



مدرس: رامتین خسروی

طراحان: امیررضا نادی، مجید صادقی نژاد، سهیل

حاجیان منش، عرفان میرشمس، امیرحسین عارف زاده،

مهدی نائینی، مهرداد لیویان، طاها مجلسی

مهلت تحویل: جمعه ۲۱ دی ۱۴۰۳، ساعت ۲۳:۵۵

مقدمه

شما در فازهای قبلی این پروژه به پیاده‌سازی UTaste 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 گرفته است.

مدل مشتری-سرویس‌دهنده

مدل مشتری-سرویس‌دهنده، ساختار توزیع‌شده برای طراحی نرم‌افزار است که در آن نرم‌افزار به چند بخش مجزا تقسیم می‌شود. یکی از این بخش‌ها سرویس‌دهنده نام دارد که وظیفه فراهم کردن منابع یا خدمات را بر عهده می‌گیرد. بخش دیگر که مشتری نام دارد منبع یا سرویس مورد نیاز خود را از سرویس‌دهنده درخواست می‌کند؛ برای مثال وب‌سایت ELearn از یک وب سرور تشکیل شده و کاربران با استفاده از مرورگر خود به آن متصل شده و درخواست‌های خود را به آن می‌فرستند. صفحاتی که پس از وارد کردن آدرس <http://elearn.ut.ac.ir> در مرورگر مشاهده می‌شود، جوابی است که از سمت سرویس‌دهنده در پاسخ به درخواست کاربر فرستاده شده است.

شرح تمرین

در این فاز شما به پیاده‌سازی یک وب‌سرور بر پایه‌ی پروتکل HTTP برای برنامه 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^۱. همچنین

^۱ در صورت لزوم می‌توانید از متدهای PUT و DELETE هم استفاده نمایید.

پارامترهای پرسمان و بدنه 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 فازهای 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 HTML² پیاده سازی کنید.

در ابتدای اجرای برنامه، شما باید یک سری اطلاعات را از روی مجموعه داده³ هایی که در قالب CSV⁴ به شما داده می شوند خوانده و در برنامه خود ذخیره کنید. توضیحات این فایل ها مشابه فازهای قبلی است، با این تفاوت که اولین عضو آرایه argv، از قبل به پارامتر port اختصاص پیدا کرده است و شما باید از آرگومان های دوم به بعد استفاده نمایید (به عبارت دیگر، index شماره صفر قابل استفاده نیست)⁵.

آنچه شما باید پیاده سازی کنید

در ادامه قابلیت هایی که برنامه ی شما باید داشته باشد به تفصیل توضیح داده خواهد شد. شما باید بر اساس هر یک از این قابلیت ها یک صفحه برای واسط کاربری آن پیاده سازی کنید و با استفاده از کدی که در فازهای قبل زده اید، منطق آن را کنترل نمایید. دقت کنید می توانید طراحی صفحه ها یا نوع صفحه های موجود را تغییر دهید اما باید تمام قابلیت های نوشته شده پشتیبانی شود، از آنجایی که این فاز به صورت خودکار چک نمی شود، نحوه و شکل نشان دادن اطلاعات به دلخواه خودتان می باشد. دقت کنید برای سادگی بعضی از قابلیت های پیاده سازی شده در فازهای قبل در این فاز وجود ندارد، اما در صورت علاقه می توانید این موارد را نیز پیاده سازی کنید (توجه کنید که پیاده سازی این بخش ها نمره امتیازی ندارد).

در هر کدام از صفحه ها در صورت رخ دادن استثنای تعریف شده در فازهای قبل، کاربر به صفحه ای با پیام مناسب آن استثنا منتقل می شود. برای نشان دادن استثناها می توانید به هر شکلی عمل کنید. مثلاً پیغام را در همان صفحه نشان دهید یا کاربر را به صفحه دیگری بفرستید. برای پاسخ 'OK' می توانید پیغامی نشان ندهید.

² HyperText Markup Language

³ Dataset

⁴ Comma-Separated Values

⁵ برای توضیحات بیشتر به ویدئوهای بارگذاری شده در مورد کتابخانه APHTTP مراجعه نمایید

صفحه ثبت نام

در این صفحه افراد می‌توانند با وارد کردن همان اطلاعاتی که در دستور ثبت‌نام در فازهای قبل از کاربران گرفته می‌شد، در سامانه ثبت‌نام کنند. دقت کنید که در صورت ثبت‌نام موفق، کاربر باید مستقیماً به صفحه اصلی مربوط به خود هدایت شود. همچنین کاربر باید بتواند از این صفحه به صفحه ی ورود برود.

صفحه ورود

در این صفحه افراد می‌توانند احراز هویت خود را انجام داده و با وارد کردن اطلاعات خود مطابق دستور ورود به سامانه در فازهای قبل، وارد سامانه شوند. پس از ورود به سیستم کاربر باید به صفحه اصلی مربوط به خود منتقل می‌شود. همچنین کاربر باید بتواند از این صفحه به صفحه ثبت نام برود. 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 ارسال‌شده متعلق به چه کاربری است. برای انجام این کار پس از وارد کردن نام کاربری و گذرواژه توسط کاربر و ارسال آن به سرور، در صورتی که این اطلاعات درست باشند، سرور شناسه یکتای مربوط به کاربر را با عنوان Session ID تولید کرده و آن را به کلاینت ارسال می‌کند. از این پس سرور با بررسی Session ID هر درخواست و مشخص کردن کاربر 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 کتابخانه APHTTP توانایی مدیریت Session ID را دارد. برای کسب اطلاعات بیشتر در این باره می‌توانید به بخش Session در [ویکی کتابخانه](#) مراجعه کنید.

صفحه اصلی

کاربر بلافاصله پس از ورود یا ثبت نام به این صفحه منتقل می‌شود و باید بتواند از این صفحه به هر صفحه دیگر وب سایت برود. همچنین تمامی صفحات باید قابلیت بازگشت به این صفحه را داشته باشند.

خروج

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

صفحه مشاهده اطلاعات یک رستوران

در این صفحه کاربر می‌تواند دقیقاً مشابه دستور مشاهده اطلاعات یک رستوران، نام رستوران را وارد کرده و اطلاعات مربوط به رستوران که در فاز های قبلی توضیح داده شد را مشاهده کند. اطلاعات نمایشی دقیقاً همان اطلاعات مربوط به فاز دوم هستند (البته اگر به هر دلیلی موفق به پیاده‌سازی فاز دوم نشده‌اید، می‌توانید اطلاعات نمایشی را مانند فاز اول نشان دهید). پیشنهاد می‌شود یک صفحه برای نمایش نام تمامی رستوران ها به صورت یکجا طراحی کرده و لینک های مورد نیاز برای دسترسی به صفحه اطلاعات و یا رزرو هر رستوران را در این صفحه قرار دهید.

صفحه رزرو رستوران

در این صفحه کاربر باید بتواند تمامی مقادیر مورد نیاز دستور رزرو رستوران شامل: نام رستوران، شماره میز، آغاز زمان رزرو، پایان زمان رزرو و غذا هایی که قصد سفارش آن ها را دارد، را وارد کند و درخواست رزرو خود را ثبت کند.

صفحه نمایش رزروهای انجام شده توسط کاربر

در این صفحه باید تمام رزرو های کاربر برای او قابل مشاهده باشد. نکته ی مهم این است که باید اطلاعات تمامی رزروهای کاربر فعلی در تمامی رستوران ها نمایش داده شوند. اطلاعات نمایشی در صفحه، دقیقاً همان اطلاعات مربوط به فاز دوم هستند (البته اگر به هر دلیلی موفق به پیاده سازی فاز دوم نشده اید، می توانید اطلاعات نمایشی را مانند فاز اول نشان دهید).

بخش امتیازی

• CSS⁶

زبان نشانه گذاری CSS یک راه برای تغییر ظاهر صفحه هایی است که از HTML بهره می برند. استفاده از این زبان علاوه بر این که می تواند ظاهر برنامه ی شما را بهتر کند، برایتان نمره ی امتیازی هم به همراه دارد. می توانید برای آشنایی بیشتر با CSS به این [لینک](#) مراجعه کنید.

• JS⁷

زبان javascript یک راه برای interactive کردن صفحه هایی است که از HTML بهره می برند. استفاده از این زبان علاوه بر این که می تواند قابلیت های زیادی به برنامه اضافه کند، برایتان نمره ی امتیازی هم به همراه دارد. می توانید برای آشنایی بیشتر با javascript به این [لینک](#) مراجعه کنید.

⁶ Cascading Style Sheet

⁷ Javascript

نکات و نحوه تحویل

- برای تحویل این پروژه، لازم است کد خود را در یک مخزن⁸ در GitHub بارگذاری کنید و سپس لینک مخزن به همراه Hash آخرین کامیت⁹ را در صفحه eLearn درس بارگذاری نمایید.
نمونه متن خواسته شده در سامانه ای‌لرن (سه بخش <username> و <repository_name> و <last_commit_hash> را جایگزین کنید):

```
https://github.com/<username>/<repository_name>  
<last_commit_hash>
```

- دقت کنید که پروژه شما باید Multi-file باشد و Makefile داشته باشد. همین‌طور در Makefile خود مشخص کنید که از استاندارد c++20 استفاده می‌کنید.
- فایل‌های خود را در مخزن قبلی به نام <SID>-F03-AP به نام **AP-UT@** را به مخزن اضافه کنید. دقت داشته باشید که برای هر سه فاز از این مخزن استفاده خواهید کرد و همه تغییرات خود را در شاخه¹⁰ main انجام دهید و نیازی به پیاده‌سازی شاخه‌ها نیست (هر چند برای کارایی بهتر توصیه می‌شود انجام دهید ولی در پایان تغییرات خود را به شاخه main نیز منتقل کنید). دقت داشته باشید که چون تحویل هر سه فاز با یکدیگر خواهد بود مشخصات کامیت خود را در هر فاز به درستی وارد کنید.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++20 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین از درستی کامل قالب خروجی برنامه خود اطمینان حاصل کنید و از دادن خروجی‌هایی که در صورت پروژه ذکر نشده است اجتناب کنید.
- دقت کنید که نام فایل اجرایی شما باید UTaste (بدون هیچ پسوندی مانند exe یا out) باشد.
- 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.
- توجه داشته باشید که حالت‌های خاصی که در صورت پروژه ذکر نشده است در تست‌های خودکار نخواهد بود و می‌توانید به هر شکلی که مد نظر دارید آن‌ها را مدیریت کنید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.
- توجه کنید که رعایت نکردن ساختار گفته شده در نام‌گذاری مخزن، فایل‌کد، فایل اجرایی و آپلود موارد خواسته شده باعث کسر 5 درصد از نمره شما خواهد شد.

⁸ Repository

⁹ Commit

¹⁰ Branch

- توجه کنید که این فاز آزمون خودکار ندارد و تصمیم در مورد نحوه نشان دادن خروجی‌ها تا زمانی که معقول باشد با خودتان است.

نمرات

- تمیزی کد
 - رعایت کردن نام‌گذاری صحیح و انسجام¹¹
 - عدم وجود کد تکراری
 - رعایت دندانه‌گذاری¹²
 - عدم استفاده از متغیرهای گلوبال
 - استفاده صحیح از متغیرهای ثابت¹³ به جای Magic Value-ها
 - ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند
- درستی کد
 - پیاده‌سازی صحیح کارکردهای خواسته شده
- طراحی
 - استفاده مناسب از استثناها برای مدیریت خطا
 - میک‌فایل
- بخش امتیازی
 - استفاده از CSS
 - استفاده از Javascript

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

¹¹ Consistency

¹² Indentation

¹³ Constant