



مقدمه

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

مدل کلاینت – سرور^۳

مدل کلاینت-سرور یک ساختار توزیع‌شده^۴ برای طراحی نرم‌افزار است که در آن نرم‌افزار به چند بخش مجزا تقسیم می‌شود. یکی از این بخش‌ها سرور نام دارد که وظیفه‌ی فراهم‌کردن منابع یا خدمات‌ها^۵ را بر عهده می‌گیرد. بخش دیگر که کلاینت نام دارد منبع یا سرویس مورد نیاز خود را از سرور درخواست می‌کند؛ برای مثال وب‌سایت CECM از یک وب‌سرور تشکیل شده و کاربران با استفاده از مرورگر^۶ خود به آن متصل شده و درخواست‌های خود را به آن می‌فرستند. صفحاتی که پس از واردکردن آدرس <http://cecm.ut.ac.ir> در مرورگر مشاهده می‌شود، جوابی است که از سمت سرور در پاسخ به درخواست کاربر فرستاده شده است.

۱ شرح تمرین

در این فاز شما به پیاده‌سازی یک وب‌سرور بر پایه‌ی پروتکل HTTP^۷ برای برنامه‌ی خود که در فازهای قبل پیاده‌سازی کرده‌اید می‌پردازید. مرورگر نیز نقش مشتری^۸ را ایفا می‌کند. پروتکل HTTP از تعدادی روش^۹ پشتیبانی می‌کند که هدف آن‌ها توصیف عملکرد درخواستی از سرور است. دو روش ارتباطی مهم در این پروتکل عبارتند از GET و POST که به ترتیب برای درخواست و ارسال اطلاعات استفاده می‌شوند. همچنین پارامترهای پرسمان^{۱۰} و بدنه‌ی^{۱۱} درخواست برای مشخص کردن اطلاعات ارسال‌شده به سوی سرویس‌دهنده مورد استفاده قرار می‌گیرند که هر کدام یک نام و مقدار دارند.

در این فاز لازم است وب‌سرور شما برخی از دستورهایی را که در فازهای قبلی پیاده‌سازی کرده‌اید پشتیبانی کند. همچنین باید برای هر یک از این عملیات‌ها رابط کاربری مخصوص به آن را با استفاده از زبان نشانه‌گذاری^{۱۲} HTML پیاده‌سازی کنید.

¹ domain logic² user interface³ client-server model⁴ distributed⁵ services⁶ web browser⁷ Hypertext Transfer Protocol⁸ client⁹ method¹⁰ query params¹¹ body¹² markup language

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 آن پیاده سازی 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
- رمز عبور
- تکرار رمز عبور
- ایمیل
- فروشنده یا خریدار بودن

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

توجه کنید که چهار ورودی اول از نوع متن هستند، اما ورودی آخر که نوع کاربری را مشخص می‌کند باید به شکل گزینه‌هایی باشد که کاربر از میان آنها انتخاب می‌کند.

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 کلاینت 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 `APTHHTTP` 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 sessionIdها نیز اعمال شود.

۴.۲ مدیر فروشگاه

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

اگر کاربری به سامانه وارد نشده باشد یا سرویسی از سرور را صدا بزند که مربوط به نوع کاربری او نیست باید خطای مناسب به او نشان داده شود.

۱.۴.۲ صفحه‌ی خانه

مدیر فروشگاه در این صفحه می‌تواند فایل کالاها، و انواع کالاهای داخل آن را مشخص کند، و کالاهای فایل مشخص شده را وارد برنامه کند. دقت کنید که در این بخش، مدیر فروشگاه، فایل گفته شده را از روی سیستم خودش، به روی سرور آپلود می‌کند. در صورت بروز مشکل در این عملیات، باید خطای مناسب به مدیر فروشگاه نشان داده شود.

۵.۲ فروشنده‌ها

۱.۵.۲ صفحه‌ی خانه

فروشنده با ورود به این صفحه می‌تواند لیست تمامی پیشنهادهایی که تا به حال ثبت کرده است را مشاهده کند.

همین‌طور در این صفحه یک لینک برای رفتن به صفحه‌ی ثبت پیشنهاد جدید وجود دارد.

اطلاعاتی که باید برای هر کدام از پیشنهادها نمایش داده شوند به شرح زیر است:

- productId ○
- offerId ○
- offerUnitPrice ○
- offerAmount ○

۲.۵.۲ صفحه‌ی ثبت پیشنهاد

در این صفحه فروشنده اطلاعات یک پیشنهاد جدید را وارد کرده و آن را به فروشگاه اضافه می‌کند.

برای ثبت پیشنهاد اطلاعات زیر وارد می‌شوند:

- productId ○
- offerUnitPrice ○

offerAmount ○

در زیر این بخش صفحه، که فروشنده می‌توانست اطلاعات پیشنهادش را وارد و ثبت کند، لیست تمامی کالاهایی که مدیر وارد کرده است، نشان داده می‌شود، تا فروشنده بتواند شناسه‌ی کالایی که قصد فروشش را دارد بدست بیاورد.

اطلاعاتی که باید برای هر کدام از کالا ها نمایش داده شوند به شرح زیر است:

productId ○

productName ○

۶.۲ خریدار

۱.۶.۲ صفحه‌ی خانه

این صفحه شامل تمام کالاهای گذاشته شده توسط مدیر سایت است و کاربر با کلیک روی هر کدام از آنها به صفحه جزئیات آن کالا منتقل می‌شود.

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

productId ○

productName ○

همین‌طور در این صفحه یک لینک برای رفتن به صفحه‌ی سبد خرید و یک لینک برای رفتن به صفحه‌ی حساب مالی وجود دارد.

۲.۶.۲ 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 به‌روزرسانی شود.

۳.۶.۲ صفحه‌ی جزئیات کالا

در این صفحه کاربر می‌تواند ویژگی‌های یک کالای خاص را به همراه تمام پیشنهادهای فروشی که توسط فروشندگان روی آن کالا گذاشته شده است ببیند و از آنها به سبد خریدش اضافه کند.

در بخش اول صفحه، برای نمایش ویژگی‌های کالای مشخص شده، باید به ازای تمام ویژگی‌های آن کالا، نام ویژگی و مقدار آن در یک جدول نمایش داده شود.

در بخش دوم این صفحه باید اطلاعات پیشنهادها نشان داده شود. اطلاعاتی که باید برای هر کدام از پیشنهادهای روی این کالا نمایش داده شوند به شرح زیر است:

offerId ○

offerUnitPrice ○

offerAmount ○

علاوه بر این موارد باید برای هر پیشنهاد، یک جعبه‌ی ورودی، از نوع مقدار عددی، که می‌تواند مقادیر ۱ تا offerAmount را بگیرد وجود داشته باشد. عدد داخل این جعبه، تعدادی که خریدار از آن پیشنهاد می‌خواهد به سبدش اضافه کند را مشخص می‌کند. در کنار جعبه‌ی توضیح داده شده، باید یک دکمه برای اضافه کردن آن تعداد مشخص شده از پیشنهاد به سبد خرید وجود داشته باشد.

دقت کنید که محدوده‌ی گفته شده باید همواره برقرار باشد و کاربر هیچگاه نتواند عددی خارج از محدوده‌ی گفته شده وارد کند. همچنین اگر offerAmount برابر ۰ است، هر دو قسمت گفته شده شامل جعبه‌ی ورودی تعداد و دکمه‌ی افزودن به سبد خرید باید غیر فعال باشند تا کاربر نتواند از آنها استفاده کند.

به این ترتیب خریدار با کلیک روی دکمه‌ی اضافه کردن به سبد خرید، می‌تواند به تعدادی که در جعبه‌ی ذکر شده وارد کرده است، از آن پیشنهاد به سبد خریدش اضافه کند که بعداً در صفحه‌ی سبد خرید قابل رؤیت خواهد بود.

۴.۶.۲ صفحه‌ی سبد خرید

هر خریدار می‌تواند با ورود به این صفحه کالاهایی که به سبد خریدش اضافه کرده است را مشاهده کند. اطلاعاتی که باید ازای هر پیشنهاد داخل سبد خرید نمایش داده شوند به شرح زیر است:

productId ○

productName ○

offerId ○

sellerId ○

totalPrice ○

amount ○

در آخر این صفحه باید مجموع هزینه کالاهای سبد، تحت عنوان مبلغ قابل پرداخت نمایش داده شود و پس از آن دکمه‌ای برای ثبت سفارش وجود داشته باشد که خریدار با کلیک روی آن، می‌تواند طبق فاز اول پروژه، سبد خرید خود را نهایی کند. نتیجه‌ی درخواست کاربر باید به درستی به کاربر نمایش داده شود. در صورت موفقیت ثبت سفارش، سبد خرید باید خالی شود.

۳ بخش امتیازی

۱.۳ نظرها

خریداران می‌توانند در صفحه‌ی جزئیات کالا نظر بدهند.

در این صفحه پس از جزئیات و پیشنهادهای فروشندگان، باید نظرهایی که برای آن کالا گذاشته شده است نمایش داده شود.

پس از بخش نمایش نظرات، بخشی برای اضافه کردن نظر وجود دارد، که نظر کاربر در آن گرفته می‌شود و پس از کلیک روی دکمه‌ی ثبت نظر، به نظرات اضافه می‌شود.

برای هر کدام از نظرهای کالا کافیست که پیام آن نظر به همراه نام صاحب نظر نمایش داده شود.

دقت کنید که در این قسمت برای اضافه شدن نظر، باید طبق فاز دوم، متن نظر ابتدا از نظر محتوای نامناسب بررسی شود و اگر نظر قابل انتشار نبود، پیام مناسب به کاربر داده شود.

۲.۳ CSS

زبان نشانه‌گذاری CSS^{۱۳} یک راه برای تغییر ظاهر صفحه‌هایی است که از HTML بهره می‌برند. استفاده از این زبان علاوه بر این که می‌تواند ظاهر برنامه‌ی شما را بهتر کند برایتان نمره‌ی امتیازی هم به همراه دارد.

همینطور کتابخانه‌های آماده‌ای از این زبان مثل **Bootstrap** و **MUI** وجود دارند که می‌توانید برای راحتی کار از آن‌ها استفاده کنید.

۳.۳ مقایسه دو کالا

خریداران می‌توانند در صفحه‌ی خانه دو کالا را انتخاب کنند و ویژگی‌های آن دو را در یک صفحه‌ی جدید، یا همان صفحه، مقایسه کنند. در این بخش باید تمام ویژگی‌های دو کالا در قالبی مرتب نمایش داده شوند تا قابلیت مقایسه فراهم باشد.

در صورت تفاوت نوع دو کالای انتخاب شده برای مقایسه، یا باید پس از زدن دکمه‌ی مقایسه، به کاربر خطای مناسب نمایش داده شود و یا از قبل، اجازه‌ی انتخاب کالای دوم را، که نوعی متفاوت از کالای اول دارد، ندهید.

۴.۳ AJAX

AJAX^{۱۴} تکنیکی برای ایجاد صفحات وب به طور سریع و پویا است. AJAX به صفحات وب این امکان را می‌دهد که به صورت غیر همزمان و تنها با تبادل اندک اطلاعات با سرور، بخشی از صفحه را بروزرسانی کنند. به این ترتیب می‌توان تنها بخش‌هایی از صفحه را بدون بارگذاری دوباره‌ی کل صفحه، تغییر داد.

در این پروژه در مواردی مانند ارسال و دریافت اطلاعات صفحه‌ی ورود یا ثبت نام، یا اطلاعات صفحه‌ی جزئیات کالا، می‌توان از Ajax استفاده کرد، تا پس از هر تغییر، نیاز به بارگذاری مجدد کل صفحه نباشد.

به عنوان مثال در صفحه‌ی جزئیات کالا می‌توان برای اضافه کردن یک پیشنهاد به سبد خرید، بدون رفتن به یک صفحه‌ی جدید برای اضافه کردن پیشنهاد به سبد خرید و سپس بازگشتن به صفحه‌ی اصلی، کاربر در همان صفحه باقی بماند و درخواست از طریق AJAX ارسال شود و نتیجه‌ی آن دریافت شود، و سپس نتیجه به کاربر نمایش داده شود.

امتیاز این بخش، بستگی به صحت استفاده شما از AJAX و مقدار استفاده‌ی شما در مواردی که مناسب است دارد.

برای دریافت اطلاعات بیشتر در این مورد، می‌توانید به این لینک مراجعه کنید.

۴ 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^{۱۵} نشان می‌دادید.

○ 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^{۱۶} 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.

¹³Cascading Style-Sheet

¹⁴Asynchronous JavaScript And XML

¹⁵Command-line interface

¹⁶Navbar

۵ 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 ۸۱۰۱۹۷۹۹۹ باشد، نام پوشه‌ی شما باید A7-3-810197999.zip باشد.

- برنامه‌ی شما باید در سیستم‌عامل 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.