



Web UTAXI!

مقدمه

شما در فازهای قبلی این پروژه به پیاده‌سازی سامانه درخواست تاکسی پرداختید و به صورت کامل منطق آن را پیاده‌سازی کردید. هدف از این فاز پیاده‌سازی واسط کاربری برای فازهای قبلی است. این پیاده‌سازی بر روی بستر وب و مبتنی بر مدل مشتری-سرویس‌دهنده صورت می‌گیرد. برای سهولت، کتابخانه‌ای تحت عنوان **APHTTP** برای پیاده‌سازی وب سرور در اختیار شما قرار گرفته‌است.



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 وبسایت 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 پیاده سازی یک وب سرور بر پایه ی پروتکل HTTP برای برنامه ی خود که در فازهای قبل پیاده سازی کرده اید می پردازید. مرورگر نیز نقش مشتری را ایفا می کند. پروتکل HTTP از تعدادی method پشتیبانی می کند که هدف آن ها 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 می شوند. همچنین پارامترهای پرسمان و بدنه ی درخواست برای مشخص کردن اطلاعات ارسال شده از سمت سرویس دهنده مورد استفاده قرار می گیرند که هر کدام یک نام و مقدار دارند.

در این فاز لازم است وب سرور شما برخی از دستورهایی را که در فازها قبلی پیاده سازی کرده اید پشتیبانی کند. همچنین باید برای هر یک از این عملیات رابط کاربری مخصوص به آن را با استفاده از زبان نشانه گذاری HTML¹ پیاده سازی کنید.

¹ HyperText Markup Language

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. برای نشان دادن استثنا ها می‌توانید به هر شکلی عمل کنید. (مثلا پیغام را در همان صفحه نشان دهید یا کاربر را به صفحه‌ی دیگری بفرستید.)

لیست مناطق شهر

این فایل مشابه فایل اطلاعات مناطق فاز دوم است.

وارد کردن لیست مناطق

این بخش مانند فاز قبل است و دستور آغاز برنامه به شکل زیر خواهد بود.

توجه کنید نام فایل اجرایی برنامه شما باید **utaxi.out** باشد.

ورودی

```
./utaxi.out <csv_file_relative_address>
```

ورودی نمونه

```
./utaxi.out folder1/folder2/info.csv
```

● صفحه‌ی ثبت نام

در این صفحه افراد می‌توانند در سایت ثبت نام کنند. در صفحه‌ی ثبت نام اطلاعات زیر از کاربر گرفته می‌شود:

- نام کاربری

- نوع کاربر (راننده یا مسافر)

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

● نحوه احراز هویت

برای راحتی کار شما، فرض می‌کنیم به ازای هر درخواستی که برای سرور فرستاده می‌شود، نام کاربری کاربر نیز در این درخواست وجود دارد. این کار به گونه‌های مختلف می‌تواند انجام پذیرد. می‌توانید از یک فایل در صفحات استفاده کنید که اطلاعات مربوط به نام کاربری را فرد برای ارسال در دستورات وارد کند.

قابلیت‌های مسافر

● صفحه درخواست سفر

در این صفحه کاربر می‌تواند یک درخواست جدید در سامانه ثبت کند. برای این منظور، یک فرم در این صفحه وجود دارد که در آن مشخصات سفر وارد می‌شود. مشخصات شامل موارد زیر می‌باشد:

- مبدا

- مقصد

- وضعیت عجله داشتن

توجه کنید که مانند روال قبل، برای هر سفر یک شناسه تعیین می شود که پس از فرستادن درخواست، برای کاربر به طرز دلخواه نمایش داده می شود.

همچنین در این بخش کاربر می تواند قبل از درخواست سفر قیمت سفر درخواستی خود را محاسبه کند.

● لغو سفر

در این بخش که به دلخواه شما می تواند یک صفحه جدا یا در همان صفحه اصلی باشد، مسافر با ثبت شناسه یک سفر می تواند سفر را لغو کند.

قابلیت های راننده

● صفحه لیست سفرها

در این صفحه راننده می تواند لیست تمام سفرها را مشاهده کند. این لیست به صورت ردیف های پشت هم آمده است که در هر ردیف اطلاعات زیر قرار دارد.

● شناسه سفر

● نام درخواست کننده (مسافر)

● مبدا

● مقصد

● وضعیت

● قیمت

همچنین توجه داشته باشید راننده می تواند با کلیک بر روی گزینه ای برای سفرهایی که در وضعیت waiting هستند، این سفر را قبول کند. اگر وضعیت سفر در حالت traveling بود نیز راننده می تواند پایان سفر را اعلام کند.

توجه کنید در این صفحه بایستی قابلیت مرتب کردن بر اساس قیمت نیز وجود داشته باشد.

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 پایانی

- 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 ۸۱۰۱۰۰۹۹۹ است، نام پرونده‌ی شما باید A7-810100999.zip باشد.

○ برای مثال، نمونه فایل مورد قبول در زیر آمده است:

A7-810100999.zip

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

- 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.

² Cascading Style Sheet

- برنامه 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.
- **دقت کنید** که نام پرونده‌ی اجرایی شما باید **utaxi.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.
- برنامه شما در این فاز به صورت دستی توسط دستیاران آموزشی اجرا و بررسی خواهد شد.