



نکات مهم

- انجام تمرین در گروه‌های دو یا سه نفره انجام خواهد شد.
- تمامی تمرین‌ها باید در یک ریپازیتوری گیت‌هاب انجام شوند که لینک آن قبل از ددلاین طی فرآیندی که اعلام خواهد شد، لازم است به دستیاران آموزشی ارسال شود. مشارکت تمامی اعضای گروه در تمرین ضروری است؛ اگر عضوی از گروه، کم‌تر از ۱۵ درصد مشارکت ثبت شده (طبق آمار اعلام شده توسط گیت‌هاب بر اساس کامیت‌ها و تعداد خطوط کامیت شده‌ی هر عضو) داشته باشد، نمره‌ای از آن تمرین به وی تعلق نخواهد گرفت.
- ریپازیتوری شما باید شامل یک فایل README باشد که اسامی و شماره‌های دانش‌جویی اعضای گروه در آن قید شده باشد. توجه کنید که فایل توضیحات شما باید شامل اسکرین‌شات‌هایی از اپلیکیشن، تکنولوژی و معماری‌ای که استفاده می‌کنید باشد. حتما توضیح کوتاهی در مورد کارکرد اپ در این فایل ارائه دهید.
- لازم است فایل زیپ شده‌ی ریپازیتوری، در کوثرای درس نیز بارگذاری شود. عدم تطابق فایل آپلود شده در کوثرا با ریپازیتوری گیت‌هاب، معادل تخلف در تمرین است. توجه کنید که کامیت‌های با معنی و تقسیم کار گروهی در این تمرین شامل نمره است.
- تحویل تمرین به صورت آنلاین است. تمامی اعضا باید در تحویل تمرین حضور داشته باشند. هر عضو موظف است به بخشی از کد که خودش پیاده‌سازی کرده است کاملاً مسلط باشد و بداند بخش‌هایی که خودش پیاده‌سازی نکرده چگونه کار میکنند. لازم نیست تمامی اعضا به تمامی بخش‌های کد مسلط باشند.
- در صورت کشف اولین تخلف یا تقلب بر اساس شرایط گفته شده در آیین‌نامه‌ی دانشکده موجود در ویکی دانشکده، نمره‌ی آن تمرین تمامی اعضای تیم صفر منظور شده و در صورت تکرار تقلب، همه‌ی اعضای تیم در درس مردود خواهند شد.
- منابع مناسبی که به شما برای حل تمرین کمک میکنند و مطالعه آنها حتما توصیه می‌شود روی کوثرا برای استفاده شما قرار داده خواهند شد.
- استفاده از هر ای پی ای که دارای قابلیت‌های خواسته شده باشد مجاز است ولی پیشنهاد میشود از ای پی ای opentdb.com ها پیشنهادی استفاده کنید.
- در این تمرین استفاده از `preferences shared` مجاز نیست و باید از یک دیتابیس برای نگه‌داری داده‌ها استفاده کنید.
- درخواست زدن به سرور نباید در ترد اصلی انجام شود. برای درخواست زدن‌ها به سرور میتوانید از کتابخانه‌هایی همچون `OKHTTP` و `Volley` و یا `Retrofit` استفاده کنید.
- تنها برای گرفتن سوالات لازم است که به سرور درخواست بزنید و باقی اطلاعات را در دیتابیس ذخیره کنید. برای مثال اطلاعات کاربر را در دیتابیس ذخیره کنید و به او اجازه دهید که اکانت بسازد یا وارد بازی شود.
- برای فعال سازی لایت مود و یا دارک مود نباید نیازی به باز و بسته کردن برنامه باشد و همچنین برای انجام این کار نیاز است که از کدهای خود اندروید استفاده شود و نباید به صورت دستی پیاده سازی شود

- توجه کنید که در این تمرین حتما باید از یک معماری مناسب برای اپ خود استفاده کنید و برای انتخاب این معماری دلیل داشته باشید. معماری‌های معروف در اپلیکیشن‌های اندروید MVVM و MVC هستند اما از معماری‌های دیگر هم می‌توانید استفاده کنید.

در این تمرین قصد داریم که یک اپ کوئیز طراحی کنیم که سوالات آن را از یک API آنلاین دریافت خواهیم کرد. شما با انجام این تمرین باید قادر باشید با استفاده از روش‌های خوب یک اپلیکیشن اندرویدی بنویسید که در **ترد جداگانه‌ای به یک API آنلاین** درخواست بزند و نتایج را که دریافت میکند به کاربر نشان دهد و همچنین در **دیتابیس آفلاین** این نتایج را نگه دارد. این تمرین شبیه‌ترین تمرین شما به کارهای واقعی صنعت برنامه‌نویسی موبایل است.

- این تمرین شامل پنج صفحه است: صفحه لاگین/رجیستر، صفحه پروفایل، صفحه تنظیمات، صفحه اصلی بازی، صفحه امتیازات و اسکوربورد

- **صفحه لاگین/رجیستر:** کاربر در این صفحه قابلیت ثبت نام و یا ورود را دارد. برای ورود کاربر باید یک ایمیل و پسورد که از قبل تعین کرده است را وارد کند. اگر ایمیل وجود نداشته باشد باید خطای مناسب به او نمایش داده شود و یا اگر پسورد او اشتباه بود باید خطای مناسب نشان داده شود اگر همه چیز درست بود کاربر باید به صفحه بازی هدایت شود. توجه کنید که با یکبار لاگین کاربر با بستن اپ نباید از اکانت خود خارج شود مگر اینکه خودش دکمه خروج را انتخاب کند.

در بخش ثبت نام کاربر باید یک ایمیل و پسورد انتخاب کند، با فشردن دکمه ثبت نام اطلاعات او باید در دیتابیس اپلیکیشن ذخیره شود و سپس به صفحه لاگین هدایت شود تا با ایمیل و پسوردش وارد اپ شود.

- **صفحه بازی:** صفحه بازی را باید طوری طراحی کنید که بازی قابلیت شروع داشته باشد و همچنین کاربر از این صفحه باید به صفحات پروفایل، تنظیمات و اسکوربورد دسترسی داشته باشد. برای مثال میتوانید یک drawer در این صفحه قرار دهید.

با فشردن دکمه شروع در این صفحه با توجه به تنظیمات انتخاب شده باید به یک ای پی ای ریکوست زده شود و سوالات را دریافت کنید. در ابتدا اگر کاربر تنظیمات را وارد نکرده بود یک تنظیمات دیفالت برای او در نظر بگیرید.

در هر سوال باید نحوه چپش گزینه ها طوری انجام شود که جواب همه سوالات یک گزینه واحد نشود به عنوان مثال جواب همه سوالات گزینه ۱ نباشد

در هر سوال بعد از انتخاب جواب پاسخ مناسب نشان داده شود (صحیح یا غلط بودن)

در انتهای بازی امتیاز بر حسب (هر سوال صحیح ۳ امتیاز و هر پاسخ غلط ۱ امتیاز منفی) محاسبه شود

هر درجه سختی با توجه به سطح آن یک ضریب دارد (آسان : ۱ ، متوسط: ۲ ، سخت: ۳)

همچنین کاربر بعد از بازی میتواند در یک لیست سوالات و پاسخ آن ها را مشاهده کند برای پیاده سازی این مورد از

Recycler View

استفاده کنید.

- اگر کاربر بتواند رکورد بازی را بزند این مورد به او نشان داده شود.

بازی های ساخته شده باید به مدت ۵ روز در cache نگه داری شوند و در صورتی که کاربر در حالت آفلاین اقدام به ساخت بازی کند یک توست در رابطه با آفلاین نبودن کاربر به او نشان داده شود و سپس بدون آنکه وارد صفحه تنظیمات شود یکی از بازی های گذشته کاربر دوباره اجرا شود.

- **صفحه تنظیمات:** کاربر در این صفحه میتواند تم اپلیکیشن را انتخاب کند.

میتواند تنظیمات بازی را در این صفحه ویرایش کند. تنظیمات شما باید حداقل شامل سختی، تعداد سوال آزمون و دنگوری سوالات باشد و این تنظیمات باید روی ریکوئست شما به API اعمال شود.

- **صفحه پروفایل:** کاربر در این صفحه می تواند اطلاعات کاربری خود را تغییر دهد یا تکمیل کند (اسم، شماره تلفن، نام کاربری و ...) این اطلاعات با تغییر کاربر باید در دیتابیس بازی ذخیره و آپدیت شوند.

- صفحه اسکور بود: کاربر در این صفحه قابلیت دیدن بهترین امتیازات بازی را خواهد داشت هر ردیف شامل نام کاربری بازیکن و امتیاز کسب شده است. با کلیک بر روی هر سطر این لیست کاربری به عنوان کارت پیرو فایل بازیکن نشان داده شود و اطلاعات کاربری بازیکن قابل مشاهده خواهد بود. تفاوت اول تا سوم با رنگ متفاوتی نشان داده شوند.