



پروژه چهارم برنامه نویسی پیشرفته

دکتر میرروشندل و آرش یوسفی

ترم ۱۴۰۳۲

IMF datacenter



هدف

کار با مفاهیم پیشرفته شی گرای و الگوی طراحی decorator و همچنین ارث بری، کپسوله سازی، چند ریختی و dunder methods در پایتون و چند نخ (threading) و همینطور آشنایی به مبانی امنیت و وب سوکت

مقدمه

شما مهندسان نرم افزار یک سازمان فوق سری به نام IMF هستید. وظیفه‌ی شما طراحی یک مرکز داده‌ی پیشرفته برای ذخیره و پردازش اطلاعات مأموریت‌های مخفی است. این مرکز داده باید بتواند درخواست‌های مختلف را مدیریت کند، تا بتواند سطح دسترسی هر فرد را به هر مفهوم مدیریت کنید.

توضیح پروژه

موجودیت های درون نرم افزار

مأموران (Agents)

- هر مأمور دارای نام، نام خانوادگی، شناسه، رمز عبور و سطح دسترسی است.
- هر مامور با توجه به سطح دسترسی خود می‌تواند فایل‌هایی را هک (مطالعه) کند و بررسی کند!

وزیر (Secretary)

- هر وزیر دارای نام، نام خانوادگی، شناسه، رمز عبور است.
 - وزیر می‌تواند ماموران را اضافه یا حذف کند.
 - وزیر می‌تواند فایل‌ها را اضافه یا حذف کند.
- دقت کنید که آبجکت وزیر از نوع singleton است یعنی فقط یک شی می‌تواند از آن ساخته شود که آن یک شی اطلاعات خود را از دیتابیس اولیه برنامه می‌خواند.

فایل‌ها (Files)

- هر فایل دارای آدرس ذخیره، سطح رمزگشایی است.
 - فایل‌ها به سه دسته تقسیم می‌شوند:
۱. فایل‌های هویت (Personnel Files) → اطلاعات کارکنان، مأموران، و مدیران IMF.
 ۲. فایل‌های رمز تسلیحات هسته‌ای (Nuclear Codes) → شامل کدهای فعال‌سازی سلاح‌های هسته‌ای.

۳. فایل‌های تحقیقات بیولوژیکی (Biological Research Files) → شامل اطلاعات ویروس‌ها، واکسن‌ها، و پروژه‌های فوق‌سری.

۴. تفاوت هر نوع فایل در سطح دسترسی و الگوریتم رمزنگاری آنان است.

سرور (Server)

- کنترل کننده اصلی برنامه شما سرور است! که باید در آن درخواست‌های کاربران رسیدگی و اجرا شوند.

کلیات نرم افزار

مامورین IMF باید یک سری سرور و یک سری فایل‌ها را هک کنند و محتوای آن را تغییر دهند و با هم صحبت کنند.

فایل‌ها: فایل‌ها دارای ۳ نوع هستند که تفاوت آن‌ها صرفاً در سطح دسترسی و الگوریتم‌های رمزنگاری آن‌هاست. (جهت آشنایی دانشجویان با الگوریتم‌های رمزنگاری انتخاب الگوریتم‌ها به عهده خودتان است - دقت کنید این بخش به این معنا است که اگر شما فایل را در سیستم خود خارج از برنامه باز کنید محتوای معنا داری را نبینید! و محتوا در صورت فراخوانی مامور decode شود).

سرور: سرور لیستی یا بهتر بگوییم (LOG) از مراجعین خود دارد، و در ازای دریافت یک دستور ابتدا آن را تجزیه تحلیل می‌کند و سپس به آن پاسخ می‌دهد! دقت کنید سرور با استفاده از چند نخ‌ی باید بتواند از چندین client پشتیبانی کند نمونه‌ای از کارهای سرور به شرح زیر است:

- پذیرش کاربران: فرآیند ورود و وارد شدن کاربر
- مدیریت ثبت و حذف مامور
- مدیریت فایل‌ها: وقتی فایلی اضافه می‌شود سرور باید یادش بماند و آن را متناسب با نوع آن رمزگذاری کند
- سرچ فایل‌ها: باید سطح دسترسی فرد درخواست دهنده را متوجه شود فایل را پیدا کند و نشان دهد
- اعمال تغییر فایل‌ها: اگر کسی فایل را ویرایش کند باید آن را دوباره رمزنگاری کند و در فایل قرار دهد
- مدیریت چت‌های کاربران
- در صورت پیاده سازی امتیازی‌ها بدانید که هر کدام از آن‌ها تابعی به سرور اضافه می‌کنند تنها امتیازی رجکس است که در سمت کلاینت انجام می‌شود.

دقت کنید که اگر سرور به هر دلیلی با شکست مواجه شد باید آن را با پیامی مناسب به کاربر گزارش دهد.

بخش های برنامه

پنل اولیه: در پنل اولیه شما به سرور اصلی IMF وصل می شوید که تنها گزینه آن ورود است.

پنل وزیر: وزیر می تواند مامور یا ماموریت جدیدی را بسازد و یا فایل های جدیدی را بسازد و در آن ها اطلاعاتی بگذارد.

- در بخش فایل او می تواند فایل جدیدی را ایجاد کند یا آدرس یک فایل را در کامپیوتر خود بدهد.
- او می تواند هر فایلی را ببیند.
- او تنها کسی است که می تواند فایل را حذف کند.

پنل مامورین: در پنل مامورین آنها دو بخش اصلی دارند! اول بخش مشاهده فایل که در آن باید نام فایل را سرچ کنند و در صورت وجود و اینکه سطح دسترسی به آن را داشته باشند باید بتوانند آن فایل را ببیند و ویرایشش کنند. بخش دوم بخش چت برنامه است! که افراد یا می توانند با زدن شناسه مامور دیگر به آن مامور (در صورت وجود) یک پیام بفرستند یا یک پیام را broad cast کنند و برای چندین اعضا بفرستند.

نکات تکمیلی

۱. در هر مرحله با استفاده با decorator (`__method__`) magic method مناسب سطح مهارت/دسترسی را بسنجید
۲. اطلاعاتی مانند پسورد باید حتما خصوصی باشند و هاش شوند.
۳. مامورین دارای توابع ساده هک هستند که صرفا اگر دسترسی داشته باشند می توانند فایل ها را ببیند.
۴. استفاده از ارث بری اجباری است اما معماری و مهندسی موجودیت ها به عهده خودتان است.
۵. دقت کنید که کلاس های مادر abstract هستند و نمی توان از آن ها مثالی ساخت.
۶. تمامی داده ها را در فایل ذخیره کنید و در ابتدای برنامه آن ها را بخوانید.
۷. پیاده سازی GUI خلاقانه و در خور اجباری است.

نکته مهم:

هدف و ارزشیابی اصلی این پروژه کار با مفاهیم سرور-سوکت و شبکه است. لازم است که تمامی درخواست ها از طریق سرور-سوکت ارسال، دریافت و روی آبجکت سرور اجرا شوند. بنابراین، به هیچ عنوان برنامه را Local پیاده سازی نکنید. عدم رعایت این نکته در پیاده سازی امکانات برنامه، منجر به بی اعتبار شدن بخش بزرگی از پروژه تحویلی خواهد شد.

امتیازی:

ایجاد هوش مصنوعی / الگوریتم برای پیام رسانی بهتر (۵۰ نمره): از الگوریتم های هوش مصنوعی برای اصلاح غلط های املائی در چت میان افراد استفاده کنید!

دیتابیس (100 نمره): دیتای موجودیت های خود را در یک دیتابیس به انتخاب خود ذخیره کنید و بخوانید.

رجکس (50 نمره): با استفاده از اصول رجکس فورمت در خواست های خود را به سرور به فورمتی درست تبدیل کنید به عنوان مثال اگر در ورودی چنین چیزی در چت مامورها دیدید آن را به این صورت بفرستید:

input : Agent1 Agent2 "Are you ready ?"

final request : {sender:Agent1, receiver : Agent 2, message : "Are you ready ?"}

ایجاد گروه برای چت (50 نمره) : یک فرد بتواند به گروه بسازد و افرادی را به آن اضافه کند و چت کنند !

داریکتوری های بازگشتی (50 نمره) : فرآیند سرچ کردن در فایل ها را برای به صورت بازگشتی پیاده کنید ... در ابتدا برنامه شما ممکن است همه فایل ها در یک پوشه نباشند! برنامه شما باید به صورت بازگشتی در پوشه مادر پروژه شما بگردد و تمامی فایل های پیدا شده را به سرور های مربوطه وصل کند.

ثبت پروژه

پروژه خود را در فایل فشرده Zip با فرمت زیر بنویسید و آن را در کوئرا ارسال کنید.

APProject4_FirstnameLastname_StudentNumber.zip

نکات عمومی پروژهها

هرگونه ایده و خلاقیت جدید، تنها با به کارگیری مباحث پروژه با هماهنگی با دستیاران آموزشی، تا ۲۵ درصد نمره مثبت به همراه خواهد داشت.

پروژه شما تنها در صورتی مشمول نمره مثبت میشود، که بخش اصلی پروژه را 80 درصد تکمیل کرده باشید.

تنها استفاده از زبان برنامه نویسی پایتون مجاز است.

کپی و استفاده کورکورانه از منابع اینترنتی، یا ابزارهایی نظیر Chat-GPT ممنوع است.

پروژه تنها به صورت انفرادی قابل انجام بوده و وجود هرگونه تشابه مشکوک بین دو کد یا عدم تسلط به ارائه، شامل جریمه خواهد بود.

رعایت اصول و آموزه های درس برنامه سازی پیشرفته، نام گذاری صحیح، کامنت نویسی مناسب، رعایت دندانده گذاری، و خوانا بودن کد الزامی است.