: با توجه به این که میتوانستیم در متد `setStyleSheet` کدهای سی اس اس بنویسیم در دکمه ها و لاین ادیت ها ما از سی اس اس های مختلف استفاده کردیم مانند `hover` در دکمه ها و `lineargradient` در لاین ادیت ها دکمه ها و بک گراند ها و ...

* قابلیت پخش صدا و کنترل صدا: پروژه ما دارای صداهاى مختلفى است مانند صدای پس زمینه صدای کلیک کردن بر روی دکمه ها و صداهاى متنوع برای هر کدام از پیام هایی که به کاربر نشان داده میشود. همچنین کاربر میتواند در قسمت تنظیمات برنامه صدای موزیک ها را قطع یا وصل کند.

* آیکون های متنوع: در هر صفحه از آیکون های متناسب با محتوای صفحه استفاده شده است و در نشان دادن تمامی پیام ها به کاربر نیز از آیکون های درست و متناسب با متن پیام استفاده شده است تا ظاهر گرافیکی برنامه بهبود یابد.

منابع استفاده شده برای ساخت پروژه:

ChatGPT-StackOverFlow-GeeksForGeeks-W3School- UiGradients-
StackExchange-Uppbeats(for sounds)