



به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



درس مهندسی نرم افزار ۲، نیمسال دوم سال تحصیلی ۱۴۰۱-۱۴۰۲

تمرین سوم

توضیحات تکمیلی:

- مباحث این تمرین از فصل هشتم تا سیزدهم کتاب پرسمان و اسلاید های مهندسی نیازمندی ها، مفاهیم طراحی، باز آرای و معماری نرم افزار می باشد.
- مهلت تحویل تمرین ۲۵ اردیبهشت در نظر گرفته شده و قابل تمدید نمی باشد.
- پاسخ به تمرین ها به صورت گروهی می باشد و همه اعضای گروه موظفند در آن مشارکت داشته باشند.
- در صورت برخورد با پاسخ های مشابه بیش از حد بین گروه های مختلف، نمره کسر خواهد شد.
- ارسال فایل مربوطه در سامانه کورسز توسط سر گروه (نام ذکر شده در گوگل شیت گروه بندی های درس) کافی می باشد و نیازی به بار گذاری توسط تمام اعضای گروه نیست.
- نظم و خوانایی تمرین از اهمیت بالایی برخوردار می باشد.
- خواهش می شود تمرین خود را در قالب یک فایل PDF با نام "SE2_HW[number]_[Group Number]" مانند: "SE2_HW3_6.pdf" در مهلت یاد شده در سایت درس بار گذاری فرمایید.
- پرسش های خود درباره این تمرین را می توانید از راه ایمیل se2springta@gmail.com بیان کنید.

سوالات تئوری

(۱)

الف) پس از نصب برنامه و آشنایی با آن، مهم ترین امکانات، ویژگی ها، و نیازمندی های برنامه را در قالب عبارت های کوتاه یا سناریو بیان کنید.

ب) هم چنین بعد از کار با برنامه و مطالعه مستندات (مطالعه ی مولفه های کلی، فایل ها و پوشه های پروژه) ساختار معماری استفاده شده را شرح دهید و برای پاسخ خود دلیل بیاورید.

۲) چرا نمی توان بلافاصله پس از به دست آوردن مدل نیازمندی ها اقدام به نوشتن کد کرد؟ چه مشکلاتی را ممکن است بوجود بیاورد و راه درست چیست؟

(۳)

الف) دو نوع متفاوت از حالت (state) وجود دارد که مدل های رفتاری (behavioral models) می توانند نمایش دهند. آن ها را توضیح دهید.

ب) بیان کنید که تفاوت دیاگرام دنباله ای (sequence diagram) با دیاگرام حالت (state diagram) چیست. شباهت هایشان چیست؟

(۴)

الف) آیا ما با نوشتن برنامه در حال طراحی نرم افزار (software design) هستیم؟ اگر خیر، طراحی نرم افزار چیست و چه چیزی آن را از کد نوشتن متفاوت میکند؟

ب) چرا طراحی برای بخش تست و بررسی اهمیت دارد؟

ج) با ذکر مثال ارتباط میان هم بستگی (Coupling) و portability نرم افزار را بیان کنید.

۵) مواردی مانند استایل معماری، الگو معماری و چارچوب ها در معماری نرم افزار مورد توجه اند. درمورد آن ها مطالعه کنید و تفاوت هایشان در مفهوم و کاربرد را ذکر کنید.

سوالات پیاده سازی

در این بخش قرار است تا یک پروژه متن باز را مورد بررسی قرار دهیم و کیفیت آن را مورد ارزیابی قرار دهیم. پروژه مورد نظر از [لینک](#) قابل دسترس هست.

این پروژه، یک پروژه متن باز است که با چارچوب جنگو پیاده سازی شده است. در این پروژه، یک سیستم سفارش غذای آنلاین پیاده سازی شده است.

مراحل نصب پروژه:

(۱) ابتدا اقدام به ساختن یک virtual env نمایید.

(۲) سپس دستورات زیر را اجرا کنید:

```
pip3 install -r requirements.txt
python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```

پس از انجام مراحل فوق، پروژه بر روی پورت 8000 بالا می آید.

ابزار SonarQube:

ابزار SonarQube یک ابزار متن باز و رایگان برای بررسی کیفیت متن برنامه ها است. این برنامه ویژگی هایی دارد که آن را با سایر ابزارهای این حوزه متمایز می کند، در این تمرین با این ابزار و ویژگی های آن آشنا خواهید شد.

راه اندازی:

برای راه اندازی این ابزار روش های متفاوتی وجود دارد. از جمله این روش ها می توان به نصب آن از طریق داکر و استفاده از SonarScanner اشاره کرد. در ادامه مراحل راه اندازی نوشته شده است. هر مرحله را اجرا کنید و خواسته های سوال را پاسخ دهید.

مرحله اول)

اجرای SonarQube با استفاده از داکر: اولین قدم اجرای SonarQube است، برای این کار نیاز است تا دستور داکر را وارد کنید. (برای دریافت ایمیج از دستور روبرو استفاده کنید: docker pull sonarqube)

(۱) درباره معماری ابزار SonarQube تحقیق کنید و در گزارش خود بیاورید.

(۲) دستور نصب و راه اندازی SonarQube در داکر را بنویسید.

مرحله دوم)

اطمینان از صحت نصب: دومین کار اجرای دستور docker ps است، این دستور فرآیندهایی که با داکر اجرا شده اند را نشان می دهد. در صورت موفقیت آمیز بودن دستور قبل باید برنامه SonarQube در این مرحله نشان داده شود.

(۳) خروجی این دستور را نشان دهید.

مرحله سوم)

نمایش گرافیکی: SonarQube دارای رابط گرافیکی است. به صورت پیش فرض این رابط در آدرس <http://localhost:9000> است (ممکن است پورت را در مرحله اول عوض کرده باشید). این آدرس را در مرورگر خود وارد کنید. به صورت پیش فرض شناسه کاربری و رمز عبور، admin است.

۴) تصویر صفحه‌ی اول SonarQube را نشان دهید.

مرحله چهارم) ساختن پروژه جدید: حال باید پروژه جدیدی بسازید، با انتخاب آیکون ساخت پروژه جدید، در صفحه جدید نام و کلید پروژه را تعیین می‌کنید. شما می‌توانید هر نام دلخواهی در این قسمت وارد کنید.

مرحله پنجم) انتخاب نحوه‌ی آنالیز: در این مرحله باید نحوه‌ی آنالیز کد را مشخص کنید که کافی است Locally را انتخاب کنید.

مرحله ششم) ساخت توکن: در این مرحله باید نامی برای توکن انتخاب کنید، توکن ایجاد شده در مراحل بعد مورد نیاز است.

مرحله هفتم) آنالیز کردن کد: در این قسمت باید زبان برنامه‌ی خود را مشخص کنید. شما باید گزینه Other را انتخاب کنید. سپس باید سیستم عامل خود را انتخاب کنید. برای این قسمت نیاز است که SonarScanner را نصب کنید. پس از نصب، دستوری که به شما داده است را باید در پوشه‌ی پروژه اجرا کنید. (برای نصب SonarScanner می‌توانید از [این](#) داکيومنت استفاده کنید)

۵) نحوه و دلیل نصب SonarScanner چیست؟

مرحله هشتم) صفحه Overall Code: در صفحه‌ی Overall Code آمار کلی‌ای از کیفیت برنامه می‌دهد.

۶) موارد زیر را تعریف کنید و تعداد آن‌ها را در برنامه‌ی داده شده گزارش نمایید. در صورت وجود، نمونه‌ای از هر مورد را بیان کنید. همچنین موردی که بیان کردید، چرا در این دسته قرار می‌گیرد؟

- Bugs
- Vulnerabilities
- Security Hotspots

۷) Code Smell چیست و به چه علتی رخ می‌دهد؟ تعداد آن را در برنامه داده شده بیابید و سه نمونه از آن را در گزارش خود توضیح دهید و آن را اصلاح (Refactor) کنید. (نمونه‌هایی که بیان می‌کنید نباید از کدهای HTML پروژه باشند)

۸) تفاوت Security Hotspots و Vulnerability چیست؟

۹) Technical Debt به چه معناست؟ مقدار آن را در برنامه داده شده مشخص نمایید.