

سامانه ایجاد و مدیریت فرم‌های پویا (داینامیک)

توضیحات

هدف این پروژه؛ ایجاد یک سامانه با قابلیت ذخیره‌سازی، نمایش و نظارت بر فرم‌ها و فرایندهای پویا و گزارش‌گیری از نتایج آن‌هاست.

با توسعه این سامانه قصد داریم سیستمی پیاده‌سازی کنیم که کاربران از طریق آن می‌توانند فرایندهای دلخواه خود را ایجاد و دسترسی افراد مختلف به آن را مدیریت کنند.

کاربر با ورود به این سامانه باید بتواند فرمی از سوالات دلخواه خود را ساخته و لینک منحصر به فرد آن را در اختیار دیگران قرار دهد. در پایان باید بتواند از خروجی نتایج دریافتی، یک گزارش نمایش دهد.

نیازمندی‌ها

• فرم:

- کاربران می‌توانند با استفاده از مدل‌های مختلف سوال، هر تعداد سوالی که می‌خواهند به فرم خود اضافه کنند. (توجه داشته باشید که پشتیبانی از انواع فیلد های مختلف به عهده شماست، اما پشتیبانی از سه فیلد Text, Select, Checkbox ضروری است).
- یک فرم میتواند به تعداد دلخواه فیلد داشته باشد.
- کاربران می‌توانند تعداد نامحدودی فرم ایجاد کنند.
- فرم ها می‌توانند عمومی یا شخصی باشند (فرم‌های شخصی با گذرواژه اختصاصی قابل دسترسی هستند).
- هنگام ایجاد فرم بر اساس نوع سوال، اطلاعات مورد نیاز از کاربر دریافت شود.

• فرایند:

- هر فرایند از تعداد دلخواهی فرم که قبلا ایجاد شده تشکیل می‌شود.
- فرایندها نیز به دو دسته عمومی یا شخصی تقسیم بندی می‌شوند (فرم‌های شخصی با گذرواژه اختصاصی قابل دسترسی هستند).
- پشتیبانی از دو نوع فرایند الزامی است:
- فرایندهای خطی: در این فرایند ها تمام مراحل پیش نیاز یکدیگر هستند و کاربر قبل از تکمیل مرحله جاری نمی‌تواند مرحله بعدی را انجام دهد.
- فرایندهای آزاد: کاربر می‌تواند تمام مراحل را بصورت دلخواه انجام دهد.

• دسته بندی:

- کاربران می‌توانند دسته بندی‌های متفاوتی ایجاد کنند و فرم‌ها و فرایندهای خود را برای مدیریت بهتر در آن قرار دهند.

• گزارش گیری:

- کاربران می‌توانند خروجی فرم‌ها را به صورت یک گزارش یکپارچه از سوالاتی که قابلیت جمع‌داده‌هایشان وجود دارد مشاهده کنند (مانند فیلد های عددی و انتخابی).
- برای هر فرم و فرایند گزارش های زیر را پیاده سازی کنید:
 - تعداد بازدید کاربران.
 - تعداد پاسخ های دریافتی.
- پاسخ‌های ارسال شده توسط کاربران برای هر فرم باید قابل مشاهده باشد.
- کاربر بتواند گزارشی از داده های دریافت شده در هر فرم را به صورت برخط (real time) مشاهده کند. به معنی که اگر دیتای جدیدی دریافت شد بدون نیاز به ارسال درخواست مجدد نتایج گزارش بروز شود.
- مدیر سایت نیز باید قادر باشد تا در صورت نیاز به صورت دوره‌ای (ماهانه/هفتگی) گزارشی از داده های درون برنامه از طریق ایمیل یا ارسال درخواست به یک API مشخص دریافت کند.

• سایر:

- استفاده از REST FRAMEWORK.
- ایجاد مستندات برای API های پیاده سازی شده.
- استفاده از کشینگ برای افزایش سرعت.
- استفاده از سیستم احراز هویت جنگو و OTP.
- قرار دادن پروژه در حالت پروداکشن. (شامل استفاده از وب سرور های محیط پروداکشن، قرار دادن تنظیمات مربوطه در حالت پروداکشن، جمع آوری فایل های استاتیک و (collect statics) ...).
- داکرایز کردن پروژه.
- برای تنظیماتی که به محیط اجرای برنامه وابسته هستند مانند کلید محرمانه، تنظیمات پایگاه داده، حالت دیباگ و ... از متغیرهای محیطی (env file) استفاده کنید.
- استفاده از ابزارهای دلخواه در زمینه تست نویسی.
- استفاده از Reverse Proxy ها مانند Nginx.

- شما میتوانید بسته به خلاقیت و توانمندی خود در پروژه از async programming و ابزار های مختلفی مانند Celery، NoSql، GraphQL و ... استفاده کنید. لازم به ذکر است که این موارد شامل امتیاز اضافه می‌باشد.
- ثبت نام و ورود کاربر از طریق سرویس‌هایی مانند گوگل و ... ([OAuth2](#)). (اختیاری و شامل امتیاز اضافه)

مستندات ضمیمه

در پوشه اصلی پروژه یک پوشه به نام documents ایجاد کنید و فایل های خواسته شده را در این پوشه قرار دهید.

- برای طراحی پایگاه داده خود از نمودار ERD استفاده کنید.
- در فایلی به نام README نحوه راه اندازی پروژه را ذکر کنید.