



## مقدمه

شما در فازهای قبلی این پروژه به پیاده‌سازی یک شبکه‌ی مجازی پرداختید و به صورت کامل منطق دامنه‌ی<sup>۱</sup> آن را پیاده‌سازی کردید. هدف از این فاز پیاده‌سازی یک واسط کاربری<sup>۲</sup> برای فازهای قبلی است. این پیاده‌سازی بر روی بستر وب و مبتنی بر مدل مشتری-سرویس‌دهنده<sup>۳</sup> صورت می‌گیرد. برای سهولت، کتابخانه‌ای تحت عنوان **APHTTP** برای پیاده‌سازی وب‌سرور در اختیار شما قرار گرفته‌است.

## مشتری-سرویس‌دهنده

مدل مشتری-سرویس‌دهنده یک ساختار توزیع‌شده<sup>۴</sup> برای طراحی نرم‌افزار است که در آن نرم‌افزار به چند بخش مجزا تقسیم می‌شود. یکی از این بخش‌ها سرویس‌دهنده نام دارد که وظیفه‌ی فراهم‌کردن منابع یا خدمات‌ها<sup>۵</sup> را بر عهده می‌گیرد. بخش دیگر که مشتری نام دارد منبع یا سرویس مورد نیاز خود را از سرویس‌دهنده درخواست می‌کند؛ برای مثال وب‌سایت CECM از یک وب‌سرور تشکیل شده و کاربران با استفاده از مرورگر<sup>۶</sup> 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 سرویس‌دهنده در پاسخ به درخواست کاربر فرستاده شده است.

## ۱ شرح تمرین

در این فاز شما به پیاده‌سازی یک وب‌سرور بر پایه‌ی پروتکل HTTP<sup>۷</sup> برای برنامه‌ی خود که در فازهای قبل پیاده‌سازی کرده‌اید می‌پردازید. مرورگر نیز نقش مشتری<sup>۸</sup> 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<sup>۹</sup> و بدنه‌ی<sup>۱۰</sup> 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 وب‌سرور شما برخی از دستورهای 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<sup>۱۱</sup> HTML پیاده‌سازی کنید.

<sup>1</sup>domain logic

<sup>2</sup>user interface

<sup>3</sup>client-server

<sup>4</sup>distributed

<sup>5</sup>services

<sup>6</sup>web browser

<sup>7</sup>Hypertext Transfer Protocol

<sup>8</sup>client

<sup>9</sup>query params

<sup>10</sup>body

<sup>11</sup>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.

### ۱.۲ صفحه‌ی ثبت نام

در این صفحه افراد می‌توانند در شبکه‌ی مجازی ثبت نام کنند. توجه کنید که هر کاربر دارای یک شناسه‌ی عددی یکتا است که این شناسه از عدد یک شروع می‌شود. این شناسه به طور خودکار در هنگام ثبت نام کاربر به آن تعلق می‌گیرد.

در صفحه‌ی ثبت نام اطلاعات زیر از کاربر گرفته می‌شود:

- نام کاربری
- سن
- رمز عبور و تکرار رمز عبور
- ایمیل
- ناشر بودن یا نبودن

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

### 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.

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.

### ۴.۲ ناشران

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 بیان 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

ناشر با اولین بار ورود به این صفحه می‌تواند لیست تمامی فیلم‌هایی که تا به حال ثبت کرده است را مشاهده کند. همین‌طور ناشر در این صفحه می‌تواند در صورت تمایل لیست فیلم‌ها را براساس نام کارگردان با وارد کردن آن فیلتر کند.

ناشر می‌تواند با کلیک بر روی دکمه‌ی حذف مربوط به هر کدام از فیلم‌ها آن فیلم را حذف کند. همین‌طور در این صفحه یک لینک برای رفتن به صفحه‌ی ثبت فیلم جدید وجود دارد.

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

- |          |   |        |   |              |   |
|----------|---|--------|---|--------------|---|
| نام      | ○ | قیمت   | ○ | سال ساخت     | ○ |
| طول زمان | ○ | امتیاز | ○ | نام کارگردان | ○ |

## ۲.۴.۲ صفحه‌ی ثبت فیلم

در این صفحه ناشر اطلاعات یک فیلم جدید را وارد کرده و آن را به شبکه اضافه می‌کند.

برای ثبت فیلم اطلاعات زیر وارد می‌شوند:

- |          |   |          |   |                     |   |
|----------|---|----------|---|---------------------|---|
| نام      | ○ | قیمت     | ○ | نام کارگردان        | ○ |
| طول زمان | ○ | سال ساخت | ○ | خلاصه <sup>۱۲</sup> | ○ |

## ۵.۲ مشتریان

تا این جا با قابلیت‌هایی که تنها ناشر می‌تواند انجام دهد، آشنا شدیم. از این پس با قابلیت‌های مشتری‌ها آشنا می‌شویم. توجه کنید که هر ناشر برای فیلم‌های دیگر می‌تواند در نقش یک مشتری باشد. یعنی تمام قابلیت‌های مشتری را نیز علاوه بر قابلیت‌های ناشر داراست.

## ۱.۵.۲ صفحه‌ی خانه

این صفحه مشابه صفحه‌ی خانه‌ی ناشر است و تنها تفاوت آن عدم توانایی حذف یا اضافه کردن فیلم توسط مشتری است. همین‌طور مشتری فیلم‌هایی که توانایی دیدن آن‌ها را دارد مشاهده می‌کند.

## ۲.۵.۲ صفحه‌ی پروفایل

در این صفحه مشتری می‌تواند لیست فیلم‌هایی را که خریداری کرده است مشاهده کند. همین‌طور مشتری می‌تواند در این صفحه اعتبار حساب خود را شارژ کند.

اطلاعاتی که در این لیست برای هر کدام از فیلم‌ها وجود دارد همان اطلاعاتی است که در صفحه‌ی خانه توسط مشتری‌ها قابل مشاهده است.

---

<sup>12</sup>summary

## ۳.۵.۲ صفحه‌ی مشاهده‌ی جزئیات فیلم

هر مشتری می‌تواند با ورود به این صفحه جزئیات یک فیلم را مشاهده کند. در انتهای این قسمت ۴ فیلم از میان فیلم‌های شبکه، پس از نمایش جزئیات فیلم مذکور به نمایش در می‌آید. با کلیک بر روی هر کدام از این فیلم‌ها مشتری به صفحه‌ی جزئیات آن فیلم هدایت می‌شود. مشتری می‌تواند در این صفحه در صورتی که توان خرید فیلم را داشت آن را خریداری کند. هر مشتری می‌تواند پس از آن که فیلمی را خرید، به آن بین ۱ تا ۱۰ امتیاز دهد. جزئیات یک فیلم شامل اطلاعات زیر است:

- نام
- امتیاز
- خلاصه
- طول زمان
- سال ساخت
- قیمت
- نام کارگردان

برای فیلم‌های پیشنهاد شده نیز باید اطلاعات زیر حتماً نشان داده شوند:

- نام
- طول زمان
- نام کارگردان

## ۳ بخش امتیازی

### ۱.۳ نظرها

در مرحله‌ی اول بعد از خریداری فیلم، مشتری می‌تواند در صفحه‌ی مشاهده‌ی جزئیات فیلم به فیلم نظر بدهد. سپس در مرحله‌ی بعد در این صفحه پس از جزئیات باید نظرهایی را که برای آن فیلم گذاشته شده نمایش داده شود. دقت کنید که در این فاز تنها نظرهای یک فیلم پیاده‌سازی می‌شوند و پاسخ به نظرها نیاز به پیاده‌سازی ندارد. برای هر کدام از نظرهای فیلم کافیس‌ت که پیام آن نظر نمایش داده شود.

## ۲.۳ CSS

زبان نشانه‌گذاری CSS<sup>۱۳</sup> 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.

<sup>13</sup>Cascading Style-Sheet

## 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 نشان می‌دادید.
- استفاده از فریم‌ورک‌های دیگر به جز APHTTP نظیر NodeJs, ReactJs, AngularJs و... غیر مجاز است.
- 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<sup>۱۴</sup> 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.
- شما مجاز به استفاده از جاوااسکریپت در این پروژه نیستید.

## 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 سیستم عامل 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.
  - در صورتی که از git استفاده می‌کنید، کدهای خود را در مخزنی<sup>۱۵</sup> که قبلاً در سایت GitHub<sup>۱۶</sup> ساخته‌اید push کنید و آدرس مخزن خود را در **بخش مربوط** در صفحه‌ی درس بنویسید. دقت کنید که شما باید تنها یک مخزن برای همه‌ی فازهای پروژه داشته باشید. همچنین، دقت کنید که علاوه بر GitHub باید کدهای مربوط به هر فاز پروژه را در محل مربوط به آن فاز در CECM نیز بارگذاری کنید.
  - هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

<sup>14</sup>Navbar

<sup>15</sup>repository

<sup>16</sup>github.com