

## ❖ گزارش پایانی گروه C2

### فعالیت‌های انجام‌شده توسط هریک از اعضای تیم:

کیمیا:

- ۱- پیاده‌سازی و طراحی نمایش تابع‌های مربوط به منوها اعم از `user_menu` , `main_menu` و غیره
- ۲- پیاده‌سازی و طراحی `console` و `Table` ها با استفاده از کتابخانه `rich`
- ۳- اجرای آزمون واحد (Unit Test) در فایل `test.py`
- ۴- اعمال قواعد PEP 8, کامنت گذاری و اعمال Exception Handling در همه فایل‌ها

مهدی:

- ۱- ایجاد کلاس‌ها و Attribute های اولیه
- ۲- ایجاد فایل `manager.py` و پیاده‌سازی تابع‌های مربوط به آن
- ۳- پیاده‌سازی `database` پروژه و ذخیره‌سازی داده‌ها
- ۴- پیاده‌سازی قسمت تاریخچه (Task History)

فعالیت‌های مشترک:

- ۱- پیاده‌سازی تابع‌های موردنیاز در کلاس‌ها، تابع‌های مستقل و اجزای داخلی آن‌ها
- ۲- اعمال سیستم لاگینگ (logging)
- ۴- نگارش گزارش پایانی پروژه
- ۳- سایر بخش‌های ذکرشده ولی اعمال شده در پروژه

### مشکلات به وجودآمده در روند توسعه پروژه:

- ۱- عدم داشتن دید کامل در ابتدای پروژه درباره نوع `database`, جزییات آن و چگونگی برقراری روابط اصولی بین داده‌ها
- ۲- وجود دشواری در پیاده‌سازی Unit Test به دلیل عدم مستقل بودن توابع و نیز ورودی گرفتن بعضی از آن‌ها

### بخش‌هایی از پروژه که می‌توانستند بهتر طراحی شوند و پیشنهاداتی برای بهبود آن‌ها:

- ۱- طراحی سیستم به صورت GUI برای دسترسی بهتر و آسان‌تر کاربر به بخشهای مختلف پروژه
- ۲- ایجاد بخشی با عنوان Notifications برای نمایش پیامها از جمله اضافه‌شدن کاربرها به یک وظیفه یا پروژه و نیز اطلاع‌رسانی به ایشان

### قابلیت‌های پروژه برای توسعه و گسترش امکانات و قابلیت‌ها:

- ۱- اضافه‌کردن قابلیت ارسال ایمیل به افراد پروژه و اطلاع‌رسانی درباره تغییرات ایجادشده
- ۲- اضافه‌کردن زمان سنج برای اطلاع‌رسانی موعد پایان وظیفه‌های تخصیص داده شده

## چالش‌های استفاده از سیستم در دنیای واقعی:

به دلیل اجرا شدن سیستم در محیط ترمینال، کاربر توانایی مشاهده تمام قابلیت‌ها و امکانات پروژه را به صورت همزمان ندارد، به همین دلیل ممکن است کمی گمراهی برای کاربر به وجود آید، لذا سیستم ارائه‌شده توسط گروه ما تا جای ممکن کاربر را در قسمت‌های مختلف با نمایش پیام و عنوان هر بخش راهنمایی کرده و کاربر امکان مشاهده تغییرات به وجودآمده را دارد.

## علت انتخاب الگوریتم‌ها و پکیج‌های مختلف در پیاده سازی کد:

### ۱- علت وجود فایل `emails_and_usernames`:

با ذخیره ایمیل و نام کاربری کاربرها در این فایل، هنگام رجیسترکردن کاربران، دسترسی به ایمیل و نام کاربری برای چک کردن unique بودن آن و همچنین نمایش افراد بسیار آسان‌تر از پیمایش در database کاربران است.

### ۲- علت افزودن ID کاربر به جای نام کاربری در database پروژه:

به دلیل پیاده‌سازی گزینه `edit username` در سیستم، در صورتی که کاربر بخواهد `username` خود را تغییر دهد، پیمایش در پروژه‌های کاربر و تغییر مقادیر `username` قدیمی به جدید بسیار زمان‌گیر است و حافظه زیادی را می‌طلبد ولی به دلیل unique و غیرقابل تغییر بودن ID، صرفاً نیاز است که `username ID` فعلی را نمایش دهیم.

### ۳- علت جداجدا بودن دیتاهای پروژه و یوزرها:

در صورت داشتن یک فایل داده واحد خوانش، تغییر و ذخیره‌سازی مجدد آن در طول روند برنامه زمان و حافظه زیادی می‌طلبد و از لحاظ اصول مطلوب‌تر است اطلاعات را به‌صورت تکه‌تکه داشته باشیم تا از لحاظ حافظه و زمان صرفه‌جویی کنیم.

### ۴- علت وجود فایل `projects.json` به صورت مستقل:

برای افزایش خوانایی پروژه و دسترسی راحت‌تر به فایل‌های کاربر.

### ۵- علت استفاده از پکیج‌های ایمپورت شده در پروژه:

استفاده از `rich` برای نمایش `table` ها و ایجاد `cutome theme` برای اضافه کردن رنگ‌ها  
استفاده از `platform, sys, tty, termios, msvcrt` برای پیاده‌سازی تابع `wait_for_press_key`  
استفاده از `pwininput` برای عدم نمایش `password` در هنگام `login` و نمایش آن به صورت \*\*\*\*\*  
استفاده از `bcrypt` برای رمزگذاری `password`  
استفاده از `uuid` برای ساختن ID برای افراد، پروژه‌ها و وظیفه‌ها  
استفاده از `os` و `json` برای کار با فایل و فولدرها  
استفاده از `re` برای تعریف نمونه `username,email,password` معتبر و بررسی آن‌ها  
استفاده از `unittest` برای پیاده سازی آزمون واحد و `shutil` برای حذف یک دایرکتوری و محتوای آن