



مقدمه

شما در فازهای قبلی این پروژه به پیاده‌سازی یک سامانه مجازی بررسی و رزرو هتل پرداختید و به صورت کامل منطق دامنه‌ی^۱ آن را پیاده‌سازی کردید. هدف از این فاز پیاده‌سازی یک واسط کاربری (گرافیکی)^۲ برای فازهای قبلی است. این پیاده‌سازی بر روی بستر وب و مبتنی بر مدل Client-server صورت می‌گیرد. برای سهولت، کتابخانه‌ای تحت عنوان **APHTTP** برای پیاده‌سازی وب‌سرور در اختیار شما قرار گرفته‌است.

۱ شرح تمرین

در این فاز شما به پیاده‌سازی یک وب‌سرور بر پایه‌ی پروتکل HTTP^۳ برای برنامه‌ی خود که در فازهای قبل پیاده‌سازی کرده‌اید می‌پردازید. مرورگر نیز نقش مشتری^۴ را ایفا می‌کند. پروتکل HTTP از تعدادی method پشتیبانی می‌کند که هدف آن‌ها توصیف عملکرد درخواستی از سرور است. دو method ارتباطی مهم در این پروتکل عبارتند از GET و POST که به ترتیب برای درخواست و ارسال اطلاعات استفاده می‌شوند. همچنین پارامترهای پرسمان^۵ و بدنه‌ی^۶ درخواست برای مشخص کردن اطلاعات ارسال شده به سوی سرویس‌دهنده مورد استفاده قرار می‌گیرند که هر کدام یک نام و مقدار دارند.

در این فاز لازم است وب‌سرور شما برخی از دستورهای را که در فازها قبلی پیاده‌سازی کرده‌اید پشتیبانی کند. همچنین باید برای هر یک از این عملیات‌ها رابط کاربری مخصوص به آن را با استفاده از زبان نشانه‌گذاری^۷ HTML پیاده‌سازی کنید.

۲ آن‌چه شما باید پیاده‌سازی کنید

در ادامه قابلیت‌هایی که برنامه‌ی شما باید داشته باشد به تفصیل توضیح داده خواهد شد. شما باید بر اساس هر یک از این قابلیت‌ها یک صفحه برای واسط کاربری (گرافیکی) آن پیاده‌سازی کنید و با استفاده از کدی که در فازهای قبل زده‌اید، منطق آن را کنترل نمایید.

دقت کنید که در پاسخ به هر درخواست از طرف کاربران، باید پیام مناسب که حداقل شامل موفق یا ناموفق بودن اجرای آن درخواست است، به کاربر نشان داده شود.

۱.۲ صفحه‌ی ثبت نام

در این صفحه افراد می‌توانند در فروشگاه ثبت نام کنند.

در صفحه‌ی ثبت نام اطلاعات زیر از کاربر گرفته می‌شود:

○ نام کاربری

¹ domain logic

² user interface

³ Hypertext Transfer Protocol

⁴ client

⁵ query parameters

⁶ body

⁷ markup language

- رمز عبور
- تکرار رمز عبور
- ایمیل

اگر نام کاربری در سیستم موجود باشد باید متناسب با آن پیغام مناسبی در صفحه‌ی مرورگر به کاربر نشان داده شود.

۲.۲ صفحه‌ی ورود

کاربر در این صفحه می‌تواند با وارد کردن نام کاربری و گذرواژه‌ی خود وارد سیستم شود.

هر درخواستی که از سمت کلاینت به سرور ارسال می‌شود، با توجه به اینکه کدام کاربر آن را ارسال کرده است، جواب متفاوتی به همراه دارد. برای همین نیاز است تا مشخص شود که درخواست ارسال شده متعلق به چه کاربری است. برای انجام این کار پس از وارد شدن نام کاربری و گذرواژه توسط کاربر و ارسال آن به سرور، در صورتی که این اطلاعات درست باشند، سرور شناسه‌ی یکتای مربوط به کاربر را با عنوان sessionId تولید کرده و آن را به مشتری ارسال می‌کند. از این پس سرور با بررسی sessionId هر درخواست و مشخص کردن نام کاربری متناظر با آن، متوجه می‌شود که این درخواست متعلق به کدام کاربر است و با توجه به آن به درخواست رسیدگی می‌کند.

به این ترتیب برای هر درخواستی که به برنامه‌ی شما می‌آید، ابتدا باید وجود sessionId را بررسی کنید و سپس در صورت وجود آن در لیست sessionId های ثبت شده، ابتدا کاربر را وارد برنامه کنید، درخواستش را اجرا کنید، و سپس آن کاربر را از سامانه خارج کنید، تا کاربران دیگر نیز بتوانند از برنامه استفاده کنند.

توجه کنید که کتابخانه‌ی APTHTTP توانایی مدیریت sessionId ها را دارد. برای اطلاعات بیشتر می‌توانید به بخش **Session** در ویکی کتابخانه مراجعه کنید.

بعد از ورود، کاربر به صفحه‌ی خانه هدایت می‌شود.

۳.۲ خروج

پس از ورود، باید یک دکمه برای خروج در همه‌ی صفحات وجود داشته باشد که به کمک آن، کاربر می‌تواند از سیستم خارج شود. پس از خروج، کاربر باید به صفحه‌ی ورود منتقل شود.

دقت کنید که خروج، تنها معادل تابع خروجی که قبلاً زده بودید نیست و باید تغییرات لازم روی sessionId ها نیز اعمال شود.

۴.۲ کاربر

صفحات قبل برای آن بود که کاربرها بتوانند به منظور استفاده از امکانات به سامانه وارد شوند. کاربران قابلیت‌هایی دارند که از این پس قابلیت‌های آن‌ها را در قالب دستورهای بیانی می‌کنیم.

اگر کاربری به سامانه وارد نشده باشد، باید خطای مناسب به او نشان داده شود.

۱.۴.۲ صفحه‌ی حساب مالی

این صفحه شامل دو بخش است. در بخش اول، اعتبار کنونی کیف پول خریدار نمایش داده می‌شود و در بخش دوم کاربر می‌تواند به مقدار دلخواه بیشتر از *، کیف پولش را شارژ کند. نتیجه‌ی درخواست کاربر باید به درستی به کاربر نمایش داده شود و در صورت موفقیت، اعتبار نمایش داده شده به روزرسانی شود. همچنین از صفحه خانه که در بخش بعد توضیح داده می‌شود، دکمه‌ای به این صفحه وجود داشته باشد.

۲.۴.۲ صفحه‌ی خانه

این صفحه شامل تمام هتل‌های موجود سایت است و کاربر با کلیک روی هر کدام از آنها به صفحه جزئیات آن هتل منتقل می‌شود.

اطلاعاتی که باید برای هر کدام از هتل‌ها در این صفحه نمایش داده شوند به شرح زیر است:

- property name : همان نام هتل است.
- hotel star rating : ستاره‌های هتل که یک عدد طبیعی بین ۱ و ۵ است.
- city : شهری که هتل در آن قرار دارد.
- image : عکس هتل

۳.۴.۲ فیلتر بازه‌ی ستاره‌های هتل

در بالای صفحه خانه دو ورودی (به شکل دلخواه) از کاربر گرفته‌شود که حد پایین و حد بالای ستاره موردنظر کاربر را تعیین می‌کند و فیلتر را با این مقادیر اعمال کند و در صفحه‌ای با همان قالب صفحه خانه نتایج را به کاربر نشان دهد. نمایش اطلاعات هتل‌ها نیز مشابه صفحه خانه است.

۴.۴.۲ صفحه‌ی جزئیات هتل

در این صفحه باید اطلاعات یک هتل خاص به طور کامل نمایش داده‌شود. اطلاعاتی که باید برای هتل‌ها در این صفحه نمایش داده شوند، به شرح زیر است:

- property name : همان نام هتل است.
- hotel overview : توضیح کوتاه درباره هتل
- property amenities : امکانات هتل که باید به صورت یک لیست نشان داده‌شوند.
- hotel star rating : ستاره‌های هتل که یک عدد طبیعی بین ۱ و ۵ است.
- city: : شهری که هتل در آن قرار دارد.
- image: : عکس هتل باید در بالای صفحه نمایش داده‌شود.
- rooms : هتل‌ها شامل ۴ نوع اتاق هستند که تعداد و قیمت به ازای هر شب برای هر کدام باید در این صفحه به صورت یک جدول نشان داده‌شوند.

۳ نکات پایانی

- اطلاعاتی که باید در هر کدام از صفحات نمایش دهید همان اطلاعاتی است که در فازهای قبلی پیاده‌سازی کرده و در رابط کاربری خط فرمان^۸ نشان می‌دادید.
- استفاده از فریم‌ورک‌های دیگر به جز APHTTP نظیر NodeJs, ReactJs, AngularJs و... غیر مجاز است.
- توجه کنید که صفحاتی که پیاده‌سازی می‌کنید باید کاربری‌های گفته شده را داشته باشند اما طراحی و جزئیات هر صفحه به عهده‌ی خود شماست؛ برای مثال، می‌تواند برای جابه‌جایی بین صفحات از نوار ناوبری^۹ یا هر روش دیگری استفاده کنید.
- منطق اصلی برنامه‌ی شما در فازهای قبلی چک می‌شود و پیاده‌سازی مواردی که در صورت پروژه ذکر شده‌است کافیست. به جز موارد گفته شده فرض می‌شود کاربر با برنامه‌ی شما رفتاری معقول دارد.

^۸Command-line interface

^۹Navbar

۴ نحوه‌ی تحویل

پرونده‌های مربوط به برنامه‌ی خود را در پوشه‌ای با نام A7-3-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۹۹۹ باشد، نام پوشه‌ی شما باید A7-3-810198999.zip باشد.

- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و اجرا شود.
- برنامه‌ی شما باید حتماً طراحی شیء‌گرا داشته باشد. همچنین باید به صورت Multifile باشد و استفاده از Makefile در این تمرین اجباری است.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.