



UTRAVEL!

مقدمه

شما در فازهای قبلی این پروژه به پیاده سازی سامانه خرید بلیت هواپیما پرداختید و به صورت کامل منطق دامنه ای آن را پیاده سازی کردید. هدف از این فاز پیاده سازی واسط کاربری برای فازهای قبلی است. این پیاده سازی بر روی بستر وب و مبتنی بر مدل مشتری-سرویس دهنده صورت می گیرد. برای سهولت، کتابخانه ای تحت عنوان APHTTP برای پیاده سازی وب سرور در اختیار شما قرار گرفته است.

مدل مشتری-سرویس دهنده

مدل مشتری-سرویس دهنده ساختار توزیع شده برای طراحی نرم افزار است که در آن نرم افزار به چند بخش مجزا تقسیم می شود. یکی از این بخش ها سرویس دهنده نام دارد که وظیفه ی فراهم کردن منابع یا خدمات ها را بر عهده می گیرد. بخش دیگر که مشتری نام دارد منبع یا سرویس مورد نیاز خود را از سرویس دهنده درخواست می کند؛ برای مثال وب سایت eLearn از یک وب سرور تشکیل شده و کاربران با استفاده از مرورگر خود به آن متصل شده و درخواست های خود را به آن می فرستند. صفحاتی که پس از وارد کردن آدرس <http://elearn.ut.ac.ir> در مرورگر مشاهده می شود، جوابی است که از سمت سرویس دهنده در پاسخ به درخواست کاربر فرستاده شده است.

شرح تمرین

در این فاز شما به پیاده‌سازی یک وب‌سرور بر پایه‌ی پروتکل HTTP برای برنامه‌ی خود که در فازهای قبل پیاده‌سازی کرده‌اید می‌پردازید. مرورگر نیز نقش مشتری 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 و بدنه‌ی درخواست برای مشخص کردن اطلاعات ارسال شده از سمت سرویس دهنده مورد استفاده قرار می‌گیرند که هر کدام یک نام و مقدار دارند.

در این فاز لازم است وب سرور شما برخی از دستورهای را که در فازها قبلی پیاده‌سازی کرده‌اید پشتیبانی کند. همچنین باید برای هر یک از این عملیات رابط کاربری مخصوص به آن را با استفاده از زبانِ نشانه‌گذاری HTML¹ پیاده‌سازی کنید.

آن چه شما باید پیاده سازی 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.

در هر کدام از صفحه‌ها در صورت رخ دادن استثناهای تعریف شده در فازهای قبل، کاربر به صفحه‌ای با پیام مناسب آن استثنا منتقل شود.

● صفحه‌ی ثبت نام

در این صفحه افراد می‌توانند در سایت ثبت نام کنند. این شناسه به طور خودکار در هنگام ثبت نام کاربر به او تعلق می‌گیرد.

در صفحه‌ی ثبت نام اطلاعات زیر از کاربر گرفته می‌شود:

● نام کاربری

● رمز عبور

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

¹ HyperText Markup Language

● صفحه‌ی ورود

کاربر در این صفحه می‌تواند با وارد کردن نام‌کاری 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 ایمیل متناظر با آن، متوجه می‌شود که این درخواست متعلق به کدام کاربر است و با توجه به آن به درخواست رسیدگی می‌کند.

توجه کنید که کتابخانه‌ی `APTHHTTP` توانایی مدیریت `session Id` ها را دارد. برای اطلاعات بیشتر می‌توانید به بخش **Session** در ویکی کتابخانه مراجعه کنید.

● 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.

قابلیت‌های کاربر

● صفحه خانه

در این صفحه لیست اطلاعات خلاصه‌ی بلیت‌های خریداری شده، اعتبار حساب کاربر و لیست اطلاعات خلاصه‌ی پروازها نمایش داده می‌شود.

برای هر بلیت باید شناسه، شناسه پرواز، تعداد و هزینه نمایش داده شود. همچنین برای هر پرواز باید شناسه پرواز، نام شرکت هواپیمایی، مبدا، مقصد و قیمت نمایش داده شود. همچنین کاربر با کلیک روی هر کدام از پروازها یا بلیت‌ها باید به صفحه‌ی مشخصات آن هدایت شود. در این صفحه یا صفحه‌ای جداگانه قابلیت افزایش اعتبار کاربر نیز بایستی وجود داشته باشد. دقت کنید که لازم نیست از این ساختار پیروی کنید و می‌توانید با توجه به خلاقیت خود، هر کدام از بخش‌های فوق را در صفحات مجزایی پیاده‌سازی کنید (برای مثال می‌توانید برای جلوگیری از شلوغی صفحه خانه، اطلاعات مربوط به بلیت‌های خریداری شده و اعتبار کاربر را در صفحه مجزایی مثل صفحه پروفایل نمایش دهید). دقت کنید که نحوه نمایش و پیاده‌سازی قابلیت‌های این صفحه به عهده خودتان است ولی لازم است که پیاده‌سازی شما، تمامی قابلیت‌های فوق را در بر داشته باشد.

○ اعمال فیلترها (امتیازی)

در صفحه‌ی خانه باید علاوه بر مواردی که توضیح داده شد، موارد لازم برای اعمال سه فیلتر مبدا و مقصد، بازه‌ی هزینه و نام هواپیمایی وجود داشته باشد. در آخر نیز باید یک دکمه برای حذف تمام فیلترها وجود داشته باشد. پس از اعمال هر کدام از فیلترها و یا حذف آن‌ها، باید پیام مناسب را که می‌تواند موفقیت یا عدم موفقیت را نشان دهد به کاربر نمایش دهید.

● صفحه‌ی مشخصات یک پرواز

در این صفحه کاربر می‌تواند جزئیات یک پرواز را مشاهده کند. همچنین در این صفحه یا صفحه‌ای جداگانه کاربر می‌تواند پرواز دلخواه خود را انتخاب کند و به تعداد مورد نیاز بلیت از نوع دلخواه خریداری کند. مشخصات شامل موارد زیر می‌باشد.

- شناسه پرواز
- نام هواپیمایی
- مبدا
- مقصد
- زمان حرکت
- زمان رسیدن
- صندلی‌های موجود
- قیمت

● صفحه‌ی مشخصات یک بلیت

در این صفحه کاربر می‌تواند جزئیات مشخصات یک بلیت را که خرید کرده است مشاهده کند. همچنین باید قابلیت برای لغو بلیت وجود داشته باشد. همچنین کاربر باید بتواند از این صفحه به صفحه‌ی پرواز با این شناسه پرواز منتقل شود.

مشخصات شامل موارد زیر می‌باشد:

- شناسه بلیت
- شناسه پرواز
- تعداد
- کلاس
- نوع بلیت

بخش امتیازی

● CSS²

زبان نشانه گذاری CSS 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. می‌توانید برای آشنایی بیشتر با CSS به این [لینک](#) مراجعه کنید.

نحوه‌ی تحویل و نکات

- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام `A7-<SID>.zip` در صفحه‌ی Elearn درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۹۹۹۹ باشد، نام پرونده‌ی شما باید `A7-810199999.zip` باشد.
- برای مثال، نمونه فایل مورد قبول در زیر آمده است:

A7-810199999.zip

```
|— main.cpp
|— makefile
|— ...
```

- استفاده از فریمورک‌های دیگر به جز APHTTP مانند NodeJS، Angular، ReactJS و ... مجاز نیست.
- منطق اصلی برنامه شما در فازهای قبل بررسی می‌شود و برای این فاز پیاده سازی بخش‌ها گفته شده کافیست. به جز موارد گفته شده می‌توانید فرض کنید کاربر رفتار معقولی با برنامه دارد.

² Cascading Style Sheet

- صفحاتی که طراحی می‌کنید باید قابلیت‌های گفته شده را داشته باشد ولی طراحی جزئیات هر صفحه به عهده شماست. برای مثال برای جابه‌جایی میان صفحات می‌توانید از نوار ناوبری یا هر روش دیگری استفاده کنید.
- بهتر است تا طراحی برنامه‌ی شما طوری باشد که کمترین وابستگی میان منطق برنامه و رابط کاربری آن وجود داشته باشد.
- دقت کنید که پرونده زیپ آپلودی شما باید پس از Unzip شدن شامل پرونده‌های پروژه شما (از جمله Makefile) باشد و از زیپ کردن پوشه‌ای که داخل آن فایل‌های پروژه‌تان قرار دارد خودداری فرمایید.
- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- دقت کنید که پروژه شما باید Multi-file باشد و Makefile داشته باشد. همین‌طور در Makefile خود مشخص کنید که از استاندارد c++11 استفاده می‌کنید.
- درستی برنامه‌ی شما در این فاز توسط دستیاران آموزشی سنجیده می‌شود. زیبایی و راحتی کار با رابط کاربری می‌تواند برای شما نمره‌ی امتیازی داشته باشد.
- دقت کنید که نام پرونده‌ی اجرایی شما باید **utavel.out** باشد.
- طراحی درست، رعایت سبک برنامه نویسی درست و تمیز بودن کد برنامه‌ی شما در نمره‌ی تمرین تأثیر زیادی دارد.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.