به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

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

تمرین سوم

توضیحات تکمیلی:

- مباحث این تمرین از فصل هشتم تا سیزدهم کتاب پرسمان و اسلاید های مهندسی نیازمندی ها، مفاهیم طراحی، باز آرایی و معماری نرم‌افزار می‌باشد.
- مهلت تحویل تمرین **۳ خرداد** در نظر گرفته شده و قابل تمدید نمی‌باشد.
- پاسخ به تمرین‌ها به صورت **گروهی** می‌باشد 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 می‌باشد و نیازی به بار گذاری توسط تمام اعضای گروه نیست.
- نظم و خوانایی تمرین از اهمیت بالایی برخوردار می‌باشد.
- خواهش می‌شود 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.
- پرسش‌های خود درباره این تمرین را می‌توانید از راه ایمیل se2springta@gmail.com بیان کنید.

۱)

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

کاربر(مشتری) باید بتواند در سایت ثبت نام کند.
کاربر باید بتواند در حساب کاربری خود وارد شود.
کاربر باید بتواند لیست غذاها را مشاهده کند.
کاربر باید بتواند غذای مورد نظر خود را ثبت سفارش کند.
و ...
سایت باید امکان نگهداری اطلاعات کاربران را داشته باشد.
سایت باید امکان نگهداری اطلاعات سفارشات را داشته باشد.
سایت باید پایداری و قابلیت اطمینان کافی داشته باشد.
سایت باید توانایی مدیریت کاربران و سفارشات متعدد و همزمان را داشته باشد.
و ...
هم‌چنین درمورد ادمین سایت هم می‌توان نیازمندی‌ها را بیان کرد.

ب) هم‌چنین بعد از کار با برنامه و مطالعه مستندات(مطالعه‌ی مولفه‌های کلی، فایل‌ها و پوشه‌های پروژه) ساختار معماری استفاده شده را شرح دهید و برای پاسخ خود دلیل بیاورید.

برنامه در چارچوب django نوشته شده است. نحوه عملکرد پروژه بدین صورت است که بعد از اینکه کاربر یک درخواست را به سرور ارسال میکند، سرور با استفاده از توزیع کننده‌ی URL درخواست را به ویو مرتبط با آن ارسال می‌کند . در نهایت ویو به کمک لایه‌ی مدل اطلاعات مورد نیاز را استخراج کرده و سپس با استفاده از لایه‌ی تمپلیت آن را به کاربر نشان می‌دهد.
با توجه به این ساختار، پروژه دارای معماری MTV می‌باشد که مشابه معماری MVC است با این تفاوت که بخش تمپلیت به ویو و بخش ویو به کنترلر تبدیل شده است.

ساختار و مولفه‌های فایل‌های اولیه‌ی جنگو به صورت زیر می‌باشد که به اختصار هر کدام را توضیح می‌دهیم. (مؤلفه بخش‌هایی از سیستم است که یک مجموعه Functionality مشابه دارند. بسیاری از مؤلفه‌های زیر که برای کارکرد سرور ما مورد نیاز است، توسط جنگو فراهم شده است.)

- __init__.py: این فایل پوشه‌ای که شاملش می‌شود را به یک پکیج پایتون تبدیل می‌کند که در این صورت می‌توان به اطلاعات فایل‌هایی که درون آن پوشه قرار دارند دسترسی داشت (از طریق ایمپورت کردن).
- manage.py: این فایل پل ارتباطی میان میان توسعه‌دهنده و پروژه‌ی جنگوی او می‌باشد. در واقع ابزار خط فرمانی می‌باشد که جنگو در اختیار توسعه دهندگان قرار داده‌است.
- settings.py: بسیاری از تنظیمات پروژه مانند تنظیمات مربوط به احراز هویت کاربران، اپ‌های نصب شده‌ی برنامه، تنظیمات مربوط به سیستم ایمیل، برخی تنظیمات مربوط به پایگاه‌داده و تنظیمات دیگر در این فایل قرار دارند.
- urls.py: در این فایل آدرس‌هایی که قابل دسترسی خواهند بود، قرار می‌گیرند.
- asgi.py و wsgi.py: این دو فایل مربوط به وب‌سرورهایی هستند که جنگو می‌تواند منطبق بر آن‌ها کار کند.

همچنین مولفه‌های دیگری دارد مانند:

- views: که مجموعه Handlerهای تمام درخواست‌ها در این مؤلفه قرار می‌گیرند.
- Admin: این فایل، در کنار ماژول‌های ادمین جنگو (django.contrib.admin) سایت ادمین جنگو را می‌سازند. اپ ادمین جنگو ابزاری است جهت ساخت، بروزرسانی، مشاهده و حذف داده‌ها و مدیریت سایت‌ها
- پایگاه داده که sqlLite می‌باشد.
- Templates که frontend پروژه را شامل می‌شود (مانند صفحات HTML پروژه).
- Media که تصاویر مورد استفاده در پروژه در آن قرار دارد.
- migrations متناظر با پوشه foodapp/migrations که فایل‌های migration برنامه جهت آماده‌سازی پایگاه داده در آن قرار دارند.
- Models: مدل‌ها در جنگو برای ذخیره‌ی اشیاء در پایگاه داده مورد استفاده قرار می‌گیرند. مدلهای جنگو کلاس‌های پایتون هستند که توسط جنگو یکسری ویژگی‌های اضافه به آن‌ها اضافه شده‌است.
- Forms: با استفاده از فرم‌ها می‌توان تبادل اطلاعات از سمت کاربر به سمت سرور انجام داد.

۲) چرا نمی‌توان بلافاصله پس از به‌دست آوردن مدل نیازمندی‌ها اقدام به نوشتن کد کرد؟ چه مشکلاتی را ممکن است بوجود بیاورد و راه درست چیست؟

یکی از اصول توسعه نرم‌افزار، مشخص کردن دقیق نیازمندی‌ها است. اگر بلافاصله بعد از به‌دست آوردن نیازمندی‌ها اقدام به نوشتن کد کنیم، ممکن است یک معماری خوب طراحی نکرده و باعث ایجاد باگ‌های مختلف در آینده شود و به طور کلی از مسیر و هدف اصلی دور شویم و ممکن است نیازمندی‌ها در آینده اضافه شوند که توسعه سیستم را ممکن است دچار مشکل کنند. پس بهتر است در ابتدا، نیازمندی‌ها به طور دقیق مشخص شده و پس از جلسات مختلف بین برنامه‌نویسان و صاحب محصول و نماینده مشتری، یک توافق بر روی نیازمندی‌ها صورت گرفته و در جلسات مختلف نیازمندی‌ها مورد تحلیل قرار بگیرند.

۳)

الف) دو نوع متفاوت از حالت(state) وجود دارد که مدل‌های رفتاری(behavioral models) می‌توانند نمایش دهند. آن‌ها را توضیح دهید.

- حالت درونی: حالت‌های درونی یک سیستم، حالت‌هایی هستند که به صورت مستقیم قابل مشاهده نیستند. درواقع حالت هر کلاس، وقتی سیستم درحال انجام یک رخداد است.
- حالت بیرونی: حالت‌هایی که به صورت مستقیم قابل مشاهده هستند. حالت‌های بیرونی شامل رفتارهای خارجی و ویژگی‌های یک سیستم می‌شود.

ب) بیان کنید که تفاوت دیاگرام دنباله‌ای(sequence diagram) با دیاگرام حالت(state diagram) چیست. شباهت‌هایشان چیست؟

شباهت آن‌ها این است که هر دو دیاگرام نشان دهنده رفتار سیستم هستند و شامل حالت‌های مختلف هستند که در اثر بروز رخدادهای مختلف ایجاد می‌شوند.

تفاوت آن‌ها در این است که در دیاگرام دنباله‌ای توالی زمانی نمایش داده می‌شود درحالی که در دیاگرام حالت توالی زمان ثبت نمی‌شود.

۴)

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

خیر. در طراحی نرم‌افزار، معماری، اجزاء، ساختار و عملکرد سیستم و نحوه ارتباط بین آن‌ها مشخص می‌شود. درحالی که مرحله نوشتن کد، شامل پیاده‌سازی طراحی‌های انجام شده با کمک زبان‌های برنامه بنویسی است.

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

طراحی بخش تست و بررسی، سبب کاهش هزینه و زمان تست، بهبود کیفیت و عملکرد نرم‌افزار، تسهیل تست خودکار، پوشش تست بهبود یافته و تشخیص خطاها می‌شود.

ج) با ذکر مثال ارتباط میان هم‌بستگی (Coupling) و portability نرم‌افزار را بیان کنید.

همبستگی بیانگر ارتباط میان اجزاء مختلف نرم‌افزار و نشان دهنده میزان ارتباط آن‌ها با یکدیگر است. درحالی که قابلیت حمل، بیانگر میزان استفاده یک نرم‌افزار در محیط‌های مختلف است. هر چقدر میزان همبستگی زیاد باشد، قابلیت حمل کم می‌شود و درصورتی که میزان همبستگی کم باشد، قابلیت حمل نیز افزایش پیدا می‌کند.

۵) مواردی مانند استایل معماری، الگو معماری و چارچوب‌ها در معماری نرم‌افزار مورد توجه‌اند. درمورد آن‌ها مطالعه کنید و تفاوت‌هایشان در مفهوم و کاربرد را ذکر کنید.

استایل معماری: یک طرح اولیه برای ساختار یک سیستم نرم افزاری است. اجزا و مولفه‌های سیستم، روابط آن‌ها و تعاملات آن‌ها را تعریف می‌کند. استایل‌های معماری اغلب بر اساس روشی که سیستم را سازمان‌دهی می‌کنند، نام‌گذاری می‌شوند، مانند یکپارچه، لایه ای، میکروسرویس، مشتری-سرور، و استایل معماری سرویس‌گرا. هر استایل روش خاصی را برای سازماندهی اجزا، تعریف تعاملات آن‌ها و دستیابی به معیارهای مورد انتظار مانند مقیاس پذیری، اصلاح پذیری یا امنیت ارائه می دهد.

الگوی معماری: هر الگو یک راه حل خاص برای یک مشکل رایج در معماری نرم افزار ارائه می دهد. در واقع یک راه حل قابل استفاده مجدد برای یک مشکل طراحی تکرارشونده در یک سبک معماری خاص است و می توان از آنها برای بهبود طراحی و پیاده سازی سیستم های نرم افزاری استفاده کرد. مفاهیمی مانند Model-View-Controller (MVC), Publisher-SubscriberPipes و Filter pattern جز الگوهای معماری هستند.

چارچوب: یک چارچوب یک محیط نرم افزاری است که مجموعه ای از مؤلفه‌ها، کتابخانه‌ها و خدمات از پیش تعریف شده را ارائه می‌دهد که می تواند برای توسعه برنامه‌های نرم افزاری مورد استفاده قرار گیرد. چارچوب‌ها می‌توانند طیف وسیعی از ویژگی‌ها مانند رابط کاربری، دسترسی به داده‌ها، ارتباطات شبکه‌ای و امنیت را ارائه دهند. چارچوب ها می توانند با ایجاد زیرساخت اولیه برای توسعه نرم افزار، در زمان و کار توسعه‌دهنده صرفه‌جویی کنند. Django، Ruby و .Net نمونه چارچوب‌های پر استفاده می‌باشند.

در کل می‌توان گفت: استایل معماری ساختار و اصول کلی یک سیستم نرم افزاری را تعریف می کند. الگوهای معماری راه حل های قابل استفاده مجدد را برای مشکلات طراحی در یک استایل خاص ارائه می‌دهند، و چارچوب‌ها زیرساخت از پیش ساخته شده برای توسعه برنامه و پشتیبانی از توسعه برنامه ارائه می‌دهند.