



تیم بک اند گاردین

طراحی api های صدور بیمه رشته عمر

سلام! امیدوارم حالتون خوب باشه. اول از همه خیلی خوشحالیم که الان دارید این برنامه ورودی تیم گاردین رو توش شرکت می کنید و امیدواریم که همتون توش قبول بشید :) اولین توضیحی که لازمه بدیم اینه که یکم سبک این تست فنی با بقیه ی شرکت ها متفاوت. توی این تست مهارت شما و همچنین سلیقه و در عین حال fast learner بودن شما توی زمان کم مهم ترین اهمیت رو داره و به همین خاطر مینی پروژه ای که در ادامه می بینید جنبه ی الگوریتمیک نداره و معیار های ارزیابی مینی پروژه در ادامه آورده شده است.

در این مینی پروژه قصد داریم یک نمونه بسیار کوچک از «رشته بیمه عمر» را پیاده کنیم. در این پیاده سازی قرار است از با استفاده از چارچوب Django یک بک اند کوچک از صدور رشته عمر را پیاده سازی نماییم که در ادامه به توضیح آن خواهیم پرداخت. دقت شود که این پروژه به صورت api خواهد بود و قرار است همه ی داده ها به صورت json جا به جا شود و به این ترتیب قرار نیست کد HTML توسط بک اند سرو شود و صرفا بک اند به عنوان api عمل می کند.

تکولوژی های مورد استفاده در پروژه

در عین حال توصیه می شود که از تکولوژی های زیر هم در پیاده سازی خود بهره بگیرید که در ارزیابی پروژه شما بسیار تاثیر خواهند داشت:

- فریم ورک Django
- استفاده از ابزار هایی مانند Sphinx برای داکيومنت کردن کد های بک اند (توجه شود که استفاده از ابزار های داکيومنت کردن در بک اند بسیار مهم است و می توان از این ابزار معرفی شده و یا هر موردی که دوست دارید استفاده کنید ولی پیاده سازی آن به صورت دقیق اجباری است)
- استفاده از ابزار های تست در جنگو و نوشتن تست برای هر یک از api ها (می توانید به شیوه های مختلفی تست را بنویسید ولی دقت کنید که از روش های استاندارد استفاده نمایید)
- استفاده از لایبرری هایی مثل Rest در پیاده سازی api ها

لیست مواردی که باید پیاده شوند به صورت زیر خواهد بود:

- ایجاد مدلی برای هر یوزر که شامل یک نام کاربری، رمز عبور و همچنین یک تلفن همراه خواهد بود.
- پیاده سازی api رجیستر برای هر کاربر که نام کاربری و رمز عبور و تلفن همراه را به صورت یک json دریافت می کند و به این ترتیب یک یوزر جدید ایجاد می کند.
- در صورتی که خطایی در رجیستر کاربر وجود داشت به صورتی مناسب به صورت json خطا به کاربر نشان داده شود. (دقت شود که در لایه ORM جنگو practice best هایی برای چک کردن برخی خطاها وجود دارد. لذا؛ بهتر است از این نوع از چک کردن خطاها استفاده کنید و این مورد در ارزیابی پروژه هم تاثیر خواهد داشت)
- در صورتی که خطایی در ثبت نام کاربر وجود نداشت، توکن ورودی به صورت json به او ارسال شود. (دقت شود که بهتر است از روش هایی مثل JWT برای ایجاد توکن ها استفاده نمایید)
- به صورت مشابه با رجیستر api لاگین کاربر نیز طراحی شود که نام کاربری و رمز عبور را می گیرد. دقت شود که همه ی مواردی که در رجیستر قرار است رعایت شود در اینجا هم باید رعایت شود.
- api برای دریافت اطلاعات یک کاربر بر اساس توکن. که token به صورت یک json به api داده می شود و در پاسخ اطلاعات او اعم از نام کاربری و تلفن همراه به او برگردانده شود.
- api برای Out Log وجود داشته باشد که باید token فرد را پاک کند تا دیگر نتواند از آن استفاده کند.
- api برای تغییر رمز عبور که در ورودی به صورت یک json توکن را به همراه رمز عبور قبلی و دو بار تکرار شدن رمز عبور جدید می گیرد. در نهایت در صورتی که توکن یا رمز عبور درست نبود و یا رمز عبور جدید و تکرار آن با هم نمی خواندند خطا خروجی دهد و در غیر این صورت پسورد را تغییر دهد و در خروجی هم به صورت json پیغام مناسب بدهد.
- یک api هم برای صدور بیمه عمر خواهیم داشت که در ورودی به صورت json موارد متفاوتی را دریافت می کنند که عبارتند از: توکن، نام، نام خانوادگی، ایمیل، تلفن همراه، سن، BMI فرد، یک عبارت boolean که نشان می دهد فرد سیگار می کشد یا نه و همچنین در مورد قلیان هم مورد مشابه وجود دارد. به علاوه تعداد نخ های مصرف سیگار فرد و تعداد دفعات مصرف قلیان هم در این api داده می شود. در صورتی که آن دو boolean نادرست باشند، این مقادیر عددی صفر خواهند بود.
- در api صدور بیمه عمر باید خطاهایی بررسی شود اعم از اینکه هیچ کدام از فیلد ها null نباشند و همچنین سن فرد بیشتر از ۴۵ سال نباشد و به عنوان نمونه اطلاعات درست باشد. مثلاً تلفن همراه ۱۱ رقمی باشد و با ۰۹ شروع شده باشد و ایمیل هم حاوی @ باشد. از طرفی فیلد ها به درستی پر شده باشند. مثلاً BMI به صورت حروف وارد نشده باشد.
- یک کد شعبه هم در api بالا در json صدور بیمه عمر داده می شود. این کد شعبه در واقع باید یک کلید خارجی در پایگاه داده در جدول «شعبات بیمه» باشد. این جدول یک پیاده سازی بسیار ساده شامل کد شعبه است که منحصر به فرد است و در کنار آن نام شعبه آورده شده است.

- دقت شود که موارد امنیتی برای جلوگیری از حملات رایج در api های بالا رعایت شود.

پنل ادمین

در ادامه کافیسٹ پنل ادمینی که جنگو دارد را به پروژه اضافه کنید.

- این صفحه کافیسٹ همان پیاده سازی پنل ادمین جنگو باشد که همه قابلیت هایی که در بالاتر توضیح دادیم را به صورت هر چهار عملیات پایه (REST) (Insert – Update – Delete – Select) برای api ها باشد و بتوان از آنها استفاده نمود.
- در این قسمت می توان از پیاده سازی documentation automatic جنگو هم استفاده نمود. به این ترتیب از django admin documentation generator استفاده کنید.

معیار های ارزیابی پروژه

معیار های ارزیابی پروژه به صورت زیر خواهد بود:

- همه ی مواردی که در بالاتر گفته شد رعایت شده باشد.
- همه ی خطاها در api ها به درستی رعایت شده باشند و نمایش داده شوند.
- در پیاده سازی مدل ها دقت بسیار زیادی شود که تا حد ممکن بهترین حالت مدل تعریف شود. به عنوان نمونه اگر جایی می توان فیلدی را به صورتی تعریف کرد که حافظه کمتری بگیرد از آن استفاده شود. در عین حال تا حد ممکن دقت شود که عدم اختلاط داده ها را داشته باشیم و هر کدام از آنها به صورت مستقلا داده ای جداگانه باشند تا از سطح نرمالیت بالایی برخوردار باشند و آنومالی نداشته باشد.
- نوشتن داکيومنتیشن به صورت جداگانه علاوه بر داکيومنتیشن پنل ادمین بسیار مهم است که در ابتدای داک در مورد ابزار آن صحبت شده است.

- نوشتن تست برای هر یک از api ها بسیار مهم است.
- تمیز بودن کد شما و Readable بودن اهمیت بسیاری دارد.
- فولدربندی مناسب و جداکردن هرکدام از قسمت های کد بسیار حائز اهمیت است.
- استفاده از practice best ها اگر جایی موردی هست بسیار مهم است.
- استفاده از لایبرری REST در پیاده سازی api ها و صحیح بودن همه ی api ها

دقت شود که ممکن است موردی از مواردی که در بالاتر و داکيومنت پروژه آمده را بلد نباشید. اما نگران نباشید (= همه ی مواردی که آورده شده اند به قدری رایج و ساده هستند که در صورتی که کمی سرچ کنید و ویدیو در یوتیوب ببینید احتمالاً در هر بخش و استفاده از هر تکنولوژی مشکلاتان حل می شود. زمان مفید این مینی پروژه در صورتی که موارد گفته شده را بلد باشید ۳ ساعت است که در صورتی که نیاز به یادگیری موردی هر یک از موارد بالاتر و تکنولوژی ها باشد ممکن است تا ۶ ساعت هم طول بکشد. لذا؛ به آخرین زمان ارسال مینی پروژه را موکول نکنید. fast learner بودن شما هم در این یادگیری بسیار مهم است و یکی از معیار های ما برای افراد کم تجربه تر همین مورد خواهد بود. امیدوارم همتون رو توی گاردین ببینیم (=)

موفق باشید :)