به نام خدا

گزارش مینی پروژه trellomize

Team "Dynamic Dou": Hamidreza Bayat, Pouria Golsorkhi

فعالیت های حمیدرضا بیات:

ساختار اولیه و منو بندی پروژه (شامل منوی اصلی، منوی ادیت، منوی ایجاد پروژه و task و...)

توابع: تابع sign up، تابع login، تابع filesWrite، تابع timeValidate، تابع checkEmail، تابع checkInUsers، تابع checkInProjects، تابع usernameCheck، تابع newTask، تابع newProject، تابع taskIndex، تابع editTasknew، تابع editAtask، تابع showProjects، تابع editProjects

(نام هر تابع بیانگر کار انجام شده توسط آن است؛ لذا از توضیحات اضافی در این باره صرف نظر کرده‌ایم)

اکثر exeption handling های مربوط به بخش های مختلف پروژه

Docstring برای توابع، ساختار بندی فایل های json و نحوه ذخیره سازی آنها، refactor کردن ساختار بعضی توابع

و به طور کلی، روند پیشروی و اجرای پروژه

فعالیت های پوریا گلسرخی:

تمام موارد مربوط به بخش manager (شامل اسکریپت manager و فایل manager.py و ذخیره سازی اطلاعات manager و منطق Active و Deactive کردن حساب های کاربری، تابع validating و...)

تمام موارد مربوط به بخش logger system (شامل فایل user_actions.log، اضافه کردن logging به تمام بخش های پروژه برای ثبت آنها، اضافه کردن تغییرات مدیر سیستم به logger و...)

استفاده از hash برای پسورد ها (شامل توابع get_hashed_password و check_password و ذخیره سازی آنها در فایل های json)

توابع showLeaderProjects و showMemberProjects برای نمایش مناسب پروژه ها

تمام موارد مربوط به یکتا بودن ID (شامل تابع checkIDUniqueness و ذخیره ID ها در فایل های json)

تمام موارد مربوط به task history (شامل اضافه کردن پیام مناسب به history یک task، اضافه کردن زمان و کاربر تغییر دهنده به history، نمایش مناسب Progress bar برای history، ذخیره سازی آن در فایل json)

تمام موارد مربوط به task comment (شامل قابلیت comment گذاری توسط هر یک از اعضای پروژه، ذخیره سازی کامنت ها با فرمت لیست مناسب در فایل json، نمایش کامنت های یک task به صورت جدول و...)

کلاس های taskStatus و taskPriority و پیاده سازی منطق ذخیره آنها

برخی exeption handling های مربوط به بخش های مختلف پروژه

در این پروژه سعی بر این بوده که حداکثر شباهت به سیستم های مدیریت پروژه وجود داشته باشد:

ابتدا کاربر می‌تواند با ساخت حساب کاربری و سپس ورود به آن، از گزینه های منو استفاده کند.
"ساخت پروژه به همراه تنظیمات مربوط به آن و اضافه کردن collaborator به پروژه، اضافه کردن یک task و اختصاص آن به اعضای پروژه، تنظیمات مربوط به task، نمایش مناسب وظایف یک پروژه با ترتیب اولویت، نمایش و تغییر وظایف یک پروژه، توانایی های مدیر سیستم و قابلیت فعال/غیرفعالسازی حساب کاربری"، خلاصه کوتاهی از قابلیت های این سیستم بود.

به طور کلی، هدف این بوده است که علی رغم اینکه پروژه و سیستم کاربری آن در فضای terminal انجام می‌شود، سرعت انجام کار ها و رابط کاربری، چیزی از سیستم های واقعی مدیریت پروژه در فضا های دیگر کم نداشته باشد! (به طور مثال، کاربر در هر جایی از برنامه، می‌تواند با چند بار زدن دکمه enter به منوی اصلی برنامه برگردد، از برنامه خارج و یا به حساب کاربری دیگری وارد شود!

–کدام بخش‌ها می‌توانستند بهتر طراحی شوند؟ پیشنهادهای خود را ارائه دهید.

+ از نظر محتوای کد و فایل های برنامه نویسی شده، بسیار جا برای این بود که کد ها تمیز تر زده شده و از تکرار بعضی کد ها جلوگیری شود. همچنین استفاده مناسب از توابع و فرم بهتر کد ها و نام متغیر ها از مواردی بود که می‌توانست بهتر پیش برود.

– معماری ارائه شده چقدر قابلیت توسعه و گسترش امکانات و قابلیت‌ها را دارد؟

+ همانطور که پیش تر ذکر شد، تلاش بر این بوده که روند برنامه، سریع و راحت باشد. بنابرین با کمی refactor کردن کد ها و توابع، می‌توان قابلیت های بسیاری به این سیستم اضافه کرد

– فرض کنید بخواهیم این سیستم را در دنیای واقعی مورد استفاده قرار دهیم. این سیستم چه چالش‌هایی برای استفاده‌های واقعی دارد؟ توضیح دهید

+ طبیعتا در دنیای کنونی، UI یک سیستم از اهمیت بالایی برخوردار است و سیستم مدیریت پروژه‌ای که رابط کاربری آن terminal است، چندان دلنشین نیست. از نظر قابلیت و امکانات، این سیستم سرعت و کیفیت کافی را برخوردار است. اما از نظر رابط کاربری و نحوه ارتباط با کاربر، چالش های زیادی وجود دارد که باید برای آنها معماری بهتری پیاده سازی شود.

– برای هر قسمت علت انتخاب الگوریتم‌ها و پکیج‌های مورد استفاده‌تان را توضیح دهید.

+ در این سیستم، از پکیج های خارجی زیادی استفاده نشده! هرچند هر آنچه استفاده شده بدین صورت است (به ترتیب import در کد):

os **برای** system cls**، کتابخانه** rich **برای زیباسازی خروجی** terminal**،** datetime & Time **برای استفاده از زمان در برنامه،** rich.progress **برای نمایش** progress bar **در بخش** history**،** json **برای ذخیره سازی اطلاعات در فایل های** json**،** Re **برای چک کردن فرمت** email**،** bcrypt **برای** hash **کردن پسورد ها،** uuid **برای ساخت** ID**،** logging **برای** logger system**،** enum **برای ساخت** enum class

۱۱ خرداد ۱۴۰۳