



مدرس: رامتین خسروی

طراحان: الهه خداوردی، نسا عباسی، عرفان عسگری،  
کوروش علی‌نقی، شهنام فیضیان، عرفان میرشمس،  
امیررضا نادی، پریسا یحیی‌پور، محمد امین یوسفی

مهلت تحویل: پنج‌شنبه ۲۴ خرداد ۱۴۰۳، ساعت ۲۳:۵۵



# UTMS

## مقدمه

شما در فازهای قبلی این پروژه به پیاده‌سازی سامانه اجتماعی جامع آموزش 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 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 HTML<sup>1</sup> پیاده‌سازی کنید.

در ابتدای اجرای برنامه، شما باید یک سری اطلاعات را از روی مجموعه داده<sup>2</sup>هایی که در قالب CSV<sup>3</sup> به شما داده می‌شوند خوانده و در برنامه خود ذخیره کنید. توضیحات این فایل‌ها مشابه فازهای قبلی است.

## 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 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، اما در صورت علاقه می‌توانید این موارد را نیز 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.

---

<sup>1</sup> HyperText Markup Language

<sup>2</sup> Dataset

<sup>3</sup> Comma-Separated Values

# قابلیت‌های مشترک

## 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 دانشجو است یا استاد یا کاربر پیش‌فرض، به صفحه اصلی مربوط به خود منتقل می‌شود. برنامه شما باید از این صفحه شروع 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 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

نحوه قرارگیری قابلیت خروج از حساب به عهده شماست. با خارج شدن از حساب کاربری بایستی به صفحه ورود برگردیم و تا زمان وارد نشدن کاربری جدید، قادر به کاری نخواهیم بود.

## صفحه اصلی

در این صفحه شخص وارد شده می‌تواند به فعالیت‌هایی که در ادامه ذکر شده‌اند و با توجه به نوع کاربر فرق می‌کند دسترسی داشته باشد. در نحوه دسترسی فرد به این قابلیت‌ها آزادی عمل دارید و به سلیقه خودتان این صفحه را طراحی کنید. مهم است که به نحوی بتوان از طریق این صفحه به قابلیت‌ها دسترسی داشت. در این صفحه اطلاعات شخص از جمله نام، رشته، شناسه و عکس پروفایل نیز نمایش داده می‌شود.

## ارسال پست

در این صفحه کاربر می‌تواند یک پست جدید ایجاد کند. برای این منظور یک فرم در صفحه وجود دارد که در آن مشخصات پست وارد می‌شود. مشخصات شامل موارد زیر می‌باشند:

- عنوان
- متن پیام
- تصویر

توجه داشته باشید که برای بخش تصویر پست باید محلی برای بارگذاری تصویر در نظر بگیرید، تضمین می‌شود که فرمت تصاویر png باشد. همچنین یک پست می‌تواند عکس نداشته باشد. راهنمایی پیاده‌سازی: در کدهای بخش handlers خود دیتای عکس را که در بدنه ریکوئست قرار می‌گیرد داخل یک مسیر مشخص و با نامی یکتا ذخیره کنید و آدرس این فایل را مانند فاز دو ذخیره کنید.

## قابلیت‌های استاد و دانشجو

### تغییر عکس پروفایل

در این صفحه باید دو قابلیت حذف و ذخیره عکس پروفایل وجود داشته باشد. در صورت استفاده از قابلیت ذخیره عکس، عکس قبلی به صورت خودکار حذف می‌شود و عکس جدید جایگزین آن می‌شود. برای دریافت عکس پروفایل باید مانند تصاویر پست‌ها عمل کنید و مکانی برای بارگذاری آن در نظر بگیرید. همچنان تضمین می‌شود که فرمت عکس‌های پروفایل png است.

### نمایش دروس ارائه شده

در این صفحه دروس ارائه شده برای کاربر نمایش داده خواهند شد.

### مشاهده صفحه شخصی کاربران

نحوه قرارگیری قابلیت جستجو و پیدا کردن صفحه شخصی دیگر کاربران به عهده شماست.

پس از پیدا شدن صفحه شخصی مورد نظر، آن را نمایش می‌دهید.

این صفحه شامل اطلاعات زیر از کاربر مورد نظر است:

- نام، عکس پروفایل و پست‌های مربوط به آن حساب. برای پست‌ها نمایش عنوان پست و عکس آن پست (در صورت وجود) ضروری است.
- همچنین باید مشخص باشد که صفحه شخصی متعلق به یک استاد است یا دانشجو و یا حساب پیش‌فرض مربوط به دانشگاه است. در صورتی که صفحه شخصی متعلق به استاد بود درجه استاد و در صورتی که شخص دانشجو بود ترم فعلی او نیز نمایش داده می‌شود. برای حساب دانشگاه نیازی به اطلاعات اضافه‌تری نیست.

## قابلیت‌های دانشجو

### ثبت نام برای درس ارائه شده

در این صفحه با وارد کردن شناسه ارائه درس مورد نظر در یک فرم دانشجو آن درس را اخذ می‌کند.

این فرم تنها شامل شناسه ارائه درس مورد نظر است.

## حذف درس برداشته شده

همانند ثبت نام، برای حذف درس برداشته شده، دانشجو لازم است شناسه آن را در یک فرم وارد کند.

## مشاهده دروس برداشته شده

در یک صفحه دروس اخذ شده توسط دانشجو نمایش داده خواهند شد. اطلاعاتی که باید برای هر درس نمایش داده شود همان اطلاعاتی است که در فاز اول بایستی به ازای وارد کردن دستور مشاهده دروس برداشته شده نمایش داده می شد.

## قابلیت های کاربر پیش فرض دانشگاه

### اضافه کردن ارائه درس

در این صفحه کاربر پیش فرض دانشگاه می تواند یک ارائه درس جدید ایجاد کند. برای این منظور یک فرم در این صفحه وجود دارد که در آن مشخصات ارائه درس وارد می شود. مشخصات شامل موارد زیر می باشد:

- شناسه درس
- شناسه استاد
- ظرفیت
- زمان برگزاری
- تاریخ امتحان
- کلاس برگزاری

برای راحتی می توانید زمان برگزاری و تاریخ امتحان را مثل فازهای قبل مانند یک رشته (string) ورودی بگیرید.

# بخش امتیازی

## • CSS<sup>4</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>5</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>4</sup> Cascading Style Sheet

<sup>5</sup> Javascript

# نکات و نحوه تحویل

- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام A6-<SID>.zip در صفحه‌ی Elearn درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۰۲۰۰۰ است، نام پرونده‌ی شما باید A6-810102000.zip باشد.
  - برای مثال، نمونه فایل مورد قبول در زیر آمده است:

A6-810102000.zip

└─ main.cpp

└─ makefile

└─ ...

- 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 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 utms.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>6</sup>
  - عدم وجود کد تکراری
  - رعایت دندانه‌گذاری<sup>7</sup>
  - عدم استفاده از متغیرهای گلوبال
  - استفاده صحیح از متغیرهای ثابت<sup>8</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

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

---

<sup>6</sup> Consistency

<sup>7</sup> Indentation

<sup>8</sup> Constant