مهندسی اینترنت

تمرین شماره‌ی ۷

مدرس: **دکتر خامس‌پناه**

طراحان: امین عارف زاده، علیرضا زارع نژاد

مهلت تحویل: جمعه ۲۴ اردیبهشت ۱۴۰۰، ساعت ۲۳:۵۵

بلبل‌ستان!

مقدمه

هدف از این پروژه، آشنایی با روش‌های احراز هویت[1] و کسب اجازه[2] و اطمینان از وجود برخی از پارامترهای امنیتی در برنامه می‌باشد.

[1] Authentication

[2] Authorization

فاز هفتم پروژه

فرآیند ثبت‌نام کاربران

در این قسمت قرار است فرآیند ثبت‌نام دانشجویان را طراحی کنید که در آن پس از وارد کردن فیلدهای لازم، یک دانشجوی جدید با مشخصات داده شده در پایگاه‌داده اضافه شود. در این صفحه علاوه‌بر فیلدهای **نام، نام‌خانوادگی، شماره دانشجویی، تاریخ تولد، رشته، دانشکده و مقطع** که در فازهای قبلی داشتید، دو فیلد جدید **ایمیل** و **کلمه‌ی عبور** نیز از کاربر دریافت می‌شود. این دو فیلد برای احراز هویت کاربران مورد نیاز است.

برای فیلد **وضعیت تحصیلی** همواره مقدار "مشغول به تحصیل" را در نظر بگیرید. همچنین برای ساده شدن کار و درگیر نشدن با مسائل مربوط به آپلود تصویر، می‌توانید مقدار ثابتی را برای فیلد **تصویر** قرار دهید.

فرض بر این است که کاربران ثبت نام شده دانشجویان جدیدی هستند که درس پاس شده ای ندارند.

نکات:

- قبل از ارسال اطلاعات به سرور، فرمت ورودی ها را در سمت کاربر[3] اعتبارسنجی کنید.
- کلمه‌ی عبور را به هیچ وجه به صورت plain text در پایگاه‌داده ذخیره نکنید بلکه از hash آن استفاده کنید.
- انجام اعتبارسنجی‌هایی نظیر تکراری نبودن ایمیل و شماره دانشجویی در سمت سرور الزامیست.

مانند فازهای قبلی، همچنان دریافت لیست دانشجویان در هنگام اجرا شدن پروژه و اضافه کردن آن‌ها به پایگاه‌داده را خواهید داشت و فرآیند ثبت‌نام صرفا برای دانشجویان جدید است. در API جدید فیلدهای ایمیل و کلمه‌ی عبور نیز به ازای هر دانشجو فرستاده می‌شود (کلمه‌ی عبور را به صورت hash شده ذخیره کنید).

http://138.197.181.131:5200

[3] client

احراز هویت به کمک JWT[4]

در این بخش به کمک JWT که یک روش بدون حالت[5] است (بدون نیاز به حافظه در سمت سرور)، احراز هویت را به برنامه‌ی خود اضافه می‌کنید. استاندارد JWT در اکثر زبان‌های برنامه‌سازی پیاده‌سازی شده و برای جاوا نیز چندین پیاده‌سازی برای آن وجود دارد.

هر JWT شامل سه بخش است:

۱. Header: شامل اطلاعات الگوریتم مورد استفاده برای signature و نوع token است.

۲. Payload: شامل claim های JWT است. در این پروژه استفاده از claim های **iss**، **iat** و **exp** (با زمان انقضای یک روز) اجباری است. در کنار آن‌ها می‌توانید از claimهای استاندارد یا غیراستاندارد دیگر نیز استفاده کنید (مثلاً یک claim به نام userId برای هویت کاربر).

۳. Signature: این قسمت شامل امضای دیجیتال سرور است که برای اطمینان از صحت JWT اضافه می‌شود. این امضا معمولا به کمک الگوریتم‌های HMAC و RSA محاسبه می‌شود. در این تمرین برای راحتی از **الگوریتم HMACSHA256** همراه با **کلید bolbolestan** استفاده کنید. در این حالت امضا به‌صورت زیر تولید می‌شود:

HMACSHA256(base64UrlEncode(header) + '.' + base64UrlEncode(payload), "bolbolestan")

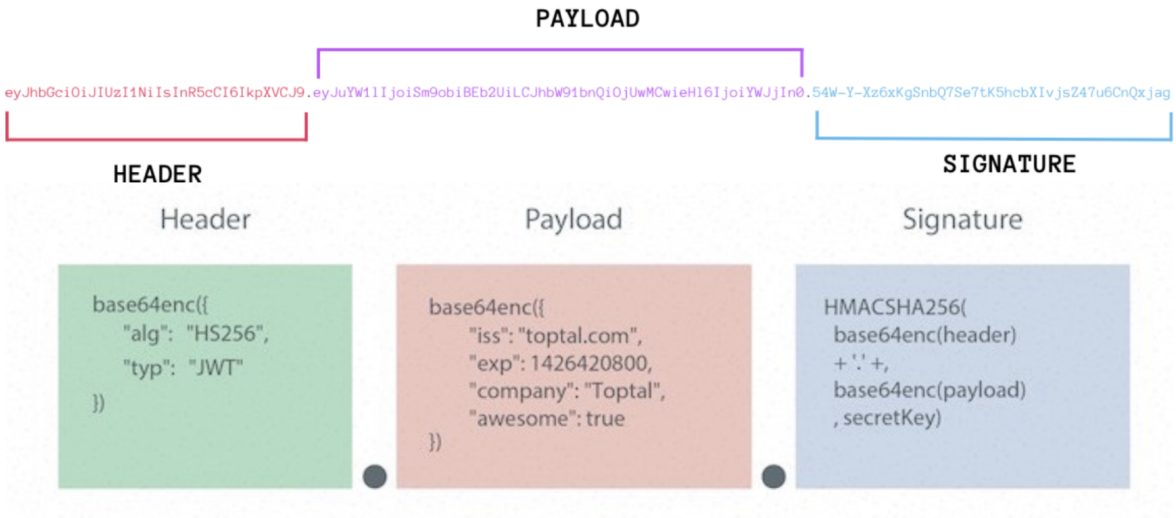
به دلیل اینکه signature تنها توسط سرور قابل تولید است (چون فقط سرور کلید را دارد)، پس میتوان صحت JWT را بر اساس آن سنجید. در صورت علاقه می‌توانید کمی در مورد امضای دیجیتال تحقیق کنید.

فرمت نهایی JWT به‌صورت زیر است:

base64UrlEncode(header) + '.' + base64UrlEncode(payload) + '.' + signature

[4] JSON Web Token

[5] Stateless



فرایند ورود کاربر

در این قسمت قرار است فرایند ورود کاربر را پیاده‌سازی کنید. ابتدا لازم است یک API طراحی کنید که ایمیل و کلمه‌ی عبور کاربر را دریافت کند و در صورت درست بودن اطلاعات، برای کاربر یک JWT صادر کند. در صورت عدم صحت اطلاعات نیز پیغام مناسب همراه با status code شماره‌ی ۴۰۳ به کاربر می‌فرستد.

در سمت کاربر نیز پس از صادر شدن JWT، این توکن را نگه‌داری کنید و در درخواستهای بعدی در header authorization بفرستید.

اعتبارسنجی کاربر

پس از موفق آمیز بودن فرآیند ورود، یک JWT برای کاربر صادر می‌شود که کاربر برای درخواست‌های بعدی خود این JWT را به سرور می‌فرستد. در نتیجه باید در درخواست‌های بعدی بر اساس JWT فرستاده شده اعتبارسنجی کاربر را انجام دهید.

برای این کار به‌جای اینکه در ابتدای هر servlet به بررسی این موضوع بپردازید، از فیلترها، که یکی از پرکاربردترین امکانات JavaEE است، استفاده کنید. یک فیلتر بسازید و در آن درستی JWT دریافت شده را بررسی‌کنید. در پیاده‌سازی این فیلتر باید سه حالت زیر را در نظر بگیرید:

- در صورت درست بودن JWT، اطلاعات کاربر را از پایگاه‌داده بگیرید و به عنوان یک attribute برای request تعیین کنید تا در ادامه به راحتی به این اطلاعات دسترسی داشته باشید.
- در صورت وجود مشکل در JWT، پاسخی با status code شماره‌ی ۴۰۳ را برای کاربر ارسال کنید.
- در صورتی‌که کاربر JWT را ارسال نکرده بود و قصد دسترسی به صفحاتی که نیازمند احراز هویت کاربر هستند را داشت، پاسخی با status code شماره‌ی ۴۰۱ به او ارسال کنید.

پس از پیاده سازی این فیلتر، آن را بر روی همه‌ی API های موجود در سیستم که نیاز به احراز هویت دارند بگذارید. دقت کنید که API های ثبت‌نام و ورود نیازی به احراز هویت ندارند.

فراموشی کلمه عبور

در این قسمت فرآیندی را باید طراحی کنید که دانشجویان در صورت فراموش کردن کلمه عبور بتوانند آن را تغییر دهند. برای اینکار ابتدا در یک صفحه ایمیل فرد را بگیرید. در صورت وجود ایمیل در پایگاه‌داده لینک مخصوصی را برای کاربر تولید کنید و به ایمیل او ارسال کنید. کاربر با کلیک روی این لینک وارد صفحه ای می‌شود که میتواند کلمه‌ی عبور خود را تغییر دهد.

دقت کنید که لینک ایجاد شده باید براساس کاربران متفاوت باشد و همچنین مسائل امنیتی در آن به خوبی رعایت شده باشد. در فرآیند تولید لینک این را در نظر بگیرید که می‌خواهیم این فرآیند نیز بدون حالت باشد (بدون نیاز به ذخیره‌سازی داده در پایگاه‌داده). با توجه به مسائلی که در مورد JWT آموختید روند مشابهی را برای تولید این لینک در نظر بگیرید. برای پیاده‌سازی این قسمت از همان کلید **bolbolestan** استفاده کنید و زمان انقضای لینک‌ها را ۱۰ دقیقه در نظر بگیرید.

جهت ساده‌تر شدن کار می‌توانید از API زیر برای ارسال ایمیل استفاده کنید. برای اینکار کافی است یک ریکوئست با متد POST به API زیر ارسال کنید. در محتوای این ریکوئست شما باید ایمیل فرد و لینکی که قرار است به او نشان داده شود را در کلید های email و url بفرستید. (نوع محتوا ریکوئست JSON است).

http://138.197.181.131:5200/api/send_mail

ممکن است ایمیل ارسال شده Spam تشخیص داده شود به همین دلیل در صورتی که آن را پیدا نکردید پوشه‌ی Spam خود را جستجو کنید.

نیازمندی های سمت رابط کاربری

در سمت کاربر دو حالت برای احراز هویت وجود دارد:

- کاربر وارد سیستم نشده و JWT ندارد که در این حالت **تنها** می‌تواند صفحات ورود، ثبت‌نام و فراموشی رمز را مشاهده‌کند (در صورت واردکردن آدرس سایر صفحات، کاربر را به صفحه‌ی ورود هدایت کنید).
- کاربر به سیستم وارد شده که در این حالت می‌تواند به تمامی صفحات به جز ورود، ثبت‌نام و فراموشی رمز دسترسی پیدا کند (در صورت واردکردن آدرس این صفحات، کاربر را به صفحه‌ی اصلی هدایت‌کنید).

در صورتی که کاربر وارد برنامه شده باشد، با بازنشانی[6] صفحه همچنان باید اطلاعات مربوط به احراز هویت او ثابت باقی بماند. برای این کار می‌توانید JWT را در حافظه‌ی محلی مرورگر نگه‌داری کنید و در هر بار بازخوانی صفحه آن را از حافظه‌ی محلی مرورگر بخوانید.

امکان خروج[7] کاربر از حسابش را نیز اضافه کنید که با توجه به بدون حالت بودن JWT، تنها کافی است این توکن را در سمت‌کاربر پاک کنید.

---

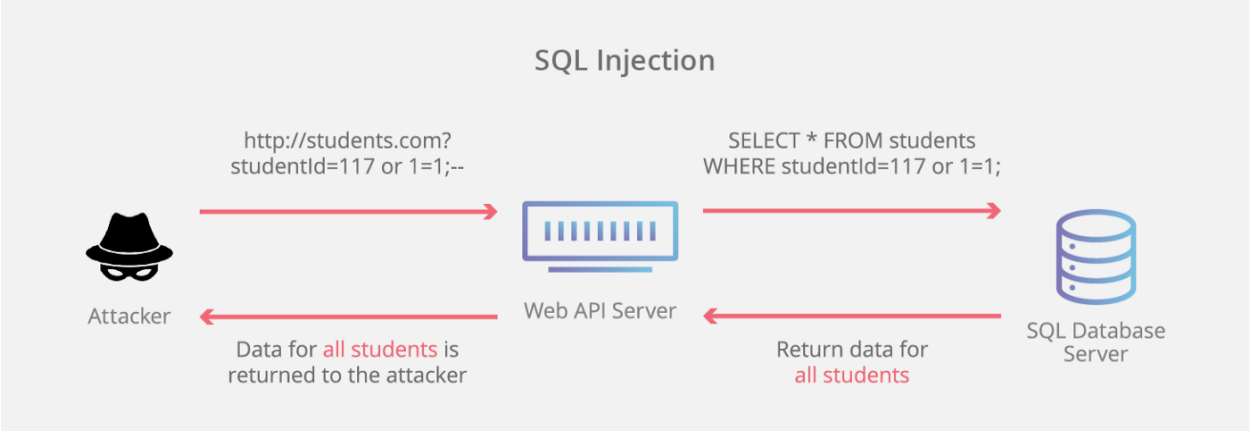
[6] Refresh

[7] Logout

هدف اصلی این پروژه یادگیری فرآیندهای مربوط به احراز هویت کاربران است. به همین دلیل مسائل مربوط به طراحی و زیبایی صفحات اهمیتی ندارد و می‌توانید طراحی دلخواه خود را داشته باشید. اما انتظار داریم نکاتی که در فازهای قبلی یاد گرفته‌اید را در این فاز هم رعایت کنید.

ایمن سازی در برابر حملات SQL Injection

در حملات Injection فرد حمله کننده در داده‌های ارسالی خود از دستورات یا کوئری‌هایی استفاده می‌کند که در صورت اجرا شدن در سرور، می‌توانند مشکل‌زا باشد. حمله‌ی SQL Injection نیز نوعی از حمله‌ی Injection است که در آن فرد حمله‌کننده کوئری‌های SQL را در داده‌های ارسالی خود به سرور می‌فرستد. به عنوان مثال می‌توانید به سناریوی زیر دقت کنید که در آن فرد حمله کننده با استفاده از حمله‌ی SQL Injection توانسته به اطلاعات تمامی دانش‌آموزان دسترسی پیدا می‌کند.



در این بخش شما باید APIهایی که در آن از کاربران ورودی دریافت می‌کنید را طوری تغییر دهید که در برابر حملات SQL Injection مقاوم باشند. برای اینکار باید از PreparedStatement ها استفاده کنید.

نکات پایانی

- کافی است که یکی از اعضای گروه Hash مربوط به آخرین کامیت پروژه سمت سرور و سمت کاربر را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت مورد ارزیابی قرار می‌گیرد.

- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این فاز پروژه‌ی شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.

- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاستی که در کلاس گفته شده است کسر خواهد شد.

- 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 دیگر شبکه‌های اجتماعی مانند تلگرام راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.

- ایمیل طراحان پروژه:

aminarefzadeh1376@gmail.com

azarenejad99@gmail.com