

مکتب شریف

اولین بوتکمپ آموزشی - استخدای ایران



پروژه نهایی

مکتب ۹۶





Phase 1 (Java Pure and Hibernate)

یک سیستم ارائه دهنده خدمات منزل را در نظر بگیرید.

این سیستم باید شامل امکانات/ویژگی های زیر باشد:

امکان افزودن کاربر جدید با توجه به نقش (متخصص/مشتري)

قابلیت تغییر رمز عبور برای کاربران

ثبت خدمات یا زیر خدمات جدید برای خدمات های تعریف شده (مدیر)

اضافه و حذف متخصصان از زیر خدمات های موجود در سیستم (مدیر)

پیاده سازی CRUD برای کلیه موجودیت ها

❖ خدمات

خدمات اصلی این سیستم عبارت اند از: دکوراسیون ساختمان، تاسیسات ساختمان، وسایل نقلیه، اسباب کشی و باربری، لوازم خانگی، نظافت و بهداشت که هر یک از این خدمات به چند زیر خدمات تقسیم می شود. به عنوان نمونه : لوازم خانگی : لوازم آشپزخانه، رختشویی، لوازم صوتی و تصویری - نظافت و بهداشت : نظافت، خشکشویی و اتوشویی، قالیشویی و مبل شویی، سمپاشی منازل این خدمات باید توسط مدیر و در حالت ران تایم به دیتابیس اضافه شوند (صرفاً مواردی برای تست).



باید متدی برای برگرداندن لیست خدمت های موجود و دیدن آنها توسط مدیر وجود داشته باشد.

برای هر زیرخدمت یک قیمت پایه و یک توضیح مختصر (در هنگام تعریف) در پایگاه داده ذخیره میشود. (توضیح و قیمت پایه باید قابلیت ویرایش داشته باشند در هر زمانی که نیاز به این مورد احساس شد).

در صورتی که خدمت انتخاب شده برای افزودن زیر خدمات وجود نداشته باشد و یا نام زیر خدمت تکراری باشد، خطای مناسب برگردانده شود.

❖ کاربران

کاربران موجود در این سیستم یکی از نقش های ، متخصص و مشتری را دارا می باشند. و همچنین یک مدیر که از قبل در دیتابیس تعریف شده است. برای هر یک از مشتریان یا متخصصان لازم است حداقل اطلاعات زیر نگهداری شود:

۱- نام و نام خانوادگی

۲- آدرس ایمیل (آدرس ایمیل باید منحصر به فرد باشد)،

۳- رمز عبور (اجباری است به طول ۸ و ترکیبی از حروف و اعداد میباشد).

۴- وضعیت متخصص (شامل <جدید>، <در انتظار تایید> و <تایید شده> میباشد).

۵- تاریخ و زمان ثبت نام

امکان افزودن کاربر جدید را با استفاده از اطلاعات ذکر شده و نقش کاربر (مشتری/متخصص) با اعتبارسنجی های مناسب سمت سرور فراهم کنید. علاوه بر اطلاعات ذکر شده ، برای هر یک از متخصصین عکس پرسنلی با حداکثر 300 کیلوبایت و فرمت عکس حتما باید jpg باشد و امتیاز مربوط به آن متخصص (که در بدو ثبت نام برابر صفر می باشد) ذخیره میشود. همچنین هر متخصص میتواند در یک یا چند زیر خدمت فعالیت داشته باشد.



قابلیت اضافه کردن متخصص به خدمات سیستم و حذف متخصصان از خدمات را پیاده کنید. (این کار باید توسط مدیر سامانه انجام شود)

مدیر در بخش لیست کاربران، بعد از ثبت نام موفق یک متخصص، کاربر ثبت نام شده را تایید خواهد کرد و متخصص مورد نظر به وضعیت "تایید شده" تغییر میکند. و سپس مدیر می تواند متخصص را به زیر خدمت های موجود اضافه کند و تا زمانی که متخصص به حالت تایید شده، در نیامده هیچ دسترسی در سیستم ندارد.

❖ ثبت سفارش

مشتری می تواند هر یک از زیر خدمات تعریف شده در سامانه را انتخاب (اول با متدی خدمت ها و سپس زیر خدمت ها را ببیند و انتخاب کند) و سفارش جدید ثبت کند. همچنین باید توضیحاتی در مورد سفارش از جمله قیمت پیشنهادی (نباید کمتر از قیمت پایه زیر خدمت انتخابی باشد)، توضیحات کار، تاریخ و زمان انجام (قبل از الان نباشد) و آدرس را وارد کند. در این مرحله سفارش در وضعیت "منتظر پیشنهاد متخصصان" قرار می گیرد.

انجام موارد زیر در حد ساخت موجودیت ها کفایت میکند.

هر سفارش میتواند در یکی از وضعیت های "منتظر پیشنهاد متخصصان"، "منتظر انتخاب متخصص"، "منتظر آمدن متخصص به محل شما"، "شروع شده"، "انجام شده" و "پرداخت شده" قرار بگیرد.

❖ پیشنهادات

برای هر یک از متخصصین، امکان ارسال پیشنهادات برای سفارشات ثبت شده وجود دارد. پیشنهاد متخصص شامل تاریخ و زمان ثبت پیشنهاد، قیمت پیشنهادی، زمان پیشنهادی برای شروع کار (قبل از زمان وارد شده توسط مشتری نباشد) و مدت زمان انجام کار است.

❖ اعتبار

هر مشتری دارای اعتبار میباشد که میتواند آن را برای پرداخت هزینه سفارشات خود خرج کنند و همچنین هر متخصص اعتبار مخصوص به خود را دارد که بعد از پرداخت سفارش توسط مشتری، مبلغ به پروفایل او افزوده میشود.

❖ نظرات

پس از اتمام کار، مشتری امتیازی برای عملکرد متخصص اعلام میکند و در صورت تمایل میتواند نظر خود را نیز بیان کند.



Phase 2(Spring Boot & Spring Data)

موارد خواسته شده در فاز قبل را با استفاده از اسپرینگ بازنویسی کرده و برای تمامی متد های سرویس ها Unit test بنویسید.

❖ ارسال پیشنهاد توسط متخصص

متخصص می تواند برای سفارشات که در زیرخدمت های مرتبط به او ثبت شده اند و در وضعیت منتظر پیشنهاد متخصصان و یا منتظر انتخاب متخصص هستند، پیشنهاد خود را ارسال کند. پیشنهاد متخصص شامل قیمت پیشنهادی، زمان پیشنهادی شروع کار و مدت زمان انجام کار است. قیمت پیشنهادی نباید از قیمت پایه ی آن زیرخدمت کمتر باشد و همچنین زمان شروع کار نباید قبل از الان وارد شود. (دقت کنید متدی که میخواهد سفارشات را به متخصص نمایش بدهد باید سفارشات در حالت «منتظر انتخاب متخصص» و یا «منتظر پیشنهاد متخصصان» را که مربوط به زیرخدمت متخصص می باشد به او نمایش دهد.

بعد از ثبت یک پیشنهاد توسط متخصصان وضعیت سفارش به "منتظر انتخاب متخصص" تغییر می یابد.

❖ مشاهده لیست پیشنهادات سفارش

مشتری می تواند لیست پیشنهاد های سفارش خود را با توجه به قیمت پیشنهادی یا امتیاز متخصص مرتب و ببیند.

❖ انتخاب متخصص برای سفارش توسط مشتری

مشتری از بین پیشنهاد های ارسالی، یکی از پیشنهاد ها را انتخاب می کند و وضعیت سفارش به "منتظر آمدن متخصص به محل شما" تغییر می کند.

❖ مسئول تغییر حالت سفارش از "منتظر آمدن متخصص به محل شما" به

حالت "شروع شده" مشتری می باشد که این تغییر را نمی توان پیش از زمانی که متخصص در پیشنهاد خود ثبت کرده است را انجام داد و باید پس از آن باشد.



❖ پس از اتمام کار توسط متخصص، مشتری می بایست حالت سفارش را از "شروع شده" به "انجام شده" تغییر دهد.

دقت کنید که در این فاز به ران شما از تست ها نمره داده می شود.
کلاس تست باید قابلیت یکجا اجرا شدن را داشته باشد و نباید متد به متد اجرا گرفت.



Phase 3(Spring MVC)

برنامه نوشته شده در تمرین قبل را با استفاده از Spring MVC بازنویسی و تکمیل کنید. همچنین لازم است در این مرحله امکانات/ویژگی های زیر را به سیستم اضافه نمایید

موارد خواسته شده تا اینجا باید قابلیت تست با postman را داشته باشند و نمره این فاز بر اساس تست های انجام شده وارد می شود.(بارم بندی برای تست های فاز قبل به وسیله postman نیز لحاظ می شود.)

❖ پس از اینکه حالت سفارش به ,انجام شده , تغییر یافت , مشتری مسئول پرداخت هزینه سفارش می باشد که در این صورت حالت سفارش از ,,انجام شده,, به ,,پرداخت شده ,, تغییر پیدا میکند.دقت کنید که در صورتی که زمان ثبت حالت ,,انجام شده,, پس از تایم مشخص شده توسط متخصص در مرحله ثبت پیشنهاد بود باید به ازای هر ساعت تاخیر یک امتیاز از امتیاز های متخصص کسر شود.

❖ دقت کنید که در صورت منفی شدن امتیاز ها متخصص، حساب او باید به حالت غیر فعال تغییر پیدا کند.

❖ پرداخت هزینه سفارش، دو حالت پرداخت برای مشتری وجود دارد :

۱-پرداخت از اعتبار : اگر مشتری این گزینه را انتخاب کرد در صورت داشتن اعتبار کافی عملیات انجام می شود و در غیر اینصورت خطای مناسب نمایش داده می شود.



۲. پرداخت آنلاین : با انتخاب این گزینه کاربر به پرداخت آنلاین هدایت می شود و با وارد کردن اطلاعات کارت اعتباری خود (این اطلاعات باید در بک اند اعتبارسنجی شوند)، اقدام به پرداخت هزینه می کند.

استفاده از captcha در صفحه ی پرداخت الزامی است. (حتما باید به صورت سمت سروری هندل شود نه سمت مشتری)

حداکثر مدت زمان حضور کاربر در صفحه پرداخت ۱۰ دقیقه است.

پس از اتمام پرداخت مشتری، ۷۰ درصد مبلغ پرداختی به پروفایل متخصص افزوده می شود.

❖ ثبت نظرات

پس از اتمام کار، مشتری بایستی امتیازی برای عملکرد متخصص اعلام کند (عددی بین ۱ تا ۵ می باشد) و در صورت تمایل میتواند نظر خود را به صورت توضیحی نیز اضافه نماید. پس از ثبت نظرات مشتریان، متخصصین امتیاز خود را بدون دیدن توضیحات جزئی میتوانند مشاهده کنند.

❖ جستجو و فیلتر لیست متخصصان/مشتریان

مدیر باید بتواند در لیست کاربران، بر اساس پارامترهای مختلفی مانند نقش کاربر (متخصص/مشتری) نام، نام خانوادگی، ایمیل، حوزه تخصصی متخصصان، امتیاز متخصصان (بیشترین یا کمترین امتیاز) فیلترهایی را اعمال کند و به جستجو بپردازد و کلیه ی رکورد هایی که شرایط تعیین شده را دارند به عنوان نتیجه مورد نظر، به او نمایش داده شود.

دقت کنید که موارد وارد شده به صورت اختیاری می باشد و مدیر بر هر اساس که لازم بداند باید بتواند روی کاربران فیلتر اعمال کند.

داشتن Postman Collection اجباری است.



Phase 4(Sprint Security)

Spring Security را به پروژه خود اضافه نمایید(مدیر و متخصص و مشتری فقط به url های مربوط به خود دسترسی باید داشته باشند).

❖ ایمیل فعالسازی ثبت نام

به منظور اعتبارسنجی آدرس ایمیل کاربر، پس از ثبت نام، ایمیل فعال سازی اکانت برای او ارسال میشود که حاوی لینکی برای نهایی سازی ثبت نام خود می باشد.

کاربر باید به ایمیل خود مراجعه نموده و روی آن کلیک نماید. در صورت رخ ندادن خطایی، با پیغام مناسب، نهایی شدن فرایند ثبت نام به کاربر اعلام شود. برای نقش متخصص، وضعیت کاربر بعد از تایید ایمیل به "در انتظار تایید" ثبت و به وی نمایش داده میشود تا زمانی که مدیر ثبت نام او را تایید نماید.

لینک نهایی سازی ثبت نام (لینک ایمیل شده) فقط یکبار قابلیت فراخوانی دارد و پس از آن معتبر نخواهد بود. اگر کاربر برای بار چندم روی آن لینک کلیک نماید باید خطای مناسب مشاهده کند.

❖ تکمیل پروفایل مدیر

در قبل بیان شد که مدیر می تواند لیست تمامی مشتریان و متخصصان را دریافت و فیلتر کند. در این فاز این امکان اضافه می شود که تاریخچه کاملی از خدمات گرفته شده/انجام شده توسط هر کاربر(کاربر مشخص) همراه با جزئیات تاریخ و هزینه و ... به مدیر نمایش داده می شود. همچنین مدیر سامانه می تواند تاریخچه ی تمامی سفارشات را بر اساس بازه ی زمانی، وضعیت سفارش، نوع خدمت و زیرخدمت مشاهده کند. همچنین امکان گزارش گیری از متخصصان و مشتریان بر اساس زمان ثبت نام در سامانه، تعداد درخواستهای ثبت کرده، تعداد سفارش های انجام داده وجود دارد.



❖ مشاهده تاریخچه سفارشات و اعتبار

مشتری و متخصص در پروفایل خود یک تاریخچه ی کامل از سفارشات که تاکنون داشته اند، با در نظر گیری وضعیت هر سفارش خواهند داشت. همچنین کاربر قادر به مشاهده اعتبار کیف پول خود می باشد.

داشتن Postman Collection اجباری است.

توضیحات کلی:

برای پیاده سازی نیازمندی های موجود، حتما نکات زیر را در نظر بگیرید:

۱. لایه بندی ها به طور کامل و درست رعایت شود در تمامی فاز ها.
۲. موجودیت ها و ارتباطات بین آنها دقیق و بهینه نوشته شوند و حتما برای هر یک از آنها حداقل اطلاعات ذکر شده نگهداری شود.
۳. خطاهای احتمالی برنامه را حتما شناسایی و رسیدگی کنید
۴. حتما پروژه را در git قرار دهید و تغییرات را در Repository مربوطه، همراه با پیام معنادار commit و push کنید.
- ۵- برای صفحه پرداخت باید یک صفحه وب طراحی کنید و این صفحه می بایست از یک زیبایی حداقلی با استفاده از بوت استرپ برخوردار باشد و اطلاعات این صفحه با استفاده از ajax به سمت سرور فرستاده شوند.
- دقت کنید مورد کپچا حتما باید به صورت سمت سروری هندل شود و در صورتی که سمت مشتری باشد هیچ نمره ای دریافت نخواهید کرد.
- ۶- تمامی کاربران سیستم شما هیچ دیدی نسبت به اطلاعات داخل دیتابیس ندارند (حتی مدیر سیستم) و اگر قرار است چیزی اضافه شود، اول نیاز به دریافت اطلاعات از دیتابیس می باشد این کار باید در متد جداگانه ای انجام شود، برای مثال



وقت اضافه کردن زیر خدمت به خدمت های موجود حتما باید متدی برای نمایش تمامی خدمت های موجود وجود داشته باشد تا مدیر بتوانید با دیدن آن اطلاعات زیر خدمت را به دیتابیس اضافه کند.

۷- در تمامی فاز ها هیچ نمره ای به کد شما تعلق نمیگیرد و کد شما حتما باید قابلیت تست داشته باشد.

۸- دقت کنید که تحت هیچ شرایطی نباید برنامه شما به اکسپشن بخورد و در صورت این اتفاق حتما کسر نمره خواهد داشت.

۹- استفاده از اکسپشن کلی مانند Exception و اکسپشن های نامربوط مانند RuntimeException کسر نمره دارد، در صورت نیاز اکسپشن کاستوم مورد نظر خود را پیاده سازی کنید.

۱۰- فیچر های فاز ها را از آن چیزی که در هر فاز مشخص شده است جلو تر نبرید.

۱۱- در صورت مشاهده کپی کردن هیچ بهانه ای پذیرفته نیست و نمره صفر برای فاز منظور خواهد شد.

۱۲- در صورت عدم ارائه مناسب نمره ای تعلق نمی گیرد و موارد خواسته شده مورد به مورد بررسی خواهد شد.

۱۳- لطفا دقت کنید که باید متد هایی برای برگرداندن سفارشات در وضعیت های انتخابی مشتری ها را داشته باشید، برای مثال مشتری که قصد پرداخت یک سفارش رو داشته باشد باید بتواند سفارش های خود را که در وضعیت انجام شده (یک وضعیت قبل از پرداخت شده) ببیند تا یکی از آنها را برای پرداخت انتخاب کند.