

شقایق فرض پور ریاضیات و کاربرد ها

شماره دانشجویی : 99441227

تیم H2

گزارش مینی پروژه:

• هر یک از اعضای تیم چه فعالیت هایی انجام داده اند؟

خودم تمام پروژه را انجام دادم.

• چه مشکلاتی در روند توسعه پروژه وجود داشته است؟

یکی از مشکلات اساسی در فرایند توسعه این پروژه، ناپایداری و عدم هماهنگی در تعریف نیازمندی ها و خواسته ها بود. به عنوان مثال، منوها و رابط کاربری به صورت کاملاً دقیق و با جزئیات کافی تعریف نشده بودند که این امر باعث سردرگمی در حین پیاده سازی شد.

در ابتدای کار، تصمیم گیری درباره نحوه ساخت کلاس ها و مدیریت داده ها با چالش هایی همراه بود. اینکه آیا پروژه ها در کلاس User ذخیره شوند یا کلاس Project، باگ هایی ایجاد کرد و در نهایت تصمیم به ایجاد کلاس جدیدی به نام UserManager گرفته شد.

کمبود تست های کافی و خودکار در فرآیند توسعه، باعث شد که برخی از باگ ها و مشکلات به موقع شناسایی نشوند و در مراحل بعدی کشف شوند که رفع آنها زمان بر بود.

• کدام بخش ها میتوانند بهتر طراحی شوند؟ پیشنهاد های خود را ارائه دهید.

یکی از بخش هایی که میتواند بهبود یابد، مدیریت کاربران و دسترسی ها است. پیشنهاد می شود سیستم نقش ها و مجوز ها برای کاربران پیاده سازی شود تا امنیت و مدیریت دسترسی ها بهتر انجام شود. همچنین، استفاده از یک پایگاه داده رابطه ای به جای فایل های JSON میتواند به بهبود مدیریت داده ها و کارایی سیستم کمک کند. افزودن تست های یکپارچه به تست های واحد نیز میتواند اطمینان بیشتری از عملکرد صحیح سیستم فراهم کند. بهبود رابط کاربری با طراحی یک رابط گرافیکی نیز تجربه کاربری را ارتقا می دهد. علاوه بر این، افزودن ویژگی هایی مانند ادغام با ابزار های دیگر مدیریت پروژه، گزارش گیری پیشرفته، و سیستم یادآوری وظایف میتواند کاربرد و بهره وری سیستم را افزایش دهد. همچنین،

پیاده‌سازی مکانیزم‌های امنیتی مناسب برای حفاظت از اطلاعات حساس کاربران و پروژه‌ها ضروری است.

• معماری ارائه شده چقدر قابلیت توسعه و گسترش امکانات و قابلیت ها را دارد؟

با توجه به اینکه اغلب کارهایی که انجام شده در قالب توابع هستند، به ما کمک می‌کند تا توسعه سریع‌تر و راحت‌تری داشته باشیم، معماری ارائه‌شده برای این پروژه به دلیل استفاده از اصول برنامه‌نویسی شیء‌گرا و طراحی ماژولار، قابلیت توسعه و گسترش بالایی دارد. طراحی ماژولار به این معناست که هر بخش از سیستم به طور مستقل پیاده‌سازی شده و می‌تواند به راحتی با افزودن ماژول‌های جدید گسترش یابد. این امر امکان افزودن ویژگی‌های جدید را به راحتی فراهم می‌کند.

علاوه بر این، استفاده از کتابخانه‌ها و ماژول‌های استاندارد پایتون مانند `argparse`، `logging` و `uuid`، انعطاف‌پذیری و قابلیت انطباق سیستم را افزایش می‌دهد. این معماری به توسعه‌دهندگان این امکان را می‌دهد تا با افزودن یا جایگزینی ماژول‌ها و کتابخانه‌های جدید، قابلیت‌ها و امکانات سیستم را بهبود بخشند. به طور کلی، ساختار فعلی پروژه به گونه‌ای است که می‌تواند به سرعت و با کمترین تغییرات اساسی، به نیازهای جدید و متغیر پاسخ دهد.

این معماری همچنین به راحتی امکان یکپارچه‌سازی با سرویس‌ها و API‌های خارجی را فراهم می‌کند. به عنوان مثال، می‌توان سیستم را به گونه‌ای گسترش داد که با سرویس‌های ابری مانند AWS یا Google Cloud تعامل داشته باشد یا با ابزارهای مدیریت پروژه مانند Jira و Slack ادغام شود. به این ترتیب، قابلیت توسعه و گسترش این سیستم نه تنها به لحاظ افزودن ویژگی‌های داخلی بلکه به لحاظ تعامل و همگام‌سازی با سیستم‌ها و ابزارهای خارجی نیز بسیار بالا است.

از دیگر مزایای این معماری می‌توان به امکان به‌روزرسانی و ارتقاء سیستم بدون نیاز به بازنویسی بخش‌های اصلی اشاره کرد. این قابلیت به توسعه‌دهندگان اجازه می‌دهد تا با افزودن ویژگی‌ها و قابلیت‌های جدید، سیستم را به‌روز نگه دارند و با نیازهای متغیر کاربران همگام باشند.

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

1. مقیاس‌پذیری: یکی از مهم‌ترین چالش‌ها، مقیاس‌پذیری سیستم است. در دنیای واقعی، سیستم باید بتواند تعداد زیادی از پروژه‌ها، وظایف و کاربران را بدون کاهش کارایی مدیریت کند. این موضوع نیازمند طراحی و پیاده‌سازی مناسب زیرساخت‌ها، مانند استفاده از پایگاه داده‌های قدرتمند و سرویس‌های مقیاس‌پذیر ابری، است تا سیستم بتواند با افزایش حجم داده‌ها و کاربران، کارایی خود را حفظ کند.

2. امنیت: حفاظت از اطلاعات حساس کاربران و پروژه‌ها یکی دیگر از چالش‌های مهم است. سیستم باید مکانیزم‌های امنیتی مناسبی مانند احراز هویت قوی، کنترل دقیق دسترسی، رمزنگاری داده‌ها، و محافظت در برابر حملات سایبری داشته باشد. عدم توجه به امنیت می‌تواند منجر به نشت اطلاعات و آسیب‌های جدی به کاربران و سازمان‌ها شود.

3. پشتیبانی و نگهداری: سیستم‌های نرم‌افزاری در دنیای واقعی نیازمند پشتیبانی و نگهداری مداوم هستند. این شامل به‌روزرسانی‌های منظم برای رفع اشکالات، بهبود کارایی، و افزودن ویژگی‌های جدید است. همچنین، نیاز به پشتیبانی فنی برای پاسخگویی به سوالات و مشکلات کاربران وجود دارد که باید به طور موثر مدیریت شود.

4. تجربه کاربری: در دنیای واقعی، تجربه کاربری نقش حیاتی در موفقیت یک سیستم دارد. سیستم باید دارای رابط کاربری ساده و کاربرپسند باشد تا کاربران با سطوح مختلف مهارت فنی بتوانند به راحتی از آن استفاده کنند. بهبود رابط کاربری و ارائه آموزش‌ها و مستندات کافی می‌تواند به افزایش پذیرش و رضایت کاربران کمک کند.

5. همزمانی در دسترسی داده‌ها: نوشتن همزمان بر روی داده‌ها توسط ادمین و کاربر می‌تواند مشکلاتی را ایجاد کند. این مسئله نیازمند مکانیزم‌های مدیریت همزمانی و کنترل دسترسی مناسب است تا از بروز مشکلات جلوگیری شود.

• برای هر قسمت علت انتخاب الگوریتم‌ها و پکیج‌های مورد استفاده تان را توضیح دهید.

انتخاب این ابزارها و الگوریتم‌ها به دلایل مختلفی انجام شده :

1. استفاده از "argparse"

علت انتخاب:

پکیج argparse یک ابزار استاندارد پایتون برای مدیریت آرگومان‌های خط فرمان است. این پکیج امکاناتی قدرتمند برای تعریف و پردازش انواع مختلف آرگومان‌ها و گزینه‌ها را فراهم می‌کند. انتخاب argparse به دلیل نیاز به ساده‌سازی تعامل با کاربر از طریق خط فرمان و مدیریت ورودی‌های مختلف بود. این ابزار به ما امکان می‌دهد که دستورات مختلف را به سادگی اجرا کنیم و اطلاعات لازم را از کاربر دریافت کنیم. به عنوان مثال، در این پروژه از argparse برای دریافت نام کاربری و رمز عبور جهت ایجاد ادمین سیستم و همچنین برای انجام عملیات مختلف مدیریت کاربران استفاده شده است.

2. استفاده از "os"

علت انتخاب:

ماژول os در پایتون به ما امکان دسترسی به عملکردهای سیستم عامل را می‌دهد. از این ماژول برای انجام عملیات‌هایی مانند بررسی وجود فایل‌ها و حذف آن‌ها استفاده شده است. انتخاب os به دلیل نیاز به انجام عملیات‌های سطح سیستم مانند بررسی وجود فایل ادمین و حذف فایل داده‌ها بود. این عملیات‌ها نیاز به تعامل با سیستم فایل دارند که توسط این ماژول به خوبی پشتیبانی می‌شوند. به عنوان مثال، در پروژه برای بررسی وجود فایل admin.txt و همچنین برای حذف فایل data.json از این ماژول استفاده شده است.

3. استفاده از "json"

علت انتخاب:

ماژول json در پایتون برای خواندن و نوشتن داده‌ها در فرمت JSON بسیار مناسب است. این فرمت به دلیل قابلیت خوانده شدن توسط انسان و ماشین و همچنین ساختاریافته بودن، انتخاب شده است. استفاده از JSON به ما امکان می‌دهد تا داده‌های کاربران و پروژه‌ها را به صورت ساختاریافته ذخیره کنیم و به راحتی آن‌ها را بخوانیم و به‌روزرسانی کنیم. در این پروژه، اطلاعات کاربران و پروژه‌ها در فایل data.json ذخیره می‌شود و برای این منظور از ماژول json استفاده شده است.

4. استفاده از "rich" (شامل Console, Table, Prompt, Confirm)

علت انتخاب:

پکیج rich یک ابزار قدرتمند برای ایجاد رابط‌های کاربری کنسولی زیبا و تعاملی است. این پکیج امکاناتی مانند نمایش جداول، چاپ متون با فرمت‌های مختلف، و تعامل با کاربر از طریق درخواست‌های ورودی و تاییدیه‌ها را فراهم می‌کند. انتخاب rich به دلیل افزایش خوانایی و زیبایی رابط کاربری کنسولی بود. با استفاده از rich می‌توانیم جداول کاربران را به شکلی زیبا نمایش دهیم و درخواست‌ها و تاییدیه‌ها را به صورتی کاربرپسند ارائه کنیم. به عنوان مثال، در پروژه از Table برای نمایش جدول کاربران و از Prompt و Confirm برای دریافت ورودی‌ها و تاییدیه‌ها از کاربر استفاده شده است.

5. تعریف کلاس‌های User و UserManager

علت انتخاب:

تعریف کلاس‌های User و UserManager برای مدیریت کاربران به صورت شی‌گرایانه انجام شده است. این رویکرد به دلیل سازماندهی بهتر کد و قابلیت توسعه آسان انتخاب شده است. با استفاده از این کلاس‌ها می‌توانیم ویژگی‌ها و رفتارهای مرتبط با کاربران را به صورت مجزا تعریف و مدیریت کنیم. این ساختار ماژولار به ما اجازه می‌دهد که کدهایمان را به صورت قابل نگهداری و گسترش‌پذیر بنویسیم و از تکرار کد جلوگیری کنیم. به عنوان مثال، کلاس User برای نگهداری اطلاعات هر کاربر و کلاس UserManager برای مدیریت کاربران و انجام عملیات‌هایی مانند فعال‌سازی و غیرفعال‌سازی کاربران استفاده شده است.

6. توابع "create_admin" و "main"

علت انتخاب:

تعریف تابع create_admin برای ایجاد ادمین سیستم به صورت متمرکز به دلیل افزایش امنیت و جلوگیری از تکرار کد بود. این تابع تضمین می‌کند که تنها یک ادمین سیستم وجود داشته باشد و اطلاعات آن به درستی ذخیره شود. همچنین، تابع main به عنوان نقطه ورود برنامه، عملیات‌های مختلف را بر اساس آرگومان‌های خط فرمان مدیریت می‌کند. این ساختار به ما اجازه می‌دهد که به راحتی دستورات مختلف را اجرا کنیم و برنامه را گسترش دهیم.

انتخاب این الگوریتم‌ها و پکیج‌ها بر اساس نیاز به مدیریت کاربران، افزایش امنیت، سهولت استفاده، و زیبایی رابط کاربری انجام شده است. هر یک از این ابزارها ویژگی‌ها و قابلیت‌های خاصی دارند که به ما در رسیدن به اهداف پروژه کمک می‌کنند. انتخاب این ابزارها باعث شد که پروژه به صورت قابل نگهداری، توسعه‌پذیر و کاربرپسند پیاده‌سازی شود.