

پروژه پایانی

ترم مهر ۱۴۰۱، مهندسی کامپیوتر، دانشگاه امام خمینی قزوین

پیاده سازی نرم افزار ساده مدیریت انبار

خلاصه نرم افزار مدیریت انبار، نسخه ما:

- یک نرم افزار ساده مدیریت انبار برای یک بیزینس کوچک به صورت چند کاربره.
- شامل امکاناتی مانند: تعریف کالا، درج ورود کالا، درج خروج کالا، و انواع و اقسام گزارش گیری.
- جهت مرور مفاهیمی مانند: ساختمان، فایل، آرایه و رشته، لیست پیوندی و طراحی واسط کاربری.

همه بخش های این فایل را خط به خط، جمله به جمله و کلمه به کلمه **باید**

دقیق و با حوصله چندین بار بخوانید و مرور کنید.

مبادا نکته ای از قلم بیفتد.

ممکن است نکات جدیدی برای توضیح دقیق تر پروژه ارائه شود. برای این نکات احتمالی گوش به زنگ باشید.

بخش های برنامه

۱- صفحه ورود و ثبت نام کاربران

منوی ورود به برنامه دارای سه گزینه است:

۱. ثبت نام کاربر جدید
۲. ورود کاربران
۳. خروج

برای پیاده سازی این منو، گزینه های موجود برای انتخاب کاربر را به نحو مناسبی نمایش داده و گزینه انتخابی کاربر را در قالب یک عدد دریافت کنید.

به عنوان مثال، گزینه ها می تواند به فرمت زیر باشد:

```
Welcome To NASTOOH Accounting Soft-ware  
  
1. Sign Up!  
2. Login  
3. EXIT  
  
>>Please Enter Your Choice: 1
```

با انتخاب گزینه خروج، برنامه بسته می شود.

دقت کنید این عکس فقط یک نمونه است و شما باید با سلیقه خود منوها را طراحی کنید...

۱- ۱: بخش ثبت نام کاربران

در این بخش باید به ترتیب اطلاعات کاربر را از وی دریافت کنید و در یک فایل مجزا ذخیره کنید.

اطلاعات ذخیره شده از کاربر باید حداقل شامل موارد زیر باشد:

۱. نام کاربری
۲. نام
۳. نام خانوادگی

۴. کد ملی

۵. شماره همراه

۶. ایمیل

۷. پسورد

چند نکته در مورد این بخش:

- با نمایش پیغامهای مناسب، داده های کاربر را به ترتیب دریافت کنید.
- پسورد را باید حتماً دوبار دریافت کنید و مقادیر وارد شده را با هم تطبیق دهید تا از اشتباهات تایپی کاربران جلوگیری کنید.
- فرمت داده های ورودی را حتماً چک کنید و اجازه ندهید کاربر اطلاعات بی معنی وارد کند.
 - به عنوان مثال کد ملی ۱۰ رقمی است.
 - در کد ملی کاراکتر جایی ندارد.
 - شماره همراه فرمتی مانند ۰۹۱۲۳۴۵۶۷۸۹ دارد.
 - حتی می توانید فرمت ایمیل را چک کنید.
 - از همه مهمتر، اجازه انتخاب پسوردهای ساده، کوتاه و قابل حدس را به کاربر ندهید. (توصیه: این موضوع را در انتهای کار خود به برنامه اضافه کنید)

یادآوری مهم: قرار نیست از ابتدا همه بخش ها و امکانات برنامه را به یکباره (و یا حتی به ترتیب این فایل) پیاده سازی کنید. توصیه می شود ابتدا اسکلت برنامه خود را بنا کنید و سپس شاخ و برگ های اضافه را پیاده سازی کنید. به عنوان مثال چک کردن فرمت داده های ورودی را می توانید در انتهای کار پیاده سازی کنید. (یا حتی وقتی در پیاده سازی بخش های دیگر برنامه به مشکل اساسی برخورد کردید، به عنوان زنگ تفریح!!! و برای آزاد شدن فکرتان، به این مسائل کوچک باقیمانده برگردید.)

۱-۲: بخش ورود کاربران

در این بخش، کاربر باید ابتدا نام کاربری خود و سپس پسورد خود را وارد کند. برنامه باید صحت نام کاربری و پسورد را چک کرده و در صورت تائید، اجازه ورود به کاربر داده شود. در صورتی که نام کاربری یا پسورد در فایل وجود نداشت و تطابق داده نشد، می توان از کاربر خواست دوباره اقدام کند.

چند نکته در این بخش:

- برای تشخیص اصالت کاربر باید به فایل مربوط به پروفایل کاربران مراجعه کنید. در این فایل مشخصات کاربران وجود دارد. باید ابتدا نام کاربری وارد شده توسط کاربر را با نام های کاربری موجود در فایل مقایسه کنید. اگر نام کاربری وارد شده پیدا نشد، که هیچی، اجازه ورود نمی دهید... اگر نام کاربری پیدا شد، باید تازه پسورد را چک کنید. روال این کار ساده و منطقی است! قدری به آن فکر کنید.
- پیشنهاد می شود هنگام تایپ پسورد توسط کاربر، به دلایل امنیتی، بر روی مانیتور فقط ستاره چاپ شود!
 - فرد نشسته بر روی صندلی کناری (که به مانیتور ما زل زده) شاید محرم نباشد!
- نکته امتیازی (تا ۵ درصد نمره): آیا می توانید جهت افزایش امنیت، پسوردهای ذخیره شده در فایل را به نحوی درهم کنید که اگر تصادفاً شخص نامحرمی به این فایل دسترسی پیدا کرد، نتواند پسوردها را به دست آورد؟؟؟
 - برای این کار بر روی غیر قابل حدس بودن روش درهم سازی خود بسیار بسیار فکر کنید. در واقع شخص نامحرم را دست کم نگیرید! شاید روش درهم سازی شما را حدس زده و پسوردها را پیدا کند. ☹
 - با انجام این کار مبانی رمزنگاری را لمس خواهید کرد.
- در نهایت: آیا می توانید تعداد تلاش ناموفق کاربران را محدود کنید؟
 - مثلاً اگر کاربری ۵ بار پسورد خود را اشتباه زد، برای ۱۵ دقیقه امکان تلاش مجدد را از وی سلب کنید و بعد از ۱۵ دقیقه دوباره اکانت وی را آزاد کنید.

۲ - منوی اصلی برنامه

کاربر پس از طی موفقیت آمیز روال تشخیص هویت، وارد برنامه می شود و منوی اصلی شامل موارد زیر نمایش داده می شود.

۱. ثبت کالای جدید
۲. ثبت ورود کالا به انبار
۳. ثبت خروج کالا از انبار
۴. گزارش گیری ها
۵. تنظیمات حساب کاربری

۶. خروج از حساب کاربری

۷. خروج از برنامه

بدیهی است گزینه انتخابی از کاربر را می توانید به صورت عددی از وی دریافت کنید.

منوی این بخش (و بخش های دیگر) را می توانید با سلیقه خود طراحی کنید.

۲-۱: بخش ثبت کالای جدید

به این نکته فکر کنید که ابتدا باید کالاها در سیستم تعریف شوند. بنابراین کاربر سیستم باید بتواند در صورت نیاز، کالای جدید در سیستم تعریف کند. برای این منظور از این بخش استفاده کنید.

هر کالا می تواند مشخصاتی مانند زیر داشته باشد:

۱. نام کالا (مثلا تلویزیون، صندلی، کفش و ...)
۲. برند کالا (مانند نایک، سونی، باربد، پارس خزر و ...)
۳. مدل کالا (مثلا XV32، H2CT99 و ...)
۴. قیمت کالا
۵. توضیحات (در حد یکی دو خط، شامل هر توضیح اضافه مانند جنس، رنگ، توان مصرفی، اندازه و ابعاد، یا هر خصوصیت دیگر)
۶. شناسه منحصر به فرد کالا در سیستم ما (در بخش های دیگر برنامه به درد میخورد، این شناسه را باید به صورت یکتا، برنامه شما به کالا تخصیص دهد، به فرمت آن خوب فکر کنید)

نکته مهم: فقط و فقط کاربری با نام کاربری admin می تواند کالای جدید تعریف کند. مابقی کاربران اجازه ثبت کالای جدید ندارند. برای این مورد حتما راهکار مناسب را لحاظ کنید.

۲ - ۲: بخش ثبت ورود کالا به انبار

در این بخش کاربر باید بتواند ورود تعداد خاصی از یک نوع کالا را به انبار ثبت کند. شامل تعداد کالا و تاریخ ورود کالا به انبار.

به فرآیند این کار در انبارهای واقعی **خوب** فکر کنید. سپس به سلیقه و با منطق خود این بخش را تکمیل کنید.

به عنوان مثال، باید این سوالها را جلوی خود قرار دهید و برای آن راهکار پیدا کنید:

کاربر میخواهد ورود تعدادی از یک کالا به انبار را ثبت کند. خب! کدام کالا؟ آیا کاربر لیست کالاهای تعریف شده در سیستم را حفظ است؟ آیا باید لیست را ابتدا به کاربر نشان داد تا برای وی یادآوری شود؟ اگر لیست خیلی بزرگ بود، چه کار باید کرد؟ آیا می توان راهکاری برای جستجوی کالای مورد نظر برای این بخش طراحی کرد؟ مثلاً کاربر فقط حرف اول یک کالا را وارد کند و فقط لیست کالاهایی که با آن حرف شروع می شوند، برای وی آورده شوند تا وی انتخاب کند؟ کاربر حرف اول کدام خصوصیت محصول را می خواهد سرچ کند؟ حرف اول مدل یا حرف اول نام کالا؟ اگر یک کاربر با تجربه و حرفه ای برای کار با نرم افزار استخدام کرده باشیم، شاید او بتواند به سرعت شناسه های کالاها را حفظ کرده و از آن استفاده کند... و سوالهای مهمی از این دست... خوب به این گونه سوالها فکر کنید.

- نکته عملی: یکی از مسائلی که برنامه نویس ها هنگام نوشتن برنامه ها باید به آن خوب و کافی فکر کنند، در نظر گرفتن شرایط کاربر نهایی برنامه است. اینکه کاربر نهایی چگونه فکر میکند؟ اینکه کاربر نهایی چگونه راحت تر است؟ اینکه کاربر نهایی چگونه با برنامه ما بهتر ارتباط برقرار می کند؟ برنامه نویس باید به همه این سوالها خوب فکر کرده و برای هر سوال، بهترین پاسخ و راه حل را انتخاب کرده و پیاده سازی کند.
- یک پیشنهاد برای پروژه حاضر: ابتدا فرض کنید کاربران شناسه منحصر به فرد کالاها را حفظ هستند و موقع ورود کالا به انبار، از آن شناسه استفاده می کنند، همین... با این فرض روال تکمیل پروژه شما سریع تر طی می شود. بعد از تکمیل مقدماتی سایر بخش ها، می توانید با فکر آزادتر به این بخش برگردید و کمی هم به فکر کاربر بیچاره باشید!

۲ - ۳: بخش ثبت خروج کالا از انبار

این بخش در واقع به کاربر کمک می کند تا خروج تعدادی از یک کالای خاص را ثبت کند.

باز هم باید به این بخش خوب فکر کنید.

سوال های مهم این بخش:

- تمام سوالهای مربوط به ثبت ورود کالا به انبار پیرامون پیدا کردن نام و مدل کالا توسط کاربر اینجا نیز صادق هستند.
- اگر کاربر بخواهد خروج ۵ عدد از یک کالا را در سیستم ثبت کند، اما در انبار کلاً ۴ عدد از آن کالا وجود داشته باشد، چه اتفاقی باید رخ دهد؟
- اصلاً ما تعداد کل یک کالای خاص را کجا ذخیره کرده ایم؟ (به فرآیند ثبت ورود کالا خوب فکر کرده اید؟)

یک پیشنهاد خوب: برای تراکنش ها (ورود/خروج) یک ساختمان یکسان تعریف کنید. مثلاً ساختمانی با فیلدهای پیشنهادی زیر:

۱. شناسه کالا
۲. نوع تراکنش (ورود یا خروج)
۳. تعداد کالا در این تراکنش
۴. تاریخ تراکنش
۵. کد کاربر ثبت کننده

حال این تراکنش ها را می توانید به ترتیب در یک فایل ذخیره کنید. بعد از آن باید در صورت نیاز بارها و بارها محتوای این فایل را تحلیل کنید. مثلاً برای پیدا کردن تعداد باقیمانده یک کالا (با شناسه خاص)، باید به ترتیب تمام تراکنش های مربوط به این کالا را پیدا کرده و پردازش کنید تا در نهایت تعداد آن کالا در انبار را به دست آورید.

به عنوان مثال موقع پردازش فایل، در میابیم برای یک کالا با کد X اولین تراکنش ورود ۱۰ عدد است. سپس چند روز بعد ۳ عدد دیگر به انبار اضافه شده است. در ادامه ۵ عدد از انبار خارج شده و بعد از آن ۱ عدد دیگر به انبار اضافه شده است. پس قاعدتاً باید در حال حاضر ۹ عدد از آن کالا در انبار وجود داشته باشد.

۲-۴: بخش گزارش ها

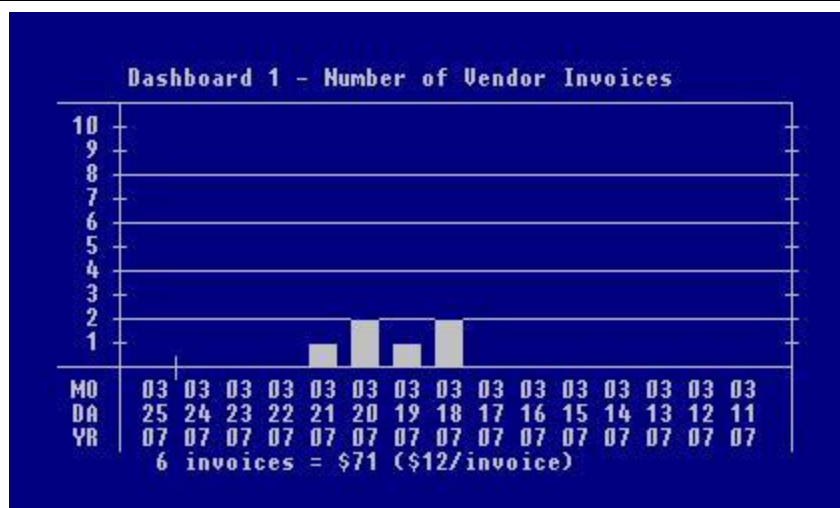
این بخش مهم ترین بخش برنامه شماسست. زیرا کاربر به دفعات زیاد نیاز به گزارش گیری دارد و به این بخش مراجعه می کند. در این بخش باید گزارش های متنوعی به کاربر ارائه شود. مثلا:

۱. تعداد موجود هر کالا در انبار.
۲. لیست کالاهایی که از آنها بیشتر از یک مقدار خاص (مثلا ۱۰۰ عدد) در انبار داریم.
۳. لیست کالاهایی که از آنها کمتر از یک مقدار خاص (مثلا ۳ عدد) در انبار داریم.
۴. ارزش ریالی کل کالاهای انبار.
- در این برنامه فرض می کنیم یک کالا را همواره به یک قیمت ثابت می خریم. (کشور بدون تورم!!!!)
۵. ارزش ریالی یک کالای خاص در انبار.
۶. تعداد ورود یک کالای خاص در یک بازه تاریخی خاص
- مثلا از اول شهریور ۱۳۹۹ تا ۲۵ خرداد ۱۴۰۰
۷. تعداد خروج یک کالا در یک بازه تاریخی خاص
۸. گزارش ورود و خروج یک کالای خاص در یک بازه تاریخی با جزئیات.
- مثلا از یک کالا در یک بازه زمانی، چه تراکنش هایی در سیستم موجود است؟ سطر به سطر...
۹. لیست تراکنش های ثبت شده توسط یک کاربر خاص. (مثلا به یکی از کاربرهای برنامه شک داریم)
- این گزارش فقط و فقط برای admin فعال است.
۱۰. لیست کاربران سیستم و تاریخ آخرین فعالیت هر کاربر.
- این گزارش فقط و فقط برای admin فعال است.
۱۱. و گزارش های متنوع دیگر... کمی فکر کنید! و با سلیقه خود گزارش اضافه کنید. قدرت بخش گزارش گیری برنامه شما، در نمره شما تاثیر زیادی دارد.
- ۱۰ گزارش عنوان شده در فوق برای برنامه اجباری است. از این به بعد نمره اضافه خواهد داشت.
- مثلا چه تعداد تلویزیون در انبار داریم؟ (مجموع مدل ها و برندهای مختلف)
- یا مثلا در یک سال خاص، چقدر برای انبار خرید کرده ایم؟
- و گزارش های مفیدی که به ذهن شما خواهد رسید...

با فکر کردن به بخش گزارش گیری، شاید مجبور شوید بخش های پیشین برنامه را تغییر دهید. به عنوان مثال شاید بعد از این مرحله ایده بهتری برای تخصیص شناسه یکتا به کالاها به ذهنتان خطور کند. مثلاً شاید بهتر باشد شناسه همه تلویزیون ها را با حرف T شروع کنیم. یا اصلاحات و بهبودهایی از این دست...

چند نکته در مورد این بخش:

- **نکته مهم:** هنگام واکنشی اطلاعات تراکنش ها از فایل مربوطه باید هر تراکنش را در یک گره لیست پیوندی قرار دهید و کلیه عملیات این بخش (شامل کلیه جستجوها) در این لیست پیوندی صورت گیرد.
- پیشنهاد می شود هنگام نیاز به گزارش، ابتدا کلیه محتویات فایل را در یک لیست پیوندی درج کنید. در واقع در ابتدا باید کل داده های تراکنش ها را از دیسک به حافظه منتقل کنیم.
- اینجا فرض می کنیم که فایل ما آنچنان بزرگ نیست، پس در حافظه جا می شود. بعدها شاید بتوان این بخش را بهینه تر و کارآتر طراحی کرد.
- هنگامی که کاربر از بخش گزارشها خارج شد و به منوی اصلی برگشت (مثلا برای ثبت تراکنش جدید)، باید حتما حافظه کل لیست پیوندی را آزاد کنید.
- یعنی با هر بار ورود به بخش گزارشها، باید نسخه جدیدی از لیست پیوندی ساخته شود و موقع خروج از بخش گزارشها، حذف شود... دلیل آن را که می دانید؟؟؟
- **دقت کنید** در این پروژه اجازه استفاده از آرایه ای از ساختمان (به هیچ نحوی) را ندارید. ☹️ برای موارد دیگر (به جز ساختمان ها) می توانید از آرایه استفاده کنید.
- استفاده از آرایه به جای لیست پیوندی موجب کسر ۳۰ درصد از کل نمره کسب شده در پروژه می شود.
- **مهم:** تعداد و تنوع این گزارش ها بر عهده شماست. لطفا خلاق باشید و گزارش های بدیعی ارائه دهید.
- نحوه دسته بندی این گزارش ها نیز بر عهده شماست. شاید بتوانید با یک دسته بندی منطقی، سر و شکل بهتری به بخش گزارش گیری دهید!
- در این بخش نیز خلاق باشید. مثلا شاید بتوانید گزارشات را دسته بندی کرده و در زیر گروههای مشخصی قرار دهید.
- گزارش ها را می توانید به صورت متنی ارائه دهید.
- اگر قدری وقت اضافه در انتهای پروژه داشتید، (و مابقی بخش های پروژه را تکمیل کرده بودید)، کمی به شکل کردن گزارش ها فکر کنید. به عنوان مثال گزارش (شبه نمودار) زیر را تحلیل کنید که کاملا با کاراکترهای خاص پیاده سازی شده است:



دقت کنید گزارش فوق فقط به عنوان نمونه از یک نرم افزار قدیمی گرفته شده است و فقط جهت دادن ایده به شما اینجا ذکر شده است. پیاده سازی چنین نموداری وقت و حوصله و دقت کافی نیاز دارد.

- حتما چنین مبحثی را به انتهای کار خود مוקول کنید و فقط در صورت اضافه آمدن زمان روی آن انرژی صرف کنید.

۲- ۴: بخش تنظیمات

در این بخش کاربر برنامه باید بتواند مشخصات پروفایل خود، از جمله پسورد، را تغییر دهد. با سلیقه خود این بخش را طراحی کنید.

- بدیهیست کاربران نمی توانند نام کاربری خود را تغییر دهند. مابقی فیلدها را می توانند اصلاح کنند.

۲- ۵: خروج از ناحیه کاربری

با انتخاب این گزینه برنامه از حساب کاربری یک کاربر خارج شده و دوباره به صفحه لاگین برمیگردد.

- دقت کنید برنامه ما چند کاربره است. پس با خارج شدن یک کاربر از حساب کاربری خود، برنامه بسته نمی شود... شاید کاربر دیگری لاگین کند.

نکات تکمیلی پیاده سازی

- برای پیاده سازی این برنامه حداقل به سه فایل نیاز دارید، یکی برای پروفایل کاربر، یکی مشخصات کالاهای موجود در انبار، و یکی هم برای ثبت تراکنش های ورود و خروج.
 - لطفاً فرمت مناسبی برای این فایل ها انتخاب کنید.
 - فایل های ذکر شده فقط یک پیشنهاد است. شما بر اساس دیدگاه خود ممکن است فایل های دیگری نیز استفاده کنید.
 - یک پیشنهاد: برای ثبت تراکنش های انبار (ورود/خروج) ها، می توانید از یک ساختمان استفاده کنید و تمام تراکنش ها را در یک فایل ثبت کنید. حال هر موقع نیاز بود، این فایل را پردازش کنید.
- برای پیاده سازی داده ها حتماً از ساختمان استفاده کنید.
 - مثلاً یک ساختمان برای کاربران با فیلدهای مناسب... یک ساختمان برای کالاها... یک ساختمان برای تراکنش ها.
- همانطور که پیش تر نیز اشاره شد، در این پروژه اجازه تعریف آرایه ای از ساختمان ندارید. در صورت نیاز به چنین مفهومی باید از تخصیص پویای حافظه (مثلاً تابع `malloc()`) و لیست پیوندی استفاده کنید.
 - استفاده از آرایه ای از ساختمان (به جای لیست پیوندی)، موجب کسر ۳۰ درصد از نمره کل پروژه خواهد شد.
 - پس در صورت نیاز می توانید ابتدا نسخه آرایه ای برنامه خود را پیاده سازی کنید و سپس آن را به صورت لیست پیوندی درآورید (اگر نمیخواهید ۳۰ درصد از نمره را از دست دهید)
- هر کجا از اجرای برنامه که کاربر باید چند گزینه انتخاب انجام دهد، ابتدا گزینه های موجود را به نحو مطلوبی برای کاربر نشان دهید، سپس با ورود یک کاراکتر گزینه وی را دریافت کنید و بر اساس کاراکتر وارد شده اجرا را دنبال کنید.
- پیشنهاد میشود از ابتدا به فکر پیاده سازی همه قابلیت های برنامه نباشید. بلکه کم کم برنامه را بسط دهید تا به نتیجه مطلوب برسید. عجله نکنید، به موقع ایده هایی برای بخش های مختلف برنامه به ذهنتان خواهد رسید.
- پیاده سازی پروژه خود را به روزهای پایانی موکول نکنید. زمانبندی برای انجام پروژه ها از نکات کلیدی محسوب می شود. سعی کنید آن را خوب یاد گرفته و انجام دهید.
- حتماً مرحله به مرحله از صحت کد خود مطمئن شوید. خطایابی را هیچگاه برای انتهای کار رها نکنید!
- در تمام برنامه باید پیام های مناسب برای کاربر نمایش داده شود. در این راه خساست به خرج ندهید!
- تا حد امکان برنامه شما باید مراقب کلیه حالت های خاص بازی باشد.
 - مثلاً شیطننت کاربر در وارد کردن کاراکتر غیر مجاز در منوها، یا مقادیر داده ها، یا موارد بی شمار دیگر...
- در طول پیاده سازی برنامه خود، باید حتماً برای هر کار خاصی یک تابع جداگانه بنویسید. این مورد اجباری است!

- تا جایی که می توانید برنامه خود را به توابع متعدد تقسیم کنید...
- پروژه خود را در مراحل مختلف آرشیو کنید تا به راحتی در صورت نیاز به نسخه های قبلی بازگردید.
- نکته امتیازی (۵٪ نمره اضافه): شما به وسیله Git می توانید در هر مرحله کد خود و تغییرات در آن را به صورت اصولی و مستند ذخیره کنید و نیازی به آرشیو گرفتن به صورت دستی نیست. در صورتی که از این امکان به صورت صحیح استفاده کنید تا سقف ۵٪ نمره امتیازی دریافت خواهید کرد. برای این منظور ابتدا به طور کامل و مبسوط در مورد Git تحقیق کنید. برای این منظور می توانید از دو سایت معروف Gitlab و یا Github استفاده کنید. فقط مواظب باشید در این سایتها از مخزن های خصوصی (و نه عمومی) استفاده کنید.
- لطفا کد خود را کاملاً خوانا و تمیز ارائه دهید. برای مثال از بخش های ۱ و ۲ از [این لینک](#) راهنمایی بگیرید.
- هر گونه نوآوری و خلاقیت برای جذاب تر شدن این برنامه نمره امتیازی خواهد داشت. این نکات را هنگام تحویل پروژه به تدریساران یادآوری کنید.
- از افزودن قابلیت های دیگر به برنامه استقبال می شود. این نکات را موقع ارائه پروژه به تدریساران گوشزد کنید.
- سعی کنید برنامه خود را هرچه کاربر-پسندتر پیاده سازی کنید.
- برای این منظور حتماً از افراد نزدیک خود بخواهید مراحل اجرای برنامه شما را قضاوت کنند. هر موردی که از دید آنها نامفهوم یا ناملموس بود را باید حتماً اصلاح کنید.
- بگذارید کاربران تازه کار با برنامه شما کار کنند. با این کار ایرادهای احتمالی را به راحتی کشف خواهید کرد و اصلاحات خوبی روی برنامه اعمال خواهید کرد. از این کار به هیچ وجه نترسید...
- برنامه خود را بارها و بارها تست کنید.
- برای اینکار حتما داده های مختلف و متنوعی در فایلها درج کنید و برنامه را بارها تست کنید. مثلاً چندین نام کاربری در فایل پروفایل داشته باشید. انواع و اقسام تراکنش ها را در فایل های مربوطه درج کنید و سعی کنید همه جور گزارش را تولید کنید.
- از طرفی سعی کنید کاربری با برنامه خود کلنجر بروید. مثلاً نام کاربری تکراری ثبت نام کنید. سعی کنید تاریخ ها و اعداد را اشتباه وارد کنید و کارهایی از این دست. در واقع باید خودتان اولین منتقد و ایرادگیر برنامه خودتان باشید.
- مواردی که در پروژه به آنها نیاز دارید و تا به حال به آنها برخورد نکرده اید (مثلاً در کلاس مطرح نشده اند) را سعی کنید با جستجو در گوگل بیابید. سعی کنید خودتان را به این جستجوها به شدت عادت دهید. ادامه عمر شما به این جستجوها خواهد گذشت.
- برای این منظور حتماً در دوران تحصیل وقت کافی به تقویت زبان انگلیسی خود اختصاص دهید.
- پروژه باید به صورت یک نفری انجام شود. کشف هر گونه مشابهت یا تقلب موجب تجدید ترم در این درس خواهد شد.

- در طول پیاده سازی پروژه می توانید با یکدیگر همفکری کنید و جلساتی را با هم برگزار کنید، اما تمام بخش های پیاده سازی این پروژه (شامل طراحی، تایپ، خطایابی، توسعه و ارائه) باید توسط خود دانشجو صورت گیرد. به عبارت دیگر تایپ حتی یک خط از برنامه توسط دیگری تقلب محسوب می شود. کشف کپی-کاری در اینجا ساده تر از آن چیزی است که فکر میکنید!
- سعی کنید جلساتتان برای به اشتراک گذاشتن ایده ها باشد، نه به اشتراک گذاشتن کدها.

تحويل و ارائه پروژه

آخرین نسخه پروژه خود را در باید در سایت کوئرا آپلود کنید. (شامل همه فایلها!!!!!!)

- بعد از اتمام مهلت سابمیت پروژه در سایت کوئرا، فرآیند تحويل پروژه شروع خواهد شد. برای این منظور تدریس‌یاران تک تک پروژه ها را از طریق جلسات حضوری تحويل خواهند گرفت. برای این منظور ابتدا یک زمانبندی دقیق توسط تدریس‌یاران به شما ارائه می شود، سپس هر دانشجو باید پروژه خود را بر اساس زمانبندی ارائه شده برای تدریس‌یارها به طور کامل ارائه داده و به پرسش ها و درخواست‌های آنها (مبنی بر اعمال تغییرات در برنامه) پاسخ دهد.
- مهلت سابمیت نهایی پروژه در سایت کوئرا پایان روز ۱۴۰۱/۱۱/۰۷ است. لطفا فعالیت های خود را به طور دقیق زمانبندی کنید.
- فرآیند تحويل حضوری پروژه احتمالا در روزهای ۱۰ و ۱۱ بهمن خواهد بود.

ورود شما را به دنیای برنامه نویسی تبریک می گویم.

در راهی که در ابتدایش قرار دارید، موفق باشید.

نستوه

