



## 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 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. همچنین پارامترهای پرسمان و بدنه درخواست برای مشخص کردن اطلاعات ارسال‌شده از سمت سرویس‌دهنده مورد استفاده قرار می‌گیرند که هر کدام یک نام و مقدار دارند.

در این فاز لازم است وب سرور شما برخی از دستورهایی را که در فازهای قبلی پیاده‌سازی کرده‌اید پشتیبانی کند. همچنین باید برای هر یک از این عملیات رابط کاربری مخصوص به آن را با استفاده از زبانِ نشانه‌گذاری HTML<sup>1</sup> پیاده‌سازی کنید.

---

<sup>1</sup> 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 پیاده‌سازی کنید و با استفاده از کدی که در فازهای قبل زده‌اید، منطق آن را کنترل نمایید. دقت کنید می‌توانید طراحی صفحه‌ها یا نوع صفحه‌های موجود را تغییر دهید اما باید تمام قابلیت‌های نوشته شده پشتیبانی شود، از آنجایی که این فاز به صورت خودکار چک نمی‌شود نحوه و شکل نشان دادن اطلاعات به دلخواه خودتان می‌باشد. دقت کنید برای سادگی بعضی از قابلیت‌های پیاده‌سازی شده در فازهای قبل در این فاز وجود ندارد (مثل مشاهده پلی‌لیست بقیه کاربران یا ...) اما در صورت علاقه می‌توانید پیاده‌سازی کنید (توجه کنید که پیاده‌سازی این بخش‌ها نمره امتیازی ندارد).

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

## قابلیت‌های مشترک کاربر عادی و هنرمند

### 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.

### 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 با توجه به اینکه کاربر عادی است یا هنرمند، 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.

## 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 اصلی

کاربر با توجه به نوع خودش باید بتواند از این صفحه به صفحه مربوط به قابلیت‌های خود برود. به طور مثال هر دو کاربر عادی و هنرمند از طریق صفحه اصلی خود باید بتوانند به صفحه مشاهده کاربران منتقل شوند. غیر از صفحات مشترک بین کاربر عادی و هنرمند، هر نوع کاربر باید بتواند به صفحات مختص به خودش هم منتقل شود. به طور مثال کاربر عادی باید توانایی رفتن به صفحه نمایش آهنگ‌های پرفرمدار و کاربر هنرمند توانایی رفتن به صفحه اشتراک‌گذاری آهنگ جدید از صفحه اصلی خودش را داشته باشد. همچنین دقت کنید که پس از ورود به سامانه و در هر صفحه دیگری غیر از صفحه اصلی باید بتواند مستقیماً به صفحه اصلی بازگردد.

## صفحه مشاهده کاربران

در این صفحه کاربر باید بتواند فهرست کاربران را به همراه اطلاعات خلاصه شده آن‌ها را مشاهده کند. همچنین کاربر باید بتواند با کلیک کردن بر روی نام هر کاربر به صفحه نمایش اطلاعات کامل آن کاربر منتقل شود.

## صفحه مشاهده اطلاعات کامل کاربر

در این صفحه کاربر باید بتواند اطلاعات کامل کاربر مورد نظر خود را مطابق فازهای قبل مشاهده کند، از جمله اطلاعاتی که نمایش داده می‌شود:

- کاربر باید بتواند اطلاعات مربوط به دنبال‌کنندگان و دنبال‌شوندگان را مشاهده کند.
- کاربر در این صفحه می‌تواند کاربر نمایش داده شده را دنبال کند یا در صورتی که در دنبال‌کنندگان او قرار داشت، عدم دنبال کردن را انتخاب کند.

## صفحه مشاهده آهنگ

در این صفحه اطلاعات مربوط به آهنگ مورد نظر نمایش داده می‌شود. قابلیت پخش آهنگ باید در این صفحه وجود داشته باشد.

کاربر عادی در این صفحه باید بتواند این آهنگ را به هر کدام از پلی‌لیست‌های خود اضافه کند. همچنین کاربر هنرمند در صورت تعلق داشتن آهنگ به خود، می‌تواند آهنگ را از طریق این صفحه حذف کند. دقت کنید که در صورت حذف موفقیت‌آمیز آهنگ، هنرمند باید به صفحه اصلی خود برگردد.

## قابلیت‌های کاربر عادی

### صفحه ایجاد پلی‌لیست

در این صفحه کاربر با وارد کردن نام پلی‌لیست می‌تواند پلی‌لیست جدیدی را برای خود ایجاد کند.

### صفحه نمایش پلی‌لیست‌های کاربر

در این صفحه کاربر می‌تواند فهرست پلی‌لیست‌های ساخته شده توسط خودش به همراه اطلاعات خلاصه درباره پلی‌لیست‌ها را مشاهده کند. همچنین کاربر باید بتواند با کلیک کردن بر روی اطلاعات هر پلی‌لیست به صفحه مشاهده آن پلی‌لیست منتقل شود.

### صفحه مشاهده پلی‌لیست کاربر

در این صفحه کاربر می‌تواند اطلاعات خلاصه شده آهنگ‌های موجود در پلی‌لیست را مطابق با خروجی مد نظر در فازهای قبلی مشاهده کرده و همچنین با کلیک کردن بر روی هر یک از این آهنگ‌ها، به صفحه مشاهده آن آهنگ منتقل شود.

کاربر می‌تواند در این صفحه پلی‌لیست را حذف کند. در صورت موفقیت‌آمیز بودن حذف پلی‌لیست، کاربر به صفحه اصلی باز می‌گردد.

### صفحه جستجوی آهنگ

در این صفحه کاربر مشابه فازهای قبل می‌تواند با وارد کردن اطلاعات مربوط به نام، نام هنرمند یا tag های آهنگ به جستجوی آهنگ‌ها بپردازد. دقت کنید که مشابه فازهای قبل، هر یک از این سه آرگومان می‌توانند خالی باشند و از طرف کاربر مقداری برای آن‌ها وارد نشود. همچنین کاربر با کلیک کردن بر روی اطلاعات نمایش داده شده در خروجی جستجو، باید به صفحه نمایش اطلاعات کامل آن آهنگ منتقل شود.

## صفحه نمایش آهنگ‌های پرترفدار

در این صفحه فهرست آهنگ‌های پرترفدار به همراه اطلاعات خروجی داده شده در فاز قبل برای آن‌ها نمایش داده می‌شود. همچنین کاربر باید بتواند با کلیک بر روی هر یک از این آهنگ‌ها به صفحه نمایش اطلاعات کامل مربوط به آن آهنگ منتقل شود.

## قابلیت‌های کاربر هنرمند

### صفحه به اشتراک‌گذاری آهنگ جدید

هنرمند در این صفحه باید بتواند با وارد کردن اطلاعات آهنگ و فایل یا آدرس آن یک آهنگ جدید به سامانه اضافه کند. نحوه دریافت ورودی‌ها و فایل آهنگ مورد نظر در این بخش بر عهده خودتان است و بسته به معماری کدتان می‌توانید این بخش را پیاده‌سازی کنید. دقت کنید که فرمت فایل آهنگ ورودی باید m4a باشد. همچنین آهنگ‌هایی که توسط برنامه پشتیبانی می‌شوند باید کم‌تر از ۱۰ مگابایت باشند؛ دلیل این موضوع به محدودیت‌های کتابخانه APHTTP برمی‌گردد.

### صفحه مشاهده آهنگ‌های هنرمند

هنرمند در این صفحه می‌تواند لیست تمام آهنگ‌های به اشتراک گذاشته شده توسط خودش به همراه اطلاعات مربوط به هر آهنگ را مشاهده کند. همچنین هنرمند باید بتواند با کلیک کردن بر روی هر آهنگ به صفحه نمایش اطلاعات کامل آن آهنگ منتقل شود.

# بخش امتیازی

## • CSS<sup>2</sup>

زبان نشانه گذاری CSS یک راه برای تغییر ظاهر صفحه‌هایی است که از HTML بهره می‌برند. استفاده از این زبان علاوه بر این‌که 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.

## • JS<sup>3</sup>

زبان javascript یک راه برای interactive کردن صفحه‌هایی است که از HTML بهره می‌برند. استفاده از این زبان علاوه بر این‌که می‌تواند قابلیت های زیادی به برنامه اضافه کند، برایتان نمره‌ی امتیازی هم به همراه دارد. می‌توانید برای آشنایی بیشتر با javascript به این [لینک](#) مراجعه کنید.

---

<sup>2</sup> Cascading Style Sheet

<sup>3</sup> Java Script

# نکات و نحوه تحویل

- 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 ۸۱۰۱۰۰۰۰۰ است، نام پرونده‌ی شما باید A7-810100000.zip باشد.

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

A7-810100000.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.
- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++20 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++20 استفاده می‌کنید.
- دقت کنید که نام فایل اجرایی شما باید sputify.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.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.



# نمرات

- تمیزی کد
  - رعایت کردن نام‌گذاری صحیح و انسجام<sup>4</sup>
  - عدم وجود کد تکراری
  - رعایت دندانه‌گذاری<sup>5</sup>
  - عدم استفاده از متغیرهای گلوبال
  - استفاده صحیح از متغیرهای ثابت<sup>6</sup> 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

---

<sup>4</sup> Consistency

<sup>5</sup> Indentation

<sup>6</sup> Constant