

هدف این تمرین، پیاده سازی یک ایمیل سرور ساده با PHP است.

### صورت مسئله

برخلاف تمرینهای قبل تمرکز این تمرین بر پیاده سازی سمت سرور است. از شما خواسته می شود که قابلیت های سمت سرور را با زبان PHP پیاده سازی کنید. در این تمرین تمامی اطلاعات افراد و ایمیل ها در پایگاه داده طراحی شده توسط خودتان ذخیره می شوند و نیازی به پیاده سازی پروتکل های SMTP و ... نیست. لازم به ذکر است که انتخاب پایگاه داده بر عهده شماست و می توانید از نمونه هایی مانند MySQL ، MariaDB ، MongoDB و ... استفاده کنید. وبسایت شما باید اطلاعات را از پایگاه داده گرفته و بر اساس نوع درخواست به کاربر نمایش دهد. اطلاعات می تواند به صورت دستی و یا با استفاده از بخشهایی از این تمرین در پایگاه داده ذخیره گردد.

همچنین در این تمرین برخلاف تمرین سوم به جای استفاده از فایل های XML از پیش ساخته شده، باید XML را با زبان PHP تولید کنید. این XML ها باید بر اساس اطلاعات موجود در پایگاه داده ساخته شوند. به عبارت دیگر نباید هیچ XML ایستایی را فراخوانی کنید، بلکه وب سرویس شما باید XML مورد نظر را تولید کند و به سمت کلاینت بفرستد.

مجددا تاکید میشود که هدف اصلی این تمرین توسعه کد در سمت سرور است! برای (تقریبا همه) موارد لازم صفحات HTML به همراه JS و CSS های لازم در اختیار شما گذاشته می شوند (البته در برخی موارد نیاز به دستکاری جزئی دارد). در این تمرین شما باید سمت سرور را توسعه دهید به نحوی که این صفحات HTML (به همراه کدهای JS و Ajax) به درستی نمایش داده شود.

مواردی که با رنگ سبز مشخص شده اند، اختیاری هستند و انجام دادن آنها نمره اضافه برای شما خواهد داشت.

### توصیف سرویس

همانطور که گفته شد، هدف این تمرین پیاده سازی یک سرویس ایمیل است. کاربر در ابتدا می تواند با استفاده از صفحه ی LoginRegister، اکانت جدید ساخته و یا با اکانت قبلی خود وارد شود. پس از ورود باید در گوشه

سمت راست header، نام کاربر نوشته شود و لینک‌هایی برای خروج و همچنین نمایه کاربر در کنار آن وجود داشته باشد.

### نکات:

- سیستم نباید اجازه‌ی ساختن اکانت جدید با username تکراری را صادر کند.
- در هنگام ورود، در صورت نادرستی نام کاربری و یا کلمه‌ی عبور، به کاربر پیغام مناسب داده شده و لینکی برای بازگشت به صفحه LoginRegister در اختیارش بگذارد.
- نمایه‌ی کاربر باید مشخصات شخصی او، **تصویرش، لیست افرادی که با آنها در تماس است<sup>1</sup>** و آخرین زمان ورود قبلی را نمایش دهد.
- با انتخاب گزینه remember me، کاربر در مراجعات بعدی به صورت login شده در نظر گرفته شده و نیازی به وارد کردن مجدد user/pass نباشد.
- **اضافه کردن captcha برای ورود به سایت اختیاری است.**

پس از ورود، هر کاربر inbox خود را مشاهده می‌کند. در این صفحه باید نام کاربر، لینکی به پروفایل او و کلیدی برای خروج قرار داشته باشد. پس از آن در بلوکی، ایمیل‌های دریافتی خود را مشاهده می‌کند. هر ایمیل شامل فرستنده، تاریخ، موضوع و متن است. با کلیک بر inbox و sent ایمیل‌های مربوطه نمایش داده می‌شوند. با کلیک بر refresh، ایمیل‌هایی که پس از تاریخ قبلی نمایش ایمیل‌ها اضافه شده‌اند، بارگذاری می‌شوند. با کلیک بر compose صفحه‌ی composeEmail باز شده و امکان فرستادن ایمیل جدید فراهم می‌شود.

### نکات:

- توجه داشته باشید در دریافت لیست نمایش داده شده در این صفحه، فقط تعداد کاراکتر محدودی از متن ایمیل باید از سرور درخواست شده و نمایش داده شود. متن کامل ایمیل تنها در صفحه‌ی مربوط به نمایش ایمیل قابل مشاهده است.

---

<sup>1</sup> contacts

- با استفاده از radio button موجود در بلوک عملیاتی، کاربر می تواند ایمیل های خود را بر اساس نام فرستنده، تاریخ و ساعت دریافت و یا بر اساس اینکه attachment دارند یا خیر دسته بندی کند. پردازش های لازم برای این منظور، باید در سمت سرور انجام شوند.
- در هر مرحله تعداد ایمیل های نمایش داده شده در صفحه با text box قرار گرفته در بالای بلوک مشخص می شود. توجه کنید فقط باید همان تعداد ایمیل از سرور درخواست و دریافت شود. (تمام ایمیل ها دریافت نشوند)
- ایمیل هایی که قبلا خوانده شده اند با رنگ سبز مشخص می شوند.
- ایمیل هایی که سرور به عنوان spam شناسایی شناسایی کرده است به رنگ زرد نمایش داده شوند. (انتخاب مکانیزم تشخیص spam به عهده ی شماست و باید مکانیزم انتخابی خود را در گزارش نهایی تشریح کنید. مثلا ایمیل هایی که با متن یکسان به تعدادی از کاربران فرستاده شده اند، متن ایمیل حاوی کلمات مشخصی باشد، فرستنده ی ایمیل توسط افراد زیادی بلاک شده باشد و ... )
- هر کاربر می تواند در صفحه ی پروفایل خود، تصویری انتخاب کرده و به عنوان background تمامی صفحات آپلود کند. در login های بعدی نیز، اگر تصویری قبلا انتخاب شده بود، به عنوان پس زمینه بارگزاری شود.
- امکان دسته بندی ایمیل ها وجود داشته باشد و کاربر بتواند دسته ی جدید تعریف کرده، دسته ای را پاک کند و یا نام آن را تغییر دهد.

با کلیک بر روی هر ایمیل، کاربر به صفحه ی readEmail فرستاده شده و محتوای ایمیل را مشاهده می کند. اگر ایمیل انتخاب شده، spam بود، ابتدا به کاربر خطاری داده می شود و در صورت تائید او ایمیل باز می شود. کاربر در صفحه ی readEmail با استفاده از کلید delete می تواند آن ایمیل را پاک کند. کاربر می تواند فایل attachment را دانلود کند.

برای نوشتن ایمیل جدید، کاربر می تواند بر compose کلیک کرده و به صفحه ی compose راهنمایی شود. در این مرحله با وارد کردن آدرس ایمیل دریافت کننده و موضوع و متن و attachment ها، ایمیل فرستاده می شود.

## نکات:

- توجه کنید پس از زده شدن کلید send ابتدا با یک ajax بررسی می شود که آیا ایمیل گیرنده‌ی مشخص شده در میان کانتکت های فرستنده وجود دارد یا خیر. در صورت عدم وجود ایمیل پیغام مناسب به کاربر داده شده و به صفحه‌ی compose بازگردانده شود.
- استفاده از ویرایشگرهای متن (که امکاناتی نظیر متن بولد و ... را فراهم می‌کند و خروجی نهایی آن به- جای رشته متن، یک فایل HTML است) اختیاری است.

هر کاربر همانطور که گفته شد می‌تواند از صفحه‌ی inbox، به پروفایل خود برود. در صفحه‌ی پروفایل علاوه بر مشخصات شخصی و لیست افراد مورد تماس، گزینه ای برای اضافه کردن فرد جدید به لیست وجود دارد که با زدن آن، کاربر به صفحه‌ی users راهنمایی می‌شود و اطلاعات کاربران مختلف را مشاهده می‌کند و با زدن کلید add to my contacts، کاربر دلخواه را به لیستش اضافه می‌کند. هر کاربر، اگر به لیست فردی دیگر اضافه شده باشد، در اولین ورود بعدی خود، با پیغامی مناسب، خبر داده می‌شود.

- کاربر می‌تواند اضافه شدن نام خود را به لیست فردی دیگر نپذیرد و در این صورت نام او از لیست آن فرد پاک می‌شود.
- کاربر بتواند در این صفحه، کاربری دیگر را بلاک کند و ایمیل‌های فرستاده شده از او را دریافت نکند. لیستی از افراد بلاک شده در پروفایل کاربر نمایش داده شود و امکان خارج کردن افراد از این لیست وجود داشته باشد.

## نکات امتیازی

علاوه بر موارد ذکرشده در صورت تمرین، موارد زیر نیز اختیاری و نمره اضافی دارند:

- در این تمرین اجازه استفاده از Framework های موجود در PHP مانند Zend و Laravel را دارید، لازم به ذکر است استفاده از این Framework ها نمره اضافی دارند.
- افزودن مکانیزم cache کردن آیتمهای دریافتی، به این صورت که اگر آیتمی در سمت کلاینت وجود دارد و از آخرین ارجاع تغییری نکرده است، دیگر از سرور دریافت نشود و فقط آیتمهای جدید گرفته شود.

## نکات پیاده‌سازی و تحویل

- این تمرین به صورت انفرادی انجام می‌شود و در صورتیکه تمرین تحویلی دو دانشجو مشابه هم باشد، برای هر دو نفر نمره صفر در نظر گرفته می‌شود.
- گزارشی از پایگاه داده مورد استفاده و جداولی که ساخته‌اید، فریم ورک مورد استفاده و ... تهیه کنید. نیازی به dump کردن پایگاه داده خود ندارید، ولی تمام فایل‌های خود را پیوست کنید.
- برای پیاده‌سازی بخش سرور می‌توانید از ابزارهایی مانند wamp, xampp و... یا از وب سرور دانشکده استفاده کنید. در هر حالت باید فایل‌های مربوط به سرور را پیوست کنید.
- فرمت اصلی صفحات و کدهای اجرایی سمت کلاینت برای شما فراهم شده‌اند. تکمیل، تصحیح و مرتبط ساختن آنها با کاربرد مورد نظر بر عهده‌ی شماست.
- توجه کنید در بسیاری از موارد، می‌توانید برای تولید ajax مناسب قسمت‌های مختلف، از نمونه‌های موجود در فایل‌های سوال راهنمایی بگیرید. مواردی دیگر نیز ممکن است نیازمند تفکر بیشتر و ایده-پردازی شما باشند. به ساختار html های فراهم شده نیز توجه کنید. (برای مثال فرم ها و نحوه ی فرستادنشان)
- ساختار xml های مورد نظر، در فایل‌های نمونه مشخص شده‌اند. در صورت نیاز به تولید xml ای دیگر با فرمتی جدید، توضیحات لازم را در گزارش خود بیاورید.
- ممکن است در برخی از موارد، نیاز به تغییر کدهای فراهم شده داشته باشید. در این صورت نیز، تغییرات پیاده شده را ذکر کنید.
- **زمان ارسال فایل‌های تمرین حداکثر تا ساعت ۲۳:۵۹ روز جمعه ۱۱ تیر است.**
- **تحویل حضوری تمرین روز شنبه ۱۲ خرداد خواهد بود.**
- فایل‌های خود را داخل پوشه ای بگذارید و پس از فشرده سازی، نام آن را مطابق الگوی زیر قرار دهید. (فقط فایل‌های فشرده شده با فرمت ZIP بررسی می‌شوند.)  
HW4-StudentID-LastName-FirstName.zip ( HW4-9631001-Ahmadi-Ahmad.zip)

موفق باشید.